

**UNISYS**

A Series  
Systems  
**Functional  
Overview**

Release 3.9.0

September 1991

Priced Item

Printed in U S America  
8600 0353-000



**UNISYS**

A Series  
Systems  
**Functional  
Overview**

Copyright © 1991 Unisys Corporation.  
All Rights Reserved.  
Unisys is a registered trademark of Unisys Corporation.

Release 3.9.0

September 1991

Priced Item

Printed in U S America  
8600 0353-000

The names, places, and/or events used in this publication are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual, living or otherwise, or that of any group or association is purely coincidental and unintentional.

**NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT.** Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this publication may be forwarded using the Product Information card at the back of the manual, or may be addressed directly to Unisys, Product Information, 25725 Jeronimo Road, Mission Viejo, CA 92691.



# Page Status

Page	Issue
iii	-000
iv	Blank
v through xvii	-000
xviii	Blank
xix	-000
xx	Blank
1-1 through 1-8	-000
2-1 through 2-21	-000
2-22	Blank
3-1 through 3-22	-000
4-1 through 4-13	-000
4-14	Blank
5-1 through 5-24	-000
6-1 through 6-10	-000
7-1 through 7-26	-000
8-1 through 8-33	-000
8-34	Blank
9-1 through 9-20	-000
10-1 through 10-6	-000
11-1 through 11-4	-000
12-1 through 12-3	-000
12-4	Blank
A-1 through A-2	-000
B-1 through B-2	-000
Glossary-1 through 24	-000
Bibliography-1 through 14	-000
Index-1 through 16	-000

Unisys uses an 11-digit document numbering system. The suffix of the document number (1234 5678-xyz) indicates the document level. The first digit of the suffix (*x*) designates a revision level; the second digit (*y*) designates an update level. For example, the first release of a document has a suffix of -000. A suffix of -130 designates the third update to revision 1. The third digit (*z*) is used to indicate an errata for a particular level and is not reflected in the page status summary.



# About This Overview

## Purpose

The *A Series Systems Functional Overview* introduces the A Series line of computer systems and their associated system software and environmental software products. This overview is designed to answer the following types of questions:

- “Should I buy an A Series system? Is it an effective solution for the types of business problems I face?”
- “I’ve just received an A Series system and a large assortment of documentation. It’s all new to me. What really is in an A Series system and what do all these products do?”
- “I have an A Series system and there is a particular task I want to perform using this system. Which software product should I use to accomplish this task?”
- “I’ve heard about a particular A Series software product, and it sounds like it might be relevant to my needs. What are the general capabilities of this software product?”

## Scope

This overview presents a functionally organized description of all the system and environmental software products Unisys sells for use on A Series systems. This overview helps you to understand how each product is used, and some of the particular strengths of each product. Where two products serve a similar function, this overview helps you to choose between them.

This overview is not a tutorial about how to administer or operate an A Series system. This overview refers you to other published Unisys documents for detailed information about how to operate the various products that are introduced here.

## Audience

If you are a new A Series system customer, are considering acquiring a system, or just want a general introduction to A Series systems, then you should read this manual.

## Prerequisites

You do not need any prior knowledge of Unisys systems to understand the information in this manual, but you should have a basic grasp of elementary computer concepts.

# How to Use This Overview

Part I of this overview, "Products," discusses A Series functions and the products that provide those functions. If you want an overview of A Series functions, or if you want to find out which product performs a particular function, you can use the table of contents. For example, suppose you are interested in database performance management. In the table of contents, you will find a section titled "Database Management." Within that section is a subsection on "Database Operations," which includes a smaller topic called "Database Performance Monitoring."

Within each topic, the text introduces the product that provides the stated function (for example, the DMMONITOR utility). The text describes the function of the product in further detail, explaining four or five noteworthy capabilities of the product. The length of the presentation varies, depending on the complexity of the product being discussed.

After reading the brief product descriptions given in this overview, you should be able to decide whether you want more details about any of the products. Each topic ends with a list of documents you can consult for further information about the product being discussed. If the online help text is available for the product, that is also mentioned.

If you have heard the name of an A Series product, and you want information about that particular product, you should go to the index first. For example, if you are interested in InfoExec™, you will find that *InfoExec* appears in the index with references to the pages where various aspects of InfoExec are discussed.

For information about A Series support services, migration policies, and documentation, refer to Part II of this overview, "Support."

Throughout this overview, B 21, B 25, B 26, B 27, B 28, and B 38 systems are referred to as the BTOS™ workstations.

Technical terms are defined and acronyms are spelled out in the text where they first appear. In addition, many of these terms and acronyms are defined in the glossary of this this overview.

Unless otherwise noted, manuals referred to in this overview are for the A Series systems.

## Organization

This overview contains the following parts, sections, and appendixes. In addition, a glossary, a bibliography, and an index appear at the end of this overview.

### Part 1. Products

The sections in this part describe the products and features of A Series systems.

### **Section 1. A Series Systems: A Design for Change**

This section provides a thumbnail sketch of the types of capabilities provided by A Series systems. These capabilities are explored in more detail in later sections of this overview.

### **Section 2. A Series Architecture**

This section introduces the basic hardware components and system software components of A Series systems. The presentation is brief and fairly nontechnical.

### **Section 3. Database Management**

This section introduces the types of database architectures available on A Series systems. This section also introduces the utilities used for database creation, operations, and inquiry or updating.

### **Section 4. Transaction Management**

This section introduces the A Series products that provide end users with easy online access to data, and the products that allow system administrators and programmers to effectively control online transactions.

### **Section 5. Program Development**

This section introduces fourth-generation and third-generation programming languages and the associated utilities that increase programmer productivity.

### **Section 6. Operational Interface**

This section discusses the products that system operators use to control the system.

### **Section 7. Basic System Management**

This section describes the products and features used to manage processes, memory, files, disk units, printing, system configuration, and system reinitialization.

### **Section 8. Data Communications Networks**

This section discusses the various types of networks that provide communication between terminals and the host system, or between two different host systems.

### **Section 9. System Monitoring and Security**

This section introduces the tools used for monitoring system performance and reliability. This section also describes the products that provide flexible levels of security on the system, including features certified by the United States Department of Defense as providing the C2 controlled access protection level.

## About This Overview

---

### **Part 2. Support**

The sections in this part describe the support Unisys provides for users of A Series systems.

#### **Section 10. CUSTOMCARE® Services**

This section describes hardware and software support services, customer education, and consulting services.

#### **Section 11. System Updates**

This section introduces the procedures used to distribute new firmware, system software, and environmental software to customers.

#### **Section 12. Documentation**

This section introduces the various types of A Series documentation.

#### **Appendix A. Inactive Software Products**

This appendix lists A Series software products that have a support-only status.

#### **Appendix B. User Groups**

This appendix describes the user groups available worldwide for users of A Series systems.

## Related Product Information

Throughout this overview, references are given to documents that provide further information about the various products described. For a list of all the documents mentioned and the form numbers used to order them, refer to the bibliography near the end of this overview. For a detailed introduction to all A Series documentation, refer to the following document:

#### ***A Series Documentation Library Overview (form 8600 0361)***

This overview describes the library of A Series software documentation. It also provides an explanation of titling conventions, the procedure for ordering documentation, and an introduction to online documentation and its role in A Series product documentation. This overview is written for all users of A Series systems.

---

CUSTOMCARE is a registered trademark of Unisys Corporation.

# Contents

About This Overview .....	v
---------------------------	---

## Part 1. Products

### Section 1. A Series Systems: A Design for Change

System Performance Range .....	1-1
Business Applications .....	1-2
Fourth-Generation Application Development Facilities .....	1-3
Advanced Database Management Facilities .....	1-4
Simplified End-User Interfaces .....	1-4
Detailed Transaction Management .....	1-5
Application Development Productivity Aids .....	1-5
Flexible Networking Options .....	1-6
Simplified System Management .....	1-7
Language Translation Capabilities .....	1-8

### Section 2. A Series Architecture

System Hardware Components .....	2-1
Central Processor Subsystem .....	2-2
Memory Subsystem .....	2-3
Input/Output (I/O) Subsystem .....	2-4
MLIP-Based Systems and HDU-Based Systems .....	2-4
Resource Management Module (RMM) Systems .....	2-5
Peripherals .....	2-5
Maintenance Subsystem .....	2-5
System Software Components .....	2-7
Master Control Program/Advanced Systems (MCP/AS) .....	2-7
Message Control Systems (MCSs) .....	2-8
Remote Print System (ReprintS) .....	2-8
System Utilities .....	2-8
Language Compilers .....	2-9
Programming Utilities .....	2-9
Operating System Concepts .....	2-9
Compiler-Oriented Hardware .....	2-10
Stack Architecture .....	2-10
Multiprogramming and Multiprocessing .....	2-11
Libraries .....	2-12
Automatic Code Segmentation .....	2-12
Reentrant Code .....	2-12
Descriptors .....	2-13
Virtual Memory .....	2-13
Actual Segment Descriptor (ASD) Memory Architecture .....	2-14
Configuration-Independent Software .....	2-14

<b>Growth Paths</b> .....	2-15
Performance Measurements .....	2-15
Hardware Compatibility .....	2-18
Mainframe Upgrades .....	2-18
I/O Subsystem Upgrades .....	2-19
Operating System Compatibility .....	2-20
Application Compatibility .....	2-21
Operations Compatibility .....	2-21

### Section 3. Database Management

<b>Database Management Environments</b> .....	3-2
<b>Structured Query Language Database (SQLDB)</b> .....	3-3
SQLDB Database Structure .....	3-3
SQLDB Database Creation .....	3-4
SQLDB Database Query and Update Facilities .....	3-4
Host Language Interfaces .....	3-4
<b>Semantic Information Manager (SIM) Databases</b> .....	3-4
SIM Database Structure .....	3-5
SIM Database Creation and Reorganization .....	3-9
SIM Database Query and Update Facilities .....	3-9
Host Language Interfaces .....	3-10
<b>InfoExec Environment Query and Update Facilities</b> .....	3-10
Interactive Query Facility (IQF) .....	3-10
Workstation Query Facility (WQF) .....	3-11
<b>Data Management System II (DMSII) Databases</b> .....	3-12
DMSII Database Structure .....	3-12
DMSII Database Creation and Reorganization .....	3-14
DMSII Database Query and Update Facilities .....	3-15
Extended Retrieval with Graphic Output (ERGO) ..	3-15
DMS.View .....	3-16
Host Language Interfaces .....	3-16
<b>Database Creation</b> .....	3-16
Advanced Data Dictionary System (ADDS) .....	3-16
Data Definition Languages (DDLs) .....	3-18
Data and Structure Definition Language (DASDL)	3-18
Semantic Information Manager Object Definition	
Language (SIM ODL) .....	3-18
Structured Query Language Data Definition	
Language (SQL DDL) .....	3-18
<b>Database Operations</b> .....	3-18
Database Backup, Auditing, and Recovery .....	3-19
Choosing Recovery Strategies .....	3-19
Recovery for Audited Databases .....	3-20
Remote Database Backup .....	3-20
Database Integrity Certification .....	3-21
Database Structure Analysis .....	3-21
Database Performance Monitoring .....	3-22



**Section 4. Transaction Management**

<b>Mainframe Transaction Environments</b> . . . . .	4-1
Generalized Transaction Control . . . . .	4-1
Flexible Configuration . . . . .	4-2
Multiple Windows into the System . . . . .	4-2
Automatic Program Initiation . . . . .	4-3
Transaction Routing . . . . .	4-3
Processing Items . . . . .	4-4
Database Recovery . . . . .	4-6
Additional Features . . . . .	4-6
Levels of Feature Content . . . . .	4-7
Migration from Other MCSs . . . . .	4-7
File Maintenance and Program Control . . . . .	4-8
<b>Workstation Integration</b> . . . . .	4-9
Windows into the Mainframe Environment . . . . .	4-9
Data Transfer and Reformatting . . . . .	4-10
DTS Release Level 3.0 . . . . .	4-11
DTS Release Level 4.1 . . . . .	4-11
DTS Release Level 5.0 . . . . .	4-12

**Section 5. Program Development**

<b>Fourth-Generation Program Development</b> . . . . .	5-2
Logic and Information Network Compiler II (LINC II) . . . . .	5-2
MAPPER . . . . .	5-5
<b>Third-Generation Program Development</b> . . . . .	5-6
Programming Languages . . . . .	5-6
ALGOL (Algorithmic Language) . . . . .	5-6
C Language . . . . .	5-7
COBOL74 . . . . .	5-7
COBOL85 . . . . .	5-7
Data Communications ALGOL (DCALGOL) . . . . .	5-7
Data Management ALGOL (DMALGOL) . . . . .	5-8
FORTRAN77 (Formula Translation ANSI-77) . . . . .	5-8
NEWP . . . . .	5-8
Pascal . . . . .	5-8
RPG (Report Program Generator) . . . . .	5-9
Work Flow Language (WFL) . . . . .	5-9
SORT Language . . . . .	5-9
Product Interfaces . . . . .	5-10
Database Interfaces . . . . .	5-10
Centralized Data Definitions . . . . .	5-10
Multilingual Interfaces . . . . .	5-11
Transaction Processing Interfaces . . . . .	5-13
Language Extensions . . . . .	5-13
Input/Output Features . . . . .	5-13
Run-Time Sharing of Program Modules . . . . .	5-14
Compile-Time Sharing of Program Modules . . . . .	5-15
Task Initiation and Control . . . . .	5-16
Interprocess Communications (IPC) . . . . .	5-16

Communication across Networks . . . . .	5-17
Sorting Capabilities . . . . .	5-17
Development Environment . . . . .	5-18
User Interface Generation . . . . .	5-18
Interactive Debugging Environment . . . . .	5-20
Workstation-Based Program Development . . . . .	5-20
Terminal-Based Program Development . . . . .	5-21
Change Control and Cross-References . . . . .	5-22
Automated Program Compilation and Testing . . . . .	5-23

**Section 6. Operational Interface**

<b>Menu Interface</b> . . . . .	6-1
Menu Mode and Command Mode . . . . .	6-2
Regulated Access to Operations Functions . . . . .	6-3
Customized Operator Interfaces . . . . .	6-3
<b>Command Interface</b> . . . . .	6-4
<b>Coexistence of Operations Interfaces</b> . . . . .	6-5
<b>PCMARC</b> . . . . .	6-6
PCMARC Terminal Emulator . . . . .	6-6
COMMLINK . . . . .	6-6
PCMARC Screen Controller . . . . .	6-6
<b>System Assistant</b> . . . . .	6-6
<b>Automation of Operator Tasks</b> . . . . .	6-7
<b>Internationalization Functions</b> . . . . .	6-7
<b>Multilingual System Interfaces</b> . . . . .	6-9
Message Translation . . . . .	6-9
Menugraph Translation . . . . .	6-10
Screen Translation . . . . .	6-10
Language Selection . . . . .	6-10

**Section 7. Basic System Management**

<b>Process Management</b> . . . . .	7-1
Work Flow Management . . . . .	7-1
Dynamic Control of Online Programs . . . . .	7-1
Process Priority Assignments . . . . .	7-2
Batch Job Controls . . . . .	7-2
Billing Utilities . . . . .	7-3
<b>Memory Management</b> . . . . .	7-4
<b>Input/Output (I/O) Management</b> . . . . .	7-5
File Maintenance . . . . .	7-5
Disk File Maintenance . . . . .	7-5
Generalized File Maintenance . . . . .	7-6
Disk File Backups . . . . .	7-6
Disk Management . . . . .	7-8
Logical Disk Groupings . . . . .	7-8
Disk Reliability Features . . . . .	7-9
Duplicate System Files . . . . .	7-9
Duplicate Disk Units . . . . .	7-9

Disk Performance Optimizations . . . . .	7-11
Physical Configuration Options . . . . .	7-11
Disk I/O Caches . . . . .	7-11
Virtual Disk Units in Memory . . . . .	7-13
Database Buffering . . . . .	7-14
Duplicate Files or Disks . . . . .	7-15
Disk Performance Modeling . . . . .	7-15
Tape Management . . . . .	7-16
Printing Management . . . . .	7-17
Printer Output Spooling . . . . .	7-17
Automatic and Manual Printing . . . . .	7-17
Additional Controls . . . . .	7-19
Data Comm Printing . . . . .	7-19
Remote Host Printing . . . . .	7-19
Viewing and Manipulating Printer Backup Files . .	7-20
Customized Character Sets . . . . .	7-20
<b>Soft Configuration</b> . . . . .	7-20
Configuration File . . . . .	7-20
Dynamic Changes to the Configuration . . . . .	7-21
Independent Groups within a System . . . . .	7-21
I/O Hardware Sharing . . . . .	7-24
<b>System Initialization</b> . . . . .	7-24
New System Initialization . . . . .	7-24
Automatic Power Schedules . . . . .	7-25
System Reinitialization . . . . .	7-25

**Section 8. Data Communications Networks**

<b>Open Systems Networking</b> . . . . .	8-1
HYPERchannel and NETEX Compatibility . . . . .	8-1
Compatibility with Timeplex Networking Products . . . .	8-2
The LINK Family . . . . .	8-2
TIMEPAC Packet Switching Products . . . . .	8-3
TIME/LAN Routers, Concentrators, and Intelligent Routing Bridges . . . . .	8-4
IBM Compatibility . . . . .	8-5
RJE/BSC Compatibility . . . . .	8-5
SNA Compatibility . . . . .	8-5
Physical Unit Types 2 and 5 Emulation . . . .	8-6
Logical Unit Type 6.2 Emulation . . . . .	8-7
3270 Terminal Emulation . . . . .	8-8
Remote Job Entry Emulation . . . . .	8-8
OSI Compatibility . . . . .	8-8
TCP/IP Compatibility . . . . .	8-10
X.25 Compatibility . . . . .	8-13
<b>Advanced Networking</b> . . . . .	8-15
Host Services . . . . .	8-15
Hardware Components . . . . .	8-15
Single-Host Networks . . . . .	8-15
Multihost Networks . . . . .	8-17
Network Definition . . . . .	8-19

## Contents

---

Decentralized Network Operations .....	8-19
Centralized Network Operations .....	8-20
Custom Line Protocols .....	8-21
Network Debugging Facilities .....	8-21
Compatibility Issues .....	8-22
<b>Basic Networking</b> .....	8-23
Single-Host Networks .....	8-23
Hardware Components .....	8-24
Simplified Data Comm Configuration .....	8-25
Custom Line Protocols .....	8-26
Debugging the Data Comm Network .....	8-26
Monitoring Data Comm Traffic .....	8-27
Multihost Networks .....	8-27
Host Services .....	8-28
Command Transfer .....	8-28
File Access .....	8-28
File Transfer .....	8-29
Interprocess Communication .....	8-29
Job Transfer and Remote Tasking .....	8-29
Print Routing .....	8-29
Station Transfer .....	8-30
Access Control .....	8-30
Logging .....	8-30
Hardware Components .....	8-30
Network Operations .....	8-32

## Section 9. System Monitoring and Security

<b>System Monitoring</b> .....	9-1
Snapshot System Analysis .....	9-1
Disk Performance Modeling .....	9-4
System Commands for Performance Monitoring .....	9-4
Logging and Log Analysis .....	9-5
Detailed System Performance Analysis .....	9-6
Data Collection .....	9-7
Reporting and Analysis .....	9-7
Data Management Utilities .....	9-8
Transaction Performance Monitoring .....	9-9
Database Statistics Monitoring .....	9-9
Customized Monitoring Programs .....	9-9
<b>Security Administration</b> .....	9-10
Basic Security .....	9-11
User Privileges .....	9-11
Station Privileges .....	9-12
Program Privileges .....	9-13
Data File Access Controls .....	9-13
Program Access Controls .....	9-13
Database Access Controls .....	9-14
Transaction Access Controls .....	9-15
Operations Interfaces Access Controls .....	9-15
Multiple Host Network Access Controls .....	9-16

Security Auditing .....	9-16
InfoGuard Security Enhancement Software .....	9-17
Simplified Security Administration .....	9-17
Security Administrator Status .....	9-17
Password Generation and Aging .....	9-18
Tape Security .....	9-18
Usercode Monitoring .....	9-18
Security Message Workstations .....	9-19
Logging Enhancements .....	9-19
C2 Security .....	9-19

**Part 2. Support**

**Section 10. CUSTOMCARE Services**

Education .....	10-1
Professional Services .....	10-2
Progression Aids .....	10-3
Unisys SURETY .....	10-4
Unisys SureNet .....	10-4
Unisys SURETY Service Options .....	10-5

**Section 11. System Updates**

Software Releases .....	11-1
Simple Installation (SI) .....	11-1
Progression Policy .....	11-2
Hardware Updates .....	11-4

**Section 12. Documentation**

CD-ROM Library .....	12-1
Online Documentation .....	12-2
Printed Documentation .....	12-2
Technical Information Documents .....	12-3

**Appendix A. Inactive Software Products**

**Appendix B. User Groups**

CUBE .....	B-1
International User Groups .....	B-2

Glossary .....	1
----------------	---

## Contents

---

<b>Bibliography</b> .....	1
<b>Index</b> .....	1

# Figures

2-1.	A Series Hardware Components .....	2-2
3-1.	SIM Schema of the Organization Database .....	3-6
3-2.	DMSII Schema of the Organization Database .....	3-13
4-1.	COMS Direct Window Interface .....	4-5
5-1.	LINC II Development Process .....	5-3
5-2.	Multilingual Interfaces .....	5-12
5-3.	Libraries and User Programs .....	5-15
5-4.	Interactive Program Development Environment .....	5-23
6-1.	MARC Home Menu .....	6-2
6-2.	ODT Automatic Display Mode .....	6-5
7-1.	Mirrored Disk I/O Handling .....	7-10
7-2.	Disk Cache Read and Write Handling .....	7-13
7-3.	Memory Disk .....	7-14
7-4.	Print Spooling .....	7-18
7-5.	Soft Configuration Example .....	7-23
8-1.	A Series Integration into an SNA Network .....	8-7
8-2.	TCP/IP Ethernet Networks and X.25 Networks .....	8-12
8-3.	BNA Version 2 Single-Host Network Example .....	8-16
8-4.	BNA Version 2 Multihost Network Example .....	8-18
8-5.	Hardware Links between BNA Version 1 and BNA Version 2 Hosts .....	8-23
8-6.	NSP/LSP and DCDLP Networks .....	8-25
8-7.	BNA Version 1 Network Connections .....	8-32
9-1.	CMF.Snapshot MONITOR Screen .....	9-2
9-2.	CMF.Snapshot SNAPSHOT Report .....	9-3
9-3.	SMFII Components .....	9-6
9-4.	SAMPLER Graphics Display .....	9-8





# Tables

2-1.	Relative Performance Measurements (RPM) .....	2-17
2-2.	A Series Upgrade Families .....	2-20
A-1.	Support-Only Products .....	A-1



Part 1  
**Products**



# Section 1

## A Series Systems: A Design for Change

Your business, whatever it is, depends on *information*: on its timeliness, accuracy, and completeness. Computer systems in recent years have assumed more and more of the burden of information management for all types of businesses. For that reason, your computer system can be one of the most crucial elements in your business strategy.

Yet today's computer market can be very confusing, with a huge assortment of competing systems and technologies that become obsolete almost as soon as they appear. Your own information needs may be evolving just as rapidly. The key to effective information management in such an environment is to choose systems that are adaptable to change.

Your computer system should be able to accommodate your changing needs without requiring costly conversions. It should allow you to develop applications solutions in time to take advantage of emerging business opportunities. And it should provide communication with as many types of networks and systems as possible, so that you are never cut off from valuable information.

The A Series line of computer systems is uniquely adapted to the needs of today's dynamic environment. The value of your A Series investment is ensured by such features as

- Compatible systems that provide a performance range of 1 to 144 times, so you can always increase your system capacity without costly conversions
- Fourth-generation development facilities that make it possible for you to implement applications in a fraction of the time required by conventional methods
- Networking facilities that allow you to participate in multivendor networks using any of several popular protocols

This section introduces you to some of the key features of the A Series family so that you can understand how the A Series is relevant to your needs. Later sections of this overview provide more detail about the types of capabilities introduced in this section.

### System Performance Range

The two basic components of any computer system, hardware and software, work together to process, store, and deliver information. The hardware provides the raw computational power. The software provides the working environment, determining what operational procedures must be followed and what applications can be run.

When your information needs grow beyond the capacity of a particular system, it is often advantageous to move to more powerful hardware. By contrast, it is very expensive and undesirable to have to move to a different software environment. Yet many vendors do not give you any choice in the matter. Some companies use a different software system

## A Series Systems: A Design for Change

---

for each product line and performance level. When you want more performance, you not only have to buy new hardware but also must convert to new software—an expensive and time-consuming change. Such an upgrade negates your substantial previous investment in software, and often forces you to spend even more money retraining your staff to use the new system.

The A Series, by contrast, is unique in the industry for providing the largest performance range of software-compatible systems. Beginning with the compact A 1 Model F, you can migrate upward to the A 17 Model N, which provides 144 times more computing power, without ever having to recompile an application program. There are many other A Series systems at intermediate levels of processing power, so you can acquire a system that is well matched to your workload at each stage in your growth. The power of each system is expressed in terms of a statistic called the *growth factor*. This statistic expresses the ratio between the capacity of each system and the capacity of the A 1-F, which has a growth factor of 1. For example, the A 6-HS/KS has a growth factor of 6.0, which means it can do 6 times as much work as an A 1-F.

The A Series systems allow you to develop application programs that can be run on any A Series system without recompilation. The A Series systems also share a common operating system, which provides a consistent operations interface for all these systems. Further, the vast majority of peripherals are compatible throughout the product line. Field technicians can accomplish many of the system upgrades by replacing components within your existing system cabinet, without ever removing the system from your installation. The result is a line of computers that speeds your company's growth, rather than obstructing it.

The A Series is also unique in providing a microcomputer-sized model that fits easily on a desktop and yet is fully software-compatible with the rest of the A Series. This model, called the Micro A, uses new processor technology to bring A Series capabilities to the microcomputer market.

The topic of A Series system performance range and compatibility is discussed further under "Growth Paths" in Section 2.

## Business Applications

Whatever line of business you are in, there is a good chance that Unisys provides an application system specifically developed for the unique needs of your industry. These applications save you the expense of developing large application programs and thus provide the most cost-effective solutions to many of your application needs. These applications are backed by Unisys Application Support Centers (ASCs), which provide telephone action lines you can call for immediate assistance. The following are some of the application systems Unisys provides for the A Series:

<b>Education</b>	Alumni Development Computerized Accounting Student Terminal System (CASTS) <sup>™</sup>
<b>Financial</b>	Global Financial System (GFS)

*continued*

---

CASTS is a trademark of Unisys Corporation.

*continued*

	Global Wholesale Banking (GWB)
<b>Health Care</b>	Hospital Management System (HMS) Professional Resource Network (PRN)
<b>Industrial/Commercial</b>	Business and Manufacturing Control System (BAMCS) Unisys Distribution Partner (BDP)
<b>Public Sector</b>	Advanced Police Information Management System (APLIMS) Budgetary Accounting Court Management System

Applications that cross industry lines are also available, such as the following:

- Executive LINC (General Accounting)
- OFIS® Link (Office Management Software)

Consult your Unisys marketing representative for details about these and other Unisys application programs. Your marketing representative can also give you information about application programs developed by third-party software companies who have Marketing Associate agreements with Unisys.

## Fourth-Generation Application Development Facilities

If you should decide that the unique needs of your business require a specialized application system, the fastest way to generate that application is by using fourth-generation application development tools. These tools, products of the latest wave of computer technology innovations, provide simplified syntax that mirrors the way people solve problems rather than the way the computer system solves problems. All the technical details of implementation are handled for you automatically.

The A Series provides two different fourth-generation programming systems: the Logic and Information Network Compiler II (LINC™ II) and MAPPER® systems. Both of these tools are capable of generating complete databases and the associated database query programs. They can vastly increase the productivity of professional programmers and can allow business analysts and end users to assume part or all of the responsibility of application development. But the value of these tools to you goes beyond the savings in development costs. LINC II and MAPPER bring you applications solutions in a time frame to suit your needs, so you can take advantage of business opportunities while they are still available.

---

OFIS is a registered trademark of Unisys Corporation.

LINC is a trademark of Unisys Corporation.

MAPPER is a registered trademark of Unisys Corporation.

For more information about LINC II and MAPPER, refer to "Fourth-Generation Program Development" in Section 5.

## Advanced Database Management Facilities

Databases can be developed either through the fourth-generation tools that were previously described, or through any of the following third-generation database management tools: the Data Management System II (DMSII), the Semantic Information Manager (SIM), and the Structured Query Language Database (SQLDB).

The Data Management System II (DMSII) provides software, including the Advanced Data Dictionary System (ADDS), that allows you to efficiently centralize data management without the complex programming effort normally associated with such a task. By using DMSII and ADDS, you describe data file structures and relationships to the system only once. Data can then be added, updated, and retrieved without concern for its placement, organization, or access method. DMSII and ADDS accomplish these functions automatically. DMSII supports the network and hierarchical data models.

The InfoExec Structured Query Language Database (SQLDB) is the newest database system offered on the A Series. SQLDB supports the relational data model and provides powerful reporting capabilities. A Series SQLDB conforms to both ANSI and International Standards Organization (ISO) standards.

The InfoExec family of software includes the features of DMSII and ADDS, together with the Semantic Information Manager (SIM), which allows you to implement databases supporting the semantic data model. SIM allows a database to capture business rules efficiently and understandably. SIM makes it unnecessary for application programmers to duplicate data validation code in each application program. The result is that database applications can be developed more quickly.

The tasks involved in database management are greatly simplified by the Operations Control Manager (OCM). This program provides a user-friendly screen interface for operating DMSII or SIM databases. OCM steps you through functions such as database auditing and recovery, integrity certification, and performance analysis. OCM reduces your operations overhead for database management and frees your personnel for other productive work.

For more information, refer to Section 3, "Database Management."

## Simplified End-User Interfaces

The A Series provides a modern generation of interactive programs that provide the end user with easy access to information in databases.

The InfoExec series of software includes the Interactive Query Facility (IQF). This is a menu-driven program that allows you to construct queries or updates for SIM or SQLDB databases without having to know anything about how the information is physically organized in the database. IQF includes many report formatting features as well.



The Workstation Query Facility (WQF) can be used for querying and updating SIM databases from a workstation environment. WQF uses workstation windowing facilities to allow you to see multiple views of a database simultaneously.

You can also use the IQF and WQF to query DMSII databases. A one-time run of the DMS.View utility makes a DMSII database intelligible to IQF and WQF without requiring changes to any application programs that use the database. The LINC.View utility allows you to perform the same conversion for DMSII databases created by LINC II.

The A Series also provides a number of products that conveniently integrate workstations with the mainframe environment. Of these, INFOVIEW™ II allows you to define multiple windows for viewing the mainframe environment and running programs on the mainframe. You can quickly traverse between these windows and other windows that access workstation environments such as MS-DOS®.

The Data Transfer System (DTS) allows you to transfer files between the workstation and mainframe, and performs the reformatting needed to ensure that the file is usable in either environment. DTS also allows workstations to access mainframe disk units and printers just as if they were attached directly to the workstation. DTS is also a fully featured terminal emulator.

For more information about IQF, WQF, DMS.View, and LINC.View, refer to Section 3, "Database Management." For more information about INFOVIEW II and LINC II, refer to "Workstation Integration" in Section 4, "Transaction Management," and "Fourth Generation Program Development" in Section 5, "Program Development."

## Detailed Transaction Management

The Communications Management System (COMS) provides a flexible environment for transaction control. COMS can provide special handling for individual users, stations, programs, and transaction types. COMS can automatically initiate online applications programs and control the number of copies of each application program that are currently running. COMS security allows you to specify the programs and transactions that are available to each user or group of users. COMS can also provide different users with customized interfaces to the same application program. And COMS statistics allow you to evaluate and optimize the performance of online applications programs.

For more information about COMS, refer to "Generalized Transaction Control" in Section 4.

## Application Development Productivity Aids

The A Series also provides a wide range of industry-standard programming languages, including C, COBOL85, COBOL74, FORTRAN77, C, Pascal, and RPG II. A number of

---

INFOVIEW is a trademark of Unisys Corporation.

MS-DOS is a registered trademark of Microsoft Corporation.

advanced tools make it easier for programmers to develop and debug programs written in these languages.

Screen Design Facility Plus (SDF Plus) is a user interface management system that provides the capabilities to manage the logic between the user interface and applications. SDF Plus helps you to implement screen interfaces easily for application programs by allowing you to "paint" screens directly on a terminal or workstation. The user interface management features allow you to specify validation for input data, to define form traversal, and to create online help text. Since SDF Plus separates the user interface logic from the data processing logic of an application, it is easy to change the interface whenever needed or to have multiple, customized interfaces to the same program. SDF Plus also allows you to distribute the user interface logic to an intelligent workstation.

The Test and Debug System (TADS) provides an interactive environment for debugging programs. Using TADS, you can trace or alter the flow of control through a program and display or alter the values of variables at any point during program execution.

For more information about programming languages and programming development tools such as SDF Plus and TADS, refer to Section 5, "Program Development."

## Flexible Networking Options

The A Series line provides a number of networking products that allow communication among A Series systems or between A Series systems and other vendors' systems.

BNA software allows A Series systems to be linked together as cooperating peers, for purposes of distributed processing or remote data communication. BNA allows programs on one host system to initiate and communicate with programs on another host. Programs can also perform I/O operations on files located on other host systems.

BNA allows operators to remotely control a host by entering system commands that are forwarded from one host to another. And end users can initiate remote sessions that connect their terminals to a remote host. BNA also includes security features that allow the administrator to limit the degree of access allowed to particular remote users of particular host systems.

A Series systems also support a number of alternative networking products that can link A Series systems together or link A Series systems into multivendor networks. These alternative types of networks are the following:

- The HYPERchannel® network, which provides for high-speed data transfer at a rate of 50 megabits per second.
- The LINK family, which provides economical high-speed data transfer across T-1 phone lines.
- TIMEPAC® Packet Switching Products, which provide effective use of X.25 packet-switching systems.

---

HYPERchannel is a registered trademark of Network Systems Corporation.

TIMEPAC is a registered trademark of Timeplex, Inc.

- TIME/LAN™ systems, which provide flexible networking alternatives to the users of local area networks (LANs).
- Systems Network Architecture (SNA), which allows A Series systems to communicate with IBM® or compatible systems.
- Transmission Control Protocol/Internet Protocol (TCP/IP), which provides station transfer, file transfer, and mail transfer using widely accepted protocols.
- X.25 networks, which allow A Series systems to communicate through a packet-switched data network.

For more information about networking products, refer to Section 8, "Data Communications Networks."

## Simplified System Management

The A Series provides a number of utilities that streamline many operational procedures, minimizing the time and effort of allocating computer resources, coordinating the work flow, and managing the communications network.

InterPro™ (Interactive Productivity) is the family name for several software facilities that increase the productivity of system administrators, operators, and programmers. The InterPro products follow a common, ergonomically developed format for the user interface, including a choice of menu-driven or command-driven modes and easy access to online help text. You can use menu mode to accomplish tasks without learning any complicated syntax. As you become more experienced, you might prefer to use command mode to enter requests in a compact form.

The InterPro family includes several data management and applications development products introduced earlier in this section: the Advanced Data Dictionary System (ADDS), the Communications Management System (COMS), the Operations Control Manager (OCM), and the Screen Design Facility Plus (SDF Plus). The InterPro versions of ADDS and OCM provide the same functions as the InfoExec versions of these products, except that they run outside the InfoExec environment and can access only DMSII databases. In addition, InterPro includes the following products that aid system operations and configuration:

- Menu-Assisted Resource Control (MARC) is a system operations interface provided by COMS. MARC presents typical operations tasks organized in a functional hierarchy within such categories as file management, job and task control, printing control, memory management, and data comm management.
- The Interactive Datacomm Configurator (IDC) simplifies the process of defining the data comm network by providing predefined protocols and example definitions of lines and stations. These example definitions can be easily duplicated and modified as needed.

---

TIME/LAN is a trademark of Timeplex, Incorporated.

IBM is a registered trademark of International Business Machines Corporation.

InterPro is a trademark of Unisys Corporation.

- Extended Retrieval with Graphic Output (ERGO) provides a menu-driven interface for both querying and updating DMSII databases.

A user-friendly interface is no help if it is not provided in a language that its users can understand. The A Series bridges the language gap by providing the MultiLingual System (MLS). This system makes it possible for you to translate system messages and application program messages into the language of your choice. The Message Translation Utility (MSGTRANS) is available to assist in the translation process. Additionally, the A Series provides features to translate screens and menus. For more information see "Internationalization Functions" in Section 6, "Operational Interface."

Moreover, many of the Unisys International locations have already performed the necessary translations, and can provide you with A Series software that communicates in the language of your choice. Contact your Unisys International marketing representative for details about your area.

For more information about MARC and MLS, refer to Section 6, "Operational Interface." For information about IDC, refer to "Simplified Data Comm Configuration" in Section 8, "Data Communications Networks." For information about ERGO, refer to Section 3, "Database Management."

## Language Translation Capabilities

The A Series now provides support of international character sets and translation of software text through various products and system-wide attributes that can be used on most A Series system software. Important product areas that are capable of supporting international character sets and translation include:

- Master Control Program
- Menu-Assisted Resource Control (MARC)
- Information Executive (InfoExec), including IQF, WQF, OCM, and ADDS.
- Structured Query Language Database (SQLDB)
- Data Management System II (DMSII)
- Screen Design Facility Plus (SDF Plus)
- Compilers for COBOL74, COBOL85, RPG, and C
- Data Communications Products

For more information about internationalization capabilities, refer to "Internationalization Functions" in Section 6, "Operational Interface."

## Section 2

# A Series Architecture

The A Series is a family of systems that offers state-of-the-art technology and the processing power to meet your organization's current and future needs. The A Series family spans a broad range of computing power, and supports a variety of peripherals including terminals, workstations, printers, and data storage devices.

The A Series line of computers developed historically from the B 5000/B 6000/B 7000 series of systems. However, these systems are no longer supported. The following discussions apply only to the A Series line of computers.

## System Hardware Components

The hardware equipment for A Series systems is modular, to allow maximum freedom to change the system configuration. The hardware and software together adjust system operation to efficiently use whatever hardware modules are available. This system-wide adjustment means the hardware configuration can change without requiring that the application software or the operating system be changed or recompiled.

Although object code, operations interfaces, and peripherals are generally compatible across the A Series, there are significant variations in the hardware design of the systems. Some of these differences can be seen by the operator in commands that display the system configuration, and also in the structure of some hardware-related system log entries. Thus it is occasionally useful to distinguish between two main types of A Series systems. The first category, Entry and Medium Systems (EMS), includes the Micro A and the systems numbered A 1 through A 10. The second category, Large Systems (LS), includes the systems numbered A 12 and above.

Within the hardware system, one or more hardware units of a particular type comprise a functional subsystem. An A Series system includes the following functional subsystems:

- Central processor subsystem
- Memory subsystem
- Input/output (I/O) subsystem
- Maintenance subsystem

Figure 2-1 shows the elements that most A Series systems have in common. Only the logical connections between the components are illustrated; the way these components are grouped into cabinets varies from one system to another. Also, while the figure shows only one of each type of component, an actual A Series system might support multiple components of some or most of these types. For example, an A Series system might have more than one I/O base. The exact configuration options depend on which type of A Series system you have.

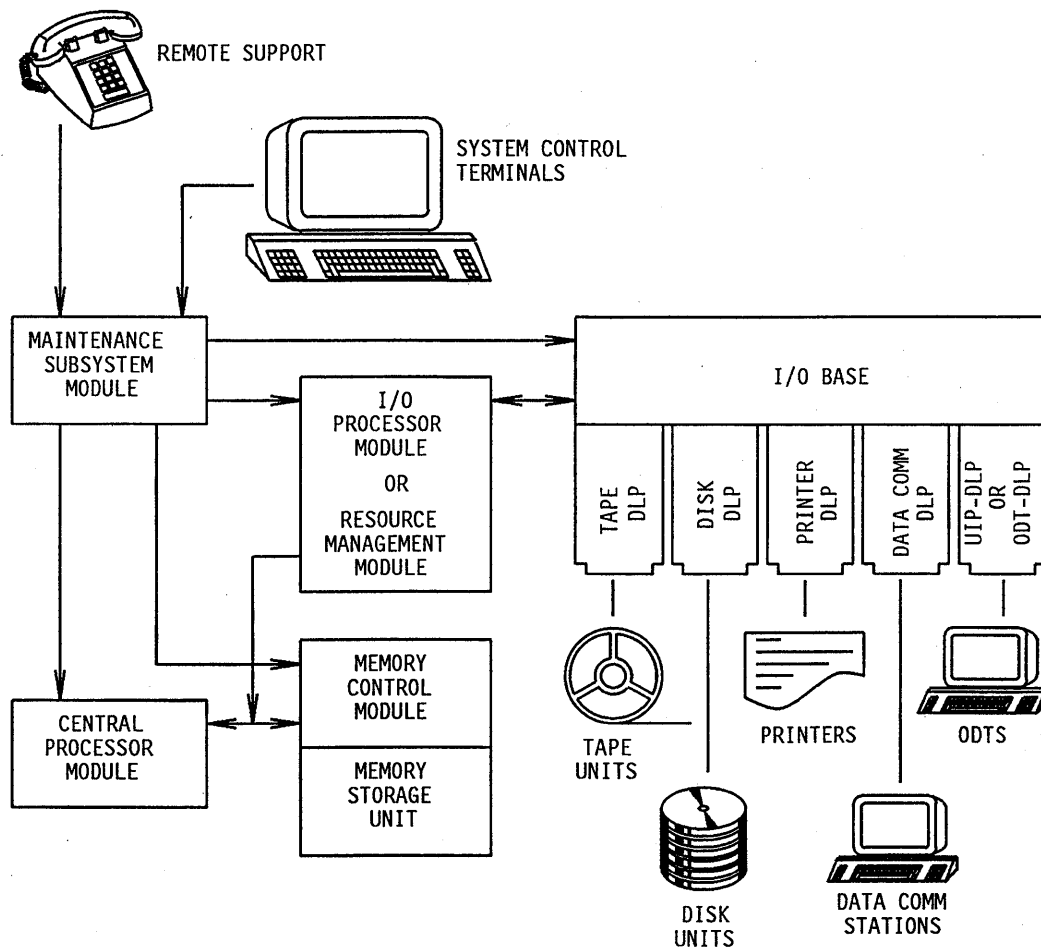


Figure 2-1. A Series Hardware Components

The following pages explain the purpose and responsibilities of each of the main components of an A Series system.

## Central Processor Subsystem

The central processor subsystem consists of one or more central processing units (CPUs) that operate together to execute system and application programs. On Large Systems (LS), a CPU is referred to as a central processing module (CPM).

The CPU incorporates the latest high-speed microchip technology. Widely different CPU designs are used for different A Series systems, to provide the best price/performance ratio at each level. However, the differences in processor design are transparent to application programs.

The following are examples of processor optimizations used in some of the A Series systems:

- *Pipelining.* The CPU is typically divided into several units, serving specific functions such as memory access or arithmetic and logic calculations. The pipelining feature allows individual units within the CPU to begin an instruction sequence before a prior sequence has finished.
- *Concurrency.* This is the ability to use the units within the processor simultaneously rather than one at a time. Concurrency is achieved by distributing operations through the various processor units, allowing execution order to be determined by resource availability rather than by code sequence order. The achievable concurrency is increased by concatenating some very simple processor instructions and by distributing some complex instructions over several units.

The behavior of the CPU is modified by a microcode program, otherwise known as the system microcode. On some A Series models, the microcode controls the processor, while on other models control is shared between the microcode and the hardware. The microcode is implemented in high-level languages and then compiled and provided as object code, which can be loaded from a tape or diskette. The use of microcode makes it possible for the processor to recognize and execute a large and flexible set of program instruction codes. A change in microcode can enable an existing A Series system to recognize newer instruction codes. This flexibility allows users to take advantage of technological improvements in hardware and software design. Architectural enhancements can be retrofitted to existing systems through new microcode releases.

## Memory Subsystem

The memory subsystem provides storage for code and data used by programs as they execute. The memory subsystem is usually divided into a memory storage unit and a memory control unit.

The memory storage unit consists of one or more memory storage boards. You can increase the amount of memory storage either by substituting a different memory storage board or by adding additional memory storage boards, depending on which type of A Series system you have. The memory storage unit features error-detection and correction circuitry, which can correct any single-bit errors and can also detect double-bit errors.

The memory control unit handles all transfers of data between main memory and the central processor subsystem. All the Large Systems (LS) feature a purgeless cache mechanism for improved performance. The purgeless cache allows data to reside in a requestor's cache until it is displaced by more recently needed data or is required by another requestor. This method eliminates the need for general cache purges, thus minimizing the transfers from main memory and maximizing the use of high-speed cache technology.

The memory architecture can address up to 4 billion words (24 billion bytes) of memory and treats all memory as a continuous region that can be allocated in a flexible manner among all the programs currently executing.

### Input/Output (I/O) Subsystem

The input/output (I/O) subsystem manages all transfers of information between the central processor subsystem and peripheral devices. Peripheral devices include disk units, terminals, workstations, and printers.

### MLIP-Based Systems and HDU-Based Systems

I/O operations are initiated by the central processor subsystem, but are immediately passed to an input/output processor (IOP), which is responsible for servicing I/O queues independently of the central processor subsystem. The I/O processors free the central processor for other work and thus increase the effective processing power of the system.

The various types of A Series systems use different types of IOPs. Entry and Medium Systems (EMS) use a single IOP called either a message level interface processor (MLIP) or a host dependent port (HDP). For this reason, Entry and Medium Systems (EMS) were formerly referred to as *MLIP systems*.

Of the Large Systems (LS), the A 12 and A 15 use IOPs called host data units (HDUs). For this reason, Large Systems (LS) were formerly referred to as *HDU systems*. The optional use of multiple HDUs on the A 15 enables this system to handle the higher rate of I/O operations initiated by its powerful central processor subsystem.

On each of these systems, the I/O processor is connected by ports to one or more I/O base units. Each I/O base contains a series of specialized intelligent microprocessors called data link processors (DLPs). The DLPs reduce the burden on the central processor and IOP by performing peripheral dependent functions. A special DLP exists for each type of peripheral subsystem, such as the tape subsystem, disk subsystem, and data comm subsystem. Some DLPs can handle more than one type of subsystem. Also, depending on the subsystem involved, a single DLP might be able to service multiple peripherals of the same type. Figure 2-1 shows an I/O base with various types of DLPs attached.

One device type that deserves particular mention is the operator display terminal (ODT). ODTs are used by operations personnel to monitor and control system operations. An ODT is actually not a special type of terminal, but any type of terminal that is connected to the system in such a way that it can communicate directly with the operating system. On systems numbered A 1 through A 6, the second device connected to the user interface processor (UIP) is treated as an ODT. On systems numbered A 9 and above, devices connected to the system through an operator display terminal data link processor (ODTDLP) are treated as ODTs. The operations interfaces provided by ODTs are discussed in Section 6, "Operational Interface."



### Resource Management Module (RMM) Systems

On A 16 and A 17 systems, the core of the I/O subsystem is a new type of unit called the resource management module (RMM). The RMM improves system throughput by executing at the hardware level a number of functions that formerly resided in the operating system and were executed by the central processing unit (CPU). The major components of the RMM and their functions are the following:

- The IOP executes at the hardware level many I/O management functions that formerly resided in the operating system and were executed by the CPU.
- The port adapters provide the interface between the IOP and I/O base units or Intelligent Peripheral Interface (IPI) channels (discussed later under this heading).
- The task control processor (TCP) executes, at the hardware level, those process-switching functions that formerly resided in the operating system and were executed by the CPU.
- The data transfer unit (DTU) is an optional component of the RMM that implements a global disk caching mechanism at the hardware level. (Disk caches are discussed under "Disk I/O Caches" in Section 7.)

Like other A Series systems, the A 16 and A 17 systems allow you to attach I/O bases that can contain DLPs supporting a variety of peripheral types. However, the A 17 also provides another type of I/O path that supports an industry-standard peripheral interface called the Intelligent Peripheral Interface (IPI). This path consists of an IPI channel that leads directly from the RMM to an IPI controller, such as the 9399-H Dual String Controller. This type of I/O path supports a substantially faster data transfer rate than traditional paths using I/O bases and DLPs.

### Peripherals

Unisys sells and supports a full line of peripheral devices for A Series systems, ranging from economical entry-level devices to the latest in high-capacity storage and communication media. The vast majority of these peripherals are compatible across the entire A Series product line. Because of the frequent additions to the A Series peripheral family, this overview does not attempt to list them. Contact your Unisys marketing representative for details about the peripherals that are available.

### Maintenance Subsystem

The maintenance subsystem includes a processor and firmware that can perform hardware diagnostic tests, initialize the system, and serve as an interface for operator commands. The maintenance subsystem is also sometimes referred to as the system control subsystem (SCS).

One or more of the terminals or workstations attached to an A Series system are designated as system control terminals (SCTs). An SCT is connected to the system in such a way that it can communicate directly with the maintenance processor. The SCT communicates with the central processor subsystem, I/O subsystem, and memory subsystem through the maintenance subsystem. By contrast, other terminals communicate with the system directly through data link processors (DLPs) and IOPs.

## A Series Architecture

---

On systems numbered A 1 through A 6, a terminal attains SCT status if it is the first device you connect to a special data link processor (DLP) called the UIP.

On systems numbered A 9 and above, the SCT is part of a workstation whose processor handles part or all of the maintenance subsystem logic. Some of these systems supply two or more of these workstations, each of which includes a single SCT. On A 9 and A 10 systems, this workstation is known as the system control processor (SCP). On systems numbered A 12 and above, this workstation is known as the System Maintenance Station (SMS I, SMS II, or SMS III).

An SCT can operate in either *maintenance mode* or *ODT mode*. The operator can use a simple command or function key to cause the SCT to change from one of these modes to another.

Most of the time, the SCT is used in ODT mode. In this mode, the SCT provides all the capabilities of an operator display terminal (ODT). The operations interfaces provided by ODTs are discussed in Section 6, "Operational Interface."

By changing the SCT to maintenance mode, the operator can access menus for running hardware diagnostic programs or for initializing the system.

All A Series maintenance subsystems include a remote support interface. The remote support interface provides a phone modem link to a Unisys Remote Support Center. Unisys support personnel can use this interface to perform a variety of diagnostic tests. This method can usually isolate failures to a replaceable hardware module. A Unisys engineer can then quickly replace the defective module. The use of remote diagnostics reduces the cost of system maintenance and provides you with a faster response to any hardware problems.

A Series systems provide various methods you can use to enable or disable the remote support interface, so that your system can never be accessed without your permission. For example, some systems provide an additional SCT mode called *remote support mode*, which you can use to enable or disable the remote support interface.

### Documentation

The following manuals provide information about the system control terminal (SCT) modes and the use of maintenance mode menus for each system:

- *A 12 System Operating Guide*
- *A 15 System Operating Guide*
- *A 16 System Operations and Programming Reference Manual*
- *A 17 System Console Operations Guide*
- *A Series A 1–A 6 Systems Software Installation Guide*
- *A Series Micro A System Installation Guide*
- *A Series System Software Installation Guide, Volume 1: A 9*
- *A Series System Software Installation Guide, Volume 1: A 10*

## System Software Components

The software provided for A Series systems falls into three major categories:

- Application software, which enables end users to accomplish specific tasks such as accounting, inventory, document processing, and so on. The system can run a variety of different application programs, but none of them is a necessary part of the system. You can use Unisys applications, purchase applications from third party software developers, or develop applications of your own.
- Environmental software, which provides a productive environment for creating applications programs and creating, operating, and querying databases. Environmental software products are not necessary for system operations, but you need at least some of them if you create application programs or use databases. Environmental software is provided by Unisys; there is no need for you to develop or modify environmental software.
- System software, which performs overall control of the system and all programs running on the system. System software also includes utilities that are essential for day-to-day system operations, as well as compilers. System software appropriate to your system model is provided by Unisys.

Applications software is beyond the scope of this overview. Environmental software products are discussed throughout this overview, most particularly in Section 3, "Database Management"; Section 4, "Transaction Management"; and Section 5, "Program Development." System software products are discussed throughout, most especially in Section 7, "Basic System Management"; Section 8, "Data Communications Networks"; and Section 9, "System Monitoring and Security."

The most essential system software products are marketed together under the title of the System Software Facility (SSF) and are required. Other system software is optional and can be ordered at any time when an installation needs the special functions they perform.

The SSF product contents vary from system to system. The following paragraphs introduce products that are included in the SSF for most systems.

### Master Control Program/Advanced Systems (MCP/AS)

The MCP/AS is the main component of the operating system. The MCP/AS manages the execution of all programs on the system, dividing processor and memory resources as needed. The MCP/AS also manages system peripherals and provides logging of errors and system utilization. And the MCP/AS provides Print System (PrintS) features, which allow you to control the output of files to printing devices such as image printers and line printers.

Further details about the functions of the MCP/AS are given under "Operating System Concepts" in this section.

### Message Control Systems (MCSs)

Message control systems (MCSs) generally control the user interface that is presented to end users at their terminals. MCSs typically provide the user with a selection of commands for manipulating files or running application programs. In addition, an MCS controls the flow of messages between end users, application programs, and the operating system.

The Communications Management System (COMS) is the most important MCS on A Series systems. COMS allows users to access the system in several ways at once, by using multiple windows and switching between them at will. COMS also allows the system administrator to control the types of applications that can be run by each user, and to modify the user interface presented to each user by an application program.

COMS also provides a transaction processor called Menu-Assisted Resource Control (MARC). This is a menu-driven interface to system operations functions, which helps operators to perform operations functions even when they are not familiar with system commands.

Another important MCS is Command and Edit (CANDE). CANDE provides a flexible environment for manipulating files and creating or executing application programs.

### Remote Print System (ReprintS)

ReprintS allows you to print output at printers that are attached to data comm lines. ReprintS also allows you to direct printed output to another host system in a BNA network.

### System Utilities

The system utilities component of the SSF package can include the following items, among others:

- **CONFIGURATOR**, which you can use to process changes to the system configuration.
- **DCSTATUS**, which you can use to determine the current status of the data comm network.
- **DUMPALL**, which you can use to copy files from one medium to another.
- **FILECOPY**, which you can use to automate the copying of large numbers of files.
- **FILEDATA**, which you can use to create selected types of reports about files.
- **LOGANALYZER**, which you can use to analyze the system log.
- **MAKEUSER**, which you can use to create or modify usercode definitions.

## Language Compilers

The programming language compilers translate the source programs written in high-level languages into machine-language programs (object code) that can be read and executed by the hardware. If you develop any application programs at your installation, you will need a compiler for each language you use for your applications. Additionally, there are certain compilers that should be available at any A Series installation.

The system software components are written in the following languages: ALGOL, DCALGOL, NEWP, and Pascal. If you license the source code for the system software, then you can use these compilers to integrate fixes supplied by Unisys into the system software.

Another important language is the Work Flow Language (WFL), which enables you to write jobs that compile or run application programs. WFL jobs pass through the job queue mechanism, which provides orderly scheduling and workload management. WFL jobs are compiled by a system library called WFLSUPPORT.

## Programming Utilities

The SSF typically includes several utilities that aid in application program development. These include the following:

- The Binder, which is a utility that allows you to combine separately compiled subprograms to produce an executable program. Binding is a convenient method of reusing program modules in separate programs, or of combining program modules written in different languages.
- The SORT Utility, which allows you to write compact but powerful sorting applications.

## Operating System Concepts

In 1961, the B 5000 computer system was introduced. Although it employed the stored logic program principle of the Von Neumann architecture, it accomplished many system objectives that were previously unattainable, including use of a software operating system, exclusive use of high-level languages, multiprogramming, multiprocessing, virtual memory, reentrant code, and other advanced architectural techniques.

These objectives were achieved by introducing new concepts radically different from what had become conventional architecture. The new architecture was completely stack-based, rather than register-based or mixed. Hardware structures were used to distinguish data from code and to eliminate the need to modify code. The result was the first multiprogramming, multiprocessing, virtual memory computer system.

This architecture has been developed and extended through successive generations of B 5000, B 6000, and B 7000 Series products, and also forms the basis for A Series design. Although other vendors still base their product design upon the conventional Von Neumann architecture, many of the features introduced by the B 5000 have subsequently been adopted by them. However, other vendors have had to adapt ideas

## A Series Architecture

---

such as multiprogramming and virtual memory to systems that were not originally designed for them. By contrast, these features are inherent in A Series system architecture.

The following pages describe some of the more significant features of the A Series systems.

### Compiler-Oriented Hardware

When the B 5000 system was first being developed, high-level languages such as COBOL, FORTRAN, and ALGOL had recently been developed and were just coming into wider use. These languages attempted to mimic the human problem-solving process rather than the structure of the computer hardware. A single statement in one of these high-level languages might correspond to 10 or more assembly language statements. This meant that programs written in these languages were 10 times shorter than assembly language programs, 10 times quicker to write, and 10 times easier to maintain. For this reason, the B 5000 designers decided to create a system that would make it practical to program exclusively in high-level languages.

With this idea in mind, the B 5000 designers provided their processor with the ability to recognize and execute instructions that corresponded to functions in high-level languages, such as COBOL, FORTRAN, and so on. These hardware innovations made it possible for the B 5000 to provide excellent performance for high-level language programs. This principle has been continued in the A Series systems. Because of compiler-oriented hardware, an assembler has never been needed or provided on A Series systems. Even the operating system is written in a high-level language.

Aside from making application programming easier, this emphasis on high-level languages has made it easier for Unisys to introduce innovations in the hardware architecture. The compilers can translate any application program into the processor instructions that are currently defined for the system. The result is that you receive the benefits of architectural innovations without any need to modify your source programs.

### Stack Architecture

A single execution of a program, from program initiation to program termination, is referred to as a *process*. In the course of executing a process, the system must allocate a number of temporary storage locations, in which the current values of program variables are kept.

In the traditional Von Neumann system architecture, the compiler is free to assign temporary storage locations from a number of general purpose process registers, as well as from memory locations scattered anywhere throughout memory. By contrast, the B 5000 introduced an innovative *stack architecture* to organize the general purpose work area for a process. In brief, the system allocates a contiguous region of memory as a work area for the process. This region is used as a last-in/first-out stack, called the *process stack*.

Each time a procedure is entered, the system creates an environment called an *activation record* for that procedure on top of the stack, and updates a register to point

to the base of this activation record. References to variables are handled through a relative addressing scheme that makes it simple for a procedure to access a different set of parameters and local variables each time it is invoked. This technique greatly increases the performance of programs with repeated procedure calls, and is ideally suited for recursive programming techniques, multiprogramming, and code reentrancy. (The latter two concepts are discussed under "Multiprogramming and Multiprocessing" and "Reentrant Code" in this section.)

The operands for any computations performed by the process are built on top of the stack. After each computation, the stack is decremented if necessary and the result is placed on top of the stack. Through the use of reverse Polish notation, which defines specific rules for the sequence in which operators and operands occur, the last-in/first-out mechanism significantly reduces the number of instructions required by other systems, to store intermediate results for a calculation. This method allows computations to be handled with maximum efficiency. To further increase the efficiency of calculations, the system stores the values needed for the calculation in hardware registers that provide rapid access.

The stack architecture also serves to simplify and speed the debugging process for applications programmers. A dump of the process stack makes it instantly clear which procedures have been entered, and in what order. Such a dump also provides the names and current values of all variables used by the process.

## Multiprogramming and Multiprocessing

*Multiprogramming* is the ability of the system to execute many different programs at the same time. The A Series hardware and software is designed to promote multiprogramming as the normal mode of operation. Each processor executes only one process at a time, but divides its time among all the processes on the system. The operating system maintains a list, called the *ready queue*, of all processes that are waiting for a processor, arranged in priority order.

A processor continues executing a particular process until one of three things happens: the process reaches a natural stopping point (for example, because it is waiting for an I/O to complete), a higher-priority process appears in the ready queue, or the process exceeds its time slice and a process of equal priority is present in the ready queue. The processor then pulls the highest-priority process from the ready queue and begins executing it. This design allows you to bias the system in favor of the processes you consider most urgent, and also ensures the most efficient use of the processor.

The process stack is optimized to store and restore process state information in a fast, efficient, and well-ordered manner. This design permits the processor to be switched from executing one process to executing other processes, and back again, without incurring the overhead that is typical of most competitive systems. A Series systems are therefore well suited for environments that require rapid process switching, including online transaction processing, timesharing, and multiprogramming environments.

*Multiprocessing* is the use of two or more central processing units in the same computing system. A Series multiple processors are equal in status, so none of the processors incurs the overhead of having to manage a master-slave multiprocessor arrangement. Use of multiple processors is flexible because each processor can access all

## A Series Architecture

---

of memory, and a process can be executed by any processor that happens to be available at the moment. This flexibility allows the most efficient use of processor resources.

On A 17 systems, the efficiency of multiprogramming and multiprocessing is further improved by a new hardware component called the task control processor (TCP). The TCP executes at the hardware level those process-switching functions that formerly resided in the operating system and were executed by the central processing unit (CPU).

## Libraries

A library is a program that provides a set of useful routines that can be shared by many other programs. The use of libraries makes it unnecessary for the programmer to duplicate code in all the programs that need to use a common type of routine.

The stack architecture of A Series systems lends itself naturally to the implementation of a dynamic library mechanism. This library mechanism is notable for its flexibility. Because the library is independent of the user program, linkage can be made at run time as needed, and multiple user programs can share the same library and communicate with each other through library procedure calls. The library mechanism is much used in the system software and is also available in most A Series application languages, such as ALGOL, COBOL74, COBOL85, FORTRAN77, and Pascal.

## Automatic Code Segmentation

High-level language compilers automatically divide the object code into variable-length segments based on the logical structure of the source program. Other vendors' systems typically divide the object code into fixed-sized pages. The A Series approach makes it unnecessary for the programmer to be concerned about segment or page sizes when writing a program, thus increasing the programmer's productivity. Variable segmentation also assures that a maximum number of segments can reside in main memory, making more efficient use of the memory resources.

## Reentrant Code

Another of the radical innovations of B 5000 design was the decision to disallow changes to the program code during execution. Previous theory had held that programmers could optimize memory usage by writing a program that modified its own code in memory, but practice had demonstrated that it is almost impossible to write reliable self-modifying code.

The principle of nonmodifiable code is maintained today by the A Series systems. The operating system uses two strategies to ensure that self-modifying code is not introduced into the system. First, because no assembler is provided, all code is generated by the compilers. The compilers do not create any self-modifying code. Second, the system marks each word in memory with a *tag field*, which identifies whether the word is related to code or data. This design protects program code from accidental corruption. More importantly, it makes possible the use of *reentrant code*, meaning that the same code segments can be used by more than one process.



Reentrant code design makes it possible for multiple users to initiate the same copy of a program and have all of their processes execute simultaneously. This design also conserves memory by allowing simultaneous executions of the same program to use the same code segments in memory.

More importantly, this design allows the code segments for the operating system to be shared by all the processes on the system. This is important because most of the work done by any process is accomplished through implicit calls on operating system procedures. On other vendors' systems, the code for these procedures typically has to be duplicated for each process, thus requiring much more memory.

Reentrant code is an inherent characteristic of A Series systems. It is language-independent and occurs automatically without any need for special programming.

### Descriptors

Special data words called, *descriptors*, contain the memory addresses of code segments and data segments. For data structures more than a few memory words long, the process stack stores a data descriptor instead of the data itself. The system stores addressing information in the descriptor. Because the descriptor is in the stack and contains the addressing information, the program can refer to the data using the same stack relative addressing it uses for all other data. Pipelining techniques allow the CPU to resolve the physical addresses of code or data segments before they are actually needed for a computation. (Pipelining is discussed under "Central Processor Subsystem" in this section.)

Code segments are never stored in the process stack, but can be scattered throughout memory. A separate stack, called the *code segment dictionary*, stores descriptors for all the code segments used by a process. Multiple executions of the same program can share the same segment dictionary.

The use of descriptors makes it possible for the system to allocate memory in a flexible manner among processes and move code or data segments around in memory as needed. For example, the system does not need to allocate memory space for code or data segments until the process actually attempts to use those segments. Further, the use of descriptors allows applications to dynamically increase the size of arrays they are using. The system can fulfill such requests through a simple change to the array descriptor.

### Virtual Memory

Virtual memory is an A Series feature that allows the system to simulate the existence of far more memory than physically exists on the system, thus allowing the efficient execution of more processes and more memory-intensive processes than would otherwise be possible. To achieve virtual memory, the system copies selected data segments out to an overlay disk file when those segments are not in use, and copies those segments back into main memory when they are needed again.

Because the code segments for a program are always available in the code file on disk, there is no need to copy code segments to the overlay file. They can simply be

## A Series Architecture

---

overwritten by other data as needed and copied back in from the code file when they are needed again. A similar optimization is used for arrays that have not been modified since the last time they were copied in from the overlay disk file. Because the data has not changed, the system can overwrite the array as needed without copying it out to the overlay disk file again.

### Actual Segment Descriptor (ASD) Memory Architecture

The actual segment descriptor (ASD) memory architecture is the latest development in A Series memory management and provides substantially greater memory addressing ability than earlier memory architectures, together with much greater ease of use. ASD memory is the memory architecture used by the A Series operating system.

ASD memory architecture treats all memory as a single resource that can be flexibly allocated among all the processes running on the system. ASD memory uses an indirect addressing mechanism to overcome the limitation on the size of the address field in a descriptor.

In ASD memory architecture, the address field in a code or data descriptor points to an entry in a structure called the ASD table. The specified ASD table entry stores the physical address of the code or data segment. The advantage of this mechanism is that the ASD table entries have 32-bit address fields, far larger than the 20-bit address field in a descriptor, which enable them to store a far greater range of physical memory addresses (more than 4 billion words or 24 billion bytes).

ASD memory architecture also includes innovations to improve system performance. Stack searches are unneeded on ASD systems because only the ASD table entry for a segment being transferred to disk must be updated; all descriptors for that segment simply continue to point to the ASD table entry, rather than to the segment itself.

Additionally, ASD memory introduces a new second-chance overlay mechanism, which causes the system to overlay the segments used least frequently. This algorithm improves system performance by reducing the number of overlays that are needed to maintain adequate available memory.

As noted under "Memory Subsystem" in this section, all A Series systems are now configured for ASD memory. Previously, the A Series supported Address Space Number (ASN) memory, a memory architecture that allowed memory to be divided into separate sections that could be manipulated to allow for greater memory usage. Migration from ASN to ASD memory is a simple process. All your application programs can run under ASD memory without any need for conversions or recompilations.

### Configuration-Independent Software

The operating system maintains tables that reflect the presence and status of all system resources, including processors, memory, I/O paths, and peripherals. The use of these tables to define system resources allows installations with completely different configurations to be represented easily by simple entries in the appropriate table, rather than by alterations in the code that makes up the operating system. Unlike competitive systems, a time-consuming system generation is never required to reflect

any configuration changes that might occur. Hence, a configuration can have resources added or removed without any effect on the operating system or applications software. Such changes can even be made without interrupting system operations.

The operating system also makes physical device control transparent to the programmer and end user. As far as a programmer is concerned, all devices are read from and written to in the same way. All files are dynamically assigned to devices by file attributes such as the file title. For example, there can be a family of two disk units called PROD. The system administrator might decide to add extra disk units to the PROD family, and the change would not affect any application programs, which merely specify that a file is located on the PROD family.

## Growth Paths

The most unique benefit of the A Series is the extremely broad range of processing power offered by this family of compatible systems. This theme was introduced under "System Performance Range" in Section 1. The following paragraphs provide more detail about performance comparisons on the A Series, the methods of upgrading systems, and the types of compatibility that are offered.

## Performance Measurements

Performance is a measure of how much work a system can do in a given period of time. Performance, along with cost, is one of the most important factors to consider when you are comparing systems offered by various vendors. However, performance can be measured in many different ways, and while some of these measures are helpful, others can be deceptive.

Part of the problem is that performance has several different aspects that depend on your point of view. An end user might perceive performance in terms of how long the system takes to complete a single transaction and deliver a response. This is called *response time*. A system administrator might perceive performance in terms of how many transactions the system can handle in an hour. This is called *throughput*. Another valid measure is the number of transactions the system can handle in an hour while still providing reasonably short response times for each transaction. This is one way of quantifying the system *capacity*.

Response time, throughput, and capacity can all be measured for a particular system running a particular set of applications. However, these are not really general measurements of system performance, because transactions and application programs vary substantially in complexity. It would be helpful if there were a simple, universal statistic that could be used to compare the performance of any two systems.

One statistic that is widely used for that purpose is *millions of instructions per second (MIPS)*. This is a measurement of the total number of instructions the central processor can execute in one second. Unfortunately, there is no direct relationship between the number of instructions executed and the amount of work performed for you.

A Series system architecture in fact uses a number of innovations to minimize the number of processor instructions needed to execute a program. Because of the A Series

## A Series Architecture

---

emphasis on high-level languages, the processor was designed to accept a large variety of instructions that perform relatively high-level functions. The processor microcode translates each of these instructions into a large number of individual steps; thus, each instruction accomplishes a lot of work for you. Other vendors may provide less powerful instructions and have to execute them at a faster rate to provide comparable performance.

Another point to bear in mind is that the A Series distributes much of the work of the central processor onto subsidiary processors, such as the I/O processor (IOP), the data link processors (DLPs), and (on some systems) the task control processor (TCP). These subsidiary processors run in parallel with the central processor. The result is that total system throughput is far greater than the central processor could provide by itself. It should be clear, therefore, that MIPs do not provide an adequate approximation for comparing A Series performance with that of other types of systems.

The most useful way to compare performance is by running the same set of application programs (called *benchmarks*) on a variety of different systems. You can then assign ratings to the systems based on the capacity demonstrated by each system when running these benchmarks. Unisys has used this method to establish a rating called a relative performance measurement (RPM) for each A Series system. The RPM rating is proportional to the performance of the system. For example, a system with an RPM of 100 has twice the performance of a system with an RPM of 50.

The RPM statistic has two great advantages. First, it relates to the task accomplished for you, rather than to the amount of work the system must do internally to accomplish the task. Secondly, it provides a fair comparison of systems, because the same benchmark programs are run on each A Series system.

On the other hand, a performance comparison is useful only if it is based on a standard of comparison that is familiar to you. Many mainframe vendors establish performance ratings relative to the IBM 370/158-3. In the RPM rating system, the IBM 370/158-3 has an RPM rating of 45. Table 2-1 shows the RPM ratings of A Series systems.

Table 2-1 also includes the *growth factor* ratings that were introduced under "System Performance Range" in Section 1. The growth factor ratings are based on the same performance measurements as the RPM ratings. The difference is that the growth factor for each system is based on a comparison with the A 1-F, which is assigned a growth factor of 1.

Table 2-1 shows all the A Series systems that are supported by the Mark 3.9.0 software release. Your Unisys marketing representative can provide you with an updated list of the available systems.

Table 2-1. Relative Performance Measurements (RPM)

System	RPM	Growth Factor
A 1-F	25	1.0
Micro A825		
Model D	33	1.2
Model F	50	
A 1-FX	40	1.6
A 4-FS	60	2.4
A 6-FS	80	3.2
Model S		
A 10-DX	110	4.4
A 6-HS	150	6.0
A 6-KS	150	6.0
A 10-FX	160	6.6
A 12-211	230	9.2
A 6-NS	280	11.2
A 10-HX	295	11.8
A 12-311	340	13.6
A 12-411	500	20.0
A 12-511	750	30.0
A 16-422	650	26.0
A 16-511	650	26.0
A 16-611	1200	48.0
A 16-522	1235	49.4
A 16-622	2280	91.2
A 17-F	1000	40.0
A 17-H	1900	76.0
A 17-J	1900	76.0
A 17-L	2750	110.0
A 17-N	3600	144.0

When referring to performance ratings of any kind, including RPMs, bear in mind that system performance varies depending on the type of work you do (commercial or scientific, online or batch), the choice of languages for your applications (COBOL, FORTRAN, and so on), and the exact configuration of the system, as well as a host of other variables. RPMs form a rough standard of comparison but are not presented as a guarantee that your applications will run on an A Series system at any particular

## A Series Architecture

---

speed. Your Unisys marketing representative can help you develop an estimate of the performance that various A Series systems can provide with your workload.

Another point to remember is that even the performance ratio of 1 to 144 for the A 1-F to the A 17-N does not express the full growth potential of the A Series. BNA architecture allows you to distribute your workload across a network of A Series systems, while facilitating data transfer among these systems. The ability to upgrade individual systems or add new systems to the network ensures that you will never outgrow the A Series line.

## Hardware Compatibility

The modular design of A Series systems makes it easy to upgrade to larger systems with a minimum of disruption to your work. Many upgrades can be accomplished on your site through the addition or replacement of particular components in your existing system cabinet. Hardware upgrades fall into two categories: mainframe upgrades and I/O subsystem upgrades.

### Mainframe Upgrades

The A Series line can be divided into several subgroups of models based on the types of field upgrades that are possible. Within each of these subgroups, any model can be upgraded to any other model in the field without changing the system cabinet. The subgroups currently being manufactured are as follows:

- Micro A
- A 1, A 4, A 6
- A 12, A 17

Customer support engineers accomplish these upgrades by applying one or more hardware kits. These kits usually involve the addition or replacement of cards in the system cabinet, including central processor, I/O processor, or memory units. Kits can also include microcode changes. Table 2-2 lists the models belonging to each subgroup and outlines the various types of units that are included in each system.

Table 2-2 includes only models that are currently being manufactured, though many additional models are still supported on the Mark 3.9 system software release. Of these additional models, A 15 models can be upgraded on site to any A 17 model. The A 2, A 3, and A 5 systems comprise another field-upgradable subgroup that is no longer manufactured.

It is important to realize that even upgrades from one subgroup to another are simple and easy to accomplish, because you can retain many of the same peripheral units. Only the processor cabinet and/or the I/O cabinet need be replaced.

### I/O Subsystem Upgrades

I/O subsystem configurations on A Series systems are flexible enough to provide you with a variety of different methods for improving I/O performance, including the following:

- You can add an additional IOP to many of the Large Systems (LS), as discussed under “Mainframe Upgrades” in this section.
- You can add additional DLPs so that each DLP has a smaller number of peripheral units to manage.
- You can upgrade peripheral units, such as disk units, to take advantage of the latest technologies. Most A Series systems support several models of disk drives, offering a greater than 1-to-15 range of storage capacity and a greater than 1-to-4 range of transfer speed. Your Unisys marketing representative can provide a detailed list of the disk drives that are available for each system.

**Table 2-2. A Series Upgrade Families**

Subgroup	System	Central Processors	I/O Processors	Memory Controls	Memory Mega-bytes
Micro A (30 RPM)	Micro A 800	1	1	1	12
	Micro A 825O	1	1	1	12
	Micro A 825F	1	1	1	12
A 1, A 4, A 6 (25-150 RPM)	A 1-F	1	1	1	12-48
	A 1-FX	1	1	1	12-48
	A 4-FS	1	1	1	12-48
	A 6-FS	1	1	1	12-48
	A 6-KS/HS	2	1	2	24-96
	A 6-NS	1	1	1	12-24
	A 12, A 16, A 17 (190-3600 RPM)	A 12-B	1	1	1
A 12-E		1	1	1	24-72
A 12		1	1	1	24-144
A 12-T		1	1	1	24-144
A 17-F		1	1	1	48-288
A 17-H		2	1	1	48-288
A 17-J		2	2	2	48-288†
A 17-L		3	2	2	48-288†
A 17-N	4	2	2	48-288†	

† The A 17-J, A 17-L, and A 17-N are partitionable and provide between 48 and 288 megabytes per partition.

## Operating System Compatibility

The operating systems for all A Series systems are compiled from the same source file. Compile-time options are used to optimize the operating system for use on particular system models. These options tailor the operating system to the hardware features of each model, thus improving performance and reducing the size of the MCP/AS code file. However, these options do not cause any changes in the operations interface. Moreover, these options do not limit your ability to run the same application on different A Series systems; refer to the following discussion of "Application Compatibility" for details.



Note that you can change the configuration of any A Series system without having to recompile the operating system. Refer to “Configuration-Independent Software” in this section.

### Application Compatibility

The compilers provide an option called TARGET, which allows you to specify the types of A Series systems on which you plan to run the compiled program. If you wish, you can specify a TARGET value that creates a program that can be run on any A Series system. Alternatively, you can specify a TARGET option that optimizes the program to make most efficient use of the processor on a particular A Series system (such as the A 6). Between these extremes are TARGET values that optimize the program for use on related groups of A Series systems.

As the A Series product line continues to expand, new hardware innovations will continue to be made and new TARGET values will be defined. It is likely that some new TARGET options will produce programs compatible only with the newest A Series systems. However, you will always have the choice of using a more general TARGET option that produces programs compatible with all A Series systems supported by the current software release. (For a discussion of compatibility across software releases, refer to “Progression Policy” in Section 11.)

### Operations Compatibility

The vast majority of system commands are identical in syntax and function on all A Series systems. Some exceptions are those commands that directly display or modify the hardware configuration of the system. The syntax of these commands is slightly extended for the larger systems, to reflect the presence of additional hardware units on those systems.

You have the option of performing operations tasks through Menu-Assisted Resource Control (MARC) menus rather than system commands. MARC menus are filtered to present only those command options that are relevant to the system you are using.



# Section 3

## Database Management

A *database* is a collection of data files and related software that is designed to provide rapid and reliable access to a large body of information. A database management system (DBMS) enables you to do the following:

- Separate the definition of data structures from the data itself.
- Keep data definitions in a centralized place where they can be maintained consistently with a high degree of integrity.
- Coordinate concurrent access of the database so that transactions do not interfere with each other.
- Ensure that data is recovered when transactions are interrupted by a system failure or program fault.
- Enhance data security.

The A Series systems support three types of databases:

- *Data Management System II (DMSII) databases (network/hierarchical models).* DMSII is a simple yet flexible system that is based on network and hierarchical data models.
- *Structured Query Language Databases (SQLDB) (relational model).* SQLDB is a database product based on the relational database model that is specified in the American National Standards Institute (ANSI) standard X3.135-1986. Because it is an ANSI standard implementation, it supports open systems application development.
- *Semantic Information Manager (SIM) databases (semantic/object-oriented models).* SIM is a data management system based on the semantic and object-oriented data models.

The A Series provides several DBMSs, or environments, to support these databases. The DMSII environment consists of a variety of utilities, each of which handles a different aspect of DMSII, SIM, or SQLDB database management. The InfoExec environment consists of utilities that support SIM databases and takes advantage of the DMSII environment to manage DMSII and SQLDB databases.

A Series fourth-generation programming environments provide the means to define and manage application systems, which often include databases. These environments are discussed under "Fourth-Generation Program Development" in Section 5.

This section describes the data organization capabilities of SQL, DMSII, and SIM databases, and describes the tools that provide each database management function in the InfoExec and DMSII environments.

# Database Management Environments

The InfoExec and DMSII environments provide two different methods of handling all aspects of database management. The functions provided by these environments include database creation, reorganization, integrity checking, structure analysis, performance monitoring, auditing, backup, recovery, querying, and updating. These functions can be accomplished through either command-oriented interfaces or menu-driven interfaces with convenient online help text.

More than one query and updating product is available in each environment. Each of these products provide slightly different functions. For example, the InfoExec Interactive Query Facility (IQF) and the InfoExec Workstation Query Facility (WQF) provide the same basic function. However, WQF uses workstation features to enhance the interface to this function.

In both the InfoExec and DMSII environments, you have a choice between accessing these utilities directly with commands or through the Operations Control Manager (OCM) interface. The OCM program provides a menu-driven interface to database utilities. The underlying utilities in the InfoExec and DMSII environments are packaged separately from the OCM product itself. If you do not acquire all the utilities, you can still use OCM as an interface to the utilities you do have.

The Advanced Data Dictionary System (ADDS) provides the means to create, modify, generate, maintain, and store database descriptions and application program descriptions for COBOL74, ALGOL, and Pascal languages, and to perform essential data administration tasks for both DMSII and InfoExec environments. ADDS is an optional product in the DMSII environment, but is required in the InfoExec environment.

The InfoExec environment uses the core DMSII environment as its underlying base. The InfoExec environment also includes advanced utilities such as DMS.View, LINC.View, and the InfoExec menu-driven interface, which serves as the entry point to these and the optional InfoExec products.

There are some basic factors to consider when choosing between the InfoExec and DMSII environments. The first point to consider is whether you use SIM databases, DMSII databases, SQLDB databases, or all three types of databases. If you use SIM databases, or a mixture of SIM and one or more of the other two types of database, you must have the InfoExec environment. Only the InfoExec environment provides support for the semantic data model.

If you use DMSII or SQLDB databases only, you can choose between the DMSII and InfoExec environments. The DMSII environment provides a complete set of tools for managing DMSII and SQLDB databases. By acquiring the InfoExec environment, you gain the following additional advantages:

- Access to the advanced querying capabilities of IQF and WQF, which are not available outside the InfoExec environment.
- Ability to use SIM host language interfaces to query your DMSII database.
- Ability to begin developing new databases in SIM, or to selectively convert existing DMSII databases to SIM, while still providing support for DMSII databases.

- Coexistence with all utilities in the DMSII environment. For example, you could continue to use ERGO after integrating your DMSII database into the InfoExec environment.

Perhaps you use DMSII databases only, and do not wish to acquire the InfoExec environment. In this case, you can still obtain many of the advantages of the InfoExec environment by acquiring InterPro ADDS and InterPro OCM. The InterPro versions of these products provide the same ease-of-use benefits as the InfoExec versions do.

The remainder of this section discusses the functions provided by the various components of the InfoExec and DMSII environments.

### Documentation

- *A Series Data Management Functional Overview*
- *A Series DMSII Utilities Operations Guide*
- *A Series DMSII Technical Overview*
- *A Series InfoExec Capabilities Manual*
- *A Series InfoExec Semantic Information Manager (SIM) Technical Overview*

## Structured Query Language Database (SQLDB)

Structured Query Language Database (SQLDB) provides an entry into the open systems approach to systems and applications development. SQLDB is a powerful, yet easily understandable database management system that takes advantage of the relational database model. A Series SQLDB offers a number of distinct advantages over other implementations of the ANSI standard:

- Strict adherence to ANSI standards, promoting open systems development
- Support from proven DMSII features and utilities
- Flexible data definition utility
- Powerful data definition language (SQL DDL) and data manipulation language (SQL DML)
- COBOL85 and ALGOL programmatic interface
- Library entry points for FORTRAN, Pascal, COBOL74, COBOL85, and RPG

### SQLDB Database Structure

SQLDB conforms to the ANSI standard for relational databases set forth in ANSI standard X3.135-1986. This standard builds and improves upon the model originally proposed by E.F. Codd in the 1970s. The relational model, based on the structure and operators of relational algebra, is tabular in nature. You define your database in terms of *domains*, which are data elements of the same type, and *tuples*, which are separate data elements that are related. Domains roughly correspond to columns in a table and tuples correspond to the rows in that table.

For example, in a database of employee information, a tuple, or row, could contain pertinent data about a particular employee, such as name, employee number, and the employee's current project. A domain, or column, could represent a particular characteristic, such as employee name or employee number for each of the employees. Thus, the domain refers to all data items that fit into a particular category. The row or tuple refers to a particular instance of related data items.

By logically joining two or more of these tables by means of a query, you can derive data from the database in a variety of ways, depending on your needs at the time. Because of this flexibility, variety, and ease of use, relational databases continue to be popular.

### SQLDB Database Creation

You create SQLDB databases by writing your database description in the SQLDB Data Definition Language (SQL DDL) and storing that description in a text file. You then use the file as input to a database utility that processes the DDL and generates the database structure.

SQLDB supports the creation of logical views of the database. Views allow you to simplify data presentation, restrict access to sensitive data, and allow concurrent access of the same data by users with different requirements.

You can also change the schema of your SQLDB database by modifying your DDL file and resubmitting it to the SQLDB system.

### SQLDB Database Query and Update Facilities

The Interactive Query Facility (IQF) can be used to query and update SQLDB databases. IQF presents you with a blank screen on which you can construct your SQLDB query, using SQLDB Data Manipulation Language (SQL DML). Any errors are immediately flagged. All of the other IQF functions, such as control breaks and report printing, are also available.

All queries can be saved in IQF for reuse.

### Host Language Interfaces

SQLDB provides extensions to COBOL85, which programmers can include in application programs to access SQLDB databases. Although extensions are not currently provided for other languages, ALGOL, RPG, FORTRAN, COBOL74, and Pascal programs can access SQLDB databases through A Series library entry points. For more information, refer to Section 5 of this overview, "Program Development."

### Semantic Information Manager (SIM) Databases

The Semantic Information Manager (SIM) enables you to design and implement databases that provide the substantial features of the semantic and object-oriented data models. The semantic data model provides all the flexibility of the relational

model, while capturing and automatically enforcing more of the real-world relationships between data elements. SIM greatly simplifies database application program development and online database queries by directly supporting constructs that allow SIM to capture and enforce business rules previously enforced by application programs.

The following pages review SIM database structures and the methods used to create, reorganize, query, and update SIM databases.

## SIM Database Structure

Information in a SIM database is organized into *classes* of related types of *entities*. Classes and entities are somewhat analogous to the data sets and data records provided by more traditional database systems. However, classes and entities support relationships that cannot be defined in conventional database systems. The possible types of relationships are illustrated by Figure 3-1, which shows a schema of a sample SIM database called the *Organization* database. The following paragraphs explain the features of this database.

In Figure 3-1, the ovals with labels (such as *PERSON*) represent classes. For each class, you can define a set of attributes that are associated with each entity in that class. Basic traits of entities in a class are represented by *data-valued attributes* (DVAs). Information about relationships with entities in the same or other classes is represented by *entity-valued attributes* (EVAs).

EVAs provide you with the means to represent complex relationships in your database. Because most of the details about the physical database level are handled automatically, you can define numerous complex relationships between classes without having to explicitly create separate tables or files, as you would do with more traditional database models.

For example, in Figure 3-1, the *MANAGER* class has an EVA called *PROJECTS-MANAGING*, that is directed at the *PROJECT* class. This is a multivalued EVA because a manager can be responsible for more than one project. However, note that the *PROJECT* class has a single-valued EVA called *PROJECT-MANAGER* that is directed at the *MANAGER* class. A project can only have one manager in this database.

What is the advantage of an EVA with these characteristics? When you define your database with EVAs in this way, then any time an application tries to assign or otherwise imply that there are two or more managers for a single project, SIM automatically enforces your predefined rule that a project can only have one manager. You no longer need to have each application program include program code to check for this possible occurrence.

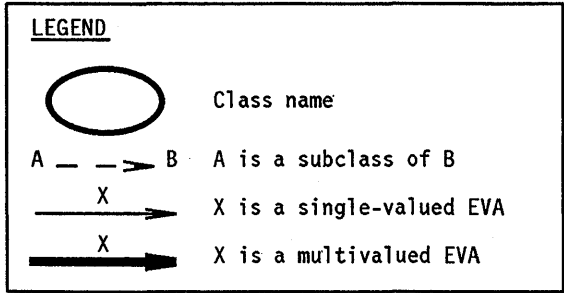
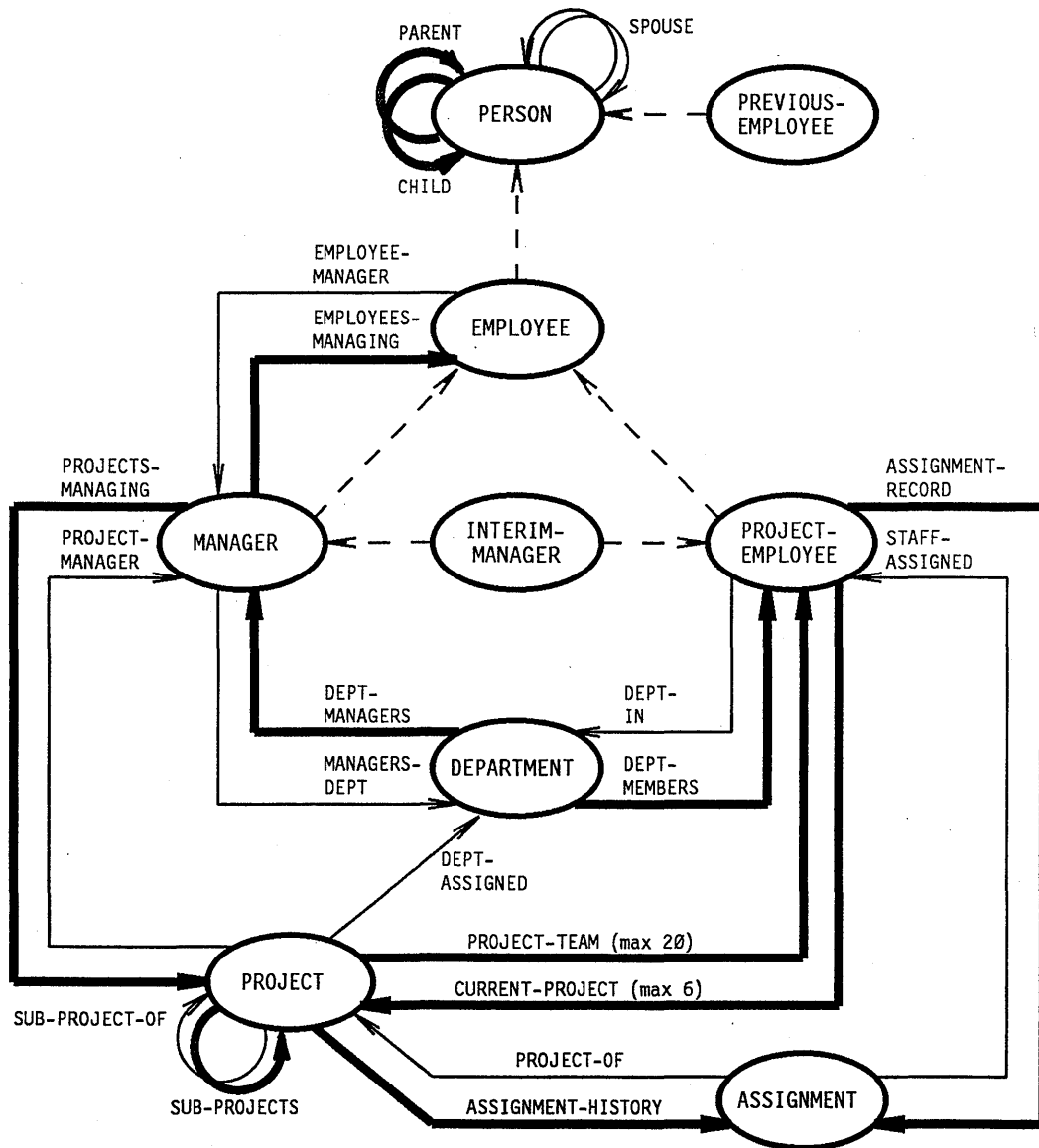


Figure 3-1. SIM Schema of the Organization Database



Advanced features of SIM include:

- *Inverse attributes.* SIM automatically creates an *inverse attribute*, which is an EVA that stores the inverse of each relationship. For example, in Figure 3-1, the EMPLOYEE-MANAGER EVA is the inverse of the EMPLOYEES-MANAGING EVA. This use of inverse attributes makes it possible for SIM to provide *referential integrity*.
- *Referential integrity.* When a database object is updated, SIM automatically updates the relationships between that object and other objects. For example, if you delete an employee entity from the sample database, the employee is automatically deleted from the EMPLOYEES-MANAGING EVA of the manager. In relational databases, the application programs have the burden of ensuring this data integrity, since the same data can be stored in multiple tables.
- *Data validation at run time.* At run time, SIM validates that data values assigned to data-valued attributes are of the correct type and within the correct range allowed for that attribute. This further reduces the need for integrity-checking code in application programs. In addition, generalized predicates, or Boolean expressions, can be associated with classes. These predicates are triggered or checked by SIM whenever necessary to guarantee that they always hold true. They can access attributes of a class and any inherited or extended attributes. This feature allows many more business rules to be captured and enforced by SIM. No other commercial DBMS provides such a powerful feature.
- *Data types.* SIM provides several predefined data types and also allows you to define data types of your own. You can define a range of values or a list of values allowed for each data type.
- *Generalization.* The use of *generalization hierarchies* allows you to organize data into meaningful structures, without duplicating the physical storage of the data. This means that each entity in a class can belong to one or more *subclasses* within that class, and each subclass can be subdivided into still finer subclasses.

For example, referring to Figure 3-1, a manager is represented as an entity in the MANAGER subclass. Of course, to be a manager, that individual must also be an employee. The EMPLOYEE class is a more *generalized* occurrence of entities in the MANAGER subclass. Entities in the MANAGER class are more *specialized*—that is, they possess attributes that make them unique from other employees represented in the EMPLOYEE class. All employees are PERSONs, but not all persons are EMPLOYEEs.

What is the advantage of this feature? The PERSON information is stored once for each entity. If the entity also participates in subclasses, such as EMPLOYEE, SIM ensures that the employee information is “related” to the PERSON record by automatically replicating the surrogate of the person in the EMPLOYEE information. The Name attribute of PERSON does not need to be replicated in the EMPLOYEE subclass. Attribute inheritance makes Name logically available from the EMPLOYEE perspective, and SIM takes care of any physical details.

- *Attribute inheritance.* The entities of each subclass inherit the attributes of the superclass. For example, the PERSON class might have the attributes Name, Social Security Number, and Age. The EMPLOYEE subclass of the PERSON class could have the attributes Profession, Hire Date, and Salary. By inheritance, entities in the EMPLOYEE subclass actually have the attributes Name, Social Security Number, Age, Profession, Hire Date, and Salary. This contributes to the strengths of data independence.
- *Set-oriented operations.* SIM supports set-oriented (rather than record-oriented) operations. This means that an application can use a single operation to retrieve all the entities that meet a specified condition, and the condition can depend on a comparison of the contents of many entities. It is therefore unnecessary for the application program to incorporate loops to find all the entities affected by a transaction.
- *Extended attributes.* SIM operations are simplified through the use of *extended attributes*. When an EVA establishes a relationship between two classes, the attributes of the second class are considered extended attributes of the entities in the first class. Because of this link, you can form simple queries to request information such as the name of the spouse of the manager of a particular employee. This innovation accomplishes the same thing as a JOIN operation in a relational database but has the advantage of being much easier to understand and use.
- *Data independence.* SIM data independence features make it possible for you to reorganize a database with minimal impact on application programs. For example, you can add attributes to an entity or change the order of the attributes of an entity without having to recompile application programs. This is possible because the application program accesses each attribute by name only. SIM matches that name with a physical location at run time.
- *Performance.* SIM is designed to support large databases and high-volume transactions. In a version of the industry standard debit/credit benchmark, SIM achieved 130 transactions per second on a dual-processor A 17. To achieve high performance, SIM uses the proven technology of the Data Management System II (DMSII) as its underlying base.
- *Coexistence.* SIM is designed to coexist with DMSII databases. With DMS.View and LINC.View, a single application program can be designed to access both DMSII and SIM databases. And the same set of operations utilities can be used for both types of databases.
- *Audit and recovery.* SIM provides comprehensive audit and recovery features, which are discussed further under “Database Backup, Auditing, and Recovery” in this section.
- *Security features.* SIM DDL includes features that allow a DBA to restrict access to various entities and/or attributes on a by-usercode basis. For details, refer to “Database Access Controls” in Section 9.

### Documentation

The *A Series InfoExec Semantic Information Manager (SIM) Technical Overview* explains SIM database structure and provides more details about the Organization Database example.

## SIM Database Creation and Reorganization

Unisys offers three tools for defining and modifying the schema of SIM databases: the SIM ODL Processor (SIM/UTILITY), the InfoExec Advanced Data Dictionary System (ADDS), and the Semantic Information Manager (SIM) Design Assistant (SDA).

ADDS is a screen-based, mainframe repository that allows you to create the database by describing the database structures and elements to the system. ADDS is available in two varieties, InfoExec ADDS and InterPro ADDS. Because you can use ADDS to define both SIM and DMSII databases, ADDS features are described later in this section.

The SIM Design Assistant (SDA) is a PC-based, graphical-interface tool for performing design, definition, and maintenance of SIM databases.

The SIM Design Assistant (SDA) is available for use on the Unisys PC family of products, the IBM PC<sup>®</sup> microcomputer, and compatible products, running on the Microsoft<sup>®</sup> Windows<sup>™</sup> graphical environment. SDA provides the capability to define, manage, and maintain SIM database schema. SDA is a lower CASE tool with a graphical multiwindow user interface. You define classes in a tree-structured graphical representation of the generalization hierarchy. Also included are graphical representations of the relationships between classes. Using these representations, SDA generates the database.

The flexible design of SDA promotes easy navigation through the database and complete freedom to choose the order in which you define the components of the database. SDA provides online error checking to indicate incomplete or invalid definitions prior to database generation, thus helping you manage multiple classes and attributes. SDA is integrated with ADDS and depicts flexible navigational aids.

### Documentation

For more information, refer to the *Personal Workstation<sup>2</sup> InfoExec Semantic Information Manager (SIM) Design Assistant (SDA) Operations Guide*.

## SIM Database Query and Update Facilities

The InfoExec environment provides two facilities that enable you to update, query, analyze, and create reports on data contained in SIM databases:

- **Interactive Query Facility (IQF)**  
IQF facilitates querying and updating SIM databases and files, by providing a menu-driven interface complete with online help and multiple means of query definition. IQF provides extensive report formatting capabilities.
- **Workstation Query Facility (WQF)**

---

IBM PC is a registered trademark of International Business Machines Corporation.

MICROSOFT is a registered trademark of Microsoft Corporation.

Windows is a trademark of Microsoft Corporation.

Personal Workstation<sup>2</sup> is a trademark of Unisys Corporation.

WQF offers many of the same capabilities as IQF, but provides them by way of a workstation, window-based environment.

### Host Language Interfaces

SIM provides extensions in various host languages that programmers can include in application programs to access SIM databases. For general information about the SIM host language interfaces, refer to "Database Interfaces" in Section 5. For information about using the SIM host language interfaces to query DMSII databases, refer to "DMS.View" in this section.

### InfoExec Environment Query and Update Facilities

The InfoExec environment offers two packages for performing query and update functions. Interactive Query Facility (IQF) is a menu-driven, mainframe resident product that supports both SQLDB and SIM databases. Workstation Query Facility (WQF) provides the same functions as IQF, but is available in a window-based workstation environment and supports only SIM databases.

### Interactive Query Facility (IQF)

The Interactive Query Facility (IQF) is a utility that provides querying and updating of SIM and SQLDB databases. Files can be queried only. IQF manages the tasks associated with the querying function, allowing you to retrieve data in a manner meaningful to you, regardless of physical file organization. IQF is an optional product of the InfoExec environment. Like other InfoExec programs, IQF is menu driven, offers extensive online help, and allows you to enter queries in a freeform manner, or to take advantage of its powerful query construction capabilities. Among its many advanced features, IQF does the following:

- Allows you to access data from up to five databases in a single query.
- Provides the means to produce tabular reports, graphs, histograms, x-y plots, or file extracts.
- Provides extensive *report painting* capabilities so that you can specify the exact format of a report. These capabilities include placement and formatting of data elements and literal text within a page. The format of the report detail line as well as control-break headings and footings are completely user-definable.
- Allows you to direct output to your terminal, to a disk file, or to a printer.
- Allows you to specify printer destination and the format of the printed output.
- Allows you to transfer data to workstations in Symbolic Link (SYLK) format, data interchange format (DIF)<sup>®</sup> format, or other formats, by using the Data Transfer System (DTS). For information about DTS, refer to "Data Transfer and Reformatting" in Section 4.

---

DIF is a registered trademark of Lotus Development Corporation.

- Allows you to save queries in ADDS for later use. Queries can also be shared with the Workstation Query Facility (WQF), which is discussed under “Workstation Query Facility (WQF)” in this section.
- Offers some compatibility with databases other than SIM. For example, DMSII databases can be updated by way of DMS.View and SQLDB databases can be queried by way of IQF. For information about IQF compatibility with DMSII databases, refer to “DMS.View” in this section. For information about IQF compatibility with LINC II databases, refer to “Logic and Information Network Compiler II (LINC II)” in Section 5.
- Allows you to translate the IQF interface into languages other than English, using the MultiLingual System (MLS)

### Documentation

The *A Series InfoExec Interactive Query Facility (IQF) Operations Guide* explains how to use IQF.

## Workstation Query Facility (WQF)

The Workstation Query Facility (WQF) is a convenient workstation-based product that provides easy access to a database for users who do not have data processing experience. With WQF, you can view and update SIM databases without using cumbersome data manipulation language syntax. The virtues of WQF are similar to those of IQF, except that WQF uses workstation windowing facilities to allow you to see multiple views of a database simultaneously.

WQF consists of a program that runs on your workstation and a cooperating program that runs on the A Series host. You can run the workstation program on the BTOS family of workstations and XE520 systems, as well as on the Unisys PC family, the IBM PC microcomputer, and compatible products. Among its many advanced features, WQF does the following:

- Provides an easily understood windowing scheme; for example, to indicate the data to be included in a report, you build a query on the Query Window.
- Provides for each window a set of commands that can be executed when that window is active.
- Provides a sophisticated menu interface and extensive online help information that makes memorizing commands unnecessary.
- Provides an optional mouse interface.
- Provides an online tutorial to teach you how to use the product productively.

- Allows you to save reports on the workstation in Symbolic Link (SYLK) or data interchange format (DIF) files. This feature makes it possible for you to manipulate the report data with workstation products such as Lotus® 1-2-3® or Multiplan® spreadsheet software.
- Offers compatibility with databases other than SIM databases. Refer to “DMS.View” in this section. For information about WQF compatibility with LINC II databases, refer to “Logic and Information Network Compiler II (LINC II)” in Section 5.

A prerequisite for WQF is the Unisys INFOVIEW II product, which provides the WQF windowing facilities. For information about this product, refer to “Windows into the Mainframe Environment” in Section 4.

### Documentation

The *Workstations InfoExec Workstation Query Facility (WQF) Operations Guide* explains how to use WQF.

## Data Management System II (DMSII) Databases

DMSII is a mature, proven database management system that continues to receive major feature enhancements and performance improvements. The following pages review DMSII database structures and the methods used to create, reorganize, query, and update DMSII databases.

### DMSII Database Structure

DMSII enables you to define multiple *data sets*, each of which consists of data *records* that are divided into named *fields*. Application programs can access information by referring to data set names and record field names, without regard to the physical location of the data sets or fields. This method of access helps ensure data independence, enabling you to add more data sets or fields to the database without modifying or recompiling any existing applications.

Even radical changes to the database organization can be made transparent to existing application programs through the use of data set *remaps*. A remap can rename fields, conceal fields, or compute a value from multiple fields in the same record. Multiple remaps can be defined for each data set, so that a different view of the data can be provided to each application program.

For each data set, you can define a variety of different *sets*, which serve as indexes into the data set. Thus, if a record includes a name, phone number, and social security number, you could define sets that would allow an application to efficiently select a record based on any of those three criteria.

---

Lotus is a registered trademark of Lotus Development Corporation.  
1-2-3 is a registered trademark of Lotus Development Corporation.  
Multiplan is a registered trademark of Microsoft Corporation.

DMSII provides a variety of data set types and set types that are optimized for different patterns of I/O, such as random I/O and sequential I/O. You can select among these types to improve database performance.

DMSII databases are record-oriented and allow databases to be described using hierarchical and network data models. However, for efficiency it is generally recommended that you implement DMSII databases that use separate data sets. Most relationships are then created in the application programs by matching values of fields with corresponding fields in different data sets. This is the same method that is used in relational databases.

Figure 3-2 shows a DMSII version of the Organization database that was introduced under "Semantic Information Manager (SIM) Databases" in this section. The boxes represent data sets. Within each box, the name below the line is the corresponding index set. The names above the line are data fields that are required to reflect the logical structure of the organization.

An example query might be to find the children of a particular employee. To do this, a program would look up the social security number of the person in the PERSON data set, search for all Parent records in the FAMILY data set that have that social security number, and read the name from the Child field of each of these records.

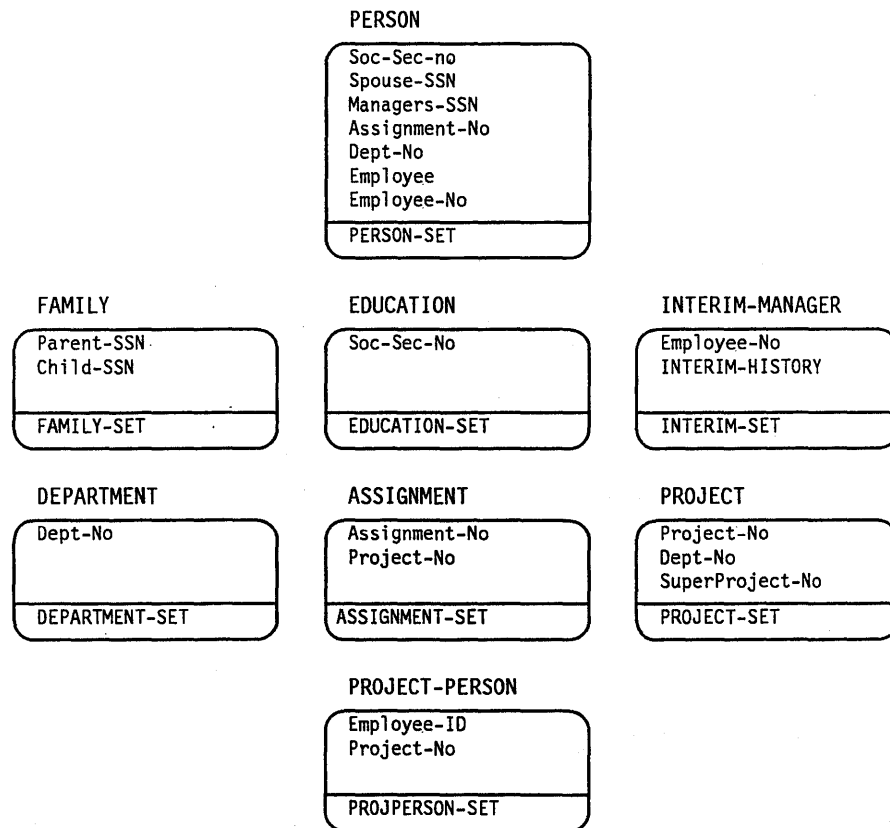


Figure 3-2. DMSII Schema of the Organization Database

DMSII provides specialized routines that allow many application programs to query and modify the database concurrently. These routines automatically maintain indices to the data that allow efficient retrieval in response to a large class of questions. These routines also perform all the details of physical file management, so the application programmer can concentrate on the logical aspects of database organization.

DMSII also provides comprehensive audit and recovery features, which are discussed further under “Database Backup, Auditing, and Recovery” in this section.

You can implement separate security restrictions for each structure in a DMSII database. For details, refer to “Database Access Controls” in Section 9.

### Documentation

- *A Series DMSII Data and Structure Definition Language (DASDL) Programming Reference Manual*  
This manual includes DMSII architecture and design considerations.
- *A Series Data Management Functional Overview*  
This manual includes a DMSII overview.

## DMSII Database Creation and Reorganization

You can define DMSII databases in either of two ways: by using the Advanced Data Dictionary System (ADDS) or by using the DMSII Data and Structure Definition Language (DASDL).

ADDS capabilities are described under “Database Creation” in this section.

DASDL is a high-level language that you can use to specify the physical and logical characteristics of a database, as well as the criteria to be used to ensure the integrity and security of the data in the database.

You can also reorganize DMSII databases in either of two ways: by using ADDS or by using the REORGANIZATION utility. REORGANIZATION is a command-oriented utility that provides many of the same database reorganization functions as ADDS, including the following:

- Adding, changing, or deleting the fields defined for the records in a data set
- Adding sets to provide new methods of access to an existing data set
- Changing attributes of the database that affect performance

### Documentation

- *A Series InfoExec Advanced Data Dictionary System (ADDS) Operations Guide*  
This guide includes ADDS documentation for DMSII users.



- *A Series DMSII Data and Structure Definition Language (DASDL) Programming Reference Manual*

This manual provides a detailed description of DASDL.

- *A Series DMSII Utilities Operations Guide*

This guide describes the REORGANIZATION utility.

## DMSII Database Query and Update Facilities

The DMSII and InfoExec environments provide facilities that enable you to update, query, analyze, and create reports on data contained in DMSII databases. These products are described in the following pages.

### Extended Retrieval with Graphic Output (ERGO)

Extended Retrieval with Graphic Output (ERGO) is an optional part of the DMSII environment that enables you to interactively query, update, and delete data items in a DMSII database.

ERGO has a menu-driven interface with help information available for all command and data items. You can also bypass the ERGO menus by entering direct commands.

ERGO allows you to form queries that are based on relationships between the data stored in two or more separate data sets. Suppose you are using the Organization database shown in Figure 3-2. You could use the RELATE statement to match records from the Person data set with records in the Project-Person data set. The matching could be done on the basis of the Employee-ID field, which occurs in both data sets.

Once this relationship is established, you can use a single query command to return information such as the social security numbers of the managers of all employees who work on a particular project.

ERGO has an extensive capability for data manipulation through built-in mathematical and statistical functions. Its report capability includes tabular and statistical formats and data extraction to both flat files and the BTOS family of workstations, plus a variety of graphic formats: discrete and continuous plot graphs, bar graphs, kivi diagrams, and histograms. You can use standard report formats included in ERGO or define your own formats through extensive report options.

ERGO allows you to access up to five DMSII databases and 10 conventional files in a single session. Also available are search and sort mechanisms, parameters for data selection, and extensive mathematical capabilities. Finished reports can be directed to a disk file, a printer, one or more terminals, or workstations running the Data Transfer System (DTS). (For information about DTS, refer to "Data Transfer and Reformatting" in Section 4.)

### Documentation

The *A Series Extended Retrieval with Graphic Output (ERGO) Operations Guide* explains how to use ERGO.

### DMS.View

You can use several components of the InfoExec environment to query and update the data in the DMSII database. The InfoExec components that can be used for this purpose are the Interactive Query Facility (IQF), the Workstation Query Facility (WQF), and the SIM host language interfaces. For an introduction to the capabilities of IQF and WQF, refer to “InfoExec Environment Query and Update Facilities” in this section. For an introduction to host language interfaces, refer to “Database Interfaces” in Section 5.

To make a DMSII database available to these InfoExec query facilities, you must first process the database with the DMS.View utility. DMS.View adds semantic directory information to the DMSII database definition. The change is transparent to DMSII utilities, and does not require changes to any DMSII application programs or operations procedures. The DMS.View utility is available only as part of the InfoExec environment.

### Documentation

The *A Series InfoExec DMS.View Operations Guide* explains how to use DMS.View.

### Host Language Interfaces

DMSII provides host language extensions and an interpretive interface to enable application programs to access databases. For further information about the programming language interfaces, refer to “Database Interfaces” in Section 5.

## Database Creation

The A Series offers several means for defining and creating databases, including the following:

- Advanced Data Dictionary System (ADDS)
- Data definition languages such as the Data and Structure Definition Language (DASDL)
- SIM Design Assistant (SDA)

### Advanced Data Dictionary System (ADDS)

ADDS is available in two versions: InfoExec ADDS and InterPro ADDS. Of these, InterPro ADDS can be run outside the InfoExec environment and is considered part of the DMSII environment. ADDS has the same capabilities in both the InfoExec and DMSII environments, except that ADDS in the DMSII environment cannot create SIM databases.

ADDS provides an easy-to-use, screen-based interface that makes it unnecessary for you to physically program your database description in a data definition language. Instead, you navigate through various data definition screens to describe the database. ADDS provides the means to generate, update, and reorganize your database structures. InfoExec ADDS enforces the structural rules of SIM, helps to eliminate inconsistencies and redundancies in data, and facilitates the definition of relationships among objects. The central storage of all data items also promotes efficient database and application design. ADDS also includes a comprehensive help facility to assist you by providing information about all aspects of data definition.

ADDS is an integral part of the InfoExec environment and provides the functions for the Data Definition, Creation, and Utilities selections on the InfoExec home menu. If you have the InfoExec environment, you can use ADDS to create or restructure both SIM and DMSII databases.

Regardless of which environment you choose to work within, each version of ADDS does the following:

- Provides centralized creation, maintenance, and update of data definitions
- Provides concurrent multiuser access for retrieval and modification of data definitions
- Supports creation of complex relationships among data elements
- Supports multiple data types
- Allows description and storage of SIM, DMSII, COBOL74, ALGOL, and Pascal data elements and programs
- Provides the means to import data descriptions directly into applications programs at run time
- Provides database generation
- Provides extensive data dictionary administration capabilities
- Allows you to link textual descriptions to data items and structures
- Provides entity and structure level security features
- Provides extensive reporting capabilities
- Allows you to assign alias names and keywords to data items
- Provides a migration utility for moving existing DMSII database descriptions into ADDS
- Provides an extensive online help facility
- Provides process definition and reporting

Both InfoExec ADDS and InterPro ADDS offer significant features for data element creation, maintenance, and control.

### **Documentation**

The *A Series InfoExec Advanced Data Dictionary System (ADDS) Operations Guide* explains how to use ADDS.

### Data Definition Languages (DDLs)

Each of the three database management products offered on the A Series includes a powerful data definition language (DDL). These are briefly described in the following text.

#### Data and Structure Definition Language (DASDL)

DASDL is the data definition language you use to describe and create DMSII databases. DASDL was designed as a high-level language that follows a structured programming approach, not unlike many of the commercial programming languages. It provides powerful tools for database definition and manipulation. You can create DMSII databases by compiling a DASDL source file that you have written, or by defining and generating the databases with ADDS.

#### Semantic Information Manager Object Definition Language (SIM ODL)

The SIM object definition language (SIM ODL) is a high-level structured language that you can use to define SIM databases in the InfoExec environment. Generally, you define SIM databases through InfoExec ADDS or SIM Design Assistant (SDA); however, you can also write SIM ODL and process it through the SIM Utility to create the database.

#### Structured Query Language Data Definition Language (SQL DDL)

The SQLDB data definition language offers you a variety of powerful data definition tools and operators for manipulating your SQLDB databases. You must write SQL DDL into a text file, which you then run through SIM Utility to create your SQLDB database.

#### Documentation

- *A Series DMSII Data and Structure Definition Language (DASDL) Programming Reference Manual*
- *A Series InfoExec Semantic Information Manager (SIM) Programming Guide*
- *A Series InfoExec Structured Query Language Database (SQLDB) Programming Guide*
- *A Series InfoExec Advanced Data Dictionary System (ADDS) Operations Guide*
- *Personal Workstation<sup>2</sup> InfoExec Semantic Information Manager (SIM) Design Assistant (SDA) Operations Guide*

### Database Operations

The Operations Control Manager (OCM) provides an integrated, menu-driven approach to the operation and control of SQL, SIM, and DMSII databases. Two versions of OCM are available. InfoExec OCM is an optional part of the InfoExec environment that provides access to SQL, SIM, and DMSII databases. InterPro OCM is an optional part of the DMSII environment that provides access to SQLDB and DMSII databases only.

You need not remember syntax to use OCM; menus guide you through commands so that you do not have to refer to manuals. In cases where you are familiar with the syntax, you can enter the command directly, without using the menus.

OCM offers both online and batch operation, thus allowing you to initiate a function online for immediate execution or specify that the function be saved and performed later through a WFL job generated by OCM.

The review mode in OCM enables you to step through the menu choices that you have made in order to verify that you specified the correct actions. You can make changes during this verification process and reverify the entries until you are satisfied that the specification is correct.

OCM performs its functions by initiating a variety of DMSII utilities. OCM does not replace these utilities, but serves as an interface to them. You can use OCM to manage SQL, SIM, or DMSII databases. OCM can be used in either the InfoExec or DMSII environments, but is a separately packaged software product that must be specifically requested.

The following pages describe the database operations features provided by OCM and identify the utilities you can use to accomplish these operations if you do not have OCM at your installation.

### Documentation

The *A Series DMSII Utilities Operations Guide* explains how to use OCM.

## Database Backup, Auditing, and Recovery

Recovery of a database must occur every time processing is halted abnormally. Recovery means reconstructing the database, ensuring that no partially completed transactions are left in it, and notifying programs where to restart processing.

### Choosing Recovery Strategies

The A Series data management software gives you the flexibility to select the recovery strategy suited to your needs. When you develop the database definition (through SQL DDL, ADDS, or DASDL), you can specify whether the database should be audited. For audited databases, the system creates an audit trail recording each database transaction. An unaudited database gives faster performance, but in the event of a system failure, it can be restored only to the point at which the last complete backup of the database was taken.

You can create a backup dump of the database by using the DMUTILITY program or the DUMPDB screen in OCM. For audited databases, the database can be online during the dump. For unaudited databases, the database must be offline during the dump.

### Recovery for Audited Databases

The following paragraphs discuss recovery for audited databases only.

If an application program aborts during a transaction, recovery occurs automatically without causing any of the other application programs to terminate. If there is a system failure (called a halt/load), the database recovers automatically the first time the database is accessed after the system recovers.

You can use manual recovery for cases where database data stored on disk has been damaged. You can initiate manual recovery by running the DMUTILITY program or by using the RECDB screen in OCM. Individual disk rows can be recovered from audit files and backup dump tape files while the database is still online. Where more major damage has occurred, the entire database can be rebuilt offline from backup dump tape files and audit files. Additionally, if you determine that an application program has updated the database with incorrect data, you can request that the database be rolled back to its state at an earlier point in time.

Concurrency control is an aspect of database recovery that is becoming more and more important as the industry evolves toward online database transactions. Concurrency control ensures that, once a transaction has completed, that transaction will not be backed out during recovery from a program abort or system halt/load.

You can achieve concurrency for DMSII databases by setting the following options: INDEPENDENTTRANS and REAPPLYCOMPLETED. The INDEPENDENTTRANS option ensures that, if one application program aborts in the middle of a transaction, the database can recover without undoing transactions submitted by other application programs. REAPPLYCOMPLETED causes all completed transactions to be reapplied from the audit trail after a system halt/load. Because these options are always implicitly set for SQLDB and SIM databases, these databases always provide concurrency control.

The Communications Management System (COMS) provides a feature called *synchronized recovery* that simplifies database recovery. You can use COMS synchronized recovery instead of, or in addition to, the database concurrency options. Refer to "Database Recovery" in Section 4.

#### Documentation

The *A Series DMSII Utilities Operations Guide* describes DMUTILITY and OCM.

### Remote Database Backup

The Remote Database Backup (RDB) product provides higher levels of database availability and can satisfy requirements for disaster recovery. RDB provides the means to create and maintain a clone database that is concurrently updated, transaction by transaction, along with a primary host database. The secondary database can be on-site with the primary database, or at a remote site. RDB provides the following benefits:

- Flexible location configuration of primary and secondary database systems
- Support of communications processor local area network (CPLAN), TCP/IP, and X.25 communication transport protocols
- Multiple methods for transferring audit blocks from primary database to secondary database
- Flexible hardware configuration requirements
- Low overhead on secondary system
- Multihost accessibility of secondary database for inquiry and reporting
- Interactive configuration capability
- Support of existing A Series hardware, system software, and database systems

### Documentation

The *A Series DMSII Utilities Operations Guide* describes how to use Remote Database Backup (RDB).

## Database Integrity Certification

The DBCERTIFICATION utility determines the integrity of a DMSII database. It provides three levels of certification:

- *Physical integrity.* The utility ensures that a file is physically intact and accessible, and isolates any problems at the data block level.
- *Internal, intrafile integrity.* The utility verifies that data in a single structure are consistent, and that storage control and internal control information are valid.
- *Interstructure integrity.* The utility crosschecks to ensure that relationships between data structures are correct.

You can select the database structures to verify and the tests to perform at each of the three levels. When you run DBCERTIFICATION, you get online or printed analysis listings as output. These listings include appropriate error messages, addresses of detected errors, and a dump of the blocks where errors occurred.

The functions of the DBCERTIFICATION utility can be accessed in OCM through the CERTIFY screen.

### Documentation

The *A Series DMSII Utilities Operations Guide* describes DBCERTIFICATION and OCM.

## Database Structure Analysis

The DBANALYZER utility is a software tool that analyzes the logical and/or the physical structure of a database and provides reports that a database manager can use to modify,

## Database Management

---

tune, reorganize, and document the database. DBANALYZER describes the structures in the database and their interrelationships, and provides structure analysis that includes physical attributes, population, available space, and data block utilization. The program does not, however, allow you to examine the contents of the database, so it does not compromise database security.

You can use DBANALYZER interactively or in batch mode. Simple commands control the scope and depth of each analysis and allow you to direct the output to a terminal or a printer.

The functions of DBANALYZER can be accessed in OCM through the ANALYZE screen.

### Documentation

The *A Series DMSII Utilities Operations Guide* describes the DBANALYZER utility and OCM.

## Database Performance Monitoring

The DMMONITOR utility is a program that enables you to monitor databases interactively. It provides access to database status and statistics information and permits you to make changes to database options and parameters. In addition, you can use this program to sample database information and capture the information in a disk file for offline analysis. The DMMONITOR program provides the following capabilities:

- Status and statistics can be displayed continuously on a screen, captured in a disk file, or printed.
- Status and statistical information can be monitored for the entire database or for selected structures of the database.
- Sampling can be selectively turned on and off during the course of a session.
- Sampled information includes the number of users, transactions performed, read operations, and write operations.

The functions of DMMONITOR can be accessed in OCM through the MONITOR screen.

### Documentation

The *A Series DMSII Utilities Operations Guide* describes the DMMONITOR utility and OCM.



# Section 4

## Transaction Management

Most installations today are developing database applications as transaction-oriented systems, with high-volume, real-time processing using extensive terminal networks. This trend is becoming increasingly important to system design, due to the rapidly growing user demand.

Unisys addresses the needs of modern transaction-oriented systems in two ways: with system software that gives the administrator control over transaction handling, and with workstation integration products that greatly simplify system access for the online user.

### Mainframe Transaction Environments

The A Series provides several environments for transaction submission and control. These environments are implemented by a number of system software programs called message control systems (MCSs).

An MCS controls the flow of messages among terminals, application programs, and the operating system. In this context, a message is typically the information transmitted between the terminal and the system, and thus includes most user communications with a computer system. In addition to controlling the flow of messages, an MCS performs the following major functions:

- Logical control of stations to provide security and secondary error handling
- Program initiation and control
- Transaction auditing and synchronized recovery

An MCS can be general, so that it can be used for a number of different functions and by all system users, or it can be specific and be designed to serve specialized users and functions.

### Generalized Transaction Control

The primary MCS on the A Series is the Communications Management System (COMS). COMS supports a network of users and provides them with a consistent, online interface to the system. COMS also manages and handles errors for the data communications network. COMS is partially integrated into the operating system, which enhances its ability to efficiently handle a high volume of transactions from multiple programs, stations, and remote files. The following pages explain some of the major functions that are provided by COMS.

### Documentation

- *A Series Communications Management System (COMS) Capabilities Manual*
- *A Series Communications Management System (COMS) Configuration Guide*
- *A Series Communications Management System (COMS) Operations Guide*
- *A Series Communications Management System (COMS) Programming Guide*

### Flexible Configuration

The hallmark of COMS is its flexibility, which makes it possible for the administrator to define different transaction handling rules for a variety of different COMS *entities*. Some of the types of entities are users, programs, stations, and transaction codes. COMS allows you to define these entities individually and to specify transaction handling rules that apply to a particular entity or a group of entities.

You define entities and their relationships using the COMS Utility. You can interface with the COMS Utility in any of three ways:

- Use menu screens that step you through the configuration process.
- Enter COMS Utility commands.
- Load a file in which you have previously stored COMS Utility commands.

To prevent illogical configurations, the COMS Utility automatically issues a warning if you define an entity that refers to a nonexistent entity. You can use the COMS Utility to create a new configuration or to modify your existing COMS configuration. Changes to the existing configuration are applied dynamically, with no interruption in system response.

### Multiple Windows into the System

Windows are one of the special types of entities provided by COMS. Each window accesses a particular program environment. COMS can map multiple windows onto each physical terminal. These windows enable you to move from one program environment to another while processing continues uninterrupted.

Additionally, COMS allows you to access up to eight copies of each window at your terminal. Each copy of a window is called a *dialog*. Window dialogs allow you to engage in multiple sessions that access the same program environment.

For example, with multiple dialogs of a single window, a data entry clerk who needs to concurrently process six customers using a single accounts receivable application can change freely from one customer's dialog to another's without interrupting the processing in other sessions of the accounts receivable application.

COMS provides the following kinds of windows:

- Direct windows, which allow you to route messages to programs you have specifically defined to COMS. Direct windows are intended for multiuser programs that were implemented to take advantage of COMS message routing and processing features. The direct window interface also provides the fastest and most efficient method of message routing, thus improving the performance of online programs.
- Remote-file windows, which enable existing online applications programs to run under COMS with few or no modifications.
- MCS windows, which provide access to a subordinate message control system, such as Command and Edit (CANDE).

The most advanced features of COMS are those provided for direct window application programs. The following direct window features are discussed in the next few pages:

- Automatic program initiation
- Transaction routing
- Processing items
- Database recovery

Figure 4-1 illustrates a number of these features.

### Automatic Program Initiation

COMS automatically initiates as many copies of an application program as necessary to meet the needs of the users of that program. You can use the COMS Utility to assign parameters that help COMS to determine when to initiate an additional copy of the program. COMS improves the average response time for transactions by dividing messages equally among the existing copies of the application program. For more information, refer to “Dynamic Control of Online Programs” in Section 7.

### Transaction Routing

You can use the COMS Utility to define the transactions that can be entered in each window. For each transaction type, you can specify an *agenda* through which the transaction will be routed. Among other things, the agenda specifies the application program to which the message will be routed. COMS can thus route the transactions from a window to a variety of different application programs, while providing the end user with a single system view.

Figure 4-1 illustrates transaction routing for a hypothetical window named W1. When an end user submits a transaction of type TR1, COMS routes the transaction through agenda A, and then to application program AP1. When an end user submits a transaction of type TR2, COMS routes the transaction through agenda B and then to application program AP2. The end user need not be aware that the TR1 and TR2 transactions are processed by different application programs.

## Transaction Management

---

### Processing Items

The agendas you create can also contain lists of *processing items* that the message is routed through on its way to or from the application program. Processing items are procedures that you can write in ALGOL and store in libraries. You do not have to create any processing items, but they are a convenient method for reformatting or filtering messages, logging messages, gathering statistics, storing information about the state of a user's transaction, or emulating various types of terminals. Because the processing items are stored in libraries, each processing item can be used by multiple programs or multiple copies of the same program.

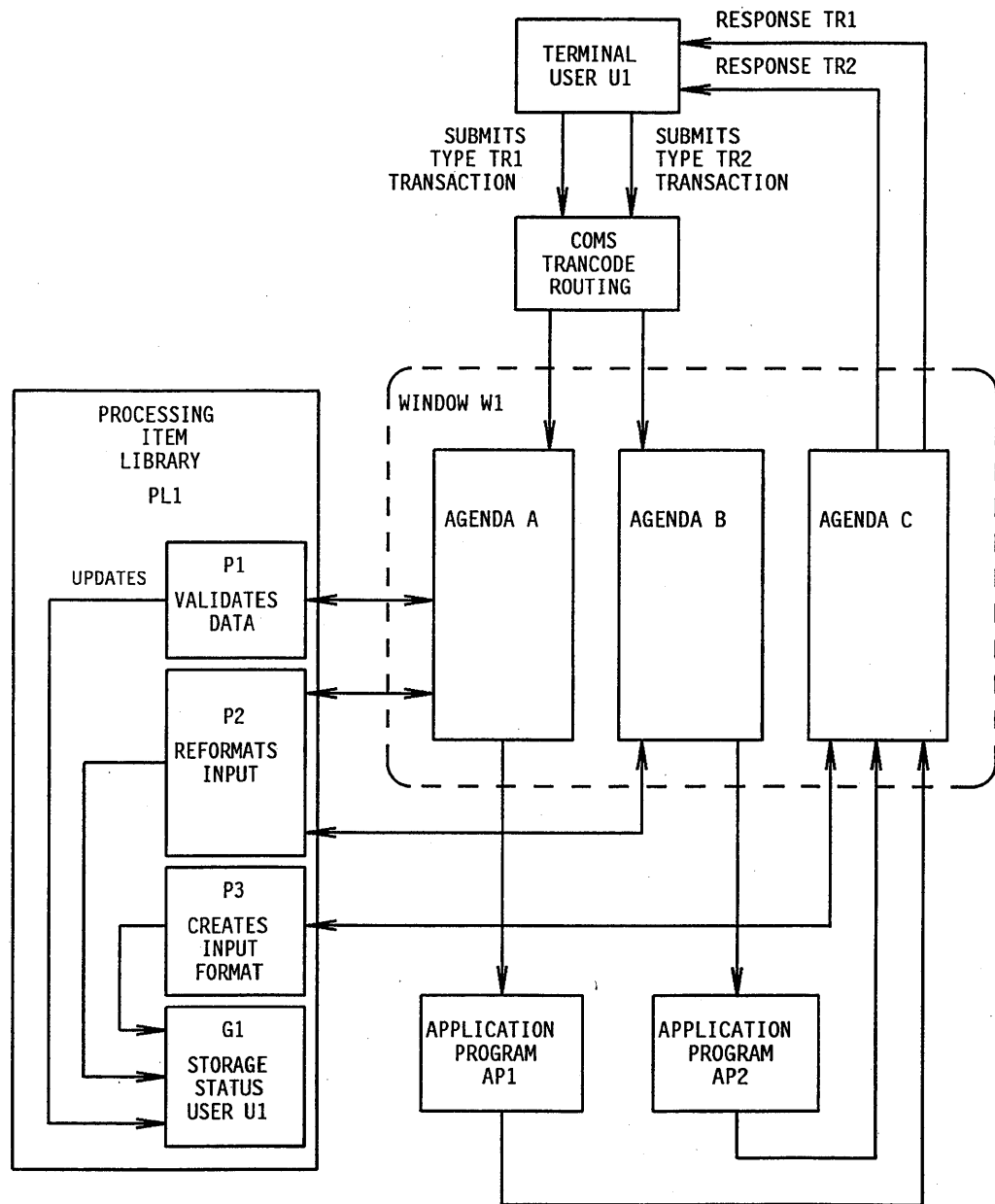


Figure 4-1. COMS Direct Window Interface

In Figure 4-1, processing item library PL1 stores the processing items P1, P2, and P3. Processing item P1 is invoked by agenda A and validates transactions of type A. Processing item P2 is invoked by agendas A and B, and reformats transaction inputs of type TR1 or TR2. Processing item P3 is invoked by agenda C and formats the output from transactions of type TR1 or TR2.

## Transaction Management

---

### Database Recovery

COMS provides a feature called *synchronized recovery* for database application programs implemented as direct window programs. Synchronized recovery is available for both SIM and DMSII applications. Synchronized recovery ensures that no completed transactions are lost during a recovery from a program abort or system halt/load.

COMS automatically maintains a *transaction trail* for all direct window database application programs. The transaction trail is separate from and in addition to any audit trails being maintained by the database software. When a database is recovering from a program abort or a system halt/load, COMS allows database recovery to proceed normally. The database software then informs COMS whether additional transactions were not recovered because of possible interdependencies with other transactions that had not yet completed. COMS resubmits each of these transactions to the database applications in the order in which the transactions were originally entered.

COMS synchronized recovery serves a function similar to that of the database concurrency options discussed under "Database Backup, Auditing, and Recovery" in Section 3. When these concurrency options are used, the data management software is normally able to reapply all completed transactions, and COMS synchronized recovery is not invoked.

COMS also provides features to protect input and output messages. The *protected input* feature writes copies of all direct program input messages to disk as soon as COMS receives them. If a system halt/load occurs before the direct window program reads them, COMS resubmits the messages after the halt/load. This feature makes it unnecessary for an online database user to type in an inquiry over again. You can use the COMS Utility to specify which agendas will use the protected input feature.

### Additional Features

COMS provides a number of other features that are discussed in other parts of this manual, as follows:

Feature	Refer To
Simplified system operations interface (MARC)	"Menu Interface" in Section 6
Security	"Transaction Access Controls" in Section 9
Statistics gathering	"Transaction Performance Monitoring" in Section 9

### Levels of Feature Content

You can choose from the following versions of COMS, depending upon the level of COMS feature content that you require:

COMS Version	Features
COMS (Kernel)	Provides a predefined configuration file that allows you to use the window feature with three windows: a Menu-Assisted Resource Control (MARC) window with eight dialogs, a CANDE window with two dialogs, and a GEMCOS window with one dialog. Additionally, you can communicate with remote-file programs.
COMS (Full-Featured)	Provides all the features of the Kernel version of COMS and also allows you to develop applications using transaction routing, processing items, and the synchronized recovery feature for Data Management System II (DMSII) and Semantic Information Manager (SIM) databases. This version also provides statistics to allow you to monitor the performance of transactions.

### Migration from Other MCSs

The historical predecessor to COMS is another MCS called the Generalized Message Control System (GEMCOS). GEMCOS was designed with many of the same goals as COMS, including flexible handling of a broad variety of transactions. The implementation of COMS introduced a large number of architectural and performance improvements that go beyond the capabilities of GEMCOS. Therefore, although GEMCOS is still fully supported for existing customers, new customers are directed to use COMS.

GEMCOS installations that wish to migrate to COMS can take advantage of migration features that will ease their transition. Unisys provides a library of COMS processing items called the format support library (FS-LIB), which emulates the message formatting and paging features provided by GEMCOS. Use of the FS-LIB minimizes the need for you to modify application programs or write new processing items of your own.

Additionally, you can make your transition from GEMCOS a gradual one by running GEMCOS as an MCS window under COMS. This method allows the GEMCOS and COMS environments to coexist on the same system. You can then move applications from GEMCOS to COMS one at a time without causing any interruptions in user service.

### Documentation

- *A Series Communications Management System (COMS) Migration Guide*
- *A Series Generalized Message Control System (GEMCOS) Operations Guide*
- *A Series Generalized Message Control System (GEMCOS) Format Generator Operations Guide*

## File Maintenance and Program Control

Another important A Series MCS is called CANDE. CANDE is a specialized MCS that provides access to a predefined set of commands called the Command and Edit (CANDE) language. CANDE provides a single, complete environment from which you can edit files and manage programs. Through CANDE, you can interact with and control programs while the programs are executing.

You can use CANDE as the only MCS on your system, but it is more typical to run CANDE as an MCS window under COMS. The dialog capabilities of COMS make it possible for each user to access several CANDE sessions at the same terminal.

CANDE file maintenance commands allow you to conveniently perform such functions as copying, removing, or displaying information about files. CANDE file editing commands allow you to change, insert, delete, move, find, and replace text within a file. CANDE makes it easy for you to undo editing changes if you have second thoughts. All your changes are made against a *work file*, which is a temporary copy of the original file. You can update the original file when you decide that you are satisfied with all your changes.

To protect against the loss of data during a system interruption, CANDE continually saves the changes made to a work file during a session. You can use the RECOVER command to recover all these changes after the system resumes execution.

CANDE provides a convenient environment for running single-user online application programs. (Multiuser programs should be implemented through the direct window feature of COMS rather than CANDE.) You can use CANDE commands to intervene in program execution or display information about your programs as they run.

CANDE also provides an ideal environment for program development. You can use CANDE commands to write the program, compile it, and test it, all in a single session. For example, compiling and executing a program can be as simple as entering the command *R*.

CANDE helps you to automate routine activities by storing sequences of CANDE commands in files, called *DO files*, for repeated use. You can schedule execution of these commands for a later time, or allow them to be executed in parallel with your current session.

Because CANDE does not make use of special characteristics of certain terminals (such as forms mode), it is able to communicate with a very wide range of device types. By contrast, COMS and its transaction processor MARC make use of specific terminal functions to achieve a more fully featured user interface, and thus make the choice of terminals more restrictive.



To facilitate accounting and security auditing, the system logs the major activities of each user in each CANDE session. CANDE provides security restrictions based on the identity of each user and each terminal. For more information about security restrictions, refer to "User Privileges" in Section 9.

### Documentation

- *A Series CANDE Configuration Reference Manual*
- *A Series CANDE Operations Reference Manual*

## Workstation Integration

There has been a growing use of microcomputers as *workstations* connected to mainframe computers during recent years. A Series workstation products increase your productivity in the following ways:

- By providing the primary user interface for certain A Series products, such as Capacity Management Facility.Snapshot (CMF.Snapshot), the Intelligent Distributed Editor (IDE), and the Workstation Query Facility (WQF). Workstation features such as graphics and pulldown menus help make these products easier to use.
- By providing multiple, customized views into the A Series environment that simplify certain types of tasks.
- By facilitating the transfer of data between the workstation and the A Series host. For example, this enables you to use workstation products to format reports based on data collected on the A Series host.
- By providing a simplified method to develop workstation-based user interfaces for applications. The product that provides this capability is the Screen Design Facility Plus (SDF Plus).
- By enabling a microcomputer to emulate various types of terminals.

For information about CMF.Snapshot, refer to "Snapshot System Analysis" in Section 9. For information about IDE, refer to "Workstation-Based Program Development" in Section 5. For information about SDF Plus, refer to "User Interface Generation" in Section 5. For information about WQF, refer to "Workstation Query Facility (WQF)" in Section 3. All other A Series workstation products are described in the following pages.

## Windows into the Mainframe Environment

The INFOVIEW II product serves as an applications support environment for the BTOS family of workstations, XE520 systems, the Unisys Personal Workstation<sup>2</sup> (PW<sup>2</sup>) family, the Unisys PC family, and IBM microcomputers and their compatibles.

The INFOVIEW II product provides multiple simultaneous sessions with the same A Series system or with as many as five different A Series systems. The INFOVIEW II environment allows you to view these sessions in a flexible manner through the use of software managed windows. You can reshape and move windows so that several of your sessions are visible at once.

You can traverse between windows by simply pressing a combination of a control key and a keypad key. The INFOVIEW II cut-and-paste feature enables you to move information between windows. If you have COMS on your A Series host, you can dedicate INFOVIEW II windows to particular COMS window dialogs. When you traverse between such INFOVIEW II windows, the INFOVIEW II environment automatically changes the COMS dialog as well.

The INFOVIEW II environment provides convenient pulldown menus for routine operations functions. You have the option of controlling the INFOVIEW II environment through a mouse interface.

The INFOVIEW II environment provides two types of terminal emulators: MT, which emulates an MT983, and EGT, which emulates a Digital Equipment Corporation VT100 or a Tektronix® 4014.

The INFOVIEW II MT terminal emulator allows you to reprogram almost all keys on the keyboard. Because key programs are stored in a file, you do not have to worry about running out of buffer space for your key programs. You can also redefine your keys so that they have different meanings depending on which INFOVIEW II window you are currently accessing.

The INFOVIEW II environment allows you to traverse between A Series host windows and *native* windows. Native windows allow you to access programs that run on the workstation itself under the MS-DOS or BTOS operating system.

The INFOVIEW II environment enables you to transfer files between A Series hosts and your workstation. The INFOVIEW II environment can transfer binary and text files; for more complex types of transfers, you can use the Data Transfer System (DTS) instead. DTS is described under "Data Transfer and Reformatting" in this section.

The INFOVIEW II environment also provides a printing feature. You can use this feature to dump the contents of INFOVIEW II screens to a printer, or to print any file stored on the workstation.

### Documentation

- *Workstations INFOVIEW II Operations Guide*
- *Workstations INFOVIEW II Programming Reference Manual*

## Data Transfer and Reformatting

The Data Transfer System (DTS) enables you to transfer data automatically between an A Series system and workstations. DTS is available in three different release levels: release level 3.0, release level 4.1, and release level 5.0.

### DTS Release Level 3.0

Release level 3.0 of DTS comes in two versions: BTOS DTS and DOS DTS. The BTOS DTS version supports the BTOS family of workstations running the BTOS operating system. The DOS DTS version supports microcomputers that run the MS-DOS operating system, including Unisys PCs, IBM PCs, and the BTOS family of workstations. (The latter line of computers can run both the BTOS and the MS-DOS operating systems). DTS software is divided into a mainframe module and a workstation module. BTOS DTS and DOS DTS use the same mainframe module but different workstation modules.

Both BTOS DTS and DOS DTS allow you to copy workstation files to the A Series mainframe or vice versa. When you copy a file to the mainframe, the file is created as a sequential file that can be used by application programs on the mainframe.

When you use BTOS DTS to copy a file to a BTOS family workstation, you can request that the file be created in any of the following formats: direct access method (DAM), indexed sequential access method (ISAM), sequential access method (SAM), Symbolic Link (SYLK), or data interchange format (DIF). Workstation products that can process BTOS DTS output files include the Multiplan and Executive WRITEone products.

When you use DOS DTS to copy a file to a Unisys or IBM PC, you can request that the file be created in any of the following formats: sequential, ASCII text, SYLK, or DIF formats. Workstation products that can process DOS DTS output files include Lotus 1-2-3 and Multiplan software.

DTS provides menus that allow you to define the characteristics of the information to be transferred, and to save this definition on the PC. Thereafter, you can initiate a transfer request using this definition whenever desired.

DOS DTS also provides a terminal emulator that allows you to use your workstation as if it were an ordinary terminal attached to the mainframe. The BTOS Terminal Emulator (BTE) provides a similar function for systems running the BTOS operating system.

### DTS Release Level 4.1

Release level 4.1 of DTS is called the Personal Workstation<sup>2</sup> Data Transfer System (PW<sup>2</sup> DTS), and it supports the following types of workstations: the Unisys PC, IBM PC, and Unisys B 38 systems running the MS-DOS operating system under the BTOS II operating system. PW<sup>2</sup> DTS provides all the capabilities of DOS DTS, plus the following additional features:

- *Additional workstation file types*, including the following types: DOS Random, BASIC Sequential, BASIC Random, dBASE III<sup>®</sup>, and No Conversion (binary) files.
- *Additional mainframe file types*, including the following types: indexed sequential access method (ISAM) and KEYEDIOII. DTS can also create source code files of types recognized by CANDE, so that you can compile these files on the mainframe without having to do any manual reformatting first.

---

dBASE III is a registered trademark of Ashton-Tate Corporation.

## Transaction Management

---

- *File format descriptions.* You can define and store file format descriptions in the Advanced Data Dictionary System (ADDS) on the mainframe. You can define fields in the file to be of different data types and use DTS to select, extract, and order specific fields of the file.
- *Batch operation.* This feature enables you to initiate a transfer of multiple files between the workstation and mainframe, instead of having to manually submit individual file transfers.
- *Virtual disk.* This feature stores information on the mainframe and makes it appear to reside on a workstation disk drive. Virtual disk enables the workstation user to access the larger capacity of mainframe disks. Virtual disk also enables the workstation user to share workstation files with other workstation users simply by using DOS file commands, such as the COPY command.
- *Virtual printer.* This feature enables you to route files on the workstation to as many as three printers on the mainframe.
- *Printer pass-through.* This feature enables you to print an A Series print file on a printer locally attached to a workstation.
- *DOS Terminal Emulator (DTE).* This enhanced terminal emulator provides key programming, flexible screen configuration, and a Dialog Manager that allows you to manage multiple dialogs to an A Series mainframe.
- *T1000 integration.* This feature gives you the option of using Intercomputer Communications Corporation (ICC) T1000 or T1000+ communications boards to link the workstation to the host.

### DTS Release Level 5.0

Release level 5.0 of DTS is called the Personal Workstation Data Transfer System (PW DTS) and can be used in the following environments: MS-DOS, BTOS, and OS/2™ operating systems. Release level 5.0 has the same feature content of release level 4.1, plus the following enhancements:

- *Macro capability.* You can create macros with high-level names and use them individually, or combine them to form your own tailored functions.
- *Dynamic key binding.* You can easily associate a function with a key and list online the key-to-function assignments that you make.
- *Color can be assigned to many terminal attributes.* If your terminal and operating system support the feature, you can assign colors to many of the terminal attributes, such as normal, highlight, underline, blink, and reverse video displays.
- *File Transfer utility (FTU) can use wild cards.* A host request for a list of mainframe file names can include a name with wild cards, as well as the optional usercode and family name.

### Documentation

The following documents apply to release level 3.0 of DTS:

- *BTOS Data Transfer System (BTOS DTS) Operations and Programming Guide*
- *DOS Data Transfer System (DOS DTS) Operations and Programming Guide*

The following documents apply to release level 4.1 and 5.0 of DTS:

- *Personal Workstation Data Transfer System (DTS) Installation and Configuration Guide*
- *Personal Workstation Data Transfer System (DTS) Operations Guide, Volume 1: Simple Transfers*
- *Personal Workstation Data Transfer System (DTS) Operations Guide, Volume 2: Advanced Operations*
- *Personal Workstation Data Transfer System (DTS) System Messages Support Reference Manual*



# Section 5

## Program Development

The A Series supports a full range of industry-standard programming languages. But the A Series provides far more than language support to the application programmer. A broad range of program development tools help you to implement, compile, test, and debug programs with the least effort possible.

To understand the A Series programming environment, you must first understand some basic terms. Computer languages are often categorized according to the “generation” to which they belong. There are four generations:

- *First generation (machine languages).* The oldest, or first-generation, computing languages are *machine languages* consisting of numeric instructions that are directly recognized by the processor. Machine language programs are very lengthy and difficult for humans to understand.
- *Second generation (assembly language).* Assembly language statements are mnemonics that each correspond to a single machine language instruction. Assembly language programs are as long as machine language programs, but somewhat more intelligible due to the use of mnemonics.
- *Third-generation.* These languages resemble English-like statements, such as *READ A RECORD*. This high-level statement corresponds to a whole series of assembly language instructions. Before a third-generation program can be executed, it must be processed by a compiler or an interpreter. The compiler or interpreter translates the program into the appropriate machine language instructions. Third-generation languages include COBOL, FORTRAN, and Pascal.
- *Fourth-generation.* To further simplify program development, a fourth-generation language allows you to use a nonprocedural phrase such as *SORT THE ADDRESS FILE*, instead of using statements to specify each step in a procedure. The language is so powerful that it knows how to tell the computer to do that for you; there is no need for you to describe the procedure step-by-step. Fourth-generation languages increase programmer productivity, and they also make it possible for business analysts and others who are not programmers to develop applications.

The A Series program development strategy maximizes productivity by heavily emphasizing the use of fourth-generation languages and by providing a variety of programmer productivity tools for use with third-generation languages. Because the A Series is optimized for the execution of high-level languages, assembly language is neither needed nor provided. For details, refer to “Operating System Concepts” in Section 2.

This section first describes the A Series fourth-generation programming environment and then describes the third-generation programming environment.

# Fourth-Generation Program Development

Fourth-generation software development tools can make it possible to develop applications in as little as a tenth of the time it would take to develop the same application through conventional methods. The benefit for your company is more than just an increase in programmer productivity. By speeding the process of developing or modifying applications, fourth-generation programming makes it possible for you to respond quickly to business opportunities before they disappear.

The A Series offers two systems for generating applications in fourth-generation programming languages: the Logic and Information Network Compiler II (LINC II) and MAPPER systems. Each has its own unique virtues. The following pages give an overview of these systems and explain the features that make these products unique among fourth-generation languages.

## Logic and Information Network Compiler II (LINC II)

Among fourth-generation language systems, the LINC II product offers the unique advantage of enabling a business analyst to create a complete, integrated business system (rather than a single application). The business analyst should have a good understanding of the business itself, but does not need extensive data processing expertise. The following description is relative to release level 14.0 of the LINC II product.

The LINC II product has two distinct but complementary functions. First, it provides a high-level business definition language that enables you to define your organization in terms that are business-oriented. Second, the LINC II product translates the business definition into the computer programs that uniquely satisfy the business needs you described.

The LINC II design philosophy is founded on the belief that an organization's information system must be based on the organization's activities rather than on a model of the data used by the organization. The first step in LINC II application development is for you to work directly with end users to describe the flow of information through the organization. You then record the results of your analysis using simple input specifications in the LINC Definition Language (LDL). You describe your information flow in LDL using three basic concepts:

- *Components.* Elements that describe the infrastructure of the business.
- *Events.* The activities and transactions that occur in the course of business.
- *Profiles.* The views of information that reflect the needs of managers and end users.

The LINC II product then uses this information to generate a complete online data management application system, including a Data Management System II (DMSII) database, COBOL74 inquiry and reporting programs, and a complete data communications network specification.

Figure 5-1 gives a conceptual overview of the LINC II development process. The business analyst writes an LDL that represents Part A1 as a component, Sale S1 as an



event, and Management Report M1 as a profile. The LINC II product uses these and other elements of the LDL specifications to generate the complete application system.

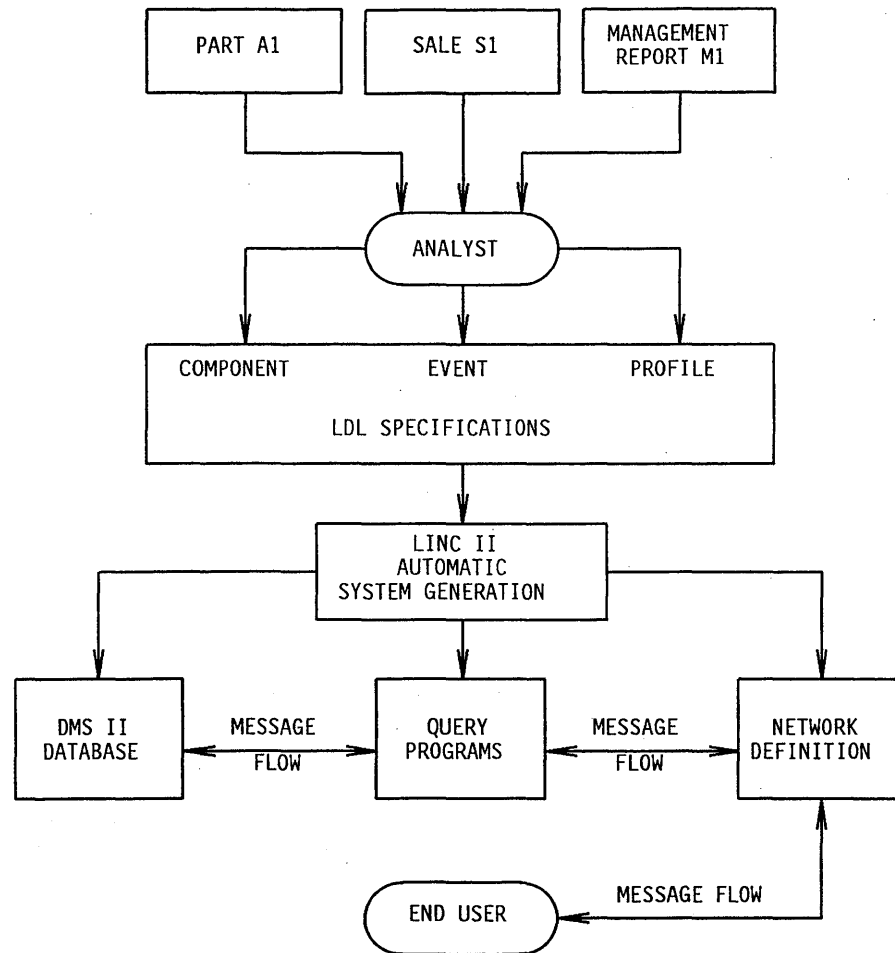


Figure 5-1. LINC II Development Process

The LINC II product produces transaction-oriented systems that facilitate online, real-time service. Because the LINC II product implements the system through existing A Series features such as DMSII, LINC II applications can offer performance comparable to application systems developed through conventional means.

Features of LINC II applications include

- Formatted screen interfaces to inquiry, update, and reporting functions.
- Full prototyping capabilities that involve the end user in all stages of the development of the system.
- Ability to display the interfaces in up to four natural languages concurrently. (A natural language is a language such as English or French.)
- Sophisticated data integrity and recovery strategies, including concurrency control.

## Program Development

---

- Distributed processing capabilities that integrate microcomputers and financial system devices (such as teller machines and magnetic card readers) into the LINC application.
- An option of using the Interactive Query Facility (IQF) or the Workstation Query Facility (WQF) to query the database. A one-time run of the LINC.View utility converts your LINC II application database to a form that is compatible with IQF and WQF and that also allows ongoing LINC II transactions.
- The LINC II product also provides its own *Ad Hoc Query*. Ad Hoc Query implements a subset of the Structured Query Language (SQL). With Ad Hoc Query, an online user can formulate his own queries against the database to supplement more formal, predefined reports.

Because the LINC II product is machine independent, the ease of progressing to another system is guaranteed as your transaction volumes increase. For example, an application system generated by the LINC II product that runs on an A 6 can be transferred to an A 12 system without any modification to the LDL specifications.

The speed and efficiency of the LINC II product facilitate enhancements, changes to the database, production of new reports, and future migrations. The LINC II product also makes customization convenient, fast, and cost-effective.

### Documentation

- *A Series LINC II Installation & Configuration Guide*
- *A Series/V Series LINC II Administration & Operations Guide*
- *A Series/V Series LINC II Customizer Programming Guide*
- *A Series/V Series LINC II Development Operations Guide*
- *A Series/V Series LINC II Network Implementation Guide*
- *A Series/V Series LINC II Programming Reference Handbook*
- *A Series/V Series LINC II Programming Reference Manual*
- *A Series/V Series LINC II Workstation Operations Guide*
- *LINC II Executive Overview*
- *LINC II Run Time Transfer Operations Guide*
- *LINC II Technical Overview*

In addition, the *A Series InfoExec LINC.View Operations Guide* is published relative to the A Series Mark 3.9 system software release.

### MAPPER

Among fourth-generation language systems, the MAPPER system is the most effective at putting the full power of a computer system directly under the control of the end users of that system. The name *MAPPER* is derived from the ability of the system to map computer capabilities onto real world tasks and concepts. The MAPPER system provides this mapping through powerful predefined functions that you can call on as needed, without having to write an application program.

Data structures in the MAPPER system are organized around concepts with familiar analogs in the real world, such as file cabinets, drawers, and reports. The MAPPER system generates a database for you using its own database management system.

The MAPPER system provides over 100 predefined functions. You can use these functions to perform tasks such as sorting, addition, comparison of values, and reformatting. You can also use one data item as an index to find the value of a related data item. The MAPPER system guides you through the process of generating queries. For example, if you specify the name of a report and a drawer, the MAPPER system displays the names of the fields that can be queried in that report.

The most common MAPPER functions can be quickly learned, as they parallel existing real-world activities. In fact, most people can do productive work with the MAPPER system after only a couple of hours of hands-on training on a test cabinet of reports, supplied with the product.

As you gain more experience, you can build customized functions, called RUNs, to easily invoke frequently used sequences of predefined MAPPER functions. RUNs can invoke other RUNs, passing parameters when necessary. You can implement entire applications using MAPPER RUNs.

The MAPPER system provides flexible data storage and retrieval capabilities, and even RUN programming directly to you. This allows more flexibility in the relationship between data processing personnel and end users. For example, the data processing personnel can create applications and allow the end users to modify these applications to suit their changing needs. The lack of ongoing maintenance work allows the data processing personnel to spend more time developing new applications.

The MAPPER system is already used by more than 2,000 organizations serving more than 500,000 end users around the world, in a wide range of industries. The A Series implementation of the MAPPER system includes the following features:

- MAPPER Connect, which is the user interface to the MAPPER system. MAPPER Connect runs on the BTOS family of workstations or on Unisys or IBM PCs. MAPPER Connect uses INFOVIEW II software to provide pulldown menus, color, support of a mouse, and integrated key programming. MAPPER Connect also includes a terminal graphics package when run on Unisys or IBM PCs.
- Full database recovery using audit trails based on disk or tape.
- Formatted display of MAPPER reports. When you request a MAPPER report, the MAPPER system presents a formatted display ready for manipulation, such as sorting and searching, or for updating, with no programming logic or special formatted screen generation required.

- The online mail feature, which enables you to send messages, results, or complete reports to other users of the same MAPPER system, users of a different MAPPER system on the same A Series system, or MAPPER users located on other A Series systems connected to this one by a BNA link.
- Security features that enable you to encrypt reports, station-to-station messages, and intersystem data transfers.
- An online help facility that displays information about all MAPPER functions, and provides helpful information in response to errors.
- Access to mainframe-based word processing capabilities, without the need for separate word processing terminals.
- Ability to copy from the MAPPER database to A Series files, or from A Series files into the MAPPER database.
- Ability to initiate A Series WFL jobs or application programs from the MAPPER system. Output files created by A Series application programs can be returned to the MAPPER system for further processing.

## Third-Generation Program Development

The A Series provides four significant types of support for third-generation program development: compilers for the most popular programming languages, language interfaces to other software products (such as databases), language extensions that take advantage of A Series architectural features, and program development tools. All these are reviewed in the following pages.

### Programming Languages

Most of the languages supplied by Unisys conform to international standards, when they exist, or to American National Standards Institute (ANSI) standards. If there are extensions or exceptions to the standards in the Unisys implementation of a language, they are usually noted as such in the reference manual for that language. This policy of adherence to standards increases the portability of application programs between the A Series and other types of systems.

The following paragraphs describe the programming languages that are available for use with A Series systems.

#### ALGOL (Algorithmic Language)

ALGOL is a general purpose, block-structured programming language. A Series extended ALGOL is used for both application and system programming and is based on the "Revised Report on the Algorithmic Language ALGOL 60."

### Documentation

- *A Series ALGOL Programming Reference Manual, Volume 1: Basic Implementation and Volume 2: Product Interfaces*
- *A Series ALGOL Test and Debug System (TADS) Programming Guide*

## C Language

C is a general-purpose programming language that offers flexible handling of data types and is useful for both high-level and low-level programming. C is widely used in the computing industry and provides a high degree of portability across different machines and vendors. The A Series implementation of C conforms to ANSI standard draft X3.159-1989.

### Documentation

For further information, refer to the *A Series C Programming Reference Manual*.

## COBOL74

COBOL is designed for the implementation of business-oriented data processing programs. The A Series supports two implementations of COBOL. The oldest supported implementation of COBOL is called COBOL74 and conforms to ANSI standard X3.23-1974.

### Documentation

- *A Series COBOL ANSI-74 Programming Reference Manual, Volume 1: Basic Implementation and Volume 2: Product Interfaces*
- *A Series COBOL ANSI-74 Test and Debug System (TADS) Programming Guide*

## COBOL85

COBOL is designed for the implementation of business-oriented data processing programs. The newest of two current A Series implementations of COBOL is called COBOL85 and conforms to ANSI standard X3.23-1985.

### Documentation

- *A Series COBOL ANSI-85 Programming Reference Manual, Volume 1: Basic Implementation and Volume 2: Product Interfaces*

## Data Communications ALGOL (DCALGOL)

DCALGOL is a specialized superset of A Series ALGOL that provides features for writing message control systems (MCSs), performing system operations functions, and monitoring system performance.

## Program Development

---

### Documentation

For further information, refer to the *A Series DCALGOL Programming Reference Manual*.

## Data Management ALGOL (DMALGOL)

DMALGOL is a specialized superset of A Series ALGOL that provides features for implementing the system software that supports DMSII, SQLDB, and SIM databases. Note that this language is not used for writing database application programs. Refer to "Database Interfaces" in this section for details about database application programs.

### Documentation

For further information, refer to the *A Series DMALGOL Programming Reference Manual*.

## FORTRAN77 (Formula Translation ANSI-77)

FORTRAN was designed for scientific applications. FORTRAN77 is the A Series implementation of FORTRAN and conforms to ANSI standard X3.9-1978.

### Documentation

- *A Series FORTRAN77 Programming Reference Manual*
- *A Series FORTRAN77 Test and Debug System (TADS) Programming Guide*

## NEWP

NEWP is a variation of ALGOL that is used to write the operating system and system utility programs. NEWP provides additional language features that are beyond what ALGOL provides, such as Modules and more extensive data typing.

### Documentation

For further information, refer to the *A Series NEWP Programming Reference Manual*.

## Pascal

Pascal is a block-structured, general-purpose language. Pascal implementation is based on the ANSI/Institute of Electrical and Electronics Engineers (IEEE) standard 770X3.97-1982 and on the International Standards Organization (ISO) 7185 Level 0.

### Documentation

For further information, refer to the *A Series Pascal Programming Reference Manual, Volume 1: Basic Implementation* and *Volume 2: Product Interfaces*.

## RPG (Report Program Generator)

RPG is a language for accessing and processing data and specifying the type and format of reports to generate using this data.

### Documentation

- *A Series Report Program Generator (RPG) Programming Reference Manual, Volume 1: Basic Implementation and Volume 2: Product Interfaces*
- *A Series Report Program Generator (RPG) Programming Template*

## Work Flow Language (WFL)

WFL is the Unisys language used for job control. WFL is a high-level, ALGOL-based language that can initiate and control compilations or the execution of production jobs. WFL features include

- Variables and expressions that can be used to manipulate parameters to programs and monitor the status of program executions.
- Flow-of-control statements such as IF statements and DO or WHILE loops, as well as subroutines and recursion. Subroutines and programs can be executed sequentially or in parallel with the WFL job.
- Statements that provide library maintenance functions, such as copying or removing files.
- Statements that specify actions to be taken after a program failure or system failure.

For further information about WFL, refer to “Automated Program Compilation and Testing” in this section and “Batch Job Controls” in Section 7.

### Documentation

- *A Series Task Attributes Programming Reference Manual*
- *A Series Task Management Programming Guide*
- *A Series Work Flow Language (WFL) Programming Reference Manual*

## SORT Language

The SORT language lets you write programs to sort or merge files on A Series systems. The SORT language allows you to sort files without going through the operating system sort intrinsic, thus providing some measure of application independence.

### Documentation

For more information, refer to the *A Series SORT Language Programming Reference Manual*.

### Product Interfaces

The A Series implementations of most standard programming languages include extensions that allow these languages to interface with various major A Series products, such as database management systems and message control systems. These product interfaces are described in the following pages. Other types of language extensions are discussed under “Language Extensions” in this section.

### Database Interfaces

If you use SIM databases at your site, you can query or update them with programs written in ALGOL, COBOL74, or Pascal. The A Series implementations of each of these languages include special data management statements that perform functions such as opening a database, selecting database records, beginning and ending transaction state, and inserting, deleting, or modifying database entries. These database management extensions are referred to collectively as *host language interfaces* to A Series databases.

The compilers for all these languages interface with the database software and are able to extract information about the database structure. The result is that the application programmer can refer to database entities simply by name, without including any physical structural descriptions. The host language interface in these languages also allows the same program to have multiple databases open concurrently.

Host language interfaces are also provided for DMSII databases. The languages that provide a DMSII interface are ALGOL, COBOL74, COBOL85, and RPG. The same program can have SIM and DMSII databases open at the same time.

DMSII databases can also be accessed indirectly through the DMSII interpretive interface. The DMSII interpretive interface enables you to write DMSII application programs in languages that do not have DMSII extensions, including FORTRAN77 and Pascal. The DMSII interpretive interface generates a tailored library for each database, which provides run-time database access for application programs.

Another product that is useful for database applications, as well as other types of programs, is the Advanced Data Dictionary System (ADDS). ADDS is discussed under “Centralized Data Definitions” in this section.

#### Documentation

- *A Series DMSII Interpretive Interface Programming Reference Manual*
- *A Series InfoExec Semantic Information Manager (SIM) Programming Guide*

### Centralized Data Definitions

You can create centralized definitions for a variety of types of data structures by using the Advanced Data Dictionary System (ADDS). You can design application programs to import these data definitions from ADDS, rather than having to repeat the data definitions in each program. The types of data definitions stored can include DMSII or



SIM record and field descriptions, Screen Design Facility Plus (SDF Plus) message field format definitions, or COBOL74 file structures. The languages that can import data definitions from ADDS are ALGOL, COBOL74, and Pascal.

For further information about ADDS, refer to “SIM Database Creation and Reorganization” and “DMSII Database Creation and Reorganization” in Section 3.

### Documentation

The *A Series InfoExec Advanced Data Dictionary System (ADDS) Operations Guide* explains how to use ADDS.

## Multilingual Interfaces

You can use the Message Translation Utility (MSGTRANS) to translate application program output messages from one natural language to another. For each program, you can store all the output messages in as many different languages as you need. The messages themselves must be stored in an ALGOL or NEWP routine. However, you can use the Binder utility to insert an ALGOL message routine into programs written in other languages. Alternatively, you could write an ALGOL or NEWP library that exports the message routine to programs written in other languages.

When the application program is run, it dynamically selects the language to be used for output messages. The administrator can set a system-wide default language, or assign default languages to particular users. The user can override the default by requesting a particular language at run time. The selection of languages for a program run is handled by a set of operating system features called the MultiLingual System (MLS).

Figure 5-2 shows an example of an application program that sends a greeting message. At run time, MLS determines the language in which to display that message for each end user.

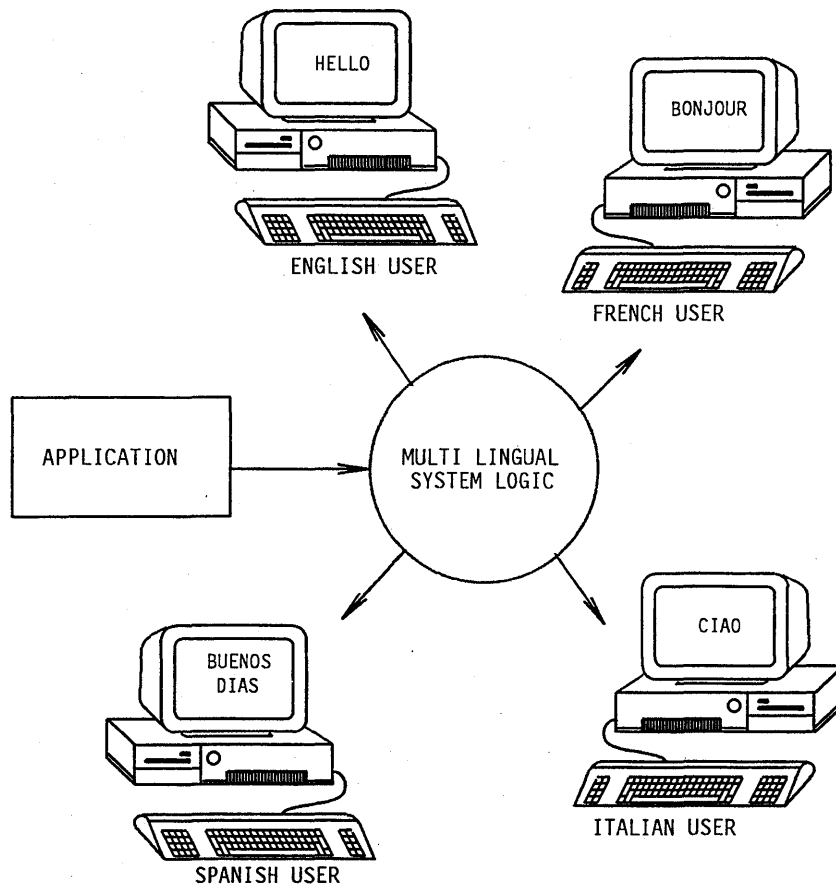


Figure 5-2. Multilingual Interfaces

You can use the Screen Design Facility Plus (SDF Plus) to design a menu-driven user interface for your program. You can choose among various language versions of the user interface at run time. SDF Plus is discussed under "User Interface Generation" in this section.

For information about multilingual operating system interfaces, refer to "Multilingual System Interfaces" in Section 6.

### Documentation

- *A Series Message Translation Utility (MSGTRANS) Operations Guide*
- *A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide*

### Transaction Processing Interfaces

The A Series implementations of ALGOL, COBOL74, COBOL85, Pascal, and RPG all feature extensions that allow applications written in these languages to take advantage of Communications Management System (COMS) transaction handling features. Some of these features include

- Message routing by transaction codes.
- Preprocessing and postprocessing of messages by *processing items*. These are library procedures that you can develop either in ALGOL or by using the Screen Design Facility Plus (SDF Plus).
- Security checking of input and output messages.

For more information about the COMS features that provide these capabilities, refer to “Generalized Transaction Control” in Section 4.

### Documentation

For detailed information about COMS programming, refer to the *A Series Communications Management System (COMS) Programming Guide*.

### Language Extensions

In addition to the major product interfaces already described, A Series implementations of programming languages include a number of major extensions that make it easy for programs to access the advanced features of A Series architecture.

### Input/Output Features

The portion of the A Series hardware system that manages program input and output is referred to as the I/O subsystem. The A Series I/O subsystem is designed to simplify program I/O operations as much as possible, allowing the programmer to concentrate on factors of real significance. For example, the programmer need not be concerned with connections to peripheral units; the I/O subsystem establishes the physical address of a file at run time. Thus, programs are not limited to running under a particular system configuration. Many of the differences among the various types of peripherals are managed completely by the I/O subsystem. Changing a program to write to a different type of peripheral (for example, to disk instead of card punch) can be as simple as a one-line change.

Files are described in A Series applications by entities called *file attributes*. A Series file attributes are available under the same names in all A Series programming languages. You can code file attribute assignments into a program, but the end user is also free to dynamically alter these attributes at run time through a feature called *file equation*. File equations give flexible program behavior without the necessity of recoding or recompiling.

The A Series supports two implementations of the indexed sequential access method (ISAM), which optimizes I/O performance for files that are accessed in both random and

serial fashion. The software that implements this method for ALGOL is called ISAM, and is accessed through a language support library.

KEYEDIO II is the name of the software that implements ISAM for COBOL74, COBOL85, and RPG. KEYEDIO II can also be accessed indirectly through library calls from languages that support libraries, such as ALGOL. The KEYEDIO II implementation uses DMSII features to maximize performance.

### Documentation

- *A Series File Attributes Programming Reference Manual*
- *A Series KEYEDIOII Programming Reference Manual*
- *A Series System Software Utilities Operations Reference Manual*

## Run-Time Sharing of Program Modules

It is sometimes useful at run time to invoke program modules from other programs. This can be done through the use of libraries.

A library is a program that provides a set of useful routines that can be shared by many other programs. The use of libraries makes it unnecessary for you to duplicate code in all the programs that need to use a common type of routine.

When a library program is initiated, it becomes memory resident; this means that while the library is in memory, one or more user programs can call on procedures located in the library. These exported procedures are executed separately from the library, as a part of the user program.

The user program can pass parameters to and receive results from these procedures. Nevertheless, these procedures can access global data that is stored only in the library. This combination of capabilities makes libraries a very powerful method for

- Providing a set of useful routines that can be shared by many different programs
- Making a routine written in one language accessible to programs written in different languages
- Controlling access to a shared resource, such as a file
- Routing interprocess communications (IPC) information between programs that are running independently of each other

Figure 5-3 shows a library that is shared by two user programs, UP1 and UP2. The procedure P2 is declared by the library program but is exported and executed separately by UP1 and UP2. Procedure P2 can do such things as write to file F1, which is declared in the library program.

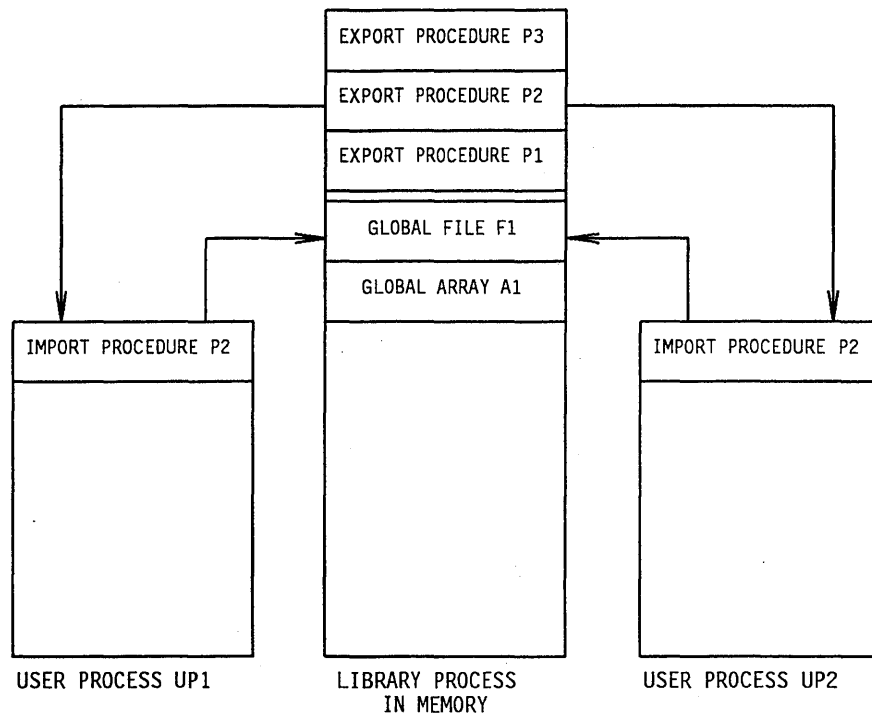


Figure 5-3. Libraries and User Programs

It is easy to update libraries without affecting user programs, because the linkage between library and user program is not established until run time.

A variety of intrinsic functions are provided on the A Series by system libraries. By linking to a system library, a user program can access functions that are not available in the language in which the user program is written.

### Documentation

For further information about libraries, refer to the *A Series System Software Utilities Operations Reference Manual*.

## Compile-Time Sharing of Program Modules

The ability to invoke a shared program module at run time was introduced under "Run-Time Sharing of Program Modules" in this section. A related capability is that of permanently inserting a module from one compiled program into another compiled program. You can accomplish this kind of module sharing by using the Binder utility.

The Binder allows you to reuse common modules in a variety of different application programs. The Binder also allows you to combine modules written in different languages in the same program. The languages that can be bound together include ALGOL,

COBOL74, COBOL85, and FORTRAN77. Pascal modules can also be bound, with some restrictions.

A utility closely associated with the Binder is the PRINTBINDINFO utility. PRINTBINDINFO can print an analysis of certain data related to binding that resides within a code file.

### Documentation

For further information, refer to the *A Series Binder Programming Reference Manual*.

## Task Initiation and Control

The A Series includes a sophisticated set of capabilities for task initiation and control. Using programs written in ALGOL, COBOL74, or Work Flow Language (WFL), you can

- Initiate internal procedures or separate programs as coroutines, asynchronous (parallel) processes, or independent processes
- Monitor and control the status of the procedure or program that was initiated, and determine whether termination was normal or abnormal
- Use synchronization facilities referred to as *events* to coordinate the activities of asynchronous processes

Task monitoring and control functions are provided by features called *task attributes*, which have the same names and functions in each language that provides them. Task attributes can be assigned or read by WFL jobs and ALGOL or COBOL74 programs. You can also assign task attributes when you initiate a program; such task attribute assignments are referred to as *task equations*.

### Documentation

- *A Series System Administration Guide*
- *A Series Task Attributes Programming Reference Manual*.
- *A Series Task Management Programming Guide*

## Interprocess Communications (IPC)

Separate programs can use any of several communication methods to exchange information as they execute. These interprocess communications (IPC) techniques make it possible for you to effectively distribute work among separate application modules while allowing these modules to interact and cooperate when necessary.

One of the primary IPC techniques is the use of *port files*. A port file dynamically establishes a link between two or more programs for the routing of messages. The programs can read from or write to the port file as if it were a disk file. However, the port file resides in memory rather than on any peripheral device. Port files can be used in any of the following languages: ALGOL, COBOL74, COBOL85, FORTRAN77, NEWP,

Pascal, and RPG. The programs that communicate through a port file do not have to be written in the same language.

Another IPC feature is the ability of one program to pass a parameter to another program at initiation time. Parameters can be passed by reference, which means that either program can change the value of the parameter, and the change will be visible to the other program. Parameters can be passed between programs written in different languages. The operating system automatically matches equivalent parameter types provided in different languages.

Other types of IPC methods include task attributes and events, which are discussed under "Task Initiation and Control" in this section. The use of libraries in IPC is discussed under "Run-Time Sharing of Program Modules" in this section.

### Documentation

- *A Series Task Attributes Programming Reference Manual*
- *A Series Task Management Programming Guide*

## Communication across Networks

The BNA Version 1 and BNA Version 2 networking products enable you to distribute application programs across multiple host systems while allowing programs to share resources and communicate with each other. Some of the important capabilities include

- The ability of end users, operators, or application programs to initiate and control programs located on other host systems.
- The ability of application programs to access files residing on other host systems. Directing an I/O to another host is as simple as setting the HOSTNAME file attribute to the name of the other host system.
- The ability of programs on separate hosts to communicate using BNA port files.

### Documentation

- *A Series Distributed Systems Service (DSS) Operations Guide*
- *A Series File Attributes Programming Reference Manual*

## Sorting Capabilities

The SORT intrinsic is a procedure in the operating system that sorts a file or a set of records into a single file of ordered records, or merges a set of presorted files into a single ordered file. SORT can be called from ALGOL and COBOL74 programs or from the SORT language. The SORT language exists to allow you to sort or merge files through direct access to the operating system SORT procedure. SORT programs can be created through CANDE or WFL and can be stored for later use.

### Documentation

- *A Series SORT Language Programming Reference Manual*
- *A Series System Software Utilities Operations Reference Manual*

This manual includes information about the SORT intrinsic.

## Development Environment

A Series program development tools speed the design and testing of application programs by providing features such as user interface generation, an interactive debugging facility, and both workstation-based and terminal-based program development.

### User Interface Generation

Modern programming design emphasizes the presentation of friendly interfaces to the user, including such features as forms with prompts, help text, and intelligent responses to user errors. Unfortunately, developing such interfaces can be a repetitive and time-consuming process. Unisys has addressed this problem by providing the Screen Design Facility Plus (SDF Plus). Using SDF Plus speeds the development of your application interfaces and results in easier-to-use applications. SDF Plus also separates the user interface logic from the data processing application, with the result that the interface can be customized for various environments without affecting the application.

Using SDF Plus, you can “paint” each user screen almost as the end user will see it. You can specify conditions that will cause various form components to be displayed or suppressed at run time. You can provide alternate language versions for each screen and for help text and messages. At run time, you can choose the language appropriate to your site. Help text definition is also an integral part of the design process for each screen.

SDF Plus provides a simple method for defining data formatting and validation rules that isolate or justify input data, compare the data with the permitted range of values, and issue appropriate error responses when necessary.

The user interface you create through SDF Plus is called a user interface system (UIS). The UIS is created and managed by the following SDF Plus modules:

- *SDF Plus Forms Editor*. This module steps you through the process of defining the components of the UIS, including form libraries, forms, and individual fields.
- *SDF Plus Archive Manager*. This module stores and retrieves the UIS definition.
- *SDF Plus Forms Processor*. This module is a driver program that interprets and executes the UIS definition at run time.

Because the SDF Plus Forms Processor interprets the UIS definition at run time, it can execute the new version of a screen immediately after the screen is modified. There is no need to regenerate or recompile the screen library. There is no need to recompile the application either, unless there are changes to the attributes of the data being sent in a



transaction. Even the screen flow can be modified without causing the application to be recompiled.

Substitution tables are another SDF Plus feature that make it easy for you to change a user interface without changing the application. Substitution tables can translate an input message into a different message that is meaningful to the application, or translate an output message into a different message that is more meaningful to the user.

A unique aspect of SDF Plus is that it can take complete responsibility for controlling the screen flow for your application. The application program does not need to contain any logic related to the display of screens. Screen display, screen flow, data validation, and error handling logic can all reside outside the application program in the UIS. SDF Plus even provides a scripting facility that is useful for testing the interface or training users.

The UIS created by SDF Plus has the flexibility to deal with workstation and terminal users. The application program is not affected by the type of device in use, so the UIS can present different interfaces to terminal and workstation users. Workstation users are presented with an interface that takes advantage of INFOVIEW II features, including color, pulldown menus, function keys, and a mouse interface.

Workstation interfaces also have the advantage of reducing the load on the mainframe. For workstation interfaces, the SDF Plus Forms Processor runs as an INFOVIEW II application on the workstation. This method can reduce data comm traffic and improve response time.

SDF Plus can optionally use the Advanced Data Dictionary System (ADDS) to store format definitions for selected message fields. ADDS makes it possible to reuse these format definitions on different screens or in different applications, thus reducing duplication of programming effort. ADDS also keeps track of which forms use each of these definitions, and can produce reports that help you to assess the effects of changing a definition.

SDF Plus is integrated with the Communications Management System (COMS) and can provide interfaces for programs that run in direct windows or remote file windows. For direct window applications, SDF Plus also implements COMS processing items to provide many of the interface functions. SDF Plus thus makes it unnecessary for you to implement these processing items in ALGOL yourself.

A previous version of the SDF Plus product, called simply Screen Design Facility (SDF), is still supported for existing customers. SDF Plus and SDF can coexist on the same system, and SDF Plus includes a migration aid that helps you to convert SDF form libraries to the SDF Plus format.

### Documentation

- *A Series Screen Design Facility Plus (SDF Plus) Capabilities Manual*
- *A Series Screen Design Facility Plus (SDF Plus) Technical Overview*
- *A Series Screen Design Facility Plus (SDF Plus) Installation and Operations Guide*

## Program Development

---

### Interactive Debugging Environment

You can interactively diagnose program errors with the Test and Debug System (TADS). Using TADS, you can control the execution of the software under test and examine or alter the values of variables at any point during program execution. You can interact with TADS in familiar programming terms, such as paragraph names, variable names, and source statement sequence numbers. With TADS, you do not need to be familiar with the details of the hardware architecture of the system.

TADS helps you to establish the completeness of a program test in two ways. First, TADS provides the ability to force execution of any procedure at any time. Second, TADS provides statistics about which statements have been executed, and how many times. TADS batch mode allows you to store a series of identical commands for use over several program runs, eliminating the need to reenter identical commands for each program run.

By using the Message Translation Utility (MSGTRANS) and the MultiLingual System (MLS), you can enable TADS to display its messages in natural languages other than English (such as Spanish and German).

TADS provides a test environment for your program, but failures can also occur in a production environment. The A Series provides the PROGRAMDUMP facility for capturing information about the status of your program at the time it failed, including the names and contents of all variables and arrays. You can use statements in the program to specify the circumstances under which a program dump should be taken and the types of information that should be included in the program dump. You can direct the program dump to a printer or to a disk file for later analysis by the DUMPANALYZER utility. DUMPANALYZER can produce reports on various aspects of the data collected.

### Documentation

- *A Series ALGOL Test and Debug System (TADS) Programming Guide*
- *A Series COBOL ANSI-74 Test and Debug System (TADS) Programming Guide*
- *A Series FORTRAN77 Test and Debug System (TADS) Programming Guide*
- *A Series System Software Support Reference Manual*

This manual provides information about DUMPANALYZER.

### Workstation-Based Program Development

The Intelligent Distributed Editor (IDE) is the most advanced tool available for creating, compiling, and correcting programs on workstations attached to A Series systems. IDE combines editing functions with access to various compilers and cross-reference information to provide an integrated program development environment. You can use IDE at any of the following types of workstations: the BTOS family of workstations and Unisys and IBM PCs and compatible systems.

IDE uses INFOVIEW II windowing capabilities to create a user-friendly environment that includes pulldown menus, overlapping windows, and function keys whose meanings

are displayed at the bottom of the screen. The word wraparound feature enables you to enter text continuously, without having to transmit on each line.

In a single IDE session, you can

- Edit and view multiple files, or multiple portions of the same file, simultaneously.
- Cause IDE to automatically place COBOL74 or FORTRAN77 programs into a standard format, with all statements correctly indented for readability.
- Initiate a compilation of the current version of the program you are editing, inspect any syntax errors that result, make corrections, initiate a new compilation, and so on. Compilers available from within IDE include ALGOL, COBOL74, and FORTRAN77.

IDE also provides the patching and cross-referencing capabilities described under “Change Control and Cross-References” in this section.

### Documentation

For further information, refer to the *A Series Intelligent Distributed Editor (IDE) Operations Guide*.

## Terminal-Based Program Development

The most advanced tool available for developing programs on terminals attached to A Series systems is the Editor. (IDE, by contrast, offers an advanced program development tool for use at workstations.) While in an Editor session, you can accomplish tasks such as

- Editing text, finding and labeling locations, replacing text targets, copying or moving text, and inserting text from other files.
- Initiating a compilation of the program you are editing, interactively displaying and correcting any syntax errors detected by the compilation, initiating a new compilation, and so on. All A Series compilers are available from within the Editor.
- Submitting WFL jobs and running programs (including the current version of the program being edited).
- Defining sequences of Editor commands or text inputs, called *macros*, which you can thereafter invoke by name.
- Loading multiple dictionaries of your choice for spelling checking.
- Automatically recovering changes made in a previous Editor session that was interrupted by a system halt/load.

The Editor also provides the features for making patches and cross-references described under “Change Control and Cross-References” in this section. Extensive online help text assists you in learning how to use these and all other Editor capabilities.

### Documentation

Refer to the *A Series Editor Operations Guide* for more information.

### Change Control and Cross-References

Certain utilities play a role both in the creation of program text through IDE or the Editor and in the process of compiling programs. Statements about IDE in the following paragraphs apply to the Editor as well.

The first of these utilities is called PATCH. The PATCH utility enables you to update program source files in a very select manner. When you conduct an IDE session in patch mode, IDE does not actually make any changes to the source file. Instead, IDE stores all the changes in a separate file called the *patch file*. This distinction is transparent to you while you are in the IDE session, because IDE displays a virtual file that combines the patch and source. When you initiate a compilation from IDE, IDE uses the PATCH utility to create a new temporary source reflecting the patch changes, and then initiates the compilation of this new source. The practice of patch mode editing makes it possible for many different people to concurrently make editing changes against the same source file. It also makes it easy for you to test the effects of your changes before permanently incorporating them in the source file.

IDE uses files produced by the XREFANALYZER utility to provide cross-reference information for the program you are editing. Using IDE cross-reference features, you can display the line where a particular program item is declared, as well as all the lines where the item is used or altered. You can also traverse directly to any of the lines that are displayed. Cross-reference information is produced in two stages: first, the compiler produces preliminary information; then, this information is processed by the XREFANALYZER utility into cross-reference files that can be used by IDE. You can either initiate XREFANALYZER after the compilation, or request the compiler to run XREFANALYZER for you automatically.

You can also use the INTERACTIVEXREF utility to view cross-reference information without entering an IDE session. INTERACTIVEXREF reads the cross-reference files produced by the XREFANALYZER utility and can display information such as the following about each identifier used by a program: the identifier name and type, the procedure it is declared in, and the sequence number of the line where the identifier is declared.

Figure 5-4 shows the relationship between IDE, the PATCH utility, the XREFANALYZER utility, and the various files used by these programs.

### Documentation

Refer to the *A Series System Software Utilities Operations Reference Manual* for further information about INTERACTIVEXREF, PATCH, and XREFANALYZER.

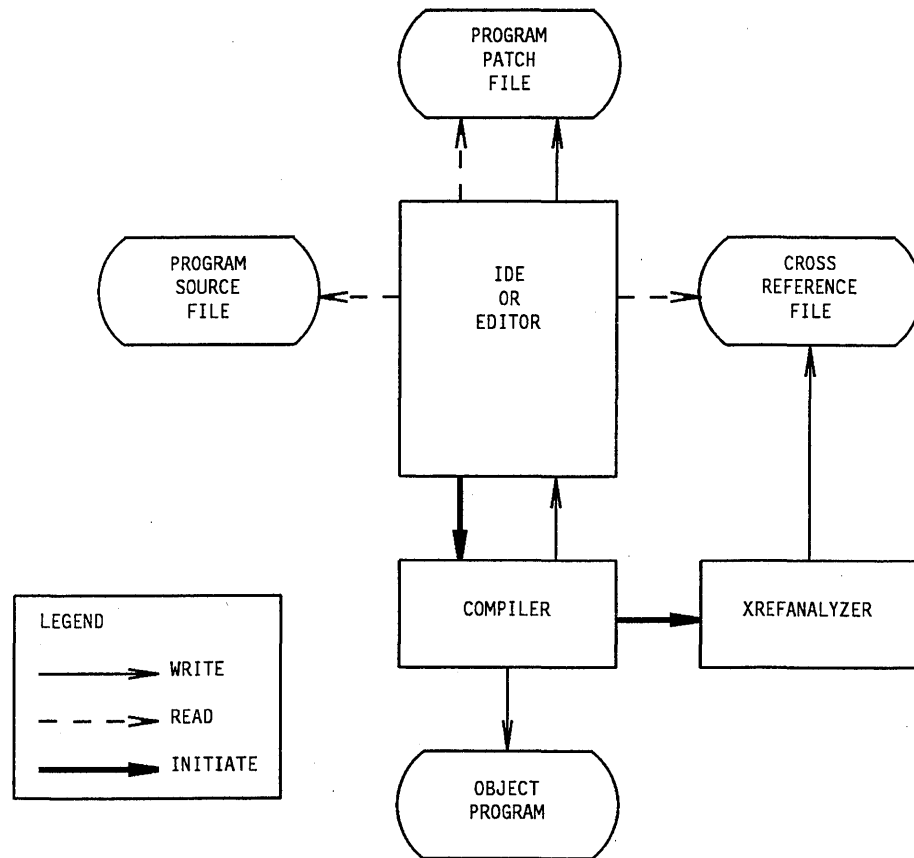


Figure 5-4. Interactive Program Development Environment

### Automated Program Compilation and Testing

You can automate the compilation and testing of programs through the use of WFL jobs. Using the features of WFL, you can write a job that runs PATCH, initiates a compilation, runs XREFANALYZER, and initiates a run of the compiled program. The WFL job can take optional actions depending on the success or failure of each of these tasks. For example, the job could skip the XREFANALYZER run if the compilation failed because of syntax errors. The WFL job can also produce a history listing (called a *job summary*) that records the success or failure of each of the tasks. The use of WFL saves you from having to personally monitor and control the compilation process, and thus frees you for more productive work.

For more information about WFL, refer to “Batch Job Controls” in Section 7.

An alternative to WFL is the use of DO files in Command and Edit (CANDE) sessions. This feature is described under “File Maintenance and Program Control” in Section 4.

### Documentation

- *A Series Task Management Programming Guide*
- *A Series Work Flow Language (WFL) Programming Reference Manual*

# Section 6

## Operational Interface

This section describes system software that handles your day-to-day system operations. These software products streamline many operational procedures, minimizing the time and effort of allocating your computer resources and coordinating the work flow.

The A Series operating system is accessible to operators who have varying levels of expertise. Those who are not familiar with A Series system commands can use menus provided by Menu-Assisted Resource Control (MARC). Those who are familiar with system commands can enter these commands in MARC or in an alternative interface called *system command mode*. An operator can use both interfaces interchangeably and concurrently. Additionally, you can translate the command interface into languages supported by a variety of character sets and cultural convention sets, if needed. For more information see “Multilingual System Interfaces” in this section.

### Menu Interface

Menu-Assisted Resource Control (MARC) provides menus for A Series systems operations functions. MARC is a feature of the Communications Management System (COMS), and is provided with all three versions of COMS. MARC is available at data comm terminals controlled by COMS, and at operator display terminals (ODTs) running in data comm mode.

You can use MARC to perform functions such as

- Monitoring and controlling all jobs and tasks on the system.
- Monitoring and controlling the status of system hardware resources, including all peripherals.
- Controlling the Print System (PrintS) and Remote Print System (ReprintS).
- Controlling the data comm network, including all message control systems (MCSs) such as COMS.
- Controlling multihost networks, including the following types of networks: BNA Version 1, BNA Version 2, Systems Network Architecture (SNA), Transmission Control Protocol/Internet Protocol (TCP/IP), and X.25.
- Performing file maintenance functions such as copying, removing, or retitling files; displaying file titles; and displaying the contents of disk files.
- Initiating programs, such as system utilities or application programs. You can submit batch or online programs, and you can interact directly with the online programs. While the program is still running, you can traverse to a screen that shows the status of the program run, or other screens that allow you to perform system operations functions.

## Operational Interface

---

### Documentation

For a detailed discussion of MARC, refer to the *A Series Menu-Assisted Resource Control (MARC) Operations Guide*.

### Menu Mode and Command Mode

You can use MARC in menu mode or command mode.

In menu mode, MARC presents you with menus displaying selections that correspond to various operations functions. Selections on the MARC home menu correspond to general categories of operations functions. (Refer to Figure 6-1 for an example of the MARC home menu.) When you enter one of these selections, MARC displays another menu that breaks down that operations function in more detail. In most cases you need traverse no more than two or three menus to reach the most specific level. Menu selections at that level cause MARC to automatically execute a system command. If the command needs parameters, MARC prompts you with a form whose fields are clearly labeled and explained.

```
MARC - MENU-ASSISTED RESOURCE CONTROL                                04:08 PM
Action: [                                                            ]
      H0me PRev GO PAREnt COmnd                                     (Press SPCFY for Help)
-----
      Session              Jobs and Tasks              System Software
TEACH What is MARC?      IE InfoExec          DMPS Dumps
NDT  News Date Time    TOOLS Application Tools  LOG  Logging
BYE  Log Off           UTIL System Utilities  MEM  Memory Management
CANDE Cande Window    RUN Run a Task        USERS Usercode/Password
ON   Change Window    START Start a WFL Job  PS   Printing System
SC   Session Control  JQ  Job Queues        SYS  System Control
                                     JD  Job Display
                                     JC  Job Control
      Data Comm              Files              Hardware
SEND Send Messages      SP  Special Programs  IO  I/O Devices
COMS COMS Displays      LIBS System Libraries MM  Memory Modules
BNA  BNA Commands      FILE File Management  PROC Processors
DC   DataComm Control  OTHER Other Devices  CONFIG System Config

Choice: [                                                            ]
```

Figure 6-1. MARC Home Menu

At any point in this process, you can obtain either short help text or longer help text information by pressing a single function key. Help is available for each menu as a whole, for each menu selection, and for each field on a form. In addition, MARC provides an



online glossary of terms. Within the online help text, each term that has a glossary definition is highlighted. You can traverse to the glossary definition by pressing a single function key, and easily return later to the text that you were reading.

MARC also provides a TEACH selection on the home menu for users who are new to MARC. The TEACH selection leads you to tutorials that introduce the A Series, the MARC interface, the online glossary of terms, and the published documentation for the A Series.

As you become more familiar with A Series operations functions, you may wish to execute operations functions without using the menus. MARC provides this capability in either of two ways. First, you can use the *GO* command to traverse to a particular menu without passing through the intermediate menus. Second, you can use command mode.

Command mode allows you to enter system commands directly. These are the commands that MARC would otherwise generate for you, based on your menu selections. It is easy to alternate between menu mode and command mode, because MARC allows you to enter system commands in the Action field of most menus and forms. MARC also helps you to learn the system commands by displaying the system command that was generated by each menu selection. The system command appears at the bottom of the Output screen that MARC displays in response to each menu selection.

## Regulated Access to Operations Functions

MARC security features allow you to prevent unauthorized access to system functions. Each MARC user is required to log on by entering a usercode and password at the start of a MARC session. MARC examines the privileges that were defined for that usercode by the system administrator, and provides any of several levels of access to system commands. If a user does not have access rights to a particular command, MARC suppresses the corresponding menu selection and rejects attempts to use the command directly.

## Customized Operator Interfaces

You can modify the MARC interface to suit the needs of your installation and of individual users at your installation. Commands, menus, and help text can all be created or modified.

MARC provides a *directive* facility that allows you to create new commands and make them available to MARC users. You define the functions of these commands by writing a library of ALGOL procedures. Thereafter, end users can enter these commands in MARC just as if they were MARC commands.

The tool used for customizing menus is the Interactive Menugraph Generator (IMG). IMG is an interactive tool that steps you through the process of creating, modifying, or deleting menus. You can change the wording that appears on existing menu screens, and you can add or delete menu selections. For each menu selection, you can specify the system command it executes or the name of another menu it causes to be displayed. In addition, you can define different sets of menus, called menugraphs, for specific users or groups of users.

IMG can also be used to customize Interactive Datacomm Configurator (IDC) menus; for information about IDC, refer to “Simplified Data Comm Configuration” in Section 8.

The tool used for customizing MARC help text is the Help Utility. Through the Help Utility, you can modify existing help text and write additional help text for any new menus and menu selections added. The Help Utility stores the help text in a file called a *help book*, and allows you to create multiple books. You can use IMG to specify the default help book to be used with each menugraph. In this way you can make different sets of help text available to different users.

You can use IMG and the Help Utility to translate the MARC interface into natural languages other than English (for example, French or Italian). Refer to “Multilingual System Interfaces” in this section.

### Documentation

- *A Series Help Utility Operations Guide*
- *A Series Interactive Menugraph Generator (IMG) Operations Guide*
- *A Series Menu-Assisted Resource Control (MARC) Operations Guide*

## Command Interface

An alternative to the MARC operations interface is the system command mode at an operator display terminal (ODT). System command mode gives you access to all the system commands that are available through MARC. In addition, system command mode gives you access to the following features that are not available in MARC:

- *Automatic display mode (ADM)*. When you initiate ADM, the system automatically displays information about various aspects of system status. The displays can include system messages, memory and processor utilization statistics, the status of peripheral units, and lists of processes that are scheduled, active, waiting, or recently completed. You can obtain the same displays through MARC menu selections or individual system commands. However, the advantage of ADM is that it displays several types of information on a single screen and updates the display automatically. You can specify which types of information will be displayed and how often the display will be updated. Figure 6-2 is an example of an ADM display.
- *Primitive commands*. These are special system commands that are used to restore normal system operations in the event that the system ceases responding to other system commands.

You can also use system command mode to submit Work Flow Language (WFL) jobs or to run utility programs.

System command mode is available at ODTs, at system control terminals (SCTs) that are running in ODT mode, and at data comm terminals that have been authorized for use as remote ODTs. System command mode automatically provides privileged access to system commands without any required log-on procedure. This interface is a convenient and appropriate one for experienced operations personnel, but is not intended for general use by all users. System security features allow you to restrict access to system

command mode if necessary; refer to “Operations Interfaces Access Controls” in Section 9 for details about these features.

### Documentation

For further information about system commands, refer to the *A Series System Commands Operations Reference Manual*.

```

---Mix-Sub-Pri--CPU Time----- 53 ACTIVE ENTRIES -----
4659  4  50      :10 JOB *SYSTEM/PRINT/ROUTER
4648  3  50      :11 (DAVID) (DAVID)OBJECT/UTIL/UPDATE ON ORDSPK
4668  3  50      :05 *SYSTEM/PRINT/BNAROUTER
5570  3  80      :58 JOB NSP109/01

----- NO WAITING ENTRIES -----

----- NO SCHEDULED ENTRIES -----

---Job-Task-Time--Hist----- COMPLETED ENTRIES -----
6212\6374 14:56 EOT (FRANKLIN) *OBJECT/DRIVER ON SYSOPS
6493\6495 14:56 EOT (DAVID) (SARAH)SYSTEM/SCHEMA ON PACK
6493\6494 14:56 EOT (ROHMER) *OBJECT/CHECKDATA ON PACK
6492/6492 14:53 E0J JOB SERVER/LP4/R#1723/J#6464
---Mix-Time----- MESSAGES -----
6492 14:53 LP4: *BD/0006464/0006486/0006487/000LINE ON PACK PRINTED & REMOVED
6486 14:53 PK64 8 FILES REMOVED IN (LINDA)TEMP/6486/= ON SOFTP
6486 14:53 PK64 (LINDA)OUTF/6486/TEMP2 CHANGED TO (LINDA)PATCH/UPDATE/CIRCULA
6486 14:53 PK64 (LINDA)PATCH/UPDATE/CIRCULAR REMOVED ON SOFTP

```

Figure 6–2. ODT Automatic Display Mode

## Coexistence of Operations Interfaces

System operators who have access to operator display terminals (ODTs) have a choice between using the ODT in system command mode or in data comm mode (which provides the MARC window). By entering a single command, you can change the terminal from system command mode to MARC, or vice versa. If you have two or more ODTs attached to your system, you can have some of them running in system command mode, and some of them running MARC. This combination brings you the benefits of menu-driven system access and automatic display mode (ADM) system displays at the same time.

# PCMARC

The A Series provides an operational environment that combines the power of MARC with the windows-style presentation of PC-based workstations. PCMARC is a PC-based workstation environment that consists of a terminal emulator, a communications interface known as COMMLINK, and the PCMARC screen controller, which manages the user interface combination of MARC and the Microsoft Windows environment.

## PCMARC Terminal Emulator

The PCMARC terminal emulator provides the capabilities to allow workstation users to accomplish host-specific functions. The emulator also provides communication with conventional A Series system utilities and application programs by way of PCMARC or CANDE.

## COMMLINK

COMMLINK is the communication interface that is responsible for managing communication between the workstation and the host computer. It controls incoming and outgoing messages between host and workstation, and among multiple workstations.

## PCMARC Screen Controller

PCMARC offers the same functions as those available on an A Series host-based MARC, but uses the presentation style of the Windows environment. The MARC menu screens are presented as pulldown menus, with tasks listed on various menus. PCMARC also provides extensive information about your position in the menugraph, helping you navigate through available MARC screens.

### Documentation

For more information, refer to the *A Series PCMARC Installation and Operations Guide*.

## System Assistant

System Assistant allows you to configure and control the operation of A Series systems. You define your system configuration using an easily understood command language. The system uses your configuration information to perform a variety of tasks that you define. You can define configurations, tasks to be performed, conditions under which to perform them, and messages to display.

During system initialization or whenever your system undergoes a halt/load, System Assistant takes over, assigning various tasks and configuring the system resources as you have previously defined them.

### Documentation

For more information, refer to the *A Series System Assistant Programming and Operations Guide*.

## Automation of Operator Tasks

The A Series allows you to write programs that automate routine operations tasks, thus minimizing the labor required of the operations staff. You can accomplish this goal by using the following products:

- *Work Flow Language (WFL)*. Using WFL, you can write jobs that perform routine file maintenance or that initiate system utilities. You can schedule these jobs to execute at any desired time or date. You can also design a WFL job that restarts itself every day at a specified time.
- *DCALGOL*. The DCALGOL programming language provides three special functions called DCKEYIN, GETSTATUS, and SETSTATUS. You can use these functions to programmatically submit any of the system commands that an operator would normally submit through MARC or system command mode. Depending on which of these functions you use, the command output is returned to the program in textual or encoded format. The program can analyze the output before issuing further system commands.
- *BNA Program Agents*. BNA allows you to write programs that monitor the status of the BNA network and respond to network problems.

### Documentation

- *A Series BNA Version 1 Program Agent Programming Guide*
- *A Series BNA Version 2 User Program Agent Programming Guide*
- *A Series DCALGOL Programming Reference Manual*
- *A Series GETSTATUS/SETSTATUS Programming Reference Manual*
- *A Series Work Flow Language (WFL) Programming Reference Manual*

## Internationalization Functions

The A Series systems provide significant features for support of a variety of languages and character sets. These features are available for use on hardware, firmware, software, and documentation, and allow you to nationalize A Series products to your native language and cultural requirements.

The following features allow you to define and control use of character sets for A Series hardware and software:

## Operational Interface

---

- **CENTRALSUPPORT library**

The CENTRALSUPPORT library provides a series of entry points that perform internationalization functions.

- **CCSVERSION**

CCSVERSION is a file attribute that supports ccsversion attributes. These attributes describe the application of a coded character set to a particular environment. The attributes of a ccsversion indicate such things as uppercase to lowercase mapping and ordering.

- **CCSFILE**

CCSFILE is a system file that serves as a repository for specific information including the ccsversion and coded character set definitions. The operating system and user programs interface to the file through entry points included in the CENTRALSUPPORT library.

- **coded character set**

A set of rules that establish a character set and the one-to-one relationship between the characters of the set and their code values. A Series systems support the following coded character sets:

- EBCDIC-based coded character set
- Double-octet coded character set
- International Alphabet 5 (IA5)
- International Organization for Standardization (ISO) coded character set
- Kanji
- Multi-octet coded character set
- Single-octet coded character set

In addition, the A Series provides the ability to internationalize cultural conventions, such as:

- Date formats
- Time formats
- Currency formats
- Decimal sign formats
- Page sizes

### **Documentation**

- *A Series Message Translation Utility (MSGTRANS) Operations Guide*
- *A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide*

## Multilingual System Interfaces

The A Series systems make it possible for you to customize system interfaces so that they can be displayed in various natural languages, such as French and German. The same A Series system can simultaneously present interfaces in many different languages to many different users. The operating system features, utilities, and compiler extensions that make multilingual interfaces possible are collectively referred to as the *MultiLingual System*.

You create a multilingual interface using the methods discussed in the following pages under the headings “Message Translation,” “Menugraph Translation,” and “Screen Translation.” You then specify the rules used to determine which language interface is presented to the user; these rules are discussed under “Language Selection” in the following pages.

Note that, if you are buying your A Series system through a Unisys distributor or a Unisys International subsidiary, they may already have used the tools described in the following text to customize the A Series software for the language used in your region. Your Unisys International marketing representative can provide you with details.

### Message Translation

You can use the Message Translation Utility (MSGTRANS) to create multiple versions of each output message used by a program and to associate a language with each version of the output message. This utility is an interactive tool that can be used by programmers with varying levels of expertise. You can use MSGTRANS to create multilingual versions of output messages originating from the following sources:

- Components of the operating system: the MCP/AS code file and various system libraries. (For example, you can translate the responses to all system commands entered through either MARC or system command mode.)
- The following programmer tools: the Test and Debug System (TADS) and KEYEDIO.
- Application programs that are written in ALGOL, NEWP, COBOL74, RPG, or Pascal, and that store their output messages in a special type of array called an OUTPUTMESSAGE array. The application programmer is responsible for using this type of array if a multilingual interface is desired.

Once you have created the multiple language versions of the application messages, you can save them and use a utility called the Binder to add them to the object code file of the program. This makes it possible for you to add message translations to a program without recompiling it. For a brief description of the Binder, refer to “Compile-Time Sharing of Program Modules” in Section 5.

### Documentation

For further information, refer to the *A Series Message Translation Utility (MSGTRANS) Operations Guide*.

### Menugraph Translation

You can use a common set of tools to provide multilingual interfaces for MARC, the Interactive Datacomm Configurator (IDC), and BNA Version 2.

Multilingual capabilities are a subset of the capabilities of the Interactive Menugraph Generator (IMG), discussed in “Customized Operator Interfaces” in this section. IMG can provide different menugraphs for different categories of users. One way of dividing these categories is by languages. You can use a standard naming convention to indicate which language a menugraph is associated with. Each menugraph can contain menus displaying text in that language.

When you use the Help Utility to develop help books, you can write the text for each of these help books in a different language and use IMG to specify the default help book for each menugraph. In this way, you can provide online help text in a variety of languages.

#### Documentation

- *A Series Help Utility Operations Guide*
- *A Series Interactive Menugraph Generator (IMG) Operations Guide*

### Screen Translation

The Screen Design Facility Plus (SDF Plus) is a tool used for implementing a forms-based user interface to an application. If you develop your applications using SDF Plus, you can provide multiple language versions of the application interface forms and associated help text. For a description of SDF Plus, refer to “User Interface Generation” in Section 5.

### Language Selection

Once you have implemented the multilingual interfaces discussed in the previous paragraphs, you can use MultiLingual System features to determine which language interface will be presented to each user. The MultiLingual system can simultaneously present an interface in a different language for each user of a product.

Programs can explicitly request the language to be used for an output message or they can allow the system to apply a default language. The system administrator can define a default language for each usercode on the system, as well as an ultimate default language for the whole system. The end user can override these defaults by assigning a language at run time. This flexibility makes it possible to automatically display the language familiar to a particular user without forcibly limiting any user to a single language.

#### Documentation

For further information, refer to the *A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide*.



# Section 7

## Basic System Management

This section provides an overview of A Series system software capabilities for configuring and managing mainframe system resources. The Unisys products described in this section include facilities for managing work flow, memory, disk usage, printing, and system configuration.

### Documentation

For a detailed discussion of system administration issues, refer to the *A Series System Administration Guide*.

## Process Management

A *process* is a single execution of a program, from program initiation to program termination. Usually, each time a user initiates a program, the system creates a new process. If many different users initiate the same program, the system must then create many different processes, which are separate, concurrent executions of the same program. The architectural concepts underlying process execution are introduced under "Stack Architecture" in Section 2.

The A Series is designed to manage many processes simultaneously and efficiently, dividing the available system resources among the processes according to priorities established by the system administrator.

## Work Flow Management

Work flow management is the art of allocating the system resources to favor those processes that are considered most urgent by the customer. On a system running both batch programs and interactive programs, a common goal of work flow management is to ensure that the batch programs do not slow the response time for interactive programs. The A Series provides three basic strategies for solving this and related work flow management problems:

- Dynamic control of online programs
- Process priority assignments
- Batch job controls

### Dynamic Control of Online Programs

The Communications Management System (COMS) provides a direct window interface that enables you to implement multiuser online programs and to control the way COMS

## Basic System Management

---

handles these programs. An online program is one that is designed to accept and respond to commands entered by a user at a terminal. A multiuser online program is one that is designed so that multiple users can be serviced by a single instance of that program (that is, by a single process).

Although the system creates a separate process for each user who initiates a program, users do not initiate COMS direct window applications; COMS initiates them itself. COMS handles all the message routing, and can direct input messages from many users to the same process. All input messages from users of a particular direct window are routed into a single input queue, where they remain until they are read by the process.

COMS monitors the average depth of the input queue. If input messages are being received faster than the process can handle them, then the depth of the input queue begins to grow, and the response time for the end user gets slower. If the input queue exceeds a specified depth, COMS can initiate an additional instance of the application program.

By initiating extra instances of an application, COMS effectively enables the application to use a greater share of the system resources. This dynamic method of performance management helps to ensure a constant service level for online users, even in the face of wide variations in the volume of transactions that are being processed.

You can use the COMS Utility to specify the minimum and maximum number of instances of the program that are allowed, the maximum desirable input queue depth, and the length of time COMS waits before responding to an excessive input queue depth. For an introduction to COMS, refer to "Generalized Transaction Control" in Section 4.

## Process Priority Assignments

The A Series allows you to use priority assignments to specify which of your processes are most urgent. When two or more processes are contending for use of the processor, the operating system assigns the processor to the process that is marked with the highest priority. The system administrator can enforce an overall priority policy by assigning priority limits to particular users or particular jobs. If you run many different types of processes, you can draw very fine distinctions between the priorities accorded to each type of process. This is possible because the A Series allows you to assign up to 100 different levels of process priority within each of two priority categories (normal and control program), for a total of 200 different priority levels.

## Batch Job Controls

In addition to using priority assignments for processes, you can further control work flow by defining resource usage for groups of related processes. To understand these resource usage controls, you must first be familiar with certain basic terms, which are introduced in the following paragraphs.

Aside from system software processes, all processes on the system are of one of two types: *jobs* or *tasks*. From the standpoint of the user, a task is typically a run of some application program that performs a data processing function for the user. A job is

usually a process whose only purpose is to initiate a series of application programs (tasks) in some particular order, perhaps also monitoring the progress of these tasks and helping to synchronize their activities. For example, a job might initiate a compilation of a particular program, initiate the execution of the resulting program code file, and initiate the printing of output generated by the program.

The tasks and jobs on the system fall into related groups called *process families*. A process family consists of a single job, all tasks initiated by that job, all tasks initiated by those tasks, and so on. Whenever a process initiates a job, that job becomes the start of a new process family rather than a member of the process family that initiated it. The process family is an important concept because you can establish resource usage limits that apply to the process family as a whole.

The Work Flow Language (WFL) allows you to write batch jobs that pass through regulatory mechanisms called *job queues*. While it is possible to write batch jobs in ALGOL, COBOL74, and COBOL85, only WFL jobs go through the job queue mechanism. WFL is therefore the preferred medium for writing batch jobs.

Through the job queue mechanism, you can define different classes of service for different types of process families. For example, you can

- Specify the maximum number of processes of each class that can be running at any given time.
- Specify a different priority to be used for each class of processes.
- Limit the total system resource usage for each process family of a particular class. You can establish separate limits for processor time, I/O time, save memory, disk sectors, lines printed, and cards punched.
- Limit the overall batch workload on the system, so that batch programs do not compete excessively with online applications.

Another useful feature of WFL relates to recovery from system halt/loads. If a WFL job is interrupted by a system halt/load, the system automatically restarts the job, which in turn can restart its tasks from the beginning.

## Billing Utilities

Billing is the practice of analyzing the system resource usage of all the users of the system and computing a monetary charge for these users. Billing is necessary at many data processing installations that are funded on the basis of services they provide to users. The A Series provides resource usage information in the form of the system log. You can use either of two utilities to analyze this information and generate a billing report: the System Management Facility II (SMFII) or `LOGGER`. Either of these utilities can analyze data collected over a long period of time and produce a summary of charges for individual users or groups of users.

`LOGGER` calls a library called `BILLINGSUPPORT` to compute the actual charges, based on the resource usage statistics `LOGGER` has collected. Unisys supplies this library in a skeleton form, together with documentation explaining the interface to `LOGGER`. This method allows you to base charges on whatever criteria are appropriate to your site.

Alternatively, you can use the SMFII LOGCONSOLIDATOR utility to collect resource usage statistics from the system log. You can then run the SMFII QUERY utility to create a report that assigns charges to specific users or groups of users, based on criteria such as processor, memory, and I/O usage. SMFII QUERY allows you to specify the rate of charges that should be used for each of these resources.

Log analysis tools are discussed further under “Logging and Log Analysis” in Section 9.

### Documentation

For further information about billing, refer to the *A Series System Administration Guide*.

## Memory Management

The basic concepts underlying A Series memory management were introduced in Section 2, under the headings “Virtual Memory” and “Actual Segment Descriptor (ASD) Memory Architecture.” The following paragraphs discuss the features a system administrator can use to optimize memory efficiency.

The memory architecture provided by the A Series operating system provides the system administrator with several types of performance parameters. The default values of these parameters were chosen to give the best performance in typical situations. However, if you wish, you can use a number of system commands to “fine-tune” these parameters. The system commands used for this purpose are ASD (Actual Segment Descriptor), SF (Set Factors), and SEGARRAYSTART (Array Segmentation Start Size).

You can use the ASD (Actual Segment Descriptor) system command to specify the size of the ASD table used to index the location of code and data segments in memory. The smaller the ASD table, the more space is left in memory for storage of program code and data. However, the ASD table must be large enough to store indexes to all the segments that are in memory. The average number and size of memory segments will vary in different environments, and thus the minimum adequate ASD table size will also vary.

You can use the SF (Set Factor) system command to set any of four different parameters related to memory management. The OLAYGOAL and AVAILMIN parameters together specify a minimum amount of memory the system should keep available at any given time. The FACTOR parameter adjusts compiler-generated estimates of the amount of memory required by each program. A fourth parameter, PRIORITY, controls the effect of process priority in memory contention situations.

You can use the SEGARRAYSTART (Array Segmentation Start Size) system command to specify the array size beyond which the operating system segments an array by default. This command can be used to improve system performance by causing more arrays to be nonsegmented. Accesses to nonsegmented arrays are faster and need less operating system intervention than accesses to segmented arrays.

## Input/Output (I/O) Management

The input/output (I/O) subsystem manages all transfers of information between processes and peripheral devices. The types of peripheral devices supported include disk units, tape drives, data comm terminals, printers, card readers, card punches, and BNA Version 1 stations.

The following pages discuss the facilities for file maintenance, as well as system features related specifically to the disk subsystem, the tape subsystem, and the print system. Data comm features are discussed in Section 8, "Data Communications Networks."

For an introduction to the physical components of the I/O subsystem, refer to "Input/Output (I/O) Subsystem" in Section 2.

### File Maintenance

A Series file maintenance features allow you to easily copy files among various types of media, and facilitate file backup procedures that help you prevent data loss.

### Disk File Maintenance

The A Series features that copy or delete disk files and modify their names or security are collectively referred to as *library maintenance* features. These functions can be performed on either system software files or files created by end users and application programs. You can perform library maintenance through various Work Flow Language (WFL) statements or through CANDE or MARC commands that have the same names as the equivalent WFL statements.

The library maintenance statements used to copy files are the COPY and ADD statements. These statements copy disk files from one disk location to another, between disk and tape, or from one tape to another. These statements differ in the following way: if a file with the same name already exists at the destination, COPY overwrites that file, but ADD does not. You can use options in these commands to direct files across a BNA or TCP/IP network to disk or tape units on a different host system. These commands can also interface to the CATALOGING feature discussed under "Disk File Backups" in this section.

Other library maintenance statements are the CHANGE statement, which changes the names of files on disk; the REMOVE statement, which deletes files from a disk; and the SECURITY statement, which changes the access security of a file on disk.

Most of these statements include the ability to specify multiple files using a single name called a *directory name*. A directory name refers to all files whose titles begin with a common element. Another advantage is that you can easily store groups of library maintenance commands in WFL programs or CANDE *DO* files for repeated use.

The system applies file security rules to these library maintenance statements, thus ensuring that they cannot be used to access files without proper authority.

### Documentation

- *A Series CANDE Operations Reference Manual*
- *A Series Work Flow Language (WFL) Programming Reference Manual*

### Generalized File Maintenance

DUMPALL is a generalized file maintenance utility used primarily for transferring files from one medium to another. DUMPALL also generates listings of files and controls the loading of files from tapes that were not created by library maintenance.

Using this utility, you can copy files from any medium to any other medium. DUMPALL can handle most file types, with a few exceptions such as Data Management System II (DMSII) files and indexed sequential-access method (ISAM) files.

DUMPALL accepts standard and nonstandard tape labels, as well as unlabeled tapes. Tape files created by DUMPALL from disk files are bit-for-bit disk images based on the record size and block size of the disk files. By contrast, tape files created by library maintenance are reformatted to optimize tape usage and to retain the label and header information from a disk file. For this reason, tape files created by DUMPALL can more easily be read by other vendors' systems than tape files created by library maintenance.

You can also use DUMPALL to perform any of the following operations:

- List selected attributes of a file or files (for a more complete listing of disk file attributes, use the FILEDATA utility).
- Concatenate files or parts of files.
- Copy one or more input files to one or more output files on the same or different devices.
- Create a display that shows a file or tape translated into ASCII, EBCDIC, hexadecimal, or other formats.
- Print a file in one of several formats.
- Test a file for I/O errors.

### Documentation

For further information about DUMPALL, refer to the *A Series System Software Utilities Operations Reference Manual*.

### Disk File Backups

Protecting information stored on disk files is a major concern of any computer installation. To achieve adequate protection, you can make copies of disk files on magnetic tape for storage at the site and in secure locations away from the site. The A Series offers three ways to automate the process of managing file backups:

- ARCHIVING
- FILECOPY Utility
- CATALOGING

The ARCHIVE process consists of three major elements:

- BACKUP

You can use the ARCHIVE BACKUP WFL command to perform three types of backup processes:

- Full backup copies all files on an indicated disk pack to tape.
- Incremental backup copies all files that changed or were created since the last backup of any kind.
- Differential backup copies all files that changed or were created since the last full backup.

- RESTORE

You can retrieve and reload tape copies of files to disk with the ARCHIVE RESTORE WFL command. You can indicate specific files you want to restore, or you can restore all of the files that have been copied to tape previously.

- ROLLOUT

The ROLLOUT command allows you to free disk space by copying files from disk to tape. Rollout is another form of backing up files, but the main difference is that you can indicate the number of sectors needed when you initiate the ROLLOUT command. The rollout process consists of building a list of candidate files, sorted by a combination of size and last access date, and then copying and removing those files in the least recently used order, until the required amount of space is available.

The FILECOPY utility is another means of backing up disk files. The FILECOPY utility automates the copying of files based on selective criteria. For example, you can pass FILECOPY parameters that cause the utility to copy all files from a particular source that were created or updated during a particular time period. These FILECOPY features allow you to minimize the number of files that are copied, but still ensure that you have a backup copy of the latest version of each file. FILECOPY can copy files from disk to tape, or from disk to another disk. FILECOPY performs its copies by creating and initiating a WFL job that contains the necessary COPY statements.

You can use the CATALOGING feature to

- Keep track of all file copies created by FILECOPY.
- Keep track of file copies made by WFL jobs you write yourself.
- Find a file or a backup copy of a file.
- Keep track of up to seven different versions of each file, and up to two copies of each version of a file.

### Documentation

- *A Series Disk Subsystem Administration and Operations Guide*
- *A Series System Software Utilities Operations Reference Manual*
- *A Series System Operations Guide*

## Disk Management

Disk units are one of the most extensively used data storage media. A *disk unit* consists of a disk pack that is mounted on a disk drive. A *disk pack* consists of a number of flat circular plates containing magnetic bits of data stored in concentric circles called tracks. Disk units provide a permanent medium for storing large volumes of data, while providing rapid random or sequential access to that data.

The means used to maintain and back up disk files were introduced under "File Maintenance" in this section. The following pages introduce A Series features that improve disk performance, reliability, and ease of use.

### Documentation

For detailed information about disk management, refer to the *A Series Disk Subsystem Administration and Operations Guide*.

## Logical Disk Groupings

The A Series allows you to group disk units logically into *families*. The system treats each family as a single entity and can spread files across all the disk units belonging to that family. Application programs and operator commands can refer to the family by its family name, without having to mention the names of the individual disk units. The use of families allows you to vary the number of disk units in a family without requiring any changes to the applications that access files on that family.

Access to disk units is further simplified by the *family substitution* feature. Family substitution establishes a primary default family and a secondary default family for individual users or programs. The primary default family setting typically indicates the family that stores the files owned by a user or program. The secondary default family setting usually indicates a family that stores system compilers, utilities, and other files that are of interest to many different users.

Family substitution makes it unnecessary to always specify the family where a requested file is located. It also simplifies application maintenance; a change to a single family substitution statement can cause an application to look for all its files on a different family. Lastly, family substitution eases system administration by making it possible to group all general-interest files on a single family and make them easily available to all users.



### Disk Reliability Features

Like any other storage medium, disk units can occasionally experience equipment failures of various kinds. One way to protect against data loss in such an event is by using the facilities described under “Disk File Backups” in this section. The A Series systems provide additional insurance against data loss or operations interruptions by allowing you to duplicate system files or even entire disk units.

#### Duplicate System Files

The operating system allows you to maintain duplicate copies of the following system files: the system directory, the catalog file, and the MCP/AS code file.

The system directory stores information about all the files stored on a family. By default, a family has only one system directory, which is located on the first disk unit in the family. If the system directory becomes unavailable, you could lose access to files throughout the family. To prevent such an eventuality, you can use the DD (Directory Duplicate) system command to specify that a duplicate copy of the system directory should be maintained on another disk unit in the family. If one system directory becomes corrupted, the system can simply use the other one.

The catalog file is used as a system-wide index into the system directories for each family on systems using the CATALOGING feature. (This feature was introduced under “Disk File Backups” in this section.) On systems that do not use cataloging, the catalog file is called the *access file*. You can protect this access information by using the AD (Access Duplicate) system command, which maintains a duplicate copy of the catalog file or the access file on a different disk unit in the same family.

The MCP/AS code file stores the code used by the operating system. You can keep multiple copies of the MCP/AS code file on different disk families. The CM (Change MCP) system command allows you to specify which of these MCP/AS code files should be initiated after a halt/load. This method ensures that the system can quickly recover from the loss of a disk family.

Alternatively, you can use the CM (Change MCP) system command to specify that duplicate copies of the MCP/AS code file should be maintained on separate disk units in the same disk family. Thereafter, if errors occur on one copy of the MCP/AS code file, the system can still use the other copy without any interruption in operations.

#### Duplicate Disk Units

The Mirrored Disk feature supports identical copies of specified disk units to provide protection against disk unit failures. By creating and maintaining duplicate copies of important disk units, the Mirrored Disk feature increases both system availability and data integrity.

The Mirrored Disk feature allows two to four disk units to function in parallel as a *mirrored set*; that is, as exact copies of one another. If one mirror in a set is destroyed, goes offline, or has an irrecoverable error, another online copy in the set allows normal functioning to proceed. In the event of an error on one mirror, application programs

using the mirrored set proceed normally, and the operator is notified of the error. Disk mirroring is totally transparent to application programs.

I/O operations for mirrored disks are handled differently from those for nonmirrored disks. While each write operation is issued to all disks within a mirrored set, each read operation is issued to only one disk within the mirrored set. The system divides read operations among all the mirrors in the set. This technique reduces the total number of read operations per disk unit and improves I/O throughput for disks that experience a high ratio of read operations to write operations. For an illustration of Mirrored Disk I/O handling, see Figure 7-1.

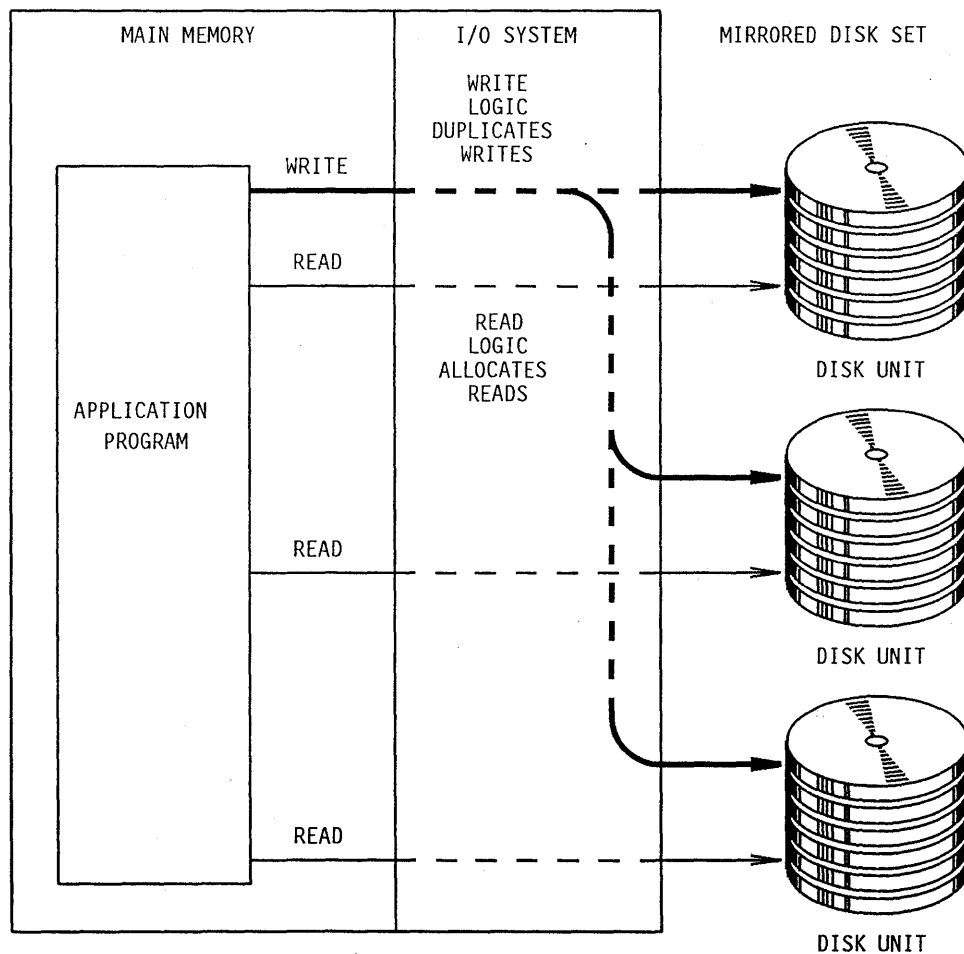


Figure 7-1. Mirrored Disk I/O Handling

The Mirrored Disk feature applies to all supported disk types except head-per-track disks. Mirrored Disk can be used on all A Series systems, regardless of whether they are using the CATALOGING feature. No special hardware is required for disk mirroring, aside from the extra disk units that are used as mirrors.

You can create mirrors of existing disk units without bringing those disk units offline or interrupting use by the system. You can also remove mirrors from a mirrored set, create new mirrors for a mirrored set, or move mirrored sets between systems or within systems without interrupting system operations.

### Disk Performance Optimizations

Recent years have seen dramatic improvements in most areas of mainframe performance, including the speed of central processors. During this time period, disk unit performance has not been improving at the same pace. The result is a situation where a computer system might be unable to fulfill its performance potential because of the inability of disk units to service I/O operations quickly enough. This problem is a general concern in the computer industry today.

The A Series solves this problem with a number of innovative techniques that can dramatically improve the performance of disk-intensive application programs. These techniques include configuration options, disk caches, Memory Disk, database buffer parameters, and duplicate system files or disks. Additionally, the A Series provides a facility to allow you to compare and evaluate the effects of these various disk performance techniques on your particular system.

### Physical Configuration Options

The physical components that affect disk I/O speed include the input/output processor (IOP), data link processors (DLPs), and the disk units themselves. The hardware options available for improving disk performance include

- Upgrading disk units to newer types of disk units that feature a higher transfer rate
- Adding extra paths, so as to reduce the number of disk units that each path has to service

Your Unisys marketing representative can give you further details about the configuration options that are available with your system.

### Disk I/O Caches

A *disk cache* is a set of buffers that hold recently requested disk data so that subsequent read requests can be satisfied from the buffers, thereby reducing the I/O load on the system and improving access time. The A Series Software Disk Cache Module brings this function to A 12 and A 15 systems. On A 17 systems, a similar function called Hardware Disk Cache is provided by the hardware. Statements in the following paragraphs apply to both types of disk caches, except where otherwise noted.

Both Software and Hardware Disk Cache are global disk caching mechanisms. That is, the disk cache resides in main memory, and the system can dynamically allocate varying amounts of disk cache space to each disk unit. By contrast, disk caches implemented by other companies generally reside at the disk controller. Such disk caches cannot respond as flexibly to changing patterns of disk usage.

The disk cache is divided into areas called *pages*. For most disk units, a complete disk track is the unit of information transfer between the disk unit and the disk cache, and each page in the cache can store a single disk track.

When a program issues a read request that refers to a disk area that is already resident in the cache, this is called a *read hit*. The request is satisfied by transferring data between the disk cache and the program buffer. This transfer requires very little time when compared to a physical disk I/O operation.

When a read request refers to a disk area that is not already resident in the cache, this is called a *read miss*. A physical I/O occurs and the requested data is transferred into the program buffer. Depending on various performance calculations, the data might be transferred into the disk cache as well. Once a data transfer to the disk cache has been initiated, the system treats any read operations that refer to that data as read hits.

Software Disk Cache can copy a disk track to any cache page, while Hardware Disk Cache uses a set-associative algorithm that defines a subset of pages to which a particular disk track can be copied. For both types of disk caches, when the system reads a disk track into the cache, the system overwrites the least recently used of the eligible pages in the cache. This "least-recently-used" algorithm allows the system to automatically respond to load changes by assigning more disk cache pages to the busiest disk units or the busiest files.

All write operations cause the relevant disk unit to be updated, thus ensuring that no essential disk data will be lost if the memory contents are lost due to a power failure or other event. If the disk tracks affected by the write are present in the disk cache, this is called a *write hit*, and the disk cache is also updated.

Figure 7-2 illustrates the handling of reads and writes by A Series disk caching mechanisms.

You do not need to rewrite or recompile any of your application programs to use disk caching; disk read and write requests are automatically directed to the disk cache by the system. The disk cache mechanism coexists with all other A Series products and features, including virtual memory, Memory Disk, and Mirrored Disk.

As part of the process of configuring your system, you can specify which disk units will be cached and how much memory will be set aside for the disk cache. Once the disk cache is in use, its function is completely automatic and requires no further attention.

The Software Disk Cache feature does not require any additional hardware on A 12 or A 15 systems, although it does require at least 4 megawords of memory to be used for the cache. Hardware Disk Cache is available on A 17 systems as a hardware option that includes a data transfer unit (DTU) within the resource management module (RMM). Hardware Disk Cache uses a minimum of 8 megawords of memory.

### **Documentation**

For further information, refer to the *A Series Disk Cache Module Installation and Operations Guide*.

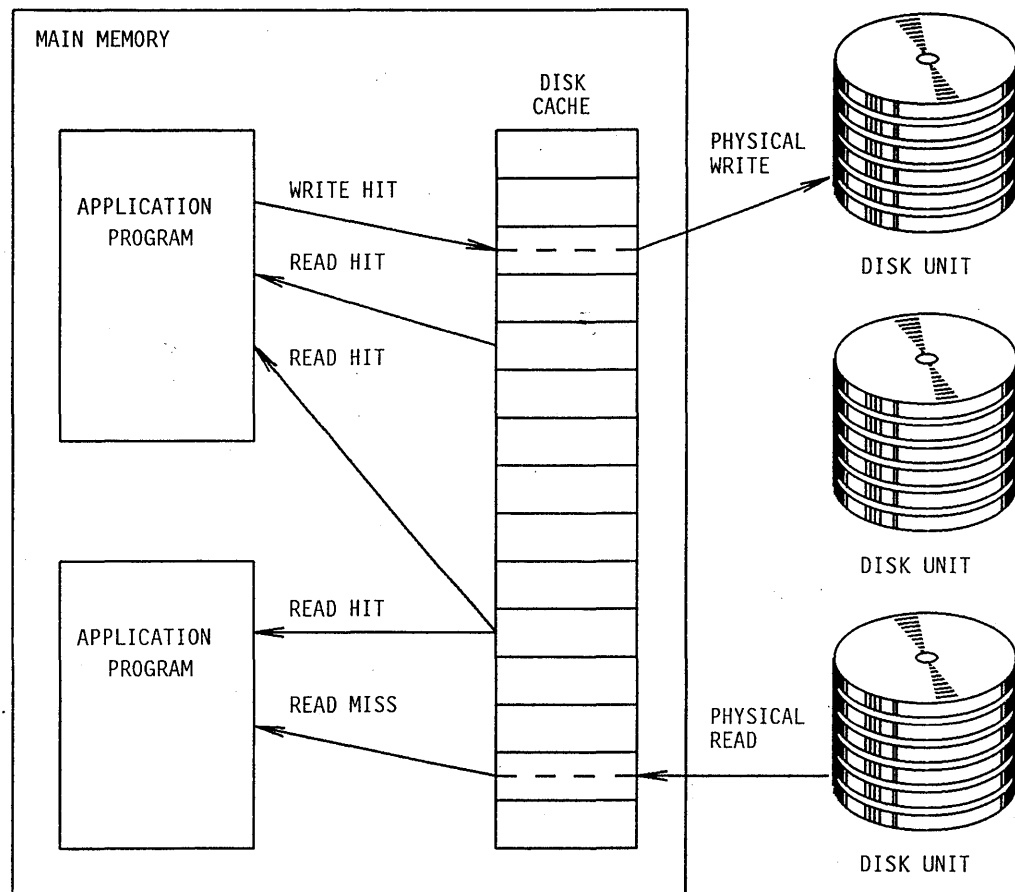


Figure 7-2. Disk Cache Read and Write Handling

### Virtual Disk Units in Memory

The A Series Memory Disk feature allows the use of a memory region as if it were a disk unit, allowing you high-speed file access. Memory Disk is like a disk cache in that it directs disk I/O operations to main memory rather than to a physical disk unit. However, whereas a disk cache essentially acts as a buffer between user processes and a physical disk unit, the Memory Disk feature completely replaces the disk unit and uses main memory as the permanent storage medium.

You can declare one or two units of Memory Disk, each with up to 60 million bytes of memory (approximately the size of a Model 206 unit in interlaced mode). Memory Disk is supported on all A Series systems. Figure 7-3 illustrates the Memory Disk concept.

Application programs use Memory Disk units just as if they were physical disk units; there is no need for special programming. A few operations restrictions apply to Memory Disk units, including the following: a Memory Disk unit cannot have a system memory

dump directed to it, nor can it store the MCP/AS code file for use after the next system halt/load.

All data stored in the Memory Disk is saved across most system halt/loads. However, memory reconfigurations or power failures cause the contents of Memory Disk units to be reinitialized. Memory Disk is therefore most useful for files such as read-only data files, code files, and temporary files. Compilers and their associated work files can be resident on Memory Disk. Memory Disk can also be used as the location for internal files created by the SORT intrinsic, thereby greatly increasing the speed of sorting.

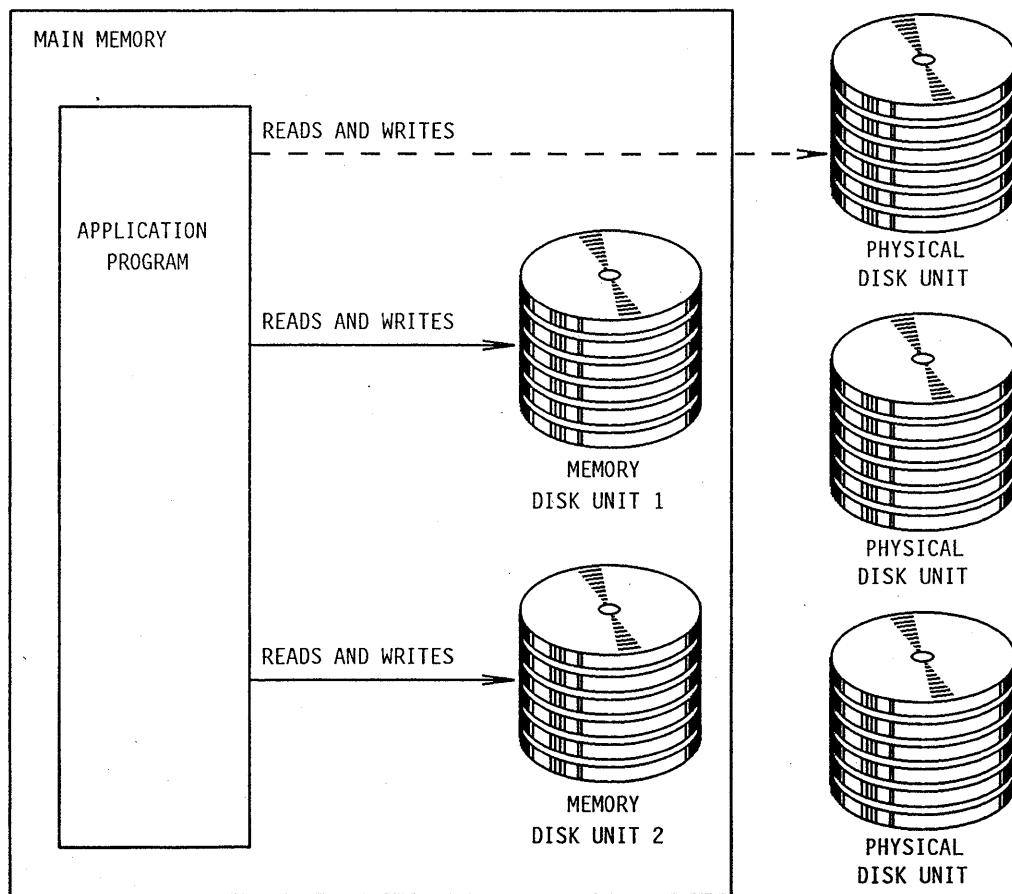


Figure 7-3. Memory Disk

### Database Buffering

Both Semantic Information Manager (SIM) and Data Management System II (DMSII) databases use buffers to store database data in memory. As long as a particular database item resides in a buffer, application programs can read and update the item directly, without the delay involved in a physical I/O operation. The more database items can be kept in the buffers, the better application program performance is likely to be.

Both SIM and DMSII enable you to dynamically specify the amount of memory that can be used for database buffers. This limit is set by a parameter called ALLOWEDCORE. You can use visible database stack (VISIBLE DBS) commands to change the ALLOWEDCORE value without causing any interruption in database availability. There is no need to recompile application programs or regenerate the database. The ALLOWEDCORE feature gives you great flexibility for optimizing performance and fine-tuning system resource usage.

### Duplicate Files or Disks

Two of the features discussed under “Disk Reliability Features” in this section can also improve disk performance in some circumstances. The use of Mirrored Disks improves performance in cases where a disk receives a high ratio of read operations to write operations. The use of duplicate MCP/AS code files can also improve system performance in some cases, because operations that cause MCP code to be read can be divided among the disk units that store the duplicate code files.

### Disk Performance Modeling

The preceding pages have introduced the wide variety of options available for optimizing disk performance on A Series systems. All these strategies can be helpful, but you might find yourself wondering how you can decide which particular techniques will work best for your system. Fortunately, Unisys provides a tool called Capacity Management Facility.Disk (CMF.Disk) that helps you to make just such decisions.

CMF.Disk operates in the following modes:

- *Collection mode.* In this mode, CMF.Disk collects statistics about disk I/O activity. The statistics are collected using a low-impact sampling method that imposes very little system overhead. The sampling mode produces a report summarizing the usage levels of each element of the disk subsystem, including all disk units and DLPs. This sampling report might reveal an unequal distribution of I/O activity, in which case you can redistribute files and DLPs to create a better balance.
- *Modeling mode.* In this mode, CMF.Disk enables you to specify various hypothetical changes to your disk subsystem configuration. These changes can include adding or subtracting DLPs, changing to different models of disk units, and using Mirrored Disk, Memory Disk, or Software Disk Cache. CMF.Disk then uses advanced simulation techniques to create a report summarizing the effect the proposed configuration would have on your disk performance.
- *Reporting mode.* In this mode, CMF.Disk enables you to create customized reports that merge selected information from multiple modeling mode runs. For example, you can create a single report that reports the performance a particular disk unit would give in each of several different disk subsystem configurations.

CMF.Disk is a menu-driven product with extensive online help, and does not require you to have any technical knowledge of the disk subsystem or modeling techniques. All necessary background information for understanding and using CMF.Disk is provided in the following documents.

### Documentation

- *A Series Capacity Management Facility.Disk Operations Guide*
- *A Series Capacity Management Facility.Disk Operations Training Guide*

## Tape Management

Magnetic tape is the least expensive medium for permanently storing large amounts of computer data. However, tape access is too slow to make it a practical medium for random reads and updates by applications. Instead, tape is typically used as a backup medium for saving copies of disk files, for recovery purposes, or as a temporary medium for transporting a file from one computer system to another.

A Series systems are able to read tapes created by a variety of types of computer systems, including all A Series systems, as well as all B 1000 Series and V Series systems. The format of the *tape label* determines whether the A Series can read a particular tape. Tape labels are special records at the start and end of a tape, and between each file on a tape, that provide details about the files stored on the tape. The I/O subsystem recognizes and reads tapes from any computer system that creates tape labels of the following types:

- All American National Standards Institute (ANSI) Level 2 standard labels
- Unisys 1100 Series labels
- IBM 360/370 format labels

A utility called RLTABLEGEN enables you to define other types of tape labels so the system can read them. In addition, the A Series can create unlabeled tapes, and can read any tape as an unlabeled tape. That is, the system can ignore the tape label format and treat it as a part of the data. The application program that reads the tape is then responsible for decoding the tape labels if they are present.

The A Series writes ANSI standard labels when creating a labeled tape. The A Series conventions for disk file names are much more flexible than the ANSI conventions for tape file names. When you use library maintenance, such as the Work Flow Language (WFL) COPY statement to copy a disk file to tape, the system automatically renames all the files to conform with the ANSI convention. At the same time, the system stores the original disk file names in a special file on the tape. When you use another COPY statement to copy the files back from tape to disk, the system automatically restores the original file names.

The A Series supports multiple files on a single tape and one file split across multiple tapes. Also supported are multifile, multireel tapes, with files split across reels as necessary.

The A Series also provides an optional tape security system that can be used to prevent users from copying data in from a tape without proper authorization. This topic is briefly described under "Tape Security" in Section 9.



### Printing Management

The Print System (also known as PrintS) is that part of the operating system and related system software that deals with the output of files to peripheral devices such as printers or card punches.

The Print System has an optional feature, the Remote Print System (or ReprintS), which provides access to printing at remote (data comm) printers and on BNA networks. To use the Remote Print System, you must have COMS (Full-Featured).

Statements in the following paragraphs about printed output generally apply to punched output as well.

#### Documentation

- *A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide*
- *A Series Printing Utilities Operations Guide*

### Printer Output Spooling

When you run a program that generates a printer file, the file can be either printed directly or *spooled* for later printing. If the program uses direct printing, then the printer file is printed as your program writes each line of output to the file. If the program uses spooling, the printer file is created as a special type of disk or tape file called a *printer backup file*. When the program finishes writing to the printer backup file, the Print System takes control of where and when the file is printed.

The practice of print spooling speeds the execution of application programs, because the program does not have to wait for a printer to become available before writing to the printer backup file. Print spooling also allows printers to be used more efficiently, because a printer is not dedicated to a program for the duration of the execution of that program.

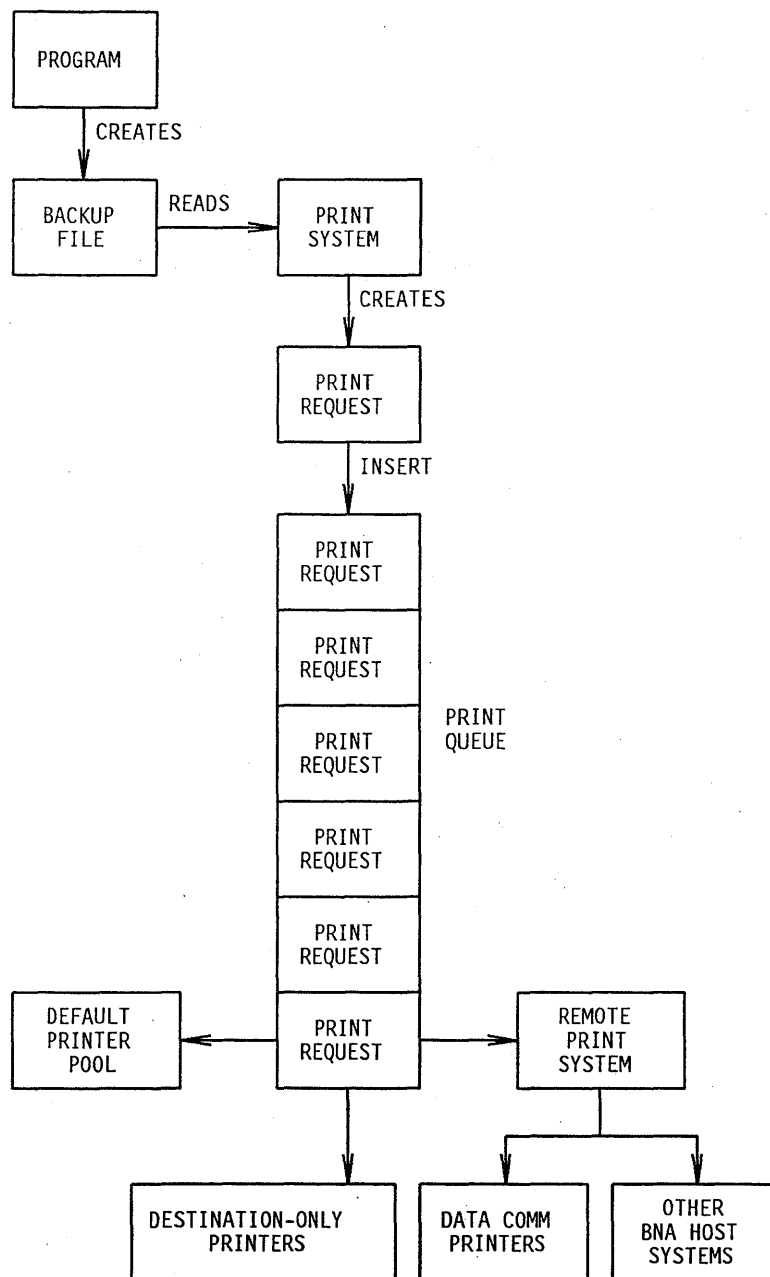
### Automatic and Manual Printing

You can allow printing to be handled automatically by the Print System, or you can save printer backup files and request their printing at a later time.

Automatic printing generally takes place as follows: once a printer backup file is closed, the Print System typically creates a *print request*, which is entered in the *print queue*. The Print System monitors the status of all printers on the system and assigns print requests to printers as they become available. When the Print System finishes printing a print request, it typically removes the associated printer backup file from disk.

The system administrator can direct the Print System to use any of several different criteria to determine the order in which print requests should be selected from the print queue for printing. Print requests can be printed in numerical order by job number, in

order of print priority, or in order of size, with the smallest being printed first. Figure 7-4 illustrates the sequence of automatic print handling.



**Figure 7-4. Print Spooling**

Optionally, the program that creates the printer backup file can request that the Print System save the printer backup file rather than immediately creating a print request. You can manually enter a **PRINT** statement later to print such files. The **PRINT** statement can be entered from a **MARC** or **CANDE** session, through system command

mode at an ODT, or from a WFL job. The PRINT statement creates a print request that goes into the print queue and is handled like other print requests.

### Additional Controls

The Print System provides a variety of print attributes that provide many different types of control over printing. The program that creates a printer backup file can set these attributes. A PRINT statement that submits a print request can also specify print attributes. For example, these print attributes can specify

- When and where the output should be printed
- How many copies of the output should be printed
- Text to be printed on header and trailer pages before and after the contents of the printer backup file

While a print request is in the print queue, you can use system commands to change any of the print attributes of that print request. For example, these commands can force or stop the printing of a print request or direct the print request to a different printer.

The system administrator can define a list of printers called the *default printer pool*. By default, the Print System routes print requests to the first printer in this pool to become available. Programs and users can use print attribute assignments to direct a print request to a printer outside the default printer pool.

### Data Comm Printing

The Remote Print System (ReprintS) extends the features of the Print System to include printers at remote destinations that are connected to the host computer through data comm lines. All Print System commands can be used for a remote device in the same way they are used for an on-site device. The remote destination must be a data comm station controlled by the Communications Management System (COMS).

### Remote Host Printing

The Remote Print System also includes BNA Print Routing, which enables you to easily route printed output to devices controlled by other host systems in a BNA network. The BNA print request entered from your local host contains all the information needed to locate a printer backup file, transfer it through BNA, and print it according to your original specifications. The printing host processes the print request, loads the printer backup file from your system, and prints the file, along with banner pages and job summary, if applicable.

Throughout the entire process, both the local and printing host systems maintain status information so that you can monitor progress if you desire, and identify and correct problems.

### Viewing and Manipulating Printer Backup Files

Two related printing utilities extend the capabilities of the Print System. You can initiate these utilities through MARC, CANDE, or a WFL job. These utilities are

- *Backup Processor*. You can use this utility to perform functions such as printing, copying, removing, or displaying the contents of printer backup files.
- *SYSTEM/BACKUP*. You can use this utility to perform functions such as printing backup files from tapes or printing selected records from a file.

### Customized Character Sets

In most cases the standard character sets supplied with the system are all that a site needs for its printer. If nonstandard sets are needed, you can use the LTTABLEGEN utility to generate custom tables for those sets.

## Soft Configuration

A configuration is the particular allocation of processor, memory, I/O devices, peripherals, and other hardware resources that together with the operating system form an A Series system. The actual physical configuration is accomplished by connecting hardware devices together.

However, the A Series also allows you to use a feature called *soft configuration* to inform the operating system of what hardware resources are available. On all A Series systems except the Micro A, you can use soft configuration to allow I/O units to be shared with another system, or to change the organization or contents of the I/O subsystem. On selected Large Systems (LS), you can also use soft configuration to divide a system into two independently running systems.

Configuration of data comm peripherals is handled separately from soft configuration. For details about data comm configuration, refer to Section 8, "Data Communications Networks."

### Documentation

For a detailed discussion of soft configuration, refer to the *A Series System Configuration Guide*.

## Configuration File

The description of the soft configuration of the system is stored in a file called the *configuration file*. To create this file, you first create a configuration source file and write in it descriptions of the hardware resources available to the system, including

- Data and I/O processors
- I/O subsystem hardware and connections

- Memory units
- System files, including data comm network information files
- Intersystem control (ISC) stations
- HYPERchannel hardware

On many A Series systems, you can use the GC (Group Configuration) system command to create a source file that shows the current configuration of the system. You can then modify this configuration source file with any changes you want to make. Alternatively, you can define your system configuration file with the Interactive Configurator, a menu-driven configuration utility. This utility eases the task of creating a configuration file.

Once you define the file, you then initiate the CONFIGURATOR utility, which processes the configuration source file and produces the configuration object file. You can use the CF (Configuration File) system command to inform the system of a new configuration file to use, and the RECONFIGURE (Reconfigure System) system command to cause the system to adopt a configuration description stored in that configuration file. This process of system reconfiguration does not require any rewriting or recompilations of application programs or system software.

### Dynamic Changes to the Configuration

The RECONFIGURE command causes an interruption in system service (called a halt/load). However, the A Series also allows you to change the soft configuration dynamically, without interrupting system operations. You can add hardware items to a configuration with the ACQUIRE (Acquire Resource) system command. You can delete hardware items from a configuration with a FREE (Free Resource) system command. These changes take effect immediately, and remain in force until the next time a RECONFIGURE command is entered. As with the RECONFIGURE command, the FREE and ACQUIRE commands do not require any changes or recompilations of your applications and system software.

On Entry and Medium Systems (EMS), FREE and ACQUIRE apply only to data link processors (DLPs) and peripherals such as tape units, disk units, and printers. On Large Systems (LS), FREE and ACQUIRE also apply to central processor, I/O processor, and memory units.

### Independent Groups within a System

If your system has at least two central processing units, at least two I/O processors, and partitionable memory, you can use soft configuration to divide your system hardware resources into *groups*. Each group runs independently, with its own separate copy of the operating system.

One reason to use separate groups is to prevent different types of applications from contending for the same hardware resources. For example, you might wish to ensure that online programs do not have to compete with batch programs for access to various resources. This division might improve the performance of the online programs if the division is carefully thought out.

Another reason to use separate groups is to provide a test environment that will not interfere with the production environment on a system. Programmers using the test environment might run unsafe software with a potential to cause system failures. A system failure of one group does not affect the other group.

Figure 7-5 shows a number of system hardware resources and three of the possible groups that could be defined for this system. You could store all three group definitions in the same configuration file. Thereafter, you could alternate between running the system as two separate groups (Groups A and B) or one complete group (Group C).

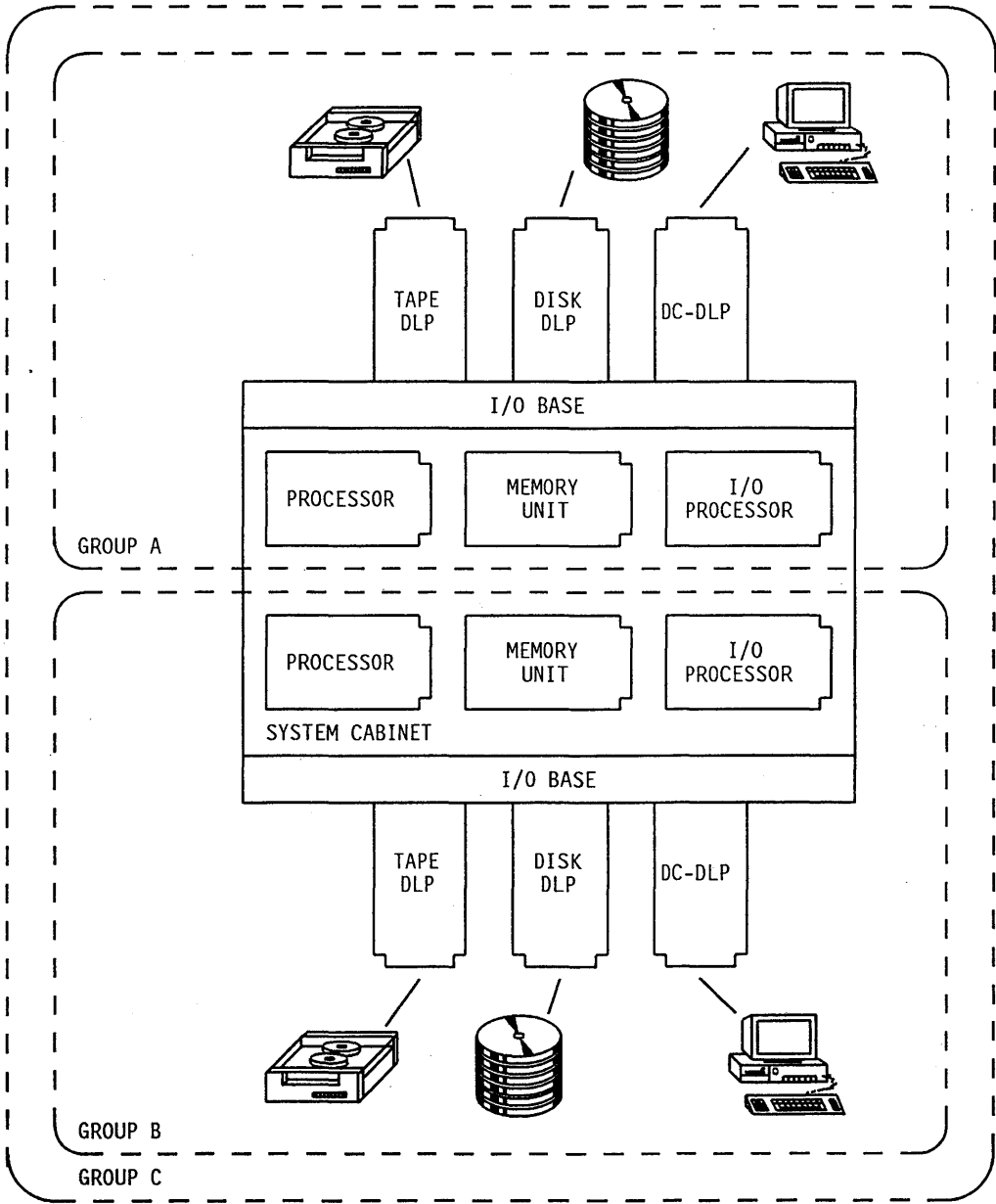


Figure 7-5. Soft Configuration Example

### I/O Hardware Sharing

Another virtue of soft configuration is that it allows separate A Series systems, or separate groups within an A Series system, to take turns using I/O bases, data link processors (DLPs), and peripheral devices. This feature makes it possible for you to allocate these I/O devices to whichever system has the greatest need at a given time. You can use the FREE and ACQUIRE commands to remove an I/O device from one group or system and add it to another. The physical links from both systems are always present; only the logical connection changes. There is no interruption in the service of either system.

### System Initialization

The A Series design makes it simple for you to initiate the operations of a new system, to schedule the operating hours for a system, and to reinitialize a system whenever necessary.

### New System Initialization

Initializing a new system involves tasks such as loading and initiating the operating system, configuring and initializing the data comm subsystem, and installing the remaining systems and environmental software. The A Series simplifies and automates system initialization through the use of the following products and features:

- *Preinitialized disk.* Many A Series systems come with a disk unit that is preinitialized with a copy of the operating system and other files that are necessary for system initialization. When you press the POWER ON button, the system goes through an automatic initialization sequence that loads the operating system into memory and initiates normal system operations.
- *DATACOMINFO file.* This file provides default definitions of several terminals and a printer, enough to make your system usable during the installation process. You can easily modify the data comm configuration later with the menu-driven Interactive Datacomm Configurator (IDC), as described under “Simplified Data Comm Configuration” in Section 8.
- *Simple Installation (SI).* This utility program assists you in copying and installing system and environmental software. SI dramatically reduces the labor involved in installing software. SI provides both a menu-driven interface and a batch interface, and can be run before data communications are established. For further information about SI, refer to “Simple Installation (SI)” in Section 11.

### Documentation

Refer to the installation guide for your particular A Series system.

- *A 12 System Operating Guide*
- *A 15 System Operating Guide*
- *A 16 System Operations and Programming Reference Manual*



- *A 17 System Console Operating Guide*
- *A Series A 1 - A 6 Systems Software Installation Guide*
- *A Series Micro A System Installation Guide*
- *A Series Micro A 825 OS/2 System Installation Guide*
- *A Series Micro A 825 UNIX® System Installation Guide*
- *A Series System Software Installation Guide, Volume 1: A 9*
- *A Series System Software Installation Guide, Volume 1: A 10*

### Automatic Power Schedules

The A 1, A 2, A 3, A 4, A 5, and A 6 systems all allow you to define an automatic power schedule with the POWER (Power Up/Down) system command. The automatic power schedule allows you to define the hours during which the system will be running. The system automatically powers itself up or down at the hours that you specified. You can create a different power schedule for each day of the week. This feature helps conserve power for systems that are used only during certain hours, and makes it unnecessary for an operator to power the system up or down manually.

Before the system turns itself off, it issues a warning to all users. You can specify how long the system waits after issuing this warning before turning itself off.

You can modify the power schedule when the system is running, without interrupting system service. You can also override the power schedule by manually powering the system up or down whenever desired.

#### Documentation

For information about the POWER command, refer to the *A Series System Commands Operations Reference Manual*.

### System Reinitialization

In the event of any unexpected system conditions that cannot be resolved through standard operating procedures, the A Series reinitialization procedures can be used to restore normal operations. The reinitialization procedures are designed to minimize the disruption of user service.

Features that ease the reinitialization process include the following:

- *PHL (Programmatic Halt Load) system command.* Entering this single command, which does not require any parameters, causes the system to go through the entire system reinitialization routine. The system configuration and most system operations parameters are retained without change.
- *Supervisor programs.* Unisys supplies a supervisor program called System Assistant (see “System Assistant” in Section 6, “Operational Interface,” for more information on System Assistant). You can optionally use the DCALGOL language to write your own programs that submit routine system commands. You can use the CS (Change Supervisor) system command to mark one of these programs as the supervisor program for the system. If a supervisor program is available, the system automatically initiates it after a halt/load.
- *Work Flow Language (WFL) job restarts.* WFL jobs are automatically reinitiated after being interrupted by a halt/load. Once a WFL job is restarted, it can resubmit its tasks from the point at which they were interrupted.
- *Database recovery.* Both Semantic Information Manager (SIM) and Data Management System II (DMSII) databases offer database abort recovery even across system interruptions such as halt/loads. Refer to “Database Backup, Auditing, and Recovery” in Section 3.
- *Synchronized recovery.* The Communications Management System (COMS) offers an alternate method of database abort recovery. COMS also can preserve input and output messages that were en route between a user and application when a halt/load occurred. Refer to “Database Recovery” in Section 4.
- *Data file protection.* Programs can use the PROTECTION file attribute to protect file data from being lost when a program is interrupted by a halt/load. This attribute affects only updates that physically complete before the halt/load.
- *Editing recovery.* If you are using CANDE or the Editor to update a file and are interrupted by a halt/load, you can recover your changes later from a CANDE or Editor recovery file.

### Documentation

- *A Series CANDE Operations Reference Manual*  
This manual discusses CANDE recovery features.
- *A Series Editor Operations Guide*  
This manual discusses Editor recovery features.
- *A Series File Attributes Programming Reference Manual*  
This manual describes the PROTECTION file attribute.
- *A Series Operating System Installation Guide*  
This manual discusses alternate reinitialization procedures.
- *A Series System Operations Guide*  
This manual discusses standard reinitialization procedures.

# Section 8

## Data Communications Networks

The data communications subsystem provides the communications link between an A Series host system and remote devices such as terminals and workstations, or between an A Series host system and other host systems.

This section is divided into the following major subject areas:

- “Open Networking” describes networking products that allow an A Series system to participate in a network with other vendors’ systems, as well as with other A Series hosts. Open Systems Interconnection (OSI), the HYPERchannel system, the LINK family, Systems Network Architecture (SNA), Transmission Control Protocol/Internet Protocol (TCP/IP), and X.25 networks are all discussed.
- “Advanced Networking” describes single-host and multihost data comm networks that use CP 2000s or CPDLPs and BNA Version 2 software. These networks are independent from those described under “Basic Networking.” These networks are called “advanced” because they provide more flexibility and better performance than the “basic networks.”
- “Basic Networking” describes single-host data comm networks that use network support processor/line support processor (NSP/LSP) combinations or data communications data link processors (DCDLPs), and multihost data comm networks that are managed by BNA Version 1 software.

### Open Systems Networking

Open systems networks are distinguished by their capabilities for not only linking together A Series systems, but for allowing A Series systems to participate in mixed networks with other vendors’ systems.

The area of open systems networks is one that is currently seeing rapid development on the A Series systems. New products in this area become available more frequently than this overview is updated. For information about the newest A Series open systems networking products, refer to your Unisys marketing representative.

### HYPERchannel and NETEX Compatibility

The HYPERchannel system is a high-speed digital communication networking system that can be used instead of BNA to connect A Series systems into a local area network (LAN). The HYPERchannel system makes it possible for you to link mixed-vendor systems into a network with A Series systems. The HYPERchannel system also provides the fastest data transfer rate available for A Series LANs: 50 megabits per second.

A HYPERchannel link consists of HYPERchannel coaxial trunks that are linked to a host system through an adapter interface specific to the system. The adapter used on

A Series systems is the A222 or A223 version, which is connected to the system through a HYPERchannel data link processor (HY-DLP). The HYPERchannel network can be accessed either programmatically through the operating system Direct HY file interface, or through the NETwork EXecutive (NETEX<sup>®</sup>) software package.

The Direct HY file is a direct method of access to the HYPERchannel network. To access the network through the Direct HY file, use a HYPERchannel adapter defined in a HYPERchannel map. As the user, you specify the format of the control messages and associated data, and initiate requests to send and receive data to and from remote hosts in the HYPERchannel network.

NETEX software, a host-to-host communications package, allows application programs in different hosts to communicate with each other, regardless of specific network configuration. Through the NETEX software, applications such as file transfer and transaction processing can be implemented across hosts. NETEX software uses the Direct HY file interface. The Direct HY file and its usage are transparent to the user when NETEX software is used.

NETEX software is provided as an ALGOL library. Access to NETEX software is available to all programming languages that can use libraries. (Libraries are discussed under "Run-Time Sharing of Program Modules" in Section 5.)

The HYPERchannel adapters and the NETEX software package are available from and supported by the Network Systems Corporation.

### Documentation

- *A Series File Attributes Programming Reference Manual*  
This manual includes an overview of HYPERchannel networks.
- *A Series System Configuration Guide*  
This guide includes information on how to create a HYPERchannel map.

## Compatibility with Timeplex Networking Products

Timeplex<sup>®</sup> Inc.—a Unisys company—designs, supplies, and services integrated voice, data, and image digital communications networks for private enterprises, communications carriers, and government agencies. Timeplex networks offer a wide range of integrated connectivity systems for T-1 and higher speed networks, X.25 packet-switching systems, and network management systems.

### The LINK Family

The LINK family of integrated connectivity systems (ICSs) manages private and hybrid T-1/E-1 communications networks, operating at speeds ranging from 4.8

---

NETEX is a registered trademark of Network Systems Corporation.  
Timeplex is a registered trademark of Timeplex Inc.

kilobits per second to 1.536 and 2.048 megabits per second. LINK products combine voice, data, and image transmissions over the same facilities, and provide economical, high-speed data transfer for A Series systems linked in a wide area network (WAN). Because they are transparent to higher level protocols, LINK ICSs can transmit BNA Version 1, BNA Version 2, Systems Network Architecture (SNA), Transmission Control Protocol/Internet Protocol (TCP/IP), and X.25 transmissions. The LINK family offers great flexibility in network configuration. For example, several output lines from a CP 2000, each using a different protocol, can be multiplexed by a LINK ICS onto a single T-1 line and then individually routed to different destinations.

A LINK ICS accepts multiple low- to high-speed inputs, and places them directly onto high-capacity T-1/E-1 digital facilities, avoiding expensive analog or low-speed digital connections. Access to channelized services—fractions of T-1 bandwidth—through a LINK ICS is currently creating new network applications and further reductions in communications costs for many installations. LINK ICSs can route traffic among hundreds of nodes and tens of thousands of attached devices. The maximum network size is currently 252 nodes.

The LINK family includes several compatible networking systems, and allows you to choose the size appropriate to your application. Starting with the microLINK/2™ system, designed for the small office environment, you can migrate upward through the miniLINK/2+™ system, to the LINK/2+™ and LINK/100+™ systems. The LINK/2+ systems accept direct connections to digital and analog voice lines as well as data. The LINK/100+ Model 28 is for large hub applications and handles up to 28 T-1 lines simultaneously.

The LINK family offers broad management capabilities including single-point network control, multiple data links, dynamic bandwidth allocation, alternate routing and rerouting, individual channel bypass, and programmable port priorities. LINK family reliability is assured by common logic redundancy, built in diagnostics, and ports that allow multiple loopbacks and remote testing by Timeplex field service engineers.

For extremely high bandwidth applications, the Timeplex TX3/SuperHub connects T-1 to T-3 facilities operating at 44.7 megabits per second. The TX3/SuperHub is a compact, highly reliable transport system that incorporates a distributed cross-connect matrix to provide connections for up to 360 T-1 circuits. A T-3 circuit carries the equivalent of 28 T-1s, but costs considerably less.

If you are interested in the LINK family, ask your Unisys marketing representative to refer you to a Timeplex representative for further information.

### **TIMEPAC Packet Switching Products**

Lower bandwidth data applications can make effective use of TIMEPAC X.25 packet-switching systems. The TIMEPAC NP100E, NP500, and NP1000 Nodal

---

microLINK/2 is a trademark of Timeplex Inc.  
miniLINK/2+ is a trademark of Timeplex Inc.  
LINK/2+ is a trademark of Timeplex Inc.  
LINK/100+ is a trademark of Timeplex Inc.

## Data Communications Networks

---

Processors use 32-bit microprocessor technology to provide local and wide area solutions for networks of all sizes from a few nodes to several thousand.

Incorporating an intelligent distributed architecture, the NPs offer outstanding flexibility in growth and configuration. The NPs are also easily integrated into large backbone voice/data/image networks based on LINK integrated connectivity systems. Higher bandwidths available through the LINK systems enhance the speed and efficiency of the TIMEPAC network segments.

A network of TIMEPAC NPs continuously monitors itself and reconfigures automatically to optimize the path for each message packet using a dynamic adaptive routing algorithm. With dynamic adaptive routing, NPs can respond quickly to changes in network conditions to avoid congestion and failure points. The result is traffic balancing and optimization over the whole network.

The TIMEPAC family includes both asynchronous and synchronous packet assemblers/disassemblers (PADs) with advanced design and the broadest protocol support of any comparable units in the industry. The TA-48 asynchronous PAD offers cost-effective connections for a large number of asynchronous terminals to a TIMEPAC network. The TS10 and TS40 synchronous PADs poll from 30 to 40 devices simultaneously for connection to packet networks.

### TIME/LAN Routers, Concentrators, and Intelligent Routing Bridges

TIME/LAN systems connect local area networks (LANs) so that host computers, personal computers, workstations, and other devices on different LANs can communicate with one another. TIME/LAN systems offer flexible networking alternatives to meet the continuing rapid growth in LANs and the need to connect LANs together for efficient use of network resources.

TIME/LAN 100 Routers connect Ethernet® LANs locally or remotely to 100 megabits-per-second (Mbps) fiber distributed data interface (FDDI) LANs. Remote connections operate at speeds of up to 1.536 Mbps. The TIME/LAN 100 FDDI Concentrator+ connects multiple stations to an FDDI network or can form its own eight-station 100-Mbps LAN.

TIME/LAN 10 Intelligent Routing Bridges join Ethernet or token-ring LANs locally or remotely at speeds up to 2.048 Mbps. Multiple LANs can be connected in any topology through TIME/LAN 10 bridges, and communication among the LANs is transparent to LAN protocol.

LINK backbone networks or TIMEPAC packet-switching networks provide the remote wide area connections for both TIME/LAN 100 and TIME/LAN 10 systems.

---

Ethernet is a registered trademark of Xerox Corporation.

## IBM Compatibility

Several A Series products are available that provide compatibility with IBM Systems Network Architecture (SNA) networks or that provide remote job entry (RJE) capabilities for bisynchronous networks.

## RJE/BSC Compatibility

Remote job entry (RJE) is a generic term for any program that allows users to submit jobs to a host system at a remote site and receive output from the jobs. In the IBM environment, the jobs are written in Job Control Language (JCL) and processed by a Job Entry System (JES) on the host.

One of the implementations of RJE on IBM systems allows the submission of jobs from certain devices using binary synchronous line protocols. These devices are collectively known as IBM Remote Job Entry/Binary Synchronous Communication (RJE/BSC) workstations. The types of devices included are the IBM 2780 RJE terminal, 3780 RJE terminal, and 360/20 RJE system. The latter system was originally called the Houston Automatic Spooling Program (HASP) and is still often referred to by the name HASP.

Unisys provides a program called RJE/BSC that allows an A Series host to emulate an IBM RJE/BSC terminal. RJE/BSC enables a user at an A Series host to submit JCL jobs that are forwarded to an IBM host for execution. RJE/BSC can print the job output or store it in a disk file.

RJE/BSC requires you to have an NSP/LSP, EDCDLP, or a DCHA. BNA software is not a prerequisite for this product.

Another RJE implementation, which communicates across SNA networks, is discussed under "Remote Job Entry Emulation" in this section.

## Documentation

For more information, refer to the *A Series Remote Job Entry/Binary Synchronous Communication (RJE/BSC) Configuration and Operations Guide*.

## SNA Compatibility

Systems Network Architecture (SNA) is a type of network widely used by IBM systems and their compatibles. A Series hosts and terminals can be linked into an SNA network through any of several SNA products, including the SNA Gateway, the SNA Logical Unit Type 6.2 (LU6.2) Service manager, the SNA 3270 Terminal Emulator, and SNA remote job entry (SNA/RJE).

Some of these products require BNA Version 2 software and CP 2000s or CPDLPs. Others can be run with DCDLPs and no supporting BNA software.

A Series SNA compatibility products are introduced in the following paragraphs. See your Unisys representative for more details about these and other IBM interface software packages for A Series systems.

### Physical Unit Types 2 and 5 Emulation

You can use SNA Gateway software to link an A Series host into an SNA network. SNA Gateway software runs on CP 2000s or CPDLPs, and requires the assistance of BNA Version 2 software on the A Series host. Two different SNA Gateway packages are available, which give two kinds of functions in the SNA network: the PUT2 Gateway and the PUT5 Gateway.

The PUT2 Gateway enables a CP 2000 or a CPDLP to act as a Physical Unit Type 2 (PUT2) node in an SNA network. A PUT2 node is a basic SNA node, capable of exchanging data with the network, but subject to the control of a System Services Control Point (SSCP) elsewhere in the network. Figure 8-1 shows an A Series host, HOSTA, that is linked into an SNA network through a CPDLP functioning as a PUT2 node.

The PUT5 Gateway enables a CP 2000 to act as a Physical Unit Type 5 (PUT5) node in an SNA network. (CPDLPs cannot run PUT5 Gateway software.) PUT5 nodes have the capabilities of a PUT2 node, and also have the ability to function as System Services Control Points (SSCPs). A CP 2000 with PUT5 can also serve as a link between IBM terminals and compatibles and an SNA network. In Figure 8-1, A Series HOSTB is linked into an SNA network through a CP 2000 functioning as a PUT5 node.

The PUT2 and PUT5 Gateway packages include command sets to control the functions of the CP 2000 or CPDLP in the SNA network. The PUT5 command set includes commands that can, for example, establish logical communication channels from the CP 2000 to other nodes in the SNA network. The PUT5 command set also includes commands for its SSCP functions. In both cases, you can enter the commands at the controlling host through MARC or through system command mode at an operator display terminal (ODT). If security permits, the commands can also be entered from terminals in the BNA Version 2 network.

### Documentation

- *A Series CP 2000 SNA PUT5 Implementation Guide*
- *A Series CP 2000 SNA PUT5 Operations Guide*
- *A Series CP 2000/CPDLP SNA PUT2 Implementation Guide*
- *A Series CP 2000/CPDLP SNA PUT2 Operations Guide*
- *CP 2000 SNA PUT5 Operations Reference Manual*
- *CP 2000/CPDLP SNA PUT2 Operations Reference Manual*



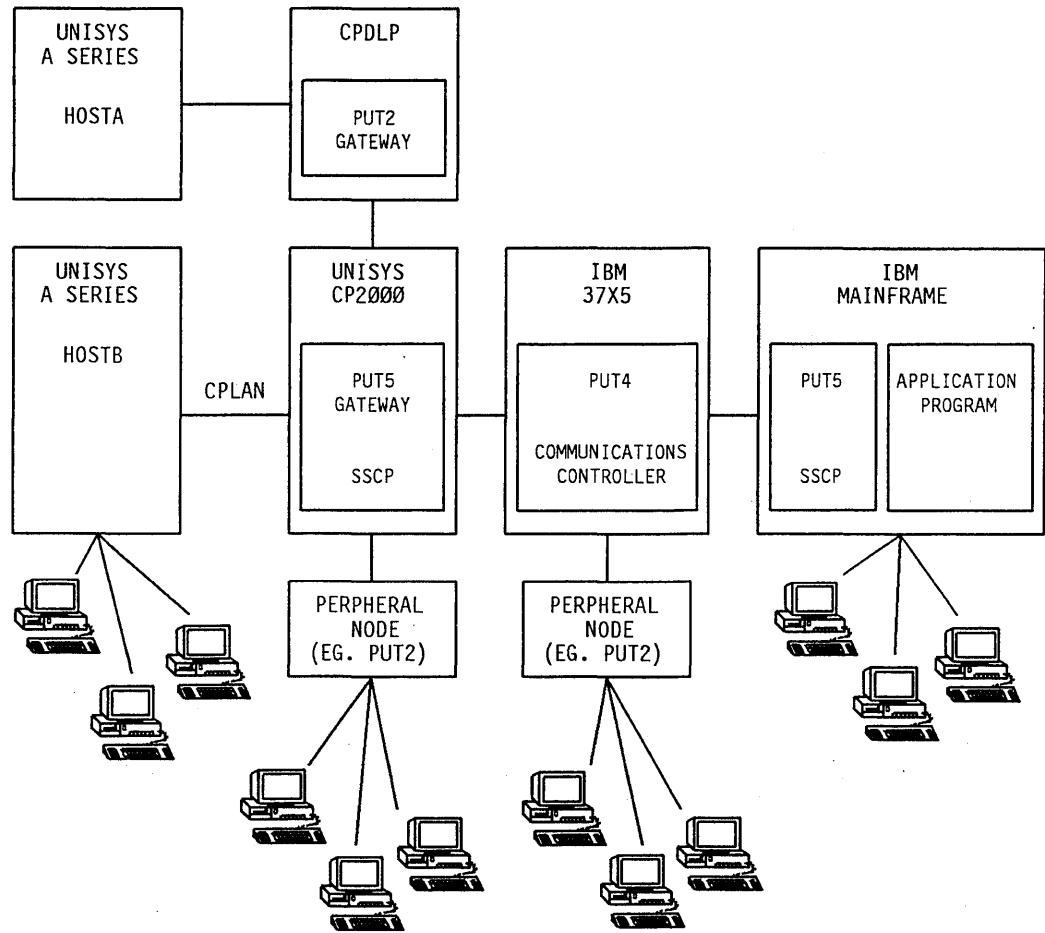


Figure 8-1. A Series Integration into an SNA Network

**Logical Unit Type 6.2 Emulation**

The SNA Logical Unit Type 6.2 Service Manager enables an A Series application program to communicate with a peer application program in a remote SNA network. The A Series host is linked to the SNA network through a CP 2000 or CPDLP. The A Series host must be running BNA Version 2 software, and the CP 2000 or CPDLP must be equipped with PUT2 Gateway or PUT5 Gateway software.

**Documentation**

- *A Series SNA LU6.2 Service Manager Implementation and Operations Guide*
- *A Series SNA LU6.2 Service Manager Programming Guide*
- *A Series SNA LU6.2 Service Manager Programming Reference Manual*

### 3270 Terminal Emulation

The 3270 series is a product family of IBM terminals and printers. The SNA 3270 Terminal Emulator allows you to use Unisys terminals in a session with an IBM host, interacting with a wide variety of IBM mainframe applications. The SNA 3270 Terminal Emulator also lets you use the system print facility to print the information received and generated during the session. The SNA 3270 Terminal Emulator can link an A Series host to an SNA network in either of the following ways:

- *Through a DCDLP* The A Series host must be running the SNA/PUT2 software module, and the DCDLP must be equipped with Synchronous Data Link Control (SDLC) protocol firmware. The A Series host can be running BNA Version 1, BNA Version 2, or neither.
- *Through a CP 2000 or CPDLP* The A Series host must be running BNA Version 2 software, and the CP 2000 or CPDLP must be equipped with PUT2 Gateway software.

#### Documentation

- *A Series SNA 3270 Terminal Emulator Implementation and Administration Guide*
- *A Series SNA 3270 Terminal Emulator Operations and Programming Guide*

### Remote Job Entry Emulation

The SNA/Remote Job Entry (SNA/RJE) product gives you the ability to submit jobs from disk for execution and/or compilation on an IBM host. SNA/RJE can link an A Series host to an SNA network using either of the methods listed for the SNA 3270 Terminal Emulator: DCDLPs or BNA Version 2 communications processors (CP 2000s or CPDLPs).

Another RJE implementation that emulates IBM RJE/BSC workstations is discussed under "RJE/BSC Compatibility" in this section.

#### Documentation

For further information, refer to the *A Series SNA RJE Implementation and Operations Guide*.

## OSI Compatibility

Open Systems Interconnection (OSI) is the data communications architectural model originally developed by the International Standards Organization (ISO) to promote interoperability among multivendor products. OSI standards continue to be developed and must undergo strict consideration by ISO members. OSI standards provide a common basis for network design, which makes it possible for different vendors to develop products that communicate with each other.

OSI standards define the following:

- Interaction between computers
- Functions necessary for communication

The Unisys A Series OSI product set operates from the platform of BNA Version 2, providing OSI interoperability while maintaining the features of BNA Version 2. Using OSI, you can link A Series systems with other A Series systems, with Unisys 1100/2200 Series, U5000 Series, or U6000 Series systems, or with other vendors' systems. The OSI product set consists of the following products:

- OSI Message Handling System (MHS)
- OSI File Transfer, Access, and Management (FTAM) software
- OSI interprocess communications (IPC)

The minimum hardware requirement for using OSI is an A Series host to which an ICP/CPLAN/CP 2000 combination is connected. The minimum software required consists of BNA Version 2 and the OSI Transport Services product. The other elements of OSI are optional.

The OSI IPC product implements the standard protocols that serve as the communication system foundation for routing messages, connecting devices to the network, and detecting transmission errors. This product also allows application programs to communicate across an OSI network using the same port file interface that is provided for BNA Version 2, with minor restrictions.

You must use one of the following software interfaces to an OSI network. Each of these interfaces allows communication with other Unisys systems and other vendors' systems.

- OSI Local Area Network (LAN) software. This product provides the capability to connect an A Series system into any IEEE 802.3 Local Area Network (LAN), such as an Ethernet LAN. An example of such a LAN is the Unisys Communications Processor Local Area Network (CPLAN).
- OSI X.25 software. This product provides the capability to operate over a Defense Data Network (DDN). For more information about X.25 networks, refer to "X.25 Compatibility" in this section.
- BNA Version 2 X.25 interface. This interface provides the ability to operate over packet-switched data networks (PSDNs), such as TELENET® or TYMNET® networks. This interface is not part of the OSI product set, but is provided by BNA Version 2 software.

---

TELENET is a registered trademark of General Telephone & Electronics Corporation.  
TYMNET is a registered trademark of Tymshare Corporation.

OSI takes advantage of some of the Unisys distributed systems software to provide enhanced capabilities beyond those already offered by your OSI network. You can use the following distributed systems software in your OSI network:

- A Series OSI Message Handling System (MHS) is more than an electronic mail system, and allows
  - Exchange of information and data between systems regardless of system type
  - Exchange of information between local applications and remote applications or users, even when the remote system type is unknown or is from another vendor
  - Distribution of files and information worldwide, over X.400 public communications services
  - Incorporation of voice, graphics, and text in messages
- File Transfer, Access, and Management (FTAM) provides a standardized framework that consists of a set of common file operations that all FTAM systems understand. Thus, you can move files to and from other systems in the network, regardless of system type, access method, or file type employed. FTAM allows you to access
  - Contents of files, including individual file records as well as whole files
  - File attributes, such as password information

Connections to LANs, DDNs, and PSDNs are all made through a CP 2000. The same CP 2000 can simultaneously link your system to a LAN, an X.25 network, and a BNA Version 2 network, or any combination of these three.

### Documentation

- *A Series Distributed Systems Service (DSS) Operations Guide*
- *A Series OSI Implementation Guide*

## TCP/IP Compatibility

Transmission Control Protocol/Internet Protocol (TCP/IP) is the name given to a family of protocols that were originally developed for use in a United States Department of Defense network. The best known United States Department of Defense networks that use TCP/IP protocols are the Advanced Research Projects Agency Network (ARPANET) and the Military Network (MILNET). The TCP/IP protocols have been published as Military Standards specifications and have become one of the most widely used non-vendor-specific networking protocols.

The Unisys A Series TCP/IP product set implements the most popular of the TCP/IP protocols. Using TCP/IP, you can link A Series systems with other A Series systems, with Unisys 1100/2200 Series, U5000 Series, or U6000 Series systems, or with other vendors' systems. The TCP/IP product set consists of the following four products: TCP/IP interprocess communications (IPC), TCP/IP local area network (LAN) software, TCP/IP X.25 software, and TCP/IP Applications Services.

The minimum hardware requirement for using TCP/IP is an A Series host to which an ICP/CPLAN/CP 2000 combination is connected. The minimum software required

consists of BNA Version 2 and the TCP/IP interprocess communications product. The other elements of the TCP/IP product set are optional.

The TCP/IP interprocess communications product implements the Transmission Control Protocol (TCP) for communication in a TCP/IP network. This product allows application programs to communicate across a TCP/IP network using the same port file interface that is provided for BNA Version 2, with minor restrictions. This product also provides support for the other TCP/IP products.

You must use one of the following software interfaces to a TCP/IP network. Each of these interfaces allows communication with other Unisys systems or other vendors' systems.

- *TCP/IP local area network (LAN) software.* This product provides the capability to connect an A Series system into an IEEE 802.3 local area network (LAN), such as an Ethernet LAN.
- *TCP/IP X.25 software.* This product provides the capability to operate over a Defense Data Network (DDN). For more information about X.25 networks, refer to "X.25 Compatibility" in this section.
- *BNA Version 2 X.25 interface.* This interface provides the ability to operate over packet-switched data networks (PSDNs), such as TELENET or TYMNET networks. This interface is not part of the TCP/IP product set, but is provided by BNA Version 2 software.

TCP/IP Application Services is an optional product that adds the standard TCP/IP application set consisting of the following features:

- *Telnet.* This feature enables you to logically connect terminals to remote hosts in the TCP/IP network.
- *File Transfer Protocol (FTP).* This feature enables you to transfer files to and from hosts across a TCP/IP network. You can initiate file transfers using the WFL *COPY* statement from a WFL job, from a MARC or CANDE session, or from system command mode at an ODT.
- *Simple Mail Transfer Protocol (SMTP).* SMTP is a protocol for electronic mail transfer. SMTP is used by the A Series Mail System, an interactive program that is discussed further in the information that follows.

Connections to LANs, DDNs, and PSDNs are all made through a CP 2000. The same CP 2000 can simultaneously link your system to a LAN, an X.25 network, and a BNA Version 2 network, or any combination of these three. Figure 8-2 shows one example of the configuration possibilities.

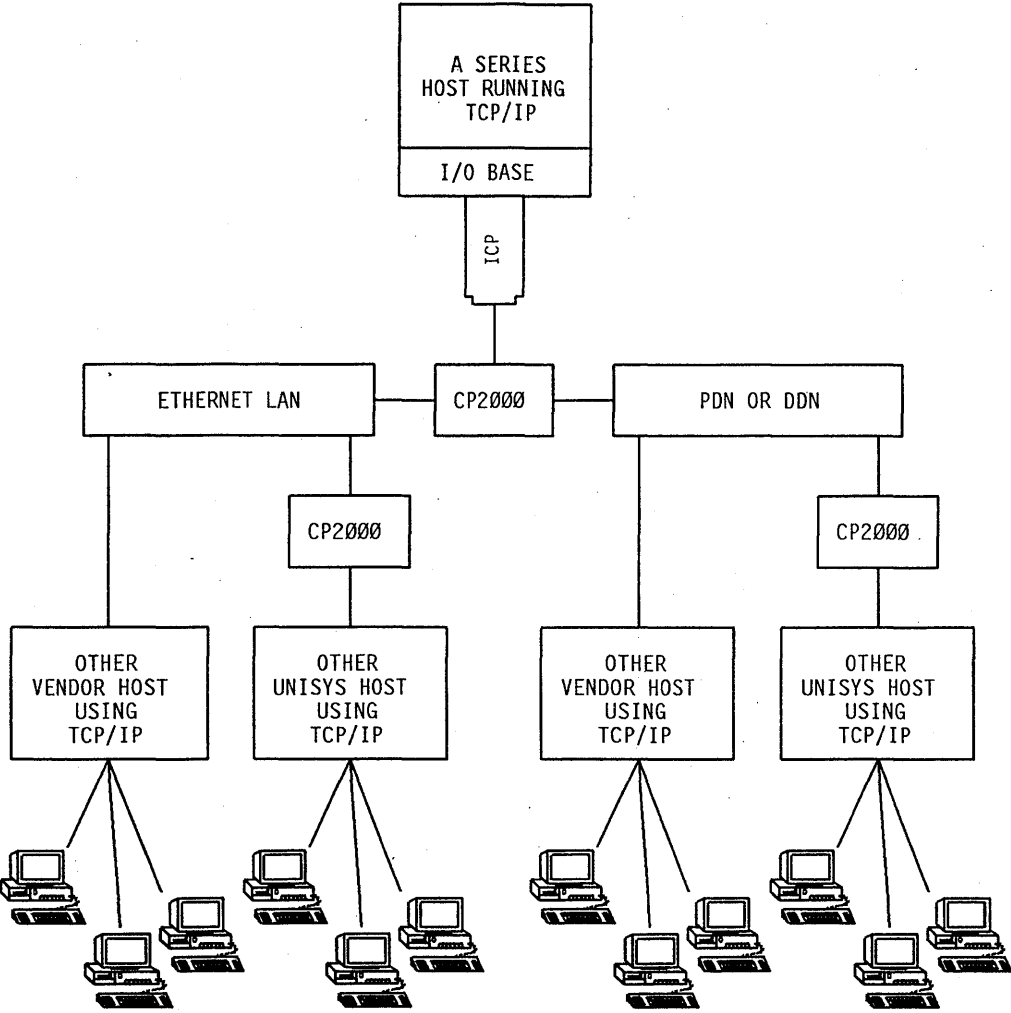


Figure 8-2. TCP/IP Ethernet Networks and X.25 Networks

The A Series Mail System is available only in a package with the TCP/IP Application Services product. However, once you have acquired the Mail System, you can also use it to send messages across a BNA Version 1 or BNA Version 2 network, or among users on a single system that is not part of a BNA or TCP/IP network. Mail System features include the following:

- *Distribution lists.* You can create these and use them repeatedly to direct single messages to a particular list of readers.
- *Folders.* You can define and name folders and store messages in them for later use.
- *News groups.* You can use these to post messages for optional viewing by readers with a common set of interests. These groups can be linked across multiple systems, so that users on different systems all see the same messages.
- *Registration facility.* This feature makes it possible for you to send mail to users based on their names, and provides assistance if you misspell someone's name.
- *Search capability.* You can use this feature to search for messages that include a particular text string.
- *Screen or command interface.* Screen mode provides the user with menus and extensive online help text. Command mode provides a concise command language for experienced users. Command mode also prompts you for missing parts of commands, thus helping you to learn the command syntax.

### Documentation

- *A Series Distributed Systems Service (DSS) Operations Guide*  
This guide includes information about File Transfer Protocol file transfer and Telnet station transfer.
- *A Series Mail System Installation and Administration Guide*
- *A Series Mail System Operations Guide*
- *A Series Transmission Control Protocol/Internet Protocol (TCP/IP) Implementation Guide*

## X.25 Compatibility

An A Series host can participate in packet-switched data networks (PSDNs) that use the level 3, packet-level protocol as defined by the Consultative Committee on International Telegraphy and Telephony (CCITT), recommendation X.25. PSDNs can be either public or private; there are many well-known public data networks that can be accessed through the X.25 protocol.

The X.25 protocol allows multiple dialogs to different remote hosts to be multiplexed through a single hardware device called an *X.25 station group*. You can also create multiple station groups to interface with different PSDNs or the same PSDN. When multiple station groups interface with the same PSDN, multiple parallel links provide fault-tolerant operation and a faster rate of information exchange.

## Data Communications Networks

---

A Series systems enable you to control X.25 networks through various extensions to the standard BNA network commands, which can be entered through the familiar operations interfaces (MARC or system command mode at an ODT), or through programs using the Program Agent protocol.

A Series systems provide the following methods of connecting to an X.25 PSDN:

- Through X.25 station groups attached to a CP 2000 or CPDLP and controlled by BNA Version 2 software. This method allows you to communicate with other A Series systems across a PSDN. The standard BNA Version 2 host services are available (refer to the discussions of "Host Services" under "Basic Networking" and "Advanced Networking" in this section).
- Through X.25 station groups attached to a CP 2000 and controlled by TCP/IP networking software. This method allows you to communicate with other A Series systems across a Defense Data Network (DDN) or a packet-switched data network (PSDN).

The system must also be running BNA Version 2 software. TCP/IP provides a number of interprocess communication and application services, which are described under "TCP/IP Compatibility" in this section.

- Through X.25 station groups attached to DCDLPs or NSP/LSPs. In this case, the X.25 network is controlled by a software product called the X.25 Message Control System (X.25 MCS), which is described in the following paragraphs. BNA software is not required.

The following considerations apply if you are using the second method in the preceding list.

The X.25 MCS is currently undergoing certification for use with a variety of public data networks, both in the United States and other countries. Contact a Unisys marketing representative for the latest information if you are interested in using the X.25 MCS with particular public data networks. .

The functions provided by the X.25 MCS include the following:

- A simple communications interface between application programs and terminals on the same or different host systems
- Ability of application programs to send commands to the X.25 MCS, and receive responses, through a port file or a remote file interface
- Ability to develop protocols higher than level 3 for specific applications
- A utility for copying files between hosts
- Ability to define closed user groups (CUGs), which provide security by ensuring that only an approved group of terminals can communicate with a particular A Series system

### Documentation

For further information about the X.25 MCS, refer to the *A Series X.25 MCS Operations and Programming Reference Manual*.



## Advanced Networking

*Advanced networking* is a term used in this overview for networks that are configured using BNA Version 2 software together with special hardware that supports BNA Version 2. Compared to the networks described under “Basic Networking” in this section, BNA Version 2 networks offer significant improvements in performance, flexibility, and reliability. Another distinction is that the same set of hardware and software features can be used to implement either single-host or multihost networks, or combinations of the two types of networks. And BNA Version 2 offers more advanced facilities for administering networks, including the Network Control Facility (NCF).

### Documentation

- *A Series BNA Version 2 Network Software Implementation Planning Guide*
- *BNA Version 2 Network Capabilities Overview*

## Host Services

BNA Version 2 offers the same host services that BNA Version 1 does, including command transfer, file access, file transfer, job transfer, station transfer, and access control. More importantly, BNA Version 2 uses the same interfaces to these features as BNA Version 1. Most commands and programming language constructs are the same in BNA Version 1 and BNA Version 2, though BNA Version 2 does provide some additional port file attributes for use by applications. And BNA Version 2 architecture provides improved performance for all these features.

For an introduction to Host Services features, refer to “Host Services” under “Basic Networking” in this section.

## Hardware Components

In a BNA Version 2 environment, expanded networking capabilities for A Series systems are provided through the following integrated products: the integrated communications processor (ICP), the Unisys communications processor local area network (CPLAN), the CP 2000 communications processor, and the communications processor data link processor (CPDLP). The following pages review the uses of these components in implementing single-host networks and multihost networks.

## Single-Host Networks

You can use either of the following types of hardware to connect terminals to a host in a BNA Version 2 network:

- ICP/CPLAN/CP 2000 combinations
- CPDLPs

The relationships between these hardware components are illustrated by Figure 8-3.

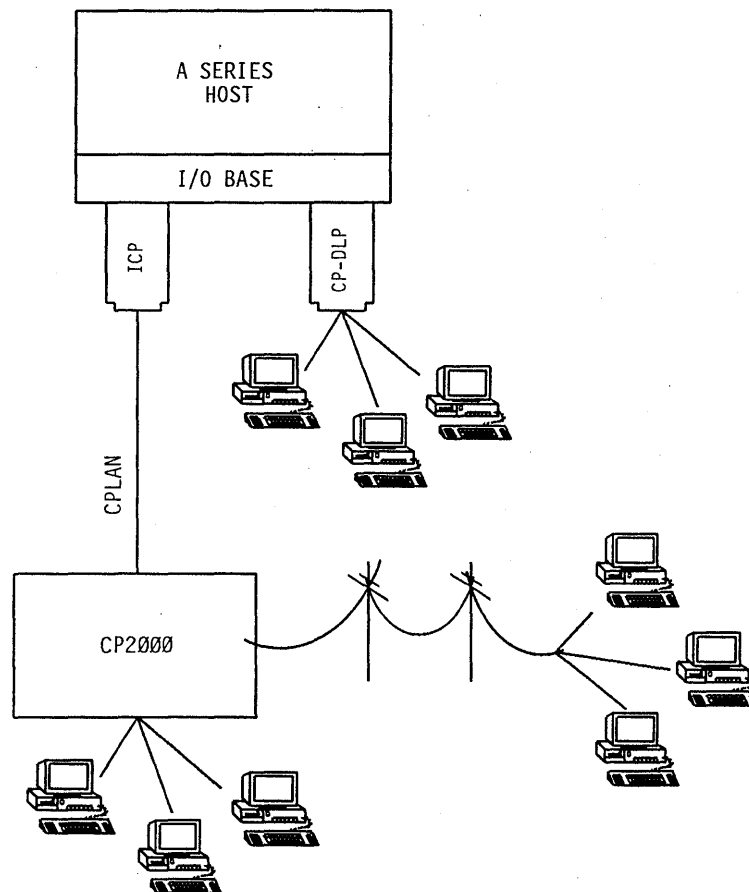


Figure 8-3. BNA Version 2 Single-Host Network Example

The ICP shown in Figure 8-3 is a type of data link processor (DLP) and plugs into a base in the host system. Among DLPs, the ICP is uniquely specialized to provide the system with an interface to CPLAN.

The CPLAN shown in Figure 8-3 connects an ICP with a CP 2000. CPLAN is a high-speed local area network (LAN) that conforms to the Institute of Electrical and Electronics Engineers (IEEE) 802.3 N standard. CPLAN operates with a 10-Mbps signaling rate over baseband coaxial cables. Because CPLAN allows each station to operate independently, you can add or remove ICPs and CP 2000s at will.

The CP 2000 shown in Figure 8-3 handles communications with individual data comm devices, such as terminals or workstations. The CP 2000 system architecture incorporates microprocessors that relieve the host of communications responsibilities and thus allow the host to process application programs faster. Each CP 2000 can be flexibly configured with up to seven different *line modules*, which can provide physical interfaces to different types of lines. The CP 2000 supports a number of predefined protocols and also allows you to implement custom protocols.

As Figure 8-3 shows, terminals can be connected to the CP 2000 either directly or over phone lines. The CP 2000 can thus act as a terminal processor for terminals located at a distant site.

The CPDLP shown in Figure 8-3 is an entry-level alternative to the use of an ICP/CPLAN/CP 2000 combination. Like the ICP, the CPDLP is a type of DLP and is plugged into a base on the host side. However, instead of providing an interface to CPLAN, the CPDLP supports direct connections to terminals.

The CPDLP supports most of the functions of a CP 2000, including most of the same predefined protocols and the ability to use custom generated protocols. However, the CPDLP supports fewer line connections and does not provide as much data comm performance improvement as the CP 2000. Further, the CPDLP supports only wide area networks; thus, you cannot connect CPLAN to a CPDLP. However, you can connect a CPDLP to a CP 2000.

### Documentation

- *A Series BNA Version 2 Network Software Implementation Planning Guide*
- *A Series CP 2000 BNA Version 2 Implementation Guide*
- *A Series CPDLP BNA Version 2 Implementation Guide*
- *A Series Integrated Communications Processor (ICP) BNA Version 2 Implementation Guide*
- *BNA Version 2 Network Attributes Data Dictionary*
- *BNA Version 2 Network Capabilities Overview*
- *Communications Processor Local Area Network (CPLAN) Planning Guide*

### Multihost Networks

The hardware components described in the previous subsection can also be used to link A Series hosts into a multihost BNA Version 2 network. Figure 8-4 illustrates three hosts that are linked into a network.

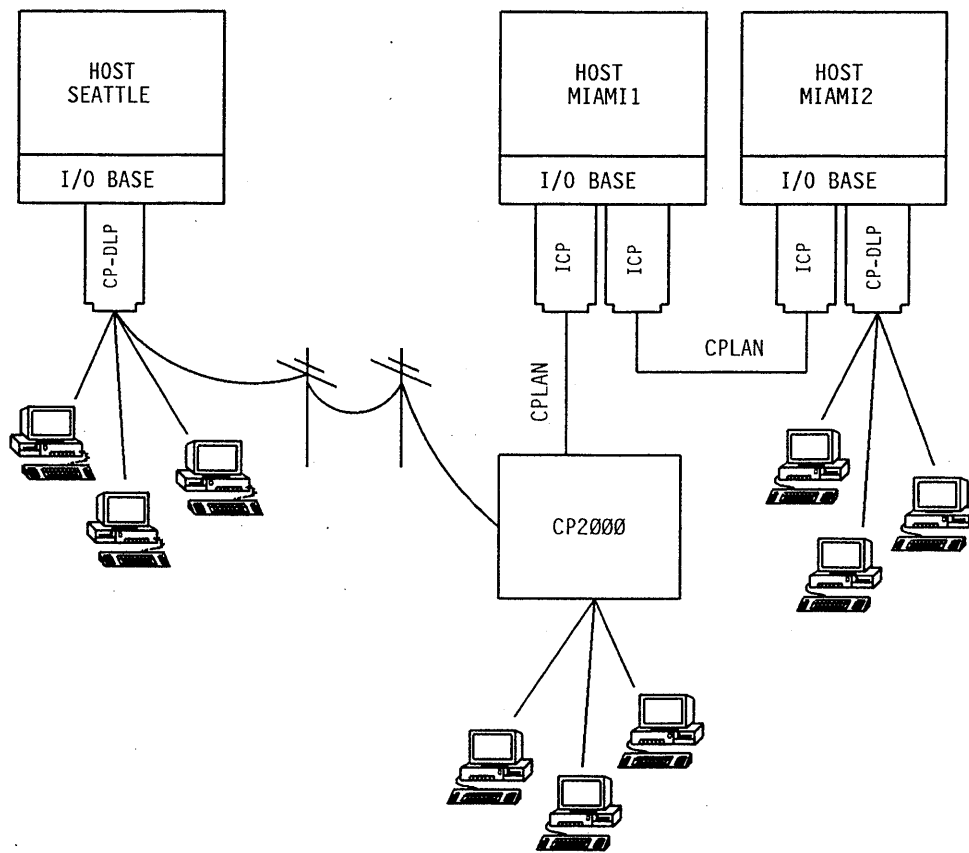


Figure 8-4. BNA Version 2 Multihost Network Example

In Figure 8-4, host SEATTLE is located at one installation, and hosts MIAMI1 and MIAMI2 are located at another installation a long distance away.

Because SEATTLE and MIAMI1 are located at separate installations, it is not possible to provide a direct cable link. Therefore, traffic is directed over phone lines instead. The interface to phone lines can be provided either by a CPDLP or an ICP/CPLAN/CP 2000 combination. Host SEATTLE uses a CPDLP, whereas host MIAMI1 uses an ICP/CPLAN/CP 2000 combination. The flexibility of CP 2000s and CPDLPs is shown by the fact that each of these units services local terminals as well as handling the link to the multihost network.

CPDLPs and CP 2000s can also be used to link an A Series host into a multihost Systems Network Architecture (SNA) or Transmission Control Protocol/Internet Protocol (TCP/IP) network. For further information about these interfaces, refer to "Open Systems Networking" in this section.

When hosts reside at the same site, you have the option of connecting them through a direct CPLAN link. In Figure 8-4, hosts MIAMI1 and MIAMI2 communicate through such a CPLAN link. An ICP at each end provides the interface between the host and

CPLAN. This is the simplest method of connecting hosts at a site into a BNA Version 2 network.

Though different types of links are used, hosts SEATTLE, MIAMI1, and MIAMI2 are considered hosts in a single BNA Version 2 network. BNA Version 2 software automatically handles the routing of messages from SEATTLE to MIAMI2 through MIAMI1.

### Documentation

- *A Series BNA Version 2 Network Software Implementation Planning Guide*
- *A Series CP 2000 BNA Version 2 Implementation Guide*
- *A Series CPDLP BNA Version 2 Implementation Guide*
- *A Series Integrated Communications Processor (ICP) BNA Version 2 Implementation Guide*
- *BNA Version 2 Network Attributes Data Dictionary*
- *BNA Version 2 Network Capabilities Overview*
- *Communications Processor Local Area Network (CPLAN) Planning Guide*

## Network Definition

The Network Administrative Utility (NAU) is a BNA Version 2 host software component that enables the entire network configuration to be specified interactively. NAU provides a set of interactive menus that you can use to build and maintain a consistent set of network initialization files for individual BNA Version 2 nodes.

NAU provides automatic consistency checking of network configuration information. NAU also automatically generates consistent network initialization files for each host and CP 2000 node in the network.

NAU initiates the CP 2000 configurator program that is used to generate code files for the CP 2000. A BNA Version 2 network with CP 2000s requires at least one NAU.

### Documentation

- *A Series BNA Version 2 Network Software Implementation Planning Guide*
- *A Series Network Administrative Utility (NAU) Operations Guide.*
- *BNA Version 2 Network Attributes Data Dictionary*

## Decentralized Network Operations

Like BNA Version 1, BNA Version 2 provides the ability to enter network monitoring and control commands from any of the host systems in a network. These commands are referred to as the BNA Version 2 Operations Interface (OI). You can enter OI commands through the familiar systems operations interfaces: MARC or system command mode at

an operator display terminal (ODT). Alternatively, you can automate operations in either of two ways:

- By storing OI commands in a disk file called an OI Message (OIM) input file. These commands can be executed by a BNA Version 2 software program called an Input File Agent.
- By writing a program that uses the Program Agent protocol. Such a program can control a BNA Version 2 network by submitting OI messages and analyzing the responses received.

BNA Version 2 has intelligent capabilities that minimize the need for operator intervention. For example, without any operator assistance, BNA Version 2 can alleviate congestion along particular routes, reroute traffic around line or node failures, and start up automatically after a system halt/load.

### Documentation

- *A Series BNA Version 2 Operations Guide*
- *A Series BNA Version 2 User Program Agent Programming Guide*
- *BNA Version 2 Network Attributes Data Dictionary*
- *BNA Version 2 Network Encoded Messages Programming Reference Manual*
- *BNA Version 2 Operations Reference Manual, Volumes 1 and 2: Commands and Inquiries*
- *BNA Version 2 Operations Reference Manual, Volume 3: Reports, Log Messages, and Error Messages*

## Centralized Network Operations

In addition to OI, BNA Version 2 provides an alternative operations interface called the network control facility (NCF). NCF extends the capabilities and enhances the interface of the standard operations interface.

NCF provides centralized monitoring of an A Series BNA Version 2 network. NCF enables the network operator to perform network analysis and diagnostics using easy-to-use reporting and control tools, and offers the following features:

- Real-time monitoring and reporting of network events and activity
- Programmatic control of network monitoring
- Color-coded summary displays of the network
- User-specified division of the network into regions and sites that you can display on color-coded site maps
- Color-coded bar graphs showing link utilization and error rates
- User-specified distributed data collection points throughout the network
- Fault-tracking system for tracking the occurrence and resolution of network problems

- Multiwindow displays of reports and responses
- Automatically updated textual and graphic network status displays
- Centralized network logging and log analysis
- Centralized monitoring of the status of communications links
- User-specified monitoring and reporting intervals
- User-specified error and utilization thresholds on communications links

### Documentation

- *A Series Network Control Facility (NCF) Implementation Guide*
- *A Series Network Control Facility (NCF) Operations Guide*

## Custom Line Protocols

You can enhance the capabilities of a CP 2000 or CPDLP by using the Custom Protocol Generator (CPG). CPG enables programmers to develop line protocols for Standard (BSTD), data link control (DLC), or Bisync station groups. These station groups, in their standard form, already provide several different protocols. However, you might need modified versions of these protocols, or entirely new ones, to create interfaces to some types of devices. Therefore, each of these station groups allows the addition of up to eight different custom protocols to its standard array of protocols. Once written, these custom protocols can be configured into the CP 2000 or CPDLP software just like the standard protocols.

The CPG allows you to

- Develop custom character-oriented and bit-oriented protocols on your host system
- Design the protocols that meet your specific needs
- Modify existing BSTD, DLC, and Bisync protocols or to create new ones
- Use source code of standard released protocols as examples
- Use Custom Protocol Generation Language (CPGL), a Pascal-like language for creating custom protocols

### Documentation

For further information, refer to the *CP 2000/CPDLP Custom Protocol Programming Guide*.

## Network Debugging Facilities

In the event of any serious problem with an integrated communications processor (ICP), a dump of its memory can be initiated automatically by the ICP or manually by an OI (Operations Interface) command. If such a dump occurs, you can use the ICP Memory Dump Analyzer program to reformat the dump into a readable form and analyze critical

portions of the dump. You can use this information yourself, or save it for a Unisys representative who can use it to analyze the problem.

The CP 2000 Memory Dump Analyzer provides a similar function for CP 2000s. During a CP 2000 dump, the contents of CP 2000 memory are written to a file on the controlling host. A program trace can also be written at the same time, to indicate some of the functions performed by the CP 2000 immediately before the dump. The CP 2000 Memory Dump Analyzer then uses the dump file as an input file, reformatting the data into a form that can be printed and used for more rapid diagnosis of CP 2000 problems.

### Documentation

- *A Series BNA Version 2 Operations Guide*
- *BNA Version 2 Memory Dump Analyzer Status Codes Support Reference Manual*

## Compatibility Issues

Any individual A Series host can be running BNA Version 1 or BNA Version 2, but not both. However, the *BNA Version 1/BNA Version 2 migration* feature provides for communication between hosts running BNA Version 1 and hosts running BNA Version 2. This feature makes it possible for you to migrate hosts in a BNA Version 1 network to BNA Version 2 one at a time.

BNA migration is provided by BNA Version 2 software, which translates messages from BNA Version 1 hosts into BNA Version 2 format and vice versa. BNA migration is a standard feature of BNA Version 2 software beginning with BNA Version 2 release level 1.1.

All the normal Host Services are available for use between BNA Version 1 and BNA Version 2 hosts. Some minor restrictions apply to the use of certain Host Services. For example, you cannot use command transfer to forward an NW (Network Prefix) system command between a BNA Version 1 host and a BNA Version 2 host. Additionally, an application on a BNA Version 2 host should refrain from using certain port file attributes exclusive to BNA Version 2 when opening a port file to communicate with an application on a BNA Version 1 host. Most port file attributes are identical in BNA Version 1 and BNA Version 2.

Figure 8-5 shows the hardware elements used to link a BNA Version 1 system with a BNA Version 2 system. Host SEATTLE is running BNA Version 1. A DCDLP connects SEATTLE to a BDLC station, which in turn communicates over a phone line with a second BDLC station. The second BDLC station is attached through an ICP/CPLAN/CP 2000 combination to host MIAMI, which is running BNA Version 2. Alternatively, a CPDLP could have been used instead of the ICP/CPLAN/CP 2000 combination.

BNA Version 2 networks can also coexist with DCDLP or NSP/LSP networks on the same host system. In Figure 8-5, host MIAMI runs BNA Version 2 software, but also supports a group of terminals by way of a DCDLP. These terminals are not controlled by BNA Version 2 software, but rather are configured and controlled through the facilities described under "Basic Networking" in this section.



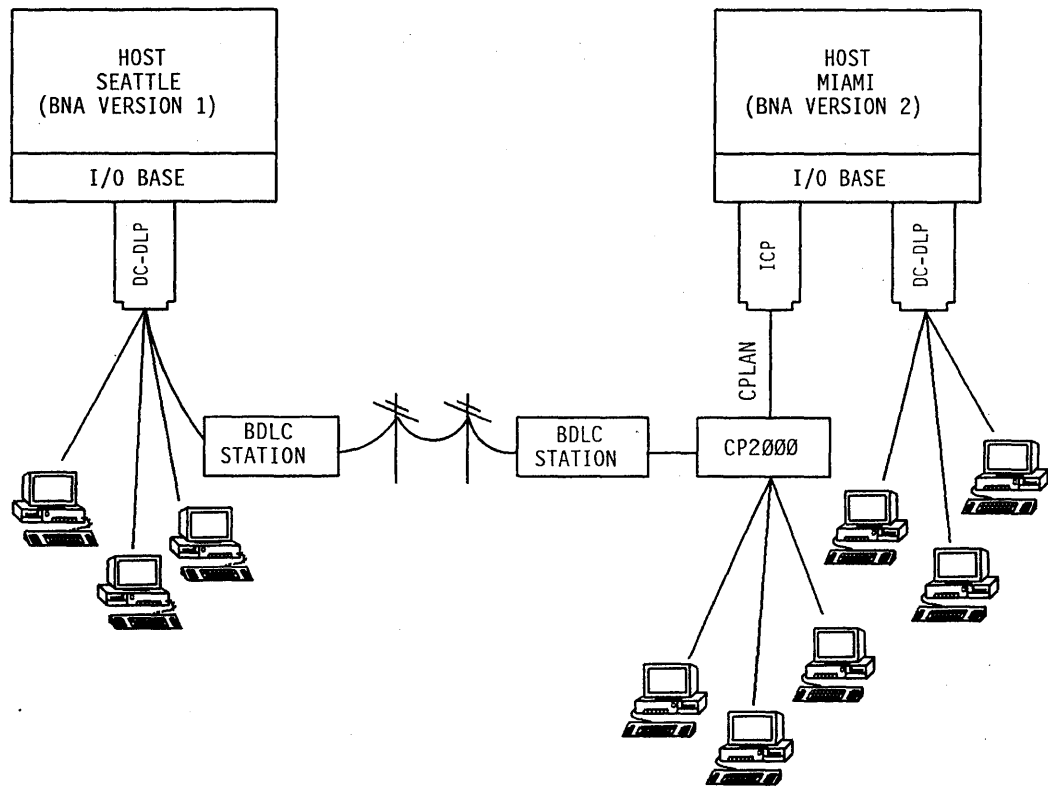


Figure 8-5. Hardware Links between BNA Version 1 and BNA Version 2 Hosts

## Basic Networking

*Basic networking* is the term used in this overview for a number of related data communications products that allow you to implement communications between an A Series host and data comm terminals, or between multiple A Series hosts. Characteristics of basic networking include flexible configuration, ability to develop custom protocols, and utility programs that help you to define or debug data comm networks.

### Single-Host Networks

Single-host networks link a single A Series system to peripheral devices such as terminals, workstations, and data comm printers. The following pages discuss the hardware components of single-host networks and the software products used to define and manage such networks.

### Hardware Components

The data communications subsystem, like other portions of the I/O subsystem, distributes communications functions from the central processor to a data link processor (DLP). There are several types of DLPs that are designed specifically for data comm. The following are the main categories: network support processors (NSPs), data communications data link processors (DCDLPs), integrated communications processors (ICPs), and communications processor data link processors (CPDLPs). Of these, ICPs and CPDLPs are discussed under "Advanced Networking" in this section. NSPs and DCDLPs are discussed in the following paragraphs.

An NSP is a programmable processor capable of supporting standard or customized line *protocols*, which consist of editing and line control procedures. Line control procedures provide user programs with services such as recovery from line errors.

On the host side, the NSP is plugged into a base that allows it to communicate with the mainframe. On the outer side, the NSP is linked to a line support processor (LSP). The LSP executes adapter control algorithms that handle the transmission of individual characters and provide error detection.

The LSP is in turn linked to up to four Quad Line Adapters (QLAs), each of which provides from one to four line adapters. Each line adapter is connected to one or more remote devices by a data comm line. If the remote devices are located at a great distance from the host, modems and a telephone line provide the link between the line adapter and the remote devices.

The DCDLP uses newer technology to combine the functions of the NSP, LSP, and QLA. On the host side, the DCDLP plugs into a base. On the outer side, the DCDLP is connected to remote devices by several data comm lines, each of which can provide connections to multiple remote devices. There are several different types of DCDLPs, which provide varying capabilities. The following are the types:

- *DCDLP II*. This is an improved version of the DCDLP that has more memory, includes more protocols, and allows you to download protocols from external firmware files.
- *Enhanced DCDLP (EDCDLP)*. This is a further improvement that has a more powerful processor, more memory, and eight lines. The EDCDLP also provides an environment similar to NSP/LSPs for implementing your own custom protocols. Moreover, the EDCDLP is more compact and costs less than NSP/LSPs.
- *Data communications host adapter (DCHA)*. The DCHA resides on a compact board that is designed for use in a Micro A system. The DCHA provides most of the capabilities of an EDCDLP, within certain limitations such as the ability to support only four unique protocols.

NSP/LSPs and the various types of DCDLPs can be used on the same host system, and can even be plugged into the same base on the system. Figure 8-6 shows a system that uses both NSP/LSPs and DCDLPs.

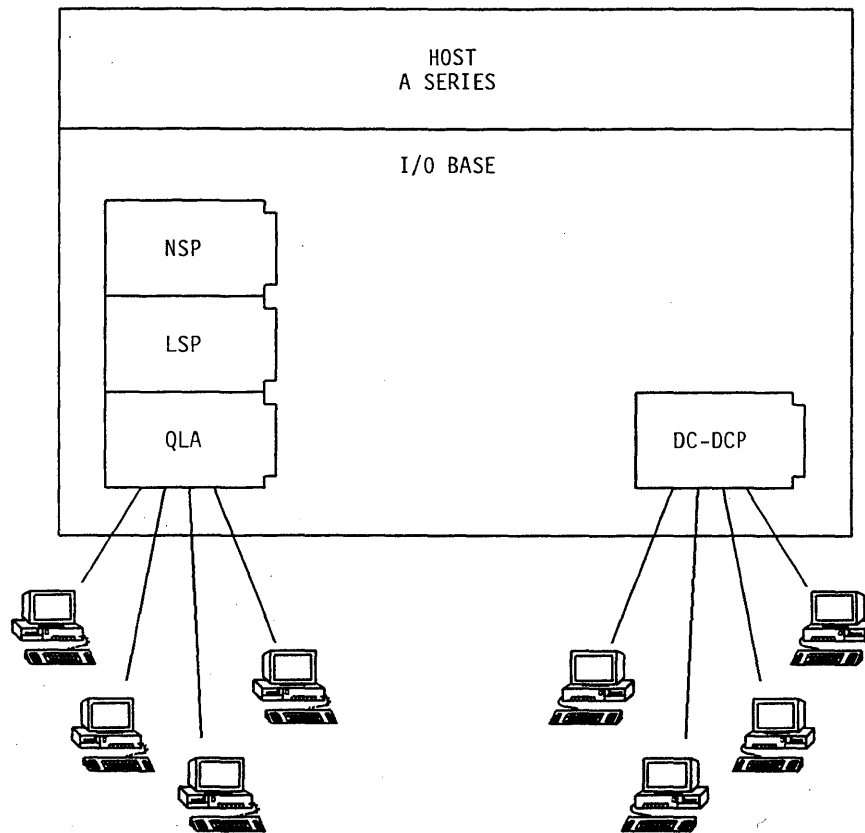


Figure 8-6. NSP/LSP and DCDLP Networks

### Simplified Data Comm Configuration

The data comm configuration for a system is stored in the DATACOMINFO file. This file contains a complete description of the configuration, including algorithms, editors, and translation tables. The DATACOMINFO file supplied with the software includes industry-standard protocols, plus example definitions for lines and stations. The definitions can serve as models for new lines and stations to be added to a configuration.

You can modify the DATACOMINFO file by using the Interactive Datacomm Configurator (IDC). IDC has a menu-driven interface with online help that greatly simplifies data comm configuration. Even if you have minimal knowledge of data communications, you can easily use IDC to define complex networks containing user-defined protocols and terminals.

IDC enables you to perform certain updates to the data comm configuration currently running on the system. This ability means that you can make changes, such as adding

and moving lines and stations, or test new algorithms and editors without having to interrupt the functioning of your data comm network.

If you need to customize the IDC menus and help text, you can do so by using the Interactive Menugraph Generator (IMG) and the Help Utility. One reason for doing this is to provide IDC with an interface in a language other than English, such as French or Spanish. Refer to “Multilingual System Interfaces” in Section 6.

### Documentation

For further information, refer to *A Series Interactive Datacomm Configurator (IDC) Operations Guide*.

## Custom Line Protocols

Although the standard DATACOMINFO file provides many industry-defined protocols, you might conceivably want to use a remote device that requires a protocol that is not provided. The data comm subsystem provides for such situations by accepting specialized, user-developed protocols. You can develop customized protocols for either an NSP/LSP, an EDCDLP, or a DCHA.

You can use the Network Definition Language II (NDLII) to develop protocols for an NSP/LSP. NDLII is a high-level language that allows you to describe a network and specify the tables and code for NSPs and LSPs. Once you have written a new protocol, you compile the program with the NDLII compiler, producing an object file called a Network Information File (NIF). You can use IDC to convert this file into a working DATACOMINFO file for the system. Alternatively, you can allow the system to automatically convert the NIF into the DATACOMINFO file when system data comm is initialized.

You can develop protocols for an EDCDLP and a DCHA in a similar way, except for an additional step involving the NDLII post compiler (NPC). After you have run the NDLII compiler and obtained a NIF file, you run NPC, which generates additional code that can run on an EDCDLP and a DCHA.

The EDCDLP also allows you to download certain protocols from firmware files that are provided separately from the DATACOMINFO file. These protocols are Data Link Control (BDLC), Synchronous Data Link Control (SDLC), and 56,000 bits per second Binary Synchronous Communication (BSC 56K BPS).

### Documentation

- *A Series Data Communications Protocols Installation and Implementation Guide*
- *A Series Network Definition Language II (NDLII) Programming Reference Manual*

## Debugging the Data Comm Network

You can use the DCDLPDUMPANALYZER utility to help you analyze dumps from any type of data communications data link processor, including the DCDLP, DCDLPII,

EDCDLP, and DCHA. You can request any of these devices to produce a dump in the event of unexpected errors in the software that controls them, including any user-defined protocols. DCDLPDUMPANALYZER provides the ability to determine the cause of these errors. You can run this utility either interactively or in batch mode. In interactive mode, you can request a hard copy of the results. Interactive mode also provides help text explaining each of the available commands.

Similarly, you can use the NSPDUMPANALYZER utility to obtain formatted listings of network support processor (NSP) memory dumps. Additionally, you can use the NDLIANALYZER utility to help in the debugging process. NDLIANALYZER combines information from the Network Information File (NIF) and an NSP dump file to produce information concerning the status of an NDLI process.

You can use the DCSTATUS utility to determine the current status of the data comm network. DCSTATUS analyzes the state of the data comm tables maintained by the operating system.

### Documentation

- *A Series Data Communications Protocols Installation and Implementation Guide*  
This guide includes information about DCDLPDUMPANALYZER.
- *A Series Software Release Installation Guide*  
This guide includes information about NSPDUMPANALYZER.
- *A Series Network Definition Language II (NDLII) Programming Reference Manual*  
This manual includes information about NDLIANALYZER.
- *A Series System Software Support Reference Manual*  
This manual includes information about DCSTATUS.

### Monitoring Data Comm Traffic

You can use the ID (Initialize Data Comm) system command to specify that the system should audit selected types of data comm activity. You use the DCAUDITOR utility to analyze the file produced by this audit. DCAUDITOR can print all traffic or selected classes, such as messages, errors, or trace messages.

### Documentation

For further information about DCAUDITOR, refer to the *A Series System Software Support Reference Manual*.

### Multihost Networks

BNA is the name of an architecture that allows the connection of multiple, independent A Series systems into a network. BNA networks give the users in a distributed processing environment the same type of access to remote resources that they have to local resources.

There are two versions of BNA: BNA Version 1 and BNA Version 2. BNA Version 2 uses some additional hardware, including CP 2000s, and is discussed under "Advanced Networking" in this section. The following paragraphs discuss BNA Version 1.

BNA Version 1 network architecture is available on several types of Unisys systems in addition to A Series systems. A Series systems can be linked in a BNA Version 1 network with any of the following types of systems: B 1000 Series systems, V Series systems, and CP9500 systems. However, not all the Host Services capabilities discussed in the following paragraphs are available on all these systems. Refer to the documentation for each of these system types for information about the BNA Version 1 capabilities they support.

### Host Services

Like BNA Version 2, BNA Version 1 software provides a variety of services to application programs, system operators, and end users. In all cases, the guiding philosophy has been to make it as simple and easy to use resources on a remote host as on a local host. These host services are outlined in the following paragraphs.

#### Documentation

Except where otherwise specified, the *A Series Distributed Systems Service (DSS) Operations Guide* is the source for information about the Host Services discussed in the following paragraphs.

### Command Transfer

A system operator can remotely control the operations of any other host system in the BNA network. By prefacing any system command with an *AT <hostname>* specification, you direct BNA to transfer that system command to another system and execute it there. BNA then routes the response back to you.

### File Access

BNA logical I/O system software allows application programs running on one host system to open, close, read from, or write to files on another host system. All these I/O operations are accomplished using the same statements that would be used to access files on the local system. A single *HOSTNAME* specification in the file declaration is all that is needed to direct all I/O operations for that file to the remote host. If this statement is not included in the original program, the end user can supply it through a file equation at run time. In this way, a program can be adapted for use across BNA without any rewriting or recompilation.

#### Documentation

For further information about BNA logical I/O, refer to the *A Series File Attributes Programming Reference Manual*.

### File Transfer

The Work Flow Language (WFL) *COPY* statement can be used to copy files between disk units on separate hosts in a BNA network. Adding a *HOSTNAME* parameter to a *COPY* statement is all that is needed to direct the transfer to a different host.

### Interprocess Communication

Application programs running on separate host systems can send messages to each other through port files. All read and write operations utilize the same syntax as for port file communication on a single system. A single assignment to the file attribute *YOURHOST* suffices to direct all I/O operations for that port file to a remote host. A single program can use port files to simultaneously communicate with programs running on many different hosts.

#### Documentation

For further information, refer to the *A Series File Attributes Programming Reference Manual*.

### Job Transfer and Remote Tasking

You can submit a WFL job from a disk file on one host, and direct it to run on another host. Alternatively, you can enter a command on one host that initiates a WFL job stored in a disk file on a different host. You can also use WFL statements or *CANDE* commands to initiate an application program that is stored on a different host system. Simply appending a *HOSTNAME* task equation to the *RUN* statement is sufficient to cause a program to be initiated across the BNA network.

#### Documentation

For further information, refer to the *A Series System Administration Guide*.

### Print Routing

Through BNA, the Remote Print System (BNA/ReprintS) provides you with integrated control over printing at remote destinations. You can use the same Print System commands to control printing at a remote device as you would for controlling an on-site device. The remote destination must be a station controlled by the Communications Management System (COMS).

#### Documentation

For further information, refer to the *A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide*.

### Station Transfer

You can dynamically connect your terminal to another A Series host system through the use of a simple *CONNECT* command. Once BNA has completed the connection, you can interface to the remote system through any of the message control systems (MCSs) available on that system, including the Communications Management System (COMS) and Command and Edit (CANDE). Your terminal will function as if it were directly connected to the remote system, and you can perform all the tasks that a local user of that system can perform. You can break the connection at any time to return to your own local host system.

If your local host system is running with COMS, you can connect one of your COMS window dialogs to another host while leaving the other dialogs attached to the local host. You can switch rapidly back and forth between two different host environments simply by moving between COMS dialogs.

### Access Control

All of the Host Services previously discussed, such as command transfer, file transfer, job transfer, and station transfer, are subject to BNA access controls. Each host has complete control over which of its resources are available to network users, and can selectively allow or deny resource use to individual hosts or users. There are two types of access control: host control and user control.

Host access control is achieved through validation and authentication. The validation process controls which hosts are allowed access, and authentication verifies that a host is what it claims to be.

The system administrator can limit access rights for individual users or programs based on both the usercode and the hostname associated with that user or program. For example, you could specify that user SMITH can access your system from the host named MIAMI, but not from the host named DALLAS. If you already have a user named SMITH on your system, you can define a *local alias usercode* that allows the SMITH at MIAMI to access your system under a different name, such as JPSMITH.

### Logging

BNA provides the option of logging specific events for purposes such as billing, maintenance, performance monitoring, reporting, scheduling, security auditing, and testing. This logging is in addition to any other system logging. The types and frequency of logging are specified by the operator.

### Hardware Components

In a BNA Version 1 network, each A Series system is considered a *host* that supports its own set of terminals through NSP/LSPs or any of the various types of DCDLPs.

At the same time, each A Series system can be a *node* in a BNA Version 1 network that links together A Series systems. Nodes directly connected to each other are known as



*neighbor nodes.* A node in a BNA Version 1 network communicates with its neighbors through entities known as *stations*. A pair of BNA Version 1 stations, one at each of the neighboring nodes, controls a single communication path between the two neighbors.

BNA stations are like other types of data comm stations, in that they communicate with the host through specialized data comm DLPs. However, a BNA station acts as an interface between systems, rather than providing a data entry point for users.

BNA Version 1 supports the following types of stations:

- *Intersystem control (ISC).* This type of station is used to link BNA Version 1 hosts into a local area network (LAN). An ISC consists of a *hub* and its associated *host control (HC) units*. The host control units are actually a special type of DLP (an HCDLP) that are plugged into a base at a host and connected by a line to the hub. A single hub can accommodate host control units attached to many different hosts. Refer to Figure 8-7 for an example of a BNA network that uses ISC stations.
- *Data Link Control (BDLC).* This type of station is used to link BNA Version 1 hosts into a wide area network through dedicated full duplex or half-duplex switched phone lines. Each BDLC station is linked to a single host by way of a DCDLP or NSP/LSP. Refer to Figure 8-7 for an example of a BNA network that uses BDLC stations.
- *X.25.* This type of station is used to link BNA Version 1 hosts into a wide area network through packet-switched data network lines that use the Consultative Committee on International Telegraphy and Telephony (CCITT) standard interface recommendation X.25. (Refer to "X.25 Compatibility" in this section.) Each host is linked to a packet switch in the X.25 network by way of a DCDLP, an NSP/LSP, or a data communications host adapter (DCHA).

Figure 8-7 illustrates three A Series hosts that each have a local data comm network and that also participate as nodes in the larger BNA Version 1 network. In this illustration, HOSTA and HOSTB communicate with each other through ISC stations, while HOSTB and HOSTC communicate through BDLC stations.

BNA Version 1 gives you the option of installing multiple parallel links between two neighbors. Each link is supported by a different pair of stations. BNA Version 1 routes the traffic between the two nodes through all of these stations, thereby increasing the rate at which information can be exchanged between the two neighbors. This concept is illustrated in Figure 8-7, which shows parallel ISC links that connect HOSTA and HOSTB.

Nonneighbor nodes can communicate because their messages are routed from neighbor to neighbor through the network. Routing tables at each node keep track of the most efficient route to any other node. These tables change dynamically as nodes or links are added to or deleted from the network, or when the physical characteristics of any node or link change. For example, in Figure 8-7, HOSTA can communicate with HOSTC through HOSTB.

### Documentation

For further information, refer to the *A Series BNA Version 1 Operations Guide*.

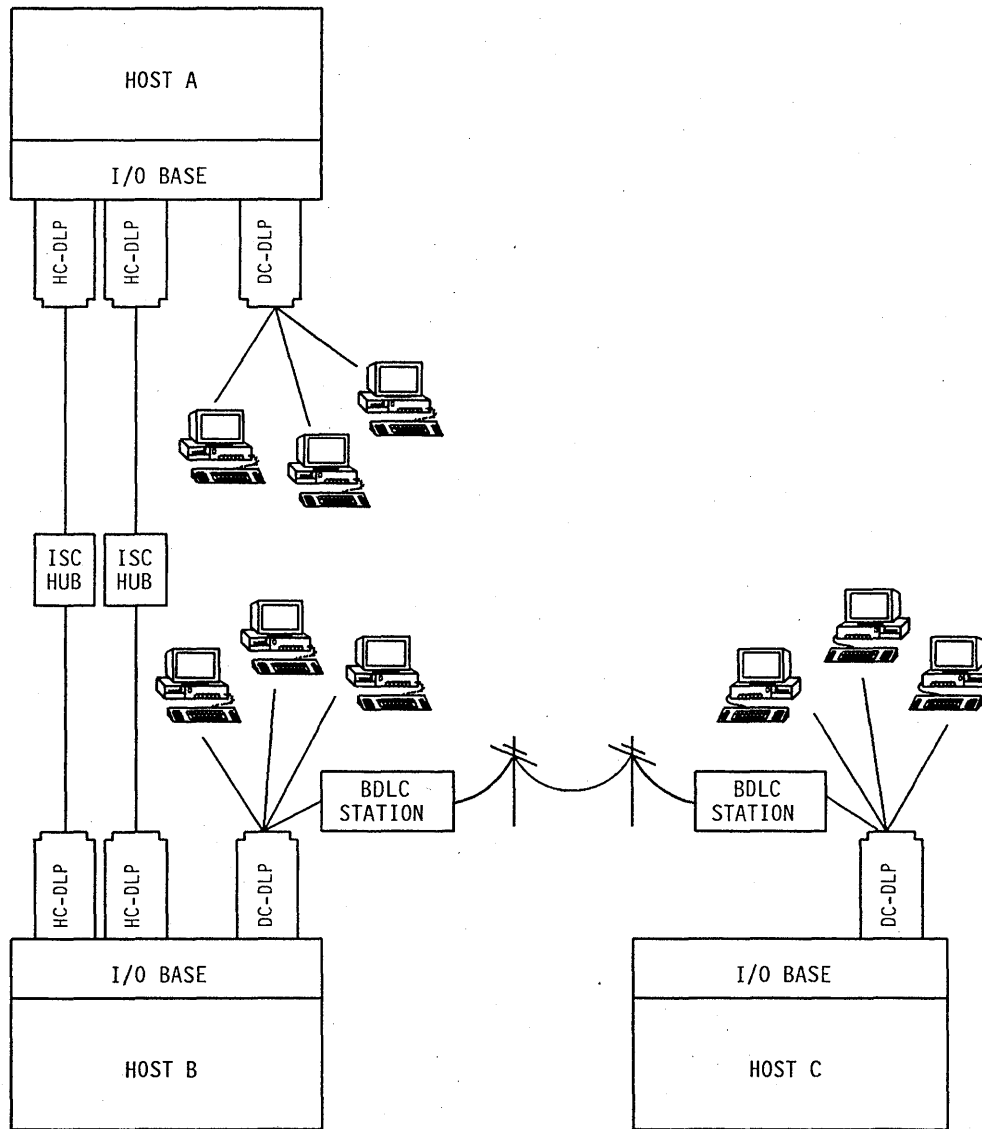


Figure 8-7. BNA Version 1 Network Connections

## Network Operations

All hosts in a BNA Version 1 network are cooperating peers, regardless of size. Each host is equal and has equal responsibility for running the network. The BNA Version 1 routing algorithms respond automatically to changes in the network configuration and characteristics.

BNA Version 1 provides a number of *network commands* you can use to control network operations. You can enter these network commands through familiar system operations

interfaces, such as Menu-Assisted Resource Control (MARC) and system command mode at an operator display terminal (ODT). You can also submit network commands from the following sources:

- *Dedicated remote stations.* These are data comm terminals that are assigned special status by the TERM network command.
- *Load files.* These are ordinary disk files in which you store network commands. You can use the LOAD command to submit a load file for execution. You can also create a disk file storing network commands that are automatically executed when you initialize a BNA Version 1 node.
- *Program Agents.* This feature allows you to write programs that can submit network commands and analyze the responses. Such programs can perform routine changes to the network configuration, or even help you analyze network problems. Program Agents can free operators from the routine chores of controlling a BNA Version 1 network.

### Documentation

- *A Series BNA Version 1 Operations Guide*
- *A Series BNA Version 1 Program Agent Programming Guide*



# Section 9

## System Monitoring and Security

System monitoring and security administration are two key elements in ensuring that your system is productive and safe from unauthorized use. These two areas also overlap in that the system monitoring tools have some applications for security auditing. The A Series provides a wide selection of tools and capabilities in these areas. You can choose the tools that provide the types of information gathering and system protection that you need.

### System Monitoring

A Series system monitoring tools collect, analyze, and report on the following types of data:

- *Hardware reliability.* You can use this information to determine whether all hardware devices are functioning normally or may require maintenance.
- *Job history.* You can use this information to determine whether or not jobs and tasks ran normally. If a job or task terminated abnormally, you can determine the general cause and use this information to help debug the job or task.
- *Resource utilization.* You can use this information to determine whether any of the system resources are overused. You can plan changes in the system workload or configuration to improve response time for your applications.
- *Security-sensitive actions.* You can use this information to determine whether anyone is attempting to breach your system security.

System monitoring information is available on either a real-time, sampling basis or through later analysis of the system log. Reports can be displayed at terminals and workstations or printed. More importantly, the A Series system monitoring tools provide a variety of analysis and formatting techniques that enable you to isolate the information that can help make your system more productive and secure.

### Snapshot System Analysis

Performance analysis is the study of the rate at which a computer system is able to execute your work. Performance can be affected by many factors that a customer site can control, including system configuration and workload scheduling. Performance analysis helps you to identify changes that can increase the productivity of your system. Performance analysis also gives you advance warning of any needed equipment acquisitions or upgrades, so you can prepare and justify the budget for these items in advance.

The most basic type of performance analysis is a “snapshot” system study. A snapshot study identifies the system resources that are overused or underused, allowing you

to balance your configuration and workload for the best performance. The A Series provides an integrated, user-friendly interface to snapshot analysis in the form of the Capacity Management Facility.Snapshot (CMF.Snapshot) product.

CMF.Snapshot has the following two major components:

- *CMF.Snapshot Workstation Component.* This is a workstation interface that runs on Unisys personal computers (PCs). The workstation component of CMF.Snapshot uses INFOVIEW II workstation management software to communicate with the mainframe and manage the user environment, including windows and graphics.
- *PROBEMANAGER.* This is a set of coordinated software modules that runs on the host. PROBEMANAGER samples the system resource utilization at regular intervals and stores the performance data. PROBEMANAGER can accommodate multiple-performance workstations that are running the workstation component of CMF.Snapshot.

The workstation component of CMF.Snapshot provides you with two types of views of your system performance: a real-time performance monitor called MONITOR, and a historical performance monitor called SNAPSHOT. Figures 9-1 and 9-2 are examples of the output provided by MONITOR and SNAPSHOT.

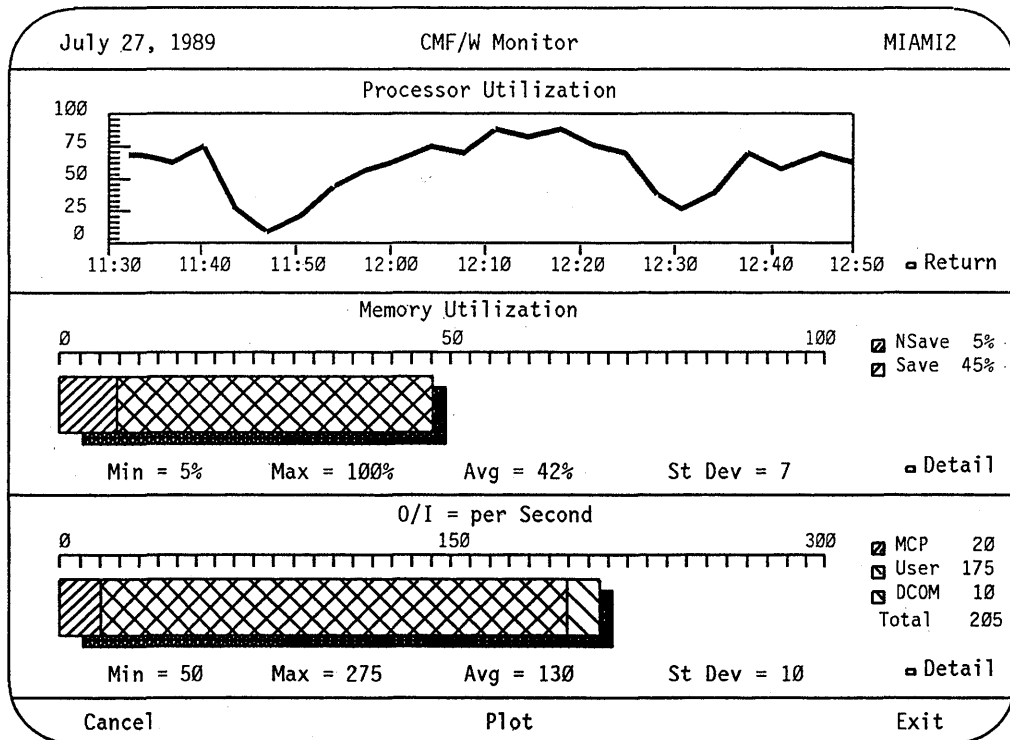


Figure 9-1. CMF.Snapshot MONITOR Screen

MONITOR is intended for use in an operations control center or an executive's office. The monitor screen is divided into three sections, or *categories*, representing three main performance concerns: processor, memory, and I/O. For each category, as many as three types of screens are available: bar graph, strip chart, and detail data. A bar graph depicts current resource utilization graphically by varying the length of a bar. A strip chart shows how resource utilization has varied over a period of hours. A detail screen lists statistics about resource utilization. You can request different types of screens for each category. Figure 9-1 shows a MONITOR screen with a strip chart for the processor category and bar graphs for the memory and I/O categories.

SNAPSHOT reports on data collected over a period of days or weeks. You can choose from a variety of predefined reports, or specify a set of reports that interest you the most and reuse that set on a regular basis. You can request reports covering specific days or specific time intervals. For example, you can request a report showing processor utilization over the first shift for a week. Figure 9-2 shows a sample of a SNAPSHOT report of processor utilization for a 24-hour period.

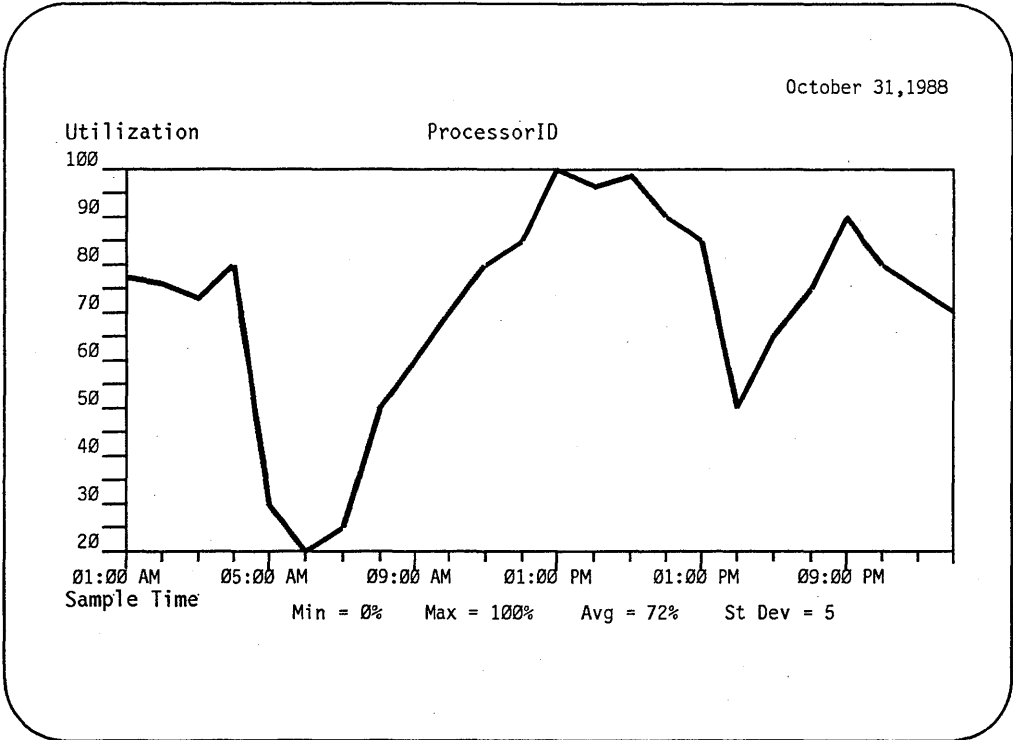


Figure 9-2. CMF.Snapshot SNAPSHOT Report

PROBEMANAGER data collection techniques are designed to make performance analysis a low-overhead exercise in terms of processor utilization and disk space. You can set parameters that control the sampling rate used by PROBEMANAGER. For MONITOR data collection, you can set additional parameters that control the amount of memory and disk resources that are used to store the data. For SNAPSHOT data collection, you can specify aggregation intervals. Aggregation is a method of storing averages, standard deviations, and variability for a set of samples in an extremely compact form. The aggregation interval is the amount of time that passes between each aggregation. A short aggregation interval allows you to generate more detailed reports later, while a long aggregation interval conserves disk space. The actual report generation logic is handled by the workstation component of CMF.Snapshot on the workstation, minimizing the impact on the mainframe.

CMF.Snapshot has an added benefit if you also have Unisys 1100/2200 Series systems at your site. The CMF.Snapshot product is available for all these systems, and it provides you with a common interface for resource utilization analysis.

### Documentation

Refer to the *Capacity Management Facility Snapshot (CMF.Snapshot) End Use Guide* for more information.

## Disk Performance Modeling

A companion to CMF.Snapshot is the Capacity Management Facility.Disk (CMF.Disk) product, which can report on disk utilization and model the effects of disk subsystem configuration changes. This product is introduced under “Disk Performance Modeling” in Section 7.

## System Commands for Performance Monitoring

The A Series provides a number of system commands that you can use to interrogate system utilization. These system commands allow you to quickly acquire utilization statistics through the MARC or system command interfaces, without having to initiate a separate utility program. Some of the system commands used for this purpose are

- *U (Utilization)*. This command displays a number of statistics about processor and I/O utilization. Utilization is divided into useful categories that make it clear how the system resources are being allocated between application programs and various system management functions.
- *CU (Core Usage)*. This command displays memory usage statistics. You can use this command to determine whether main memory is saturated.
- *DU (Disk Utilization)*. This command displays statistics about the amount of space available on a disk family and the size of the available spaces. This information can help you decide how to respond to shortages of disk space.
- *A (Active Mix Entries)*. This command displays all the currently running processes on the system. The display includes information about the accumulated processor usage of each process. This information can help you to determine whether any of these processes is using an exorbitant amount of processor time.



### Logging and Log Analysis

The A Series provides comprehensive logging of system events and accumulates statistics in a variety of areas, including data comm activity, file usage, hardware errors, job and task activity, security, and system configuration changes. This information is stored in the system log, also known as the SUMLOG.

You can use the following utility programs to analyze the information in the system log:

- **LOGANALYZER.** This utility produces a report consisting of all system log entries that correspond to parameters that you set. You can use LOGANALYZER to extract any or all of the types of entries that occur in the system log.

You can use LOGANALYZER to examine the current system log or archived system logs. You can also limit LOGANALYZER's search to entries in the system log that were made during a particular period. LOGANALYZER can display reports, print them, or store them in a disk file for later use.

- **LOGGER.** This utility generates reports that aid in the analysis of system performance and utilization. LOGGER can produce reports on job statistics, file statistics, or overall system statistics. LOGGER can summarize, total, average, and sort the data to produce a report. LOGGER can draw its data from current or old system log files. LOGGER can also update files that summarize system activity over a period of months.
- **LOGCONSOLIDATOR.** This utility extracts information from the system log into a form that can be analyzed by the QUERY utility. Both LOGCONSOLIDATOR and QUERY are part of the System Management Facility II (SMFII), and are discussed under "Detailed System Performance Analysis" in this section.

These log analysis utilities can also be used for security auditing, as described under "Security Auditing" in this section.

The LOGANALYZER utility has an interface to an optional library called SITESUPPORT, which you can implement to perform custom analysis of log entries. For example, you could design this library to translate log entries from English into other languages. Thereafter, all LOGANALYZER reports would show log information in the language of your choice. SITESUPPORT, if it is present, also preprocesses information that appears in job summaries.

You can also write application programs of your own to analyze the system log.

#### Documentation

The *A Series System Software Support Reference Manual* includes sections discussing LOGANALYZER and LOGGER. In addition, the SUMLOG section discusses log record formats and the SITESUPPORT library.

## Detailed System Performance Analysis

The System Management Facility II (SMFII) is a software system that monitors and provides data about various aspects of system performance. The function of SMFII overlaps with that of CMF.Snapshot to some extent. However, CMF.Snapshot is specialized to make resource utilization analysis as easy as possible. SMFII offers a broader range of information, including workload analysis, hardware reliability, and system availability.

SMFII consists of a number of related programs, which fall into three general categories: data collection, reporting and analysis, and data management utilities. These programs and their relationships are shown in Figure 9-3.

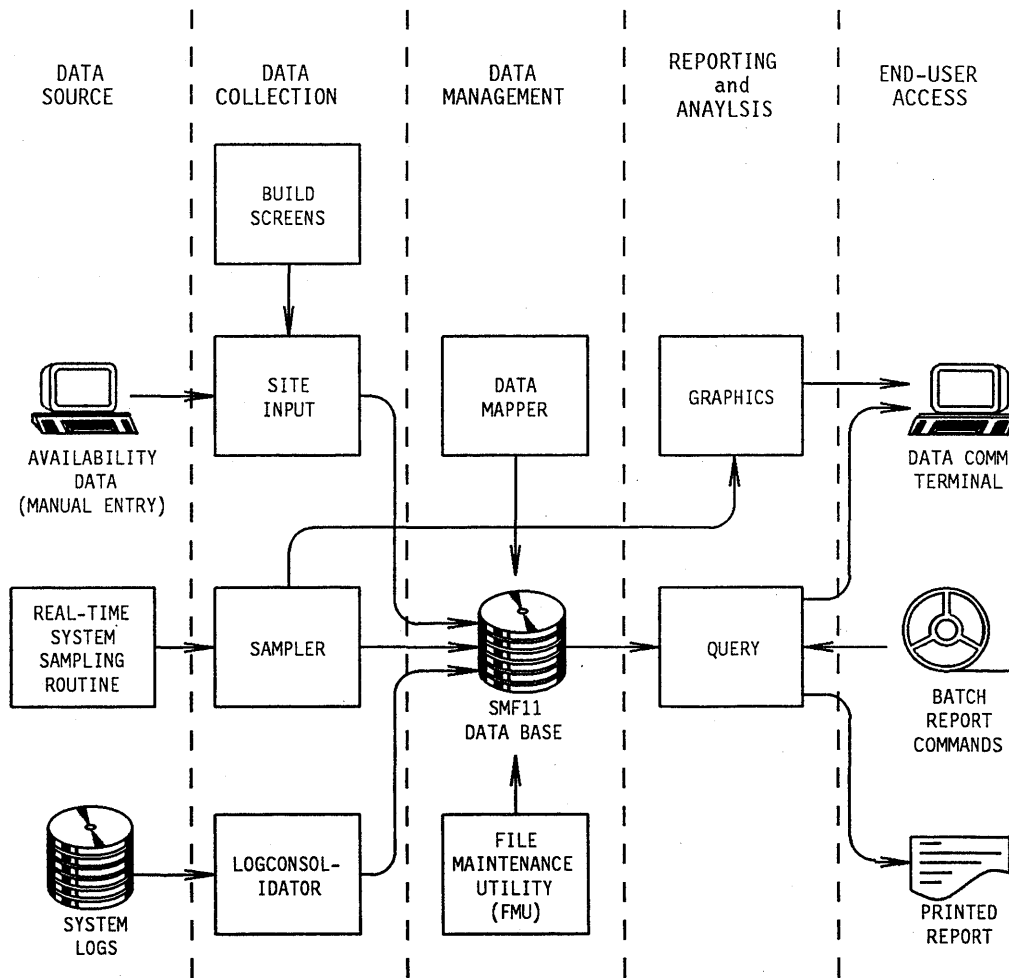


Figure 9-3. SMFII Components

### Data Collection

You can collect data using any or all of the following SMFII utilities:

- **LOGCONSOLIDATOR.** This utility extracts hardware or workload information from one or more system summary logs (SUMLOGs). The hardware information helps you to analyze the reliability of hardware devices. The workload information helps you to determine how the system resources are being divided among the various programs that run on the system.
- **SAMPLER.** This utility collects real-time performance information from the operating system. SAMPLER is designed to use a minimum of system resources so that it can be left running all the time. The types of data that can be collected include processor, memory, I/O, and data comm utilization statistics. Applications programs can also transmit transaction statistics to SAMPLER through the SAMPLER port file interface.

In addition to its data collection function, SAMPLER has a GRAPHICS facility that can optionally display patterns representing current rates of system resource utilization. Operators can learn to recognize utilization patterns and anticipate system performance problems. Figure 9-4 shows a SAMPLER GRAPHICS display for a system with heavy I/O activity.

- **SITEINPUT.** This utility collects system availability data from site personnel. You define SITEINPUT screens that ask system operators to enter particular types of information about disruptive events and the actions taken in response to these events.

### Documentation

For further information, refer to the *A Series System Management Facility II (SMFII) Resource Management Operations Reference Manual*.

### Reporting and Analysis

The LOGCONSOLIDATOR, SAMPLER, and SITEINPUT utilities all produce database modules. You can use the QUERY utility to analyze and report on the information in any or all of these modules.

Like LOGGER, QUERY can summarize, total, average, and sort the data to produce a report. However, QUERY also enables you to create expressions that filter the input data or that use the values of several data items to calculate a single report item. QUERY can produce reports in a variety of tabular or graphic formats, and gives you detailed control over the formatting of these reports. With QUERY, you can print the reports or display them at a terminal.

You can use QUERY interactively or run it as a batch program. You can also store sequences of QUERY commands in disk files for repeated use.

## Documentation

For further information, refer to the *A Series System Management Facility II (SMFII) Query Operations Guide*.

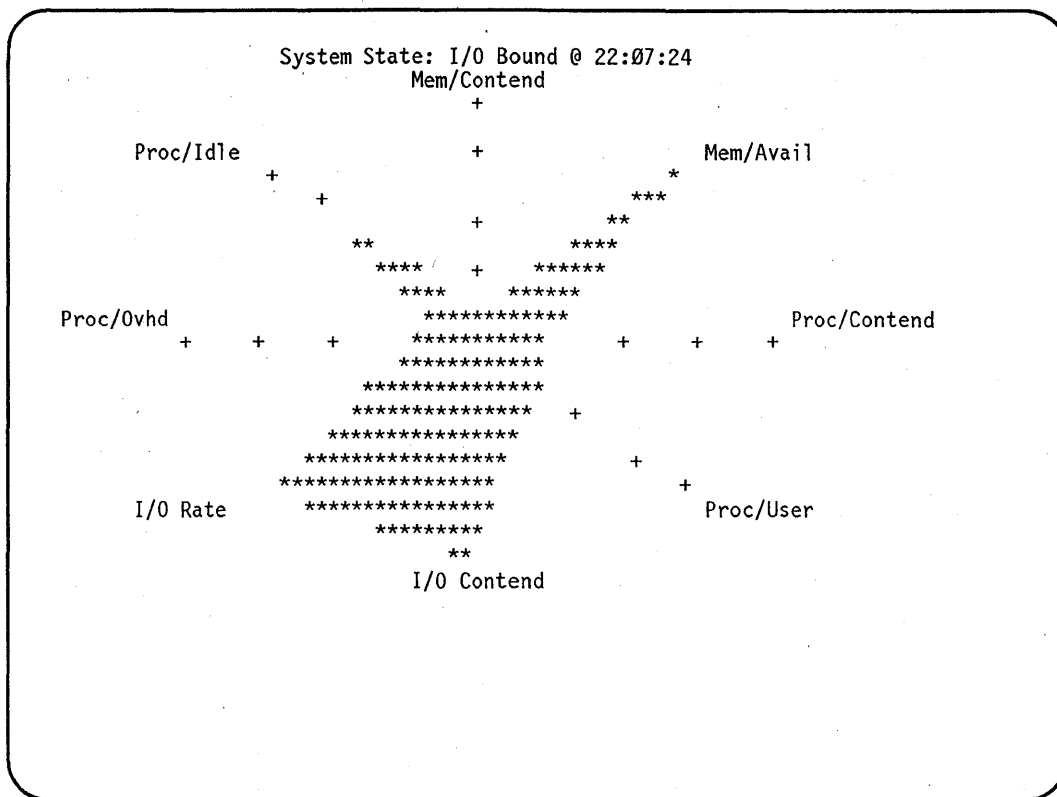


Figure 9-4. SAMPLER Graphics Display

## Data Management Utilities

SMFII provides the following utilities to assist in the management of the database files created by LOGCONSOLIDATOR, SAMPLER, and SITEINPUT:

- *Datamapper*. This utility allows you to migrate to a newer version of the SMFII software without losing access to old SMFII database files. Datamapper converts data collected by prior SMFII releases into a form acceptable to new SMFII releases.
- *File Maintenance Utility (FMU)*. As you run SMFII repeatedly over a period of time, the SMFII database files gradually increase in size. FMU conserves disk space by allowing you to selectively delete or archive old data and reorganize the data that remains.
- *SITEINPUT Buildscreens*. This utility is a development tool that helps you to create a screen interface for the SITEINPUT utility.

### Documentation

For further information, refer to the *A Series System Management Facility II (SMFII) Resource Management Operations Reference Manual*.

## Transaction Performance Monitoring

The Communications Management System (COMS) provides the Statistics facility to help you optimize and identify bottlenecks in processing transactions. You can use the Statistics facility to determine the average response time for direct window programs or particular direct window transactions.

COMS provides the Statistics window as a menu-driven interface for gathering and reporting statistics. The Statistics window allows you to specify which of the direct window programs should be monitored. You can use the Statistics window to request online reports or printed reports.

If you request an online report, COMS displays a set of statistics and periodically updates the display. You can specify how often the display should be updated. The display includes the average response time for transactions sent to that program, and the average number of input messages waiting for processing by the program. The display also includes processor, I/O, and elapsed time statistics for each program.

If you request a printed report, you can choose between several levels of detail. The COMS System Statistics report summarizes performance of all direct window programs on the system. The COMS Program Statistics report concentrates on a particular program. And the COMS Transaction Statistics report gives statistics about a particular type of transaction. You can also create customized reports, and you can optionally direct reports into disk files for later analysis by Extended Retrieval with Graphic Output (ERGO).

COMS direct window programs can also access COMS statistics through COMS service functions.

### Documentation

- *A Series Communications Management System (COMS) Configuration Guide*
- *A Series Communications Management System (COMS) Programming Guide*

## Database Statistics Monitoring

You can monitor database performance by using the DMMONITOR utility or the corresponding function in the Operations Control Manager (OCM). For an introduction to this topic, refer to "Database Performance Monitoring" in Section 3.

## Customized Monitoring Programs

The statistics gathering interfaces used by SMFII and other utilities are also available to application programs written in DCALGOL. These DCALGOL functions enable you to

write your own system monitoring programs, which can be tailored to the needs of your installation.

The following DCALGOL functions allow you to submit system commands from a program: DKEYIN, GETSTATUS, and SETSTATUS. The system commands submitted by the program can include commands that display system resource usage. Some of these commands are introduced under “System Commands for Performance Monitoring” in this section.

SYSTEMSTATUS is another important DCALGOL system monitoring function, which can gather many different types of information concerning the activity and environment of the system. SYSTEMSTATUS returns groups of related information, locking the system while it extracts the data. You use SYSTEMSTATUS when you want comprehensive information concerning a particular area of the system, such as job queue or hardware configuration information.

### Documentation

- *A Series DCALGOL Programming Reference Manual*
- *A Series GETSTATUS/SETSTATUS Programming Reference Manual*
- *A Series SYSTEMSTATUS Programming Reference Manual*

## Security Administration

The A Series provides a coordinated set of security features that allow you to

- Prevent unauthorized persons from gaining access to the system
- Ensure that authorized users of the system can access only those system resources that they are authorized to use

The keynote of A Series security features is flexibility. You can evaluate the security needs of your installation and adopt a security policy that meets those needs. For an installation handling very sensitive information, you can institute strict access controls and auditing procedures. For installations that handle less sensitive information, you can relax security controls and thus facilitate the exchange of information between users.

On another level, A Series flexibility allows you to assign different levels of access rights to various users. The users, in turn, can choose whether to protect their files from access by other users.

All A Series systems provide a common set of security features, which is discussed under “Basic Security” following in this section. An additional, optional set of security features is described under “InfoGuard Security Enhancement Software” later in this section.

### Documentation

All the security features introduced in the following pages are discussed in detail in these two manuals:

- *A Series Security Administration Guide*
- *A Series Security Features Operations and Programming Guide*

### Basic Security

Basic system security involves the use of three types of security features:

- *User, station, and program attributes.* These definitions specify the access rights accorded to particular users and programs.
- *Resource access controls.* These controls restrict access to resources such as data files, programs, databases, system commands, and application transaction types.
- *Security auditing tools.* These tools allow you to determine if security was violated, determine who violated it, and monitor the further activities of the responsible user.

You have great flexibility in choosing your system security policy because of the wide variety of user definitions, resource access controls, and security auditing methods that are available. The combination of these features determines the overall security environment for your system.

### User Privileges

The first control on access to system resources is user identification based on a *usercode*. Usercodes identify the user to the system in two different contexts, as follows:

- *Terminals controlled by an MCS.* Users access the system primarily through terminals that are controlled by message control systems (MCSs), such as the Communications Management System (COMS) or Command and Edit (CANDE). These MCSs typically require the user to enter a usercode at the start of each session. The MCSs examine the privileges that were previously defined for the usercode before allowing the user to enter any particular command.
- *Processes run with a usercode.* If a program is initiated by a user in a CANDE or Menu-Assisted Resource Control (MARC) session, the resulting process typically inherits the usercode of the session. If the process initiates another program, the resulting process typically inherits the usercode of the original process. In this way, all the processes that were directly or indirectly initiated by a user remain associated with that user. The operating system evaluates the rights of a particular process by examining the privileges of the associated usercode.

Each usercode can have an associated password to ensure that the usercode is used only by the intended person. The usercode itself can be displayed publicly without compromising security, so long as the password is kept confidential. When the user logs on to a CANDE or MARC session, the user must provide a password in addition to a usercode. The MCS denies access to the system if the usercode is not valid or the password does not belong with the usercode.

When a user at a CANDE or MARC session initiates a process, the user can append a USERCODE task attribute assignment that causes the process to run under a usercode different from the session usercode. A process can also use USERCODE task attribute

assignments to change its own usercode while it is running, or to initiate a different process under a different usercode. In each of these cases, however, the USERCODE assignment must also include the associated password, or the assignment is not accepted.

The most basic privilege granted by a usercode is the ability to access files whose titles begin with that usercode. For example, a program running under usercode SMITH could access a file titled (SMITH)FILEA ON DISK. (This topic is discussed further under "Data File Access Controls" in this section.)

However, the use of a particular usercode can grant access to a wide range of other capabilities, depending on the attributes that are defined for that usercode. Usercodes are typically defined by a system administrator or senior members of the operations staff. You can create usercodes and define attributes for them with the MAKEUSER utility. You can define attributes for each usercode that assign, among other things, the following privileges to the user:

- The ability to perform certain restricted functions, such as accessing private files stored under other usercodes; this ability is referred to as *privileged status*.
- The ability to enter most system commands in MARC.
- The ability to enter CANDE network control commands.
- The ability to run the COMS Utility.

To create an open system, you assign all the possible privileges to all users. But to create a secure system, you restrict most privileges to a small number of personnel who need those privileges to perform their duties.

You can also require particular users to furnish a secondary identification in addition to the usercode. This secondary identification consists of an *accesscode* and *accesscode password*. The use of accesscodes allows you to enforce different access rights to a file for multiple users who share a common usercode. The accesscode password verifies a user's right to use an accesscode, just as a password verifies the right to use a usercode. You can use guard files to specify which accesscodes can use a particular file, as discussed under "Data File Access Controls" in this section.

### Station Privileges

Another way of defining access rights to the system is on a station-by-station basis. This method can be useful if there are physical restrictions at your site that enable you to control who has access to each station. For example, if a particular station is located in a room where only operations personnel are admitted, it can be convenient to assign special privileges to that station. The following are some of the special privileges that can be assigned to a station:

- The ability to run the COMS Utility
- The ability to enter CANDE network control commands
- The ability to log on to a MARC session with a null usercode and access all system commands



### Program Privileges

While a program is running, it has the privileges associated with the usercode it is running under. However, it is also possible to permanently assign privileged status to a program. The program then assumes privileged status each time it is run, regardless of which user initiates it. The advantage of privileged programs is that they can perform a particular privileged action for a user without the necessity of granting the user access to any other privileged actions. You can limit access to privileged programs with the techniques discussed under "Program Access Controls" in this section.

### Data File Access Controls

The owner of a file can specify the level of protection that is needed for that particular file. By default, files are treated as *private*. A private file can be accessed by a particular agent (for example, a CANDE user or a program run) only if that agent has the same usercode as the file, or else has privileged status. If the owner of a file wishes to make it available to everyone, he or she can make the file *public* instead.

The owner can also specify the ways in which a public file can be accessed by other users. For example, the owner can specify that other users can read from the file, but cannot write to it.

More detailed access control can be achieved through the use of *guard files*. A guard file is a special type of file whose only purpose is to store information about access restrictions for another, associated file. The guard file can include lists of the usercodes, accesscodes, and programs that are allowed to access a file.

The user can use the SECURITY command to specify whether a file is public, private, or protected by a guard file. The SECURITY command is available in CANDE, MARC, or Work Flow Language (WFL). Users can use the GUARDFILE utility to create guard files. Programs that create output files can use file attribute assignments to specify the file security.

### Program Access Controls

Programs are files of a special type, called *object code files*. The system allows you to protect programs by using the same techniques used to protect data files. You can make a program public or private, and assign a guard file if desired. These controls limit the ability of other users to run or copy the program.

Additionally, the A Series assigns a special security status to object code files that has no parallel for data files. The rule is that only compilers can write to object code files. A particular program attains compiler status only if a system operator uses the MC (Make Compiler) system command to mark it with that status. If a program that is not a compiler attempts to write to an object code file, the operating system aborts the write operation with an error.

The system also allows you to prevent users from importing programs from other systems without authorization. You can use the RESTRICT (Restrict Unit or Volume) system command to specify that files copied from particular sources, such as particular

tape units, are restricted. If a code file originates from a restricted source, then no one can run that code file, no matter where the file is currently located. Only a system operator can remove the restriction by using another form of the RESTRICT command.

On systems where security administrator status is enabled, use of the MC and RESTRICT commands is limited to those with security administrator status. Refer to “Security Administrator Status” in this section.

The restrictions against writing to code files and importing code files have the natural side effect of thwarting many computer *viruses*. A computer virus is a program that breaches system security and also creates many copies of itself. These copies can continue executing and breaching system security long after the original virus is terminated. Computer viruses have been a topic of increasing concern in recent years in the computer industry. The A Series restrictions against writing to code files make it difficult for a computer virus to reproduce itself. The restrictions against importing code files make it difficult for anyone to introduce unsafe code onto the system.

### Database Access Controls

Access to databases can be controlled through system-level security features and through security features within the Semantic Information Manager (SIM), Structured Query Language Database (SQLDB), and Data Management System II (DMSII) software. System-level security is achieved by assigning a guard file to the database or to particular database files. Guard files selectively limit access to these files, as described under “Data File Access Controls” in this section.

For Semantic Information Manager (SIM) databases, you can supplement the system-level security by implementing security restrictions through the Advanced Data Dictionary System (ADDS) or directly through SIM ODL. The SIM schema allows the inclusion of accesses and permissions. The combined effect of these security features is that a usercode/accesscode/program combination can be restricted from accessing, modifying, or deleting various entities, classes, and attributes.

For SQLDB databases, the ANSI standard defines a system of *views* and *grants*. With these features, any usercode can be given various access to any row or column of information in an SQLDB database.

For DMSII databases, you can use ADDS or the Data and Structure Definition Language (DASDL) to define database features that supplement the system-level security. ADDS and DASDL allow you to create multiple *logical databases* for each physical database. A logical database provides a view into selected data sets in the database. Application programs that open the database under the logical database name can access only the data sets that are allowed for that logical database. You can then restrict particular users to particular logical databases by associating separate guard files with the database itself and with each logical database. As a result, you can make selected data sets visible to some users but not to others.

Additionally, ADDS provides security features that protect database structure definitions from being viewed by unauthorized users. Thus, users are not only prevented from accessing selected data structures, but they are also prevented from finding out that those data structures exist.

### Documentation

- *A Series Security Features Operations and Programming Guide*
- *A Series InfoExec Advanced Data Dictionary System (ADDS) Operations Guide*

### Transaction Access Controls

The Communications Management System (COMS) plays a major role in regulating security for online transaction processing programs. COMS is introduced under “Generalized Transaction Control” in Section 4.

COMS allows you to define relationships between several types of entities, including usercodes, stations, windows, programs, and transaction types. You can use the COMS Utility to create these relationships. The COMS Utility also allows you to create special security categories that can be associated with various entities. The flexibility of COMS configuration enables you to regulate resource access on any of several levels, including the following:

- *Usercode level.* You can specify the windows that can be used by a particular user, and the stations from which the user can access COMS. You can also associate a usercode with a particular security category.
- *Station level.* You can specify privileges for particular stations, such as the ability to run the COMS Utility. You can also associate a station with a particular security category.
- *Program level.* You can specify that only usercodes or stations with a particular security category can access a particular direct window program.
- *Transaction level.* You can specify that only usercodes or stations with a particular security category can submit a particular type of transaction.
- *Processing item level.* You can develop COMS processing items that perform security checks on usercodes and station names.

COMS also allows you to define security stations, as discussed under “Security Message Workstations” in this section.

### Documentation

- *A Series Communications Management System (COMS) Configuration Guide*
- *A Series Communications Management System (COMS) Programming Guide*

### Operations Interfaces Access Controls

System commands can affect many system resources, including data files, programs, system hardware devices, and system software operations. Accordingly, for a system to be secure, access to these commands must be restricted to system operators. The A Series allows you to place such restrictions on all the sources that can submit system commands: MARC sessions, DCALGOL programs that use functions such as DCKEYIN, and terminals running in system command mode.

MARC requires users to log on, and consults each usercode definition in order to allow or restrict access to particular system commands. DCALGOL programs must be privileged or running under a privileged usercode to submit most system commands.

Because system command mode does not require the user to log on, it is usually restricted to terminals that are physically protected from all but authorized personnel. Alternatively, you can use the RESTRICT (Restrict Unit or Volume) system command to specify that a particular terminal can operate only in MARC mode rather than system command mode. You can associate a security key with the restricted terminal so that only users who know the security key can restore the terminal to system command mode.

### Multiple Host Network Access Controls

When A Series hosts are linked together in a BNA network, each host has the ability to protect its resources from unauthorized use by any of the other host systems. This topic is introduced under "Access Control" in Section 8.

### Security Auditing

Security auditing allows you to detect security breaches, resource misuse, and other security-relevant activities. When attempted security violations are detected by an audit, you can take steps to remedy the problem. Such steps might involve changes in the system security policy, changes of passwords that might have been compromised, intensive auditing of the actions of particular users, changes in the system access rights allowed certain users, and, in extreme cases, denial of physical access to the system for some users.

Auditing is accomplished through analysis of the system log. The system log includes entries for user MCS session activities as well as actions committed by running programs. A special category of system log entries records all security violations that occur on the system, and indicates the usercode and originating station of the offender. Security violations include attempts to log on with an incorrect usercode or unauthorized attempts to change or copy a file.

You can generate reports on security-related log entries by using any of several utilities: LOGANALYZER, LOGGER, and LOGCONSOLIDATOR. These utilities are introduced under "Logging and Log Analysis" in this section.

The system log information could be considered sensitive in itself, because it records all major activities on the system. Accordingly, you can protect system log files from unauthorized access by using the same controls as for ordinary data files. For example, you can declare system logs to be private, or you can associate guard files with them.

### Documentation

- *A Series System Software Support Reference Manual*  
This manual describes the system log and the LOGANALYZER and LOGGER utilities.
- *A Series System Management Facility II (SMFII) Resource Management Operations Reference Manual.*  
This manual describes the LOGCONSOLIDATOR utility.

## InfoGuard Security Enhancement Software

InfoGuard is an optional software product that provides enhancements and additions to security for A Series systems. The InfoGuard features increase the level of system security and simplify the task of configuring a secure system.

### Simplified Security Administration

InfoGuard provides a centralized interface for assigning or displaying system security options. The MARC *SECOPT* screen includes prompts and online help for all security options. Alternatively, you can use the SECOPT (Security Options) system command.

Although the SECOPT screen and command are also available without InfoGuard, InfoGuard greatly increases the range of options that can be controlled through this interface. For example, InfoGuard allows you to request *disk scrubbing*, which erases the contents of disk files before allowing the disk space to be reused. You can also declare tape scrubbing, which serves a similar purpose for tapes.

You can also use SECOPT to assign an overall security class to the system, rather than assigning values to individual security options. Each security class establishes a different set of defaults and restrictions for the security options. There are five different security classes, ranging from the MINIMAL class to the S2 class. The MINIMAL class suspends most security checking and causes all users to be treated as privileged. The S2 class provides the most stringent security controls.

### Security Administrator Status

InfoGuard provides a special level of user privilege referred to as *security administrator status*. This status can be assigned to usercodes or programs. When security administrator status is enabled on the system, actions that affect system security policy are available only to the security administrator. On systems not running InfoGuard, these actions would be available to any privileged user or system operator.

Some of the actions that require security administrator status are

- Creating, modifying, or displaying usercode definitions
- Using the SECOPT (Security Options) system command
- Controlling system logging activity

## System Monitoring and Security

---

- Assigning privileged status or compiler status to a program

### Password Generation and Aging

InfoGuard enables you to designate a lifetime and a warning period for passwords. When a password reaches a certain age, the user is warned that the password will expire in so many days. At the end of that time, the password can no longer be used to access the system.

You have the software-supported option of allowing users with expired passwords to generate a new password for themselves, or of requiring them to apply to the security administrator for a new password. If you allow users to generate their own passwords, you can specify a minimum length for passwords and prevent users from reusing any of their recent passwords.

You can also require all users to use passwords that were generated by the computer. A MARC password screen provides the user with an easy-to-use interface to the system, which generates a random password 7 to 16 characters long. For ease of memorization, the password is constructed by concatenating short English words.

### Tape Security

InfoGuard enables you to use tape security features similar to those provided for disk files. This means that a tape volume can be owned by a particular user, that it can be public or private, that usage can be restricted to read-only or write-only, and that the tape volume can be guarded with a guard file. You can assign tape security restrictions using the *WFL VOLUME* statement.

### Usercode Monitoring

InfoGuard provides several features that make it easier to determine if a particular usercode is being abused.

When a user logs on to CANDE, InfoGuard causes the system to display information about the most recent occasion the same usercode was used, either as a log-on value or as an assignment to a program run. This information is also available in MARC if the user requests it. This information can alert a user to the fact that someone else is using his or her usercode. The user can prevent further misuse of the usercode by suspending the usercode with the *MARC SUSPENDUSERCODE* command or the corresponding MARC menu selection. Only someone with security administrator status can restore the usercode.

InfoGuard allows you to request that the system keep a running count of the total number of security violations committed by a particular user. (Note that a violation is simply an attempt to do something that is unauthorized on the system. The system does not allow such attempts to succeed, but it does log them.) The system also keeps track of the dates when the violations occurred. You can specify that a certain usercode should automatically be suspended if it exceeds a certain number of security violations in a day.

InfoGuard also provides a special MAKEUSER command that generates reports on password usage, suspended usercodes, and violation counts for usercodes. This command, called *ACCESS*, makes it simple for you to determine whether usercodes are being misused.

### Security Message Workstations

InfoGuard allows you to designate a station to act as a security message workstation. All system security messages are routed to the security message workstation, allowing for easier monitoring of security-related activities.

The security message workstation reports both security violations and other events relevant to security. Examples of this latter type of message are initiations of privileged programs, the use of commands that are critical to security, and the logging on of users with privileged or security administrator status. To help you trace the origin of security violations, the security messages provide the station name or unit number from which the job causing the violation was started.

### Logging Enhancements

With InfoGuard you can use the LOGGING (Logging Options) system command to specify the types of information that will be recorded in the system log. By restricting the types of information that are logged, you reduce system overhead and improve performance. You can selectively restore the logging of any types of activity that you decide are important. InfoGuard also allows you to use the LG (Log for Mix Number) system command to cause additional logging of the activities of a particular job or task. Both the LOGGING and LG commands are also available as MARC screens.

Alternatively, you can cause additional logging for the activities of a particular user by running the MAKEUSER utility and assigning a LOGSELECT attribute to a usercode.

The InfoGuard System Data Access (IGSDA) support library provides controlled access to the system log. If the system log is public, IGSDA allows all users to read any part of the system log. If the system log is private, IGSDA provides nonprivileged users with filtered access to the system log. This means that IGSDA allows these users to read only log entries pertaining to their own activities and certain system activities that are considered nonsensitive. IGSDA thus allows nonprivileged users to obtain information they need without allowing them to obtain information about the activities of other users.

All log analysis utility programs access the system log through IGSDA, and thus provide filtering. These utility programs include LOGANALYZER, LOGGER, and LOGCONSOLIDATOR.

### C2 Security

Unisys A Series systems running standard software and employing InfoGuard have been evaluated at the C2 controlled-access protection level by the United States Department of Defense National Computer Security Center.

## System Monitoring and Security

---

For a system to be approved for class C2 operation, it must be evaluated and certified by the United States Department of Defense National Computer Security Center. To meet the requirements for class C2 certification, as described in the *Department of Defense Trusted Computer System Evaluation Criteria* DoD 5200.28-STD, a system must have features such as the following:

- Identification of individual users through a log-on procedure
- Accountability of action at the level of the individual user
- Secured reuse of system resources such that no user gains unauthorized access to data

The C2 certification means that you can use A Series systems in environments that have high security requirements, such as defense contracting. Currently, the Mark 3.7 system software release has been evaluated and certified.



# Part 2

## **Support**



# Section 10

## **CUSTOMCARE Services**

A Series systems are backed by a broad range of education services, implementation and consulting services, and services for hardware and software support. These services help you to make the most effective use of A Series products, tailoring the system to your needs and maximizing system performance. These services also ensure that qualified personnel are available to help you with any problems that might arise.

The customer support services offered by Unisys are known collectively as CUSTOMCARE service.

This section describes services that are available in the United States. Unisys International customers should consult their Unisys representative for information about the services available in their country. United States customers can also consult their Unisys representative for details about any of the services described in this section.

### **Education**

The better you and your end users know your systems, the more value these systems have for your company. The fastest way for you to learn about A Series products is through personal interaction with experts in each product area. Unisys provides this interaction in the form of customer education courses.

Unisys customer education provides training for all the people who manage, design, program, operate, or use A Series systems. Courses are available to teach you how to use system software and line-of-business applications. Other courses teach you how to develop applications of your own through facilities such as the LINC II or MAPPER systems.

Unisys regularly holds education classes in about 20 major U.S. cities, including Atlanta, Boston, Chicago, Cleveland, Dallas, Los Angeles, Minneapolis/St. Paul, New York, Philadelphia, Princeton, San Francisco, and Washington, D.C. You can also make special arrangements for classes to be held at your site or at another mutually agreed-upon location.

Unisys offers educational consultation to help you determine what training your personnel need. An educational consultant can assist you in assessing skill levels of personnel, determining training requirements, and developing training plans to meet your particular needs.

You can also request special courses when standard courses do not meet unique training requirements. The objective of a special course is to provide maximum training at the lowest cost. As an example, an assessment of a particular data processing staff might show that skill levels are above the entry-level position for which a 10-day standard course was designed. By emphasizing the needed skills and deleting unnecessary or

repetitive material, a Unisys instructor is able to teach the same course in fewer days, or cover additional materials in the same period.

Another option is the use of self-paced instruction programs, which include various media such as workbooks, audiotapes, videotapes, and computer tapes.

## Professional Services

Unisys professional services provide you with consultants in a wide variety of fields including application development and system management.

Unisys consultants can implement entire application systems according to your specifications, or can modify an existing application system. Unisys consultants can also help you with fourth-generation languages such as LINC II and MAPPER systems.

These applications consultants have specialized training and expertise in the requirements of your line of business, including

- Airlines and travel
- Communications
- Defense
- Distribution
- Education
- Engineering/Scientific
- Finance
- Government
- Health care
- Manufacturing

Unisys applications consultants follow a family of development methodologies and tools called Professional Project Practices (PPP). This approach keeps you involved at every phase. With PPP, you follow the status and approve the direction at all times. You also gain a thorough knowledge of the end product long before delivery so that you can ensure that the results meet your objectives on time and within budget.

Unisys systems consultants can help you configure, implement, manage, or support your system. These consultants can also help you implement multivendor environments so that diverse products work together as a seamless entity.

One popular program is the A1/A4/A6 Starter Plan, in which a Unisys consultant, following a well-defined plan, analyzes the needs of a new A1, A4, or A6 customer, then works with the customer to install systems software on the new system. The two key benefits to the customer are a configured, working system and improved knowledge of system administration and operations gained by working with a Unisys consultant.

Unisys performance consultants can perform a variety of studies for you. A basic snapshot study identifies the utilization levels for system resources and helps to isolate bottlenecks. More detailed studies can measure and analyze system workloads and help you to optimize your system for maximum performance. Capacity planning studies can compare the available system capacity with your growth plans and provide you with a plan specifying when equipment should be upgraded or supplemented.

## Progression Aids

Unisys provides consulting services and, in some cases, specialized tools to simplify progression from other Unisys systems to A Series systems.

One common progression path is from B 1000 Series systems to A Series systems. The following progression tools that aid this process are included in the A Series system software release:

Progression Tool	Function
Data Base Data Transfer Utility (SYSTEM/DBTGGEN)	Uploads data from a B 1000 Series DMSII database to an A Series DMSII database
SYSTEM/BCT Translator Filter	Translates B 1000 Series COBOL68 and B 1000 Series COBOL74 to A Series COBOL74
SYSTEM/BRT Translator Filter	Translates B 1000 Series RPG to A Series RPG
SYSTEM/B1000COPY	Reads B 1000 SYSTEM/COPY tapes on an A Series system
SYSTEM/CMSCOPY	Reads tapes created on a CMS mode system on an A Series system

B 1000 products that provide an easy conversion route to the A Series include

- *On-Line Data Entry System (ODESY)*. The A Series ODESY provides functional and visual equivalence to B 1000 Series ODESY. A program for progressing ODESY format dump files from B 1000 to A Series is also provided.
- *LINC II system*. You can migrate LINC II application systems from B 1000 Series to the A Series without the necessity of any LINC Definition Language (LDL) changes. LINC II is introduced in Section 5.

### Documentation

- *B 1000 Series to A Series Progression Guide*
- *A Series Data Base Data Transfer (DBT) Utility User's Guide*
- *A Series On-Line Data Entry System (ODESY) Installation Reference Manual*

## Unisys SURETY

Unisys tests all software products before shipment to ensure the highest standards of reliability. Typically, more than half the total time, effort, and cost of producing a software or hardware product is spent testing and debugging. However, in the event that Unisys software fails to perform to specifications, you can resolve the problem through Unisys software support services.

The Unisys SURETY Service Program integrates coverage for hardware and software services and support, thus customers call only one telephone number for all needed services.

With Unisys SURETY, you can choose the level of maintenance and support coverage that you receive. This flexibility allows you to arrange for the type of support that is most cost-effective for your site.

Among the services provided by Unisys SURETY are the Unisys SureNet electronic bulletin board and preferred prices on environmental products, supplies, and microproduct accessories.

## Unisys SureNet

The Unisys SureNet electronic bulletin board system is designed for both system and workstation users and provides the following features:

- Online help screens, menu-drive commands, and queries to guide you through the system
- Common data communication software that is compatible with most personal computers and workstations
- Connection to SureNet through standard telephone line and asynchronous modem with speeds of 300 to 9600 bps

Unisys SureNet provides access to two major categories of information, Public Service and Support Service.

The *Public Service* modules are available at no charge through a toll-free telephone number. They contain a broad spectrum of information about Unisys products and services, including

- Product information that is organized so you can access only that which interests you
- Unisys services directory that provides a quick reference to help you find the resources you need
- Customer education listings that include a course catalog, descriptions, schedules, prerequisites, and registration information
- New product and service announcements that keep you apprised of the latest technological innovations and service solutions
- Press releases that keep you aware of innovative solutions, unique applications for products and services, and news about Unisys Corporation

- A feedback mailbox that solicits your comments about current features and requests or suggestions for the future development of the Unisys SureNet bulletin board system

The *Support Service* modules provide rapid problem resolution by guiding you to implement corrective actions in the event of system problems. The Support Service modules provide the following information:

- Problem-specific support, including problem symptoms and either corrective or temporary solutions.
- Product-specific information, featuring fourth-generation languages, application software, local area networks, and data communications.
- System operations support, including technical bulletins and publication updates, open systems interconnect specifications, and compatibility matrixes for the integration of Unisys products and products from other companies.
- Additional system operations support, including recent product performance improvements, solutions to commonly asked questions, and configuration and portation information. You can also check the status of correction requests through the Unisys Communication Forms (UCFs).
- Electronic mail and technical conferencing that allow you to ask technical questions and to suggest solutions through the online SureNet system. You can either submit your queries to Unisys for action and dissemination, or you can share them with other system users.

The Support Service modules are available only to customers who have a service agreement and have purchased a Unisys SureNet subscriptions. Support Service is directed primarily to system analysts, programmers, and other technically proficient system users.

## Unisys SURETY Service Options

The available Unisys SURETY prepackaged service options are:

- *Intro services.* Among the services offered with this package are centralized telephone support for both hardware and system environmental software, essential engineering changes and Unisys SureNet.
- *Basic services.* Among the services offered with this package are all the Intro services, as well as replacement parts, carry-in/ship-in service (where applicable), and preferred rates on Unisys supplies and cables.
- *BasicPlus services.* Among the services offered with this package are all the Basic services, as well as central support through a toll-free telephone number, on-call hardware support, preventive maintenance, software maintenance releases, and preferred rates on environmental products.
- *Comprehensive services.* Among the services offered with this package are all the BasicPlus services, as well as on-site systems operations reviews, guaranteed response and telephone access times, and installation support. Comprehensive services are available during your site's hours of operation.

## **CUSTOMCARE Services**

---

Unisys SURETY is available either directly from Unisys or through the many Unisys value-added resellers (VARs). For customer-specific requirements that are beyond the Unisys SURETY program, separately priced services are available.



# Section 11

## System Updates

Unisys has established regular procedures for issuing updated versions of hardware and software products, as well as entirely new products. These procedures allow you to integrate new capabilities into your system at your own pace, with minimal disruption to your normal operations.

### Software Releases

Updated versions of the software are periodically offered to all customers as *support releases*. The support release software includes all fixes that have been made to critical and noncritical problems since the latest Mark release. A support release is identified by a three-part number whose last digit is incremented by 1 for each support release. For example 3.9.2 would be the second support release issued after the Mark 3.9 system software release.

*Kit releases* are occasionally issued to provide support for a new hardware device, or to provide a major new software product. The purpose of kits is to enable customers to use the new hardware device or software product before the next Mark release.

The *Individual Kit* introduces a new product and most often cannot be used in conjunction with another Individual Kit. When any given Individual Kit is qualified, that kit is then merged with the next appropriate support release. Usually, an Individual Kit name refers to the hardware or software product that the kit makes available; for example, A4-6 Model S.

A *critical support object tape* is issued to you if you submitted a User Communication Form (UCF) reporting a critical system problem. The critical support object tape includes only those software products involved in the reported problem. The software incorporates corrections for the reported problem, as well as corrections for all other critical problems reported for this product since the latest Mark or support release. (If you acquired a kit release, the software corrections are relative to all critical problems reported for this product since the kit release.) The software is issued as precompiled object code files. The source code patches are also sent if your organization licenses source code.

#### Documentation

For an overview of the Mark 3.9 system software release, refer to the *A Series Mark 3.9 Software Release Capabilities Overview*.

### Simple Installation (SI)

The process of copying and installing a new A Series software release is greatly expedited by use of the Simple Installation (SI) product. By automating and simplifying

## System Updates

---

the installation process, SI minimizes installation mistakes and reduces the time required to install a release.

Normally you run SI interactively, using SI menus and online help. You can also use SI in batch mode or access it indirectly by using the INSTALL (Install Software) system command. The INSTALL system command allows you to install your software before your data communications have been established, such as after a cold start.

The following are some of the capabilities of SI:

- SI can install a Mark or support release, or just selected files. SI also can install by product or software categories (compiler, library, MCS, and so forth).
- SI can copy the software without installing it or install the software without copying it. This allows you to prepare for installation at one time, but execute it at another.
- SI can check a family for files from the software release, verify that the files are from the correct release level, and copy and/or install all the missing software on that family.
- SI can move software from one family to another. SI maintains a record of the family where each software file currently resides.
- SI allows you to install for testing and verification purposes. Software installed for such purposes can be placed on a test family, or it can be assigned a usercode or naming convention that distinguishes it from the production software in use on the system.

SI does not install critical support object tapes. These tapes include only the software changed by the correction, and are accompanied by directions for installation.

### Documentation

For further information, refer to the *A Series Software Release Installation Guide*.

## Progression Policy

A Series design policies and support policies ensure that you can progress to new releases at the time that is most convenient to you, and with the least possible disruption of your normal system operations.

Each Mark release stream continues to be fully supported for at least 90 days after the second subsequent Mark release becomes available. For example, the 3.7 release would be supported for at least 90 days after the 3.9 system software release becomes available. Thus, a substantial period passes between the time a new release becomes available and the time you have to progress to it.

A Mark release stream might be supported even longer if it is the last stream that can be run on a particular system model. In such a case, the stream continues to be supported on that model only. The period of extended support can be as long as several years, depending on the model involved.

Release compatibility policies for the A Series include the following:

- Any feature deimplementations that would require modifications to application source code are announced two Mark release streams in advance. For example, deimplementations that occurred in the Mark 3.9 release stream were originally announced in the Mark 3.7 release stream.
- A program compiled under one Mark release stream can be run without recompilation under the next three Mark release streams. The only restriction is that the program must not use any deimplemented features. For example, programs compiled under the Mark 3.6 release stream can be run under the Mark 3.9 release stream. The system issues a warning whenever you run a program that needs to be recompiled before it can run on the next Mark release.

Additional restrictions apply when moving to new A Series models. Each new A Series model is associated with a Mark release stream, which is the first stream that provides system software that can run on that model. For example, 3.7 was the first Mark release stream to support A 17 systems. An effort is made to provide compatibility with application programs compiled under previous releases, but the exact amount of backward compatibility provided varies from one model to another.

- A program compiled under one Mark release stream can also be run without recompilation under the previous Mark release stream. The only restriction is that the program must not use any new features that were not available under the previous Mark release stream. For example, a program compiled under the Mark 3.9 release stream would also run under the Mark 3.8 release stream, as long as it uses no new Mark 3.9 features.
- The previous rule also applies to compilers. Thus, the Mark 3.9 COBOL74 compiler can be run on a Mark 3.8 system. Though the compiler is running under Mark 3.8, any code files created by the compiler are considered to be Mark 3.9 code files. This rule allows you to recompile all your applications before moving to the next release. The compiler flags any deimplemented features in the program with syntax errors.

Note, however, that the compiler does not support features any newer than the release stream it is running under. For example, the 3.9 COBOL74 compiler running on a Mark 3.8 system does not generally support new 3.9 features in the programs it compiles. If a product has any new features that can be compiled under the previous release, this capability is explicitly discussed in the product documentation.

- Both Semantic Information Manager (SIM) and Data Management System II (DMSII) databases can be explicitly updated from the previous Mark release stream to the new Mark release stream, or moved back from the current Mark release stream to the previous Mark release stream. Typically, the update does not require you to make any changes to the database definition, but does require that the database definition be reprocessed by the Data and Structure Definition Language (DASDL) compiler. If any conversion of physical database files is also necessary, appropriate automated utilities and documentation are provided for forward and backward migration.

Application programs that access a database on one Mark release stream do not need to be recompiled to access the updated database on the next Mark release stream.

## System Updates

---

Every effort is made to conform to these rules; however, when these rules cannot be met, Unisys reserves the right to make exceptions. The documents in the following list include information about any such exceptions.

### Documentation

- *A Series Mark 3.9 Software Release Capabilities Overview*

This manual provides brief descriptions of all new products and major feature enhancements associated with the Mark 3.9 release. This manual also lists all current and future deimplementations of products and features. Finally, it lists compatibility issues and gives special installation instructions for certain products. A new version of this manual is published for each Mark release.

- *Release Notes*

This letter is issued for each Mark or support release. For Mark releases, this letter provides last-minute additions to the information in the *A Series Mark 3.9 Software Release Capabilities Overview*. For support releases, this letter outlines the corrections that are made available by this support release.

- *Mark 3.9.0 Documentation Changes List (D-Notes)*

This collection of notes describes new features in more detail than the *A Series Mark 3.9 Software Release Capabilities Overview* does. These notes are shipped with each Mark release on a special tape called the DOCUMENTS tape. The DOCUMENTS tape includes a program called LISTNOTES, which you can use to print these notes.

- *Mark 3.9.x Resolved Problems List (P-Notes)*

This collection of notes describes corrections that have been made to software problems. These notes are shipped with each Mark release and support release on the DOCUMENTS tape. The *x* in the title shown here represents the support release number (or a zero for notes issued with a Mark release). You can use the LISTNOTES program to print these notes.

## Hardware Updates

As part of an ongoing effort to further enhance reliability, availability, and supportability, Unisys, from time to time, initiates design changes to hardware equipment and associated firmware and microcode. Changes are provided to customer support engineers as Field Change Notices (FCNs). FCNs provide detailed instructions that enable customer support engineers to install hardware changes. To minimize disruption to the system, FCNs are usually incorporated into the system during preventive maintenance periods.

Microcode for A Series systems is supplied separately from other elements of the A Series system software.

If you have not been notified that your A Series system requires new microcode, you should continue operating the microcode level currently running in your machine.

# Section 12

## Documentation

A complete range of supporting documentation is available for A Series equipment and software products. Documentation can come in various forms, whether integrated into a software product as online help or printed and published as a manual. The following paragraphs review the types of documentation that are available. Two additional types of documentation, the system notes and the Release Notes, are introduced under "Progression Policy" in Section 11.

### CD-ROM Library

The *A Series System Software Documentation CD-ROM Library* provides an easy, quick, and accurate way to find information about Unisys products. The CD-ROM Library uses CD-ROM technology and advanced online search techniques to allow you to access any of the books within the A Series documentation library.

The CD-ROM Library runs under the Microsoft Windows environment, letting you have multiple applications open simultaneously, and allowing you to search, copy, or print Unisys product information without leaving an application.

All A Series manuals are combined into one database. The concept of a CD-ROM Library database is different from conventional, field-oriented and record-oriented databases. A CD-ROM Library database contains free-form text, images, and links (cross-references) to other parts of the database.

To use the CD-ROM Library, you need the following hardware and software:

- A Unisys Personal Workstation<sup>2</sup> (PW<sup>2</sup>) microcomputer or other fully compatible PC with a hard disk. A PC with a 386 microprocessor is strongly recommended.
- MS-DOS operating system, release 3.1 or later.
- The MS-DOS CD-ROM extensions to the MS-DOS operating system.
- Microsoft Windows software version 3.0 or later.
- An operational minimum of 640K conventional random-access memory (RAM).
- Two megabytes or more of extended memory.

**Note:** *Extended memory is not the same as expanded memory.*

- A video graphics array (VGA), enhanced graphics adapter (EGA), or multicolor graphics array (MCGA) monitor and graphics card. You can use either a color or monochrome display.
- A mouse device that is supported by the Windows environment. Although you can use the Windows environment without a mouse, a mouse is required for the CD-ROM Library.

- A CD-ROM drive that supports the ISO 9660 format for compact discs. Most current models of CD-ROM drives support this format.
- A printer that is supported by the Windows environment, if you want to print portions of the product information library.

For information about the CD-ROM documentation package, refer to the *A Series/V Series System Software Documentation CD-ROM Library Operations Guide* that is shipped with the package. To order the complete package, you must contact your Unisys representative. You must indicate the form numbers for both the CD-ROM and the guide.

## Online Documentation

Online documentation in the form of help text is available as part of the software for many screen-based and window-based A Series products. Online documentation is part of the A Series philosophy of software design, which holds that the product should adapt itself to the customer rather than vice versa.

While you are using the product, you can access help text to learn how to perform particular tasks, what to enter in the fields of a window or screen, and what the window or screen terms mean. In many cases, you can translate the help text into natural languages other than English, as described under “Multilingual System Interfaces” in Section 6.

## Printed Documentation

Printed documentation is available for all A Series hardware and software products. A Series printed documentation is prepared using a standard Unisys methodology that is based on modern ergonomics studies of documentation usage. Document design proceeds from audience profile studies and task analyses that help to isolate the information that is truly relevant to each user.

For major products, libraries of manuals are often prepared to provide each type of user with descriptions of how to perform the tasks relevant to his or her job. Convenient reference manuals are also published for the benefit of more experienced users.

You can familiarize yourself with the printed documentation that is available by reading the *A Series Documentation Library Overview*. This book lists all A Series printed manuals alphabetically by product, by subject, and by title. An explanation of the contents, scope, and audience for each manual is also given.

When you acquire a new A Series system, you are automatically provided with a core group of manuals that arrives with the system. These documents contain information needed to start the system and install the software. You receive one copy of each of the manuals associated with the software and hardware you ordered. The documents are shipped in the same boxes with the software and hardware.

If you want additional copies, or want copies of other documents, you can order them through Corporate Software and Publication Operations (CSPO).

Manuals are updated regularly to reflect changes to the software product that is documented. You can obtain these updates individually, or you can subscribe to a service that will automatically send you updates to all your manuals. When you order a new manual, you are automatically provided with the version that matches the software level you are running.

## Technical Information Documents

Technical information documents provide timely solutions to software problems and keep customers current on the status of software corrections and updates. They are issued as needed and serve as temporary documentation until formal changes to product documentation are made. The following are the types of technical information documents and their purposes:

- *Flashes* are used to communicate information about the availability of software corrections or procedural advice for resolving software product problems.
- *Technical Information Papers* are used to communicate technical information about installation or operational procedures.
- *Product support information manuals* inform users of the status of A Series system software. They include new software features and their availability dates, new publications, an index of software Flashes, and a listing of known software problems and qualified solutions. A product support information manual is available for each product family.





# Appendix A

## Inactive Software Products

A number of A Series software products are available that have a “support-only” status. These products are fully supported, and are available in Mark 3.9 versions. However, no further enhancements or new features are planned for these products. The following are some reasons why a software product might have support-only status:

- It is an older product with a well-developed feature content.
- A newer and more advanced product is available that serves a similar purpose.
- It depends on a hardware product that is no longer manufactured (such as the ET2000).

These products can still provide effective solutions for existing customers. However, in cases where a newer product is available that provides a similar function, new customers are encouraged to use the newer product. Most of the support-only products are not discussed in the main body of this manual. Table A-1 lists these support-only products. The table serves two purposes:

- If you are an existing user of one of these products, this table is here to assure you that the product is still supported on the Mark 3.9 system software release. The third column of the table lists possible alternative products that you might wish to investigate as part of your long-term planning.
- If you are a new A Series user, this table introduces you to some product names you might hear referred to at user group meetings, or that you might find as cross-references in some of our documents. To give you a general idea what people are talking about, the second column of the table indicates the basic function of the product.

**Table A-1. Support-Only Products**

Product	Function	Alternative
Activity Reporting System (BARS)	System performance monitoring	Capacity Management Facility.Snapshot (CMF.Snapshot)
APLB	Application development in A Programming Language (APL)	None
BASIC	Application development in BASIC	None

continued

## Inactive Software Products

Table A-1. Support-Only Products (cont.)

Product	Function	Alternative
BNA Version 1	Distributed processing in a multihost network	BNA Version 2
B 1000 to A Series progression tools	Migrating databases, data files, and programs from B 1000 to A Series systems	None
COBOL(68)	Application development in ANSI-68 COBOL	COBOL74, COBOL85
Diagnostic MCS	Data comm subsystem debugging	None
DMSII Inquiry	Querying and updating DMSII databases	Interactive Query Facility (IQF)
Extended Retrieval with Graphic Output (ERGO)	Querying and updating DMSII databases	IQF/Workstation Query Facility (WQF)
FORTRAN	Application development in ANSI-66 FORTRAN	FORTRAN77
Generalized Message Control System (GEMCOS)	Transaction management	Communications Management System (COMS)
KEYEDIO	Indexed sequential access to files	KEYEDIOII
On-Line Data Entry System (ODESY)	Simplified forms for data entry	None
Online Reporter III	Data analysis and reporting	None
PL/I	Application development in PL/I	None
Reporter III	Data analysis and reporting	None
Remote job entry (RJE)	Job submission from remote stations	BNA
Screen Design Facility (SDF)	User interface development for application programs	Screen Design Facility Plus (SDF Plus)
Security Support Library	Site-implemented security controls	InfoGuard security options
Transaction processing system (TPS)	Centralized transaction definitions	None

# Appendix B

## User Groups

User groups are a good source of specialized information concerning the use of computer systems. There are several groups available worldwide for A Series system users.

### CUBE

CUBE, Incorporated, is an independent association composed of Unisys customers in North America. The group operates as a clearinghouse for the exchange of information among customers and with Unisys. The type of information exchanged concerns use of Unisys equipment, software products, and services.

The group conducts its own activities with the assistance of a full-time executive secretary. The other officers are volunteers elected annually from the membership.

The principal activity of CUBE is conducting semiannual conferences for Unisys customers. These meetings are three to four days long and are usually held in April and October in major cities in North America. The conferences consist of more than 150 separate sessions for presentation of papers and for discussions on effective computer management and technical hardware and software topics. The speakers can be other customers, Unisys representatives, or outside experts.

In addition, there are displays featuring new product offerings from CUBE members and from Unisys Corporation. Members have the opportunity to meet and talk with other users with the same kind of Unisys equipment and from the same or a related line of business. Members benefit from presentations by representatives of Unisys Corporation.

CUBE membership is open to any customer or any organization that uses Unisys computer systems. To become a member, send your name, company name and mailing address, and the name of the Unisys equipment installed or purchased to

CUBE Incorporated  
Maple Park Office Plaza  
100 Maple Park Blvd.  
Suite 152  
St. Clair Shores, MI 48081  
(313) 771-9300  
FAX (313) 771-9598

After your request is received, your name is entered in the CUBE membership list. You receive all subsequent mailings about CUBE conferences and meeting reports. Your

## User Groups

---

membership remains active indefinitely, unless you fail to attend two consecutive CUBE conferences. You can renew your membership by writing to the CUBE secretary.

There are no annual dues or fee schedules. CUBE finances all its activities from the registration fees that each participant is charged at the conferences.

## International User Groups

There are many other user groups for international customers of Unisys. The Unisys Users Association (UUA) is the organization of European and African Unisys computer users. For information about joining UUA, write to

UUA Secretariat  
Woodside, Over Lane, Baslow, Bakewell  
Derbyshire, DE4 1RT  
UNITED KINGDOM

There are also a number of other U.S. and international user groups. You can contact your Unisys representative for information about joining the user group in your area.

# Glossary

In this glossary, definitions obtained from outside sources are preceded by an abbreviation enclosed in parentheses. Definitions from *Dictionary of Computing* are preceded by DOC. Definitions from *American National Dictionary for Information Processing Systems* are preceded by ANDIPS. Definitions from *Vocabulary for Data Processing, Telecommunications, and Office Systems* are preceded by VDP.

## A

### **accesscode**

An identification code, subordinate to a usercode, which sometimes has an associated password of its own. An accesscode can be specified in the USERDATAFILE as required along with a usercode/password combination. An accesscode further establishes a user's identity, controls security, and restricts access to disk files.

### **Activity Reporting System (BARS) utility**

A real-time utility program that monitors the performance of the system and displays performance statistics as numeric values and bar graphs. The items displayed are those concerning current central processing unit (CPU), memory, I/O, and disk pack use.

### **actual segment descriptor (ASD)**

A pointer to the location of a data or code item in memory or on a disk.

### **ADDS**

See Advanced Data Dictionary System.

### **Advanced Data Dictionary System (ADDS)**

A software product that allows for the centralized definition, storage, and retrieval of data descriptions.

### **ALGOL**

Algorithmic language. A structured, high-level programming language that provides the basis for the stack architecture of the Unisys A Series systems. ALGOL was the first block-structured language developed in the 1960s and served as a basis for such languages as Pascal and Ada. It is still used extensively on A Series systems, primarily for systems programming.

### **ANSI**

American National Standards Institute. A nongovernmental, nonprofit organization that is the central body for coordinating voluntary standards in the United States. ANSI also serves as the United States member of the International Standards Organization (ISO).

### **APL**

A Programming Language. A procedure-oriented language that can produce very short but powerful programs.

## Glossary

---

### **APLB**

A Programming Language B. A second-generation extended version of A Programming Language (APL).

### **application software**

Programs written to provide specific functions to solve specific problems for end users.

### **ASD**

See actual segment descriptor.

### **ASD memory**

The memory architecture used on A Series systems. In this memory architecture, memory is treated as a single continuous region that is indexed by the ASD table. Memory management is very flexible and is handled automatically by the operating system.

### **assembly language**

A programming language consisting of mnemonic symbols, most of which correspond on a one-to-one basis to machine language instructions. An assembly language program is translated to machine language through an assembler, which is an automatic translator program.

## **B**

### **BARS**

See Activity Reporting System (BARS) utility.

### **base**

A hardware unit in the I/O subsystem that contains the components necessary for handling message routing between the data link processors (DLPs) and the host I/O processor (IOP). The base itself contains one base control card (BCC), one maintenance card (MC), and at least one distribution card (DC). If there is more than one DC in the base, there is also one path select module (PSM). A base can contain up to seven DLPs.

### **Binder**

A program that enables separately compiled subprograms to be joined with a host object code file to produce a single object code file.

### **BNA**

The network architecture used on A Series, B 1000, and V Series systems as well as CP9500 and CP 2000 communications processors to connect multiple, independent, compatible computer systems into a network for distributed processing and resource sharing.

### **BTOS**

The operating system on the BTOS family of workstations.

### **BTOS family of workstations**

Any of the Unisys microcomputers that can run the BTOS operating system, including the B 21, B 22, B 26, B 27, B 28, and B 38 systems. Of these, the systems numbered B 26 and above can optionally run the MS-DOS operating system under BTOS.

**byte**

(1) (ANDIPS) A binary character string operated upon as a unit and usually shorter than a computer word. (2) On Unisys A Series systems, a measurable group of 8 consecutive bits having a single usage. In data communications, a byte is often referred to as a character or an octet.

**C****CANDE**

*See* Command and Edit.

**CATALOGING**

A system option that, when enabled with the OP (Options) system command, allows the system to keep track of copies of files that have been backed up onto a tape or a disk.

**central processing unit (CPU)**

The computer hardware unit that controls and executes the instructions contained in object code files.

**central processor subsystem**

All of the central processing units (CPUs) on a particular system. The total number of CPUs varies from one A Series model to another.

**COBOL**

Common Business-Oriented Language. A widely used, procedure-oriented language intended for use in solving problems in business data processing. The main characteristics of COBOL are the easy readability of programs and a considerable degree of machine independence. COBOL is the most widely used procedure-oriented language.

**COBOL74**

A version of the COBOL language that is compatible with the American National Standard X3.23-1974.

**Command and Edit (CANDE)**

A time-sharing message control system (MCS) that enables a user to create and edit files, and to develop, test, and execute programs, interactively.

**Communications Management System (COMS)**

A general message control system (MCS) that controls online environments on A Series systems. COMS can support the processing of multiprogram transactions, single-station remote files, and multistation remote files. *See also* COMS (Full-Featured) and COMS (Kernel).

**communications processor data link processor (CPDLP)**

A data link processor (DLP) that enables a host running BNA Version 2 software to communicate with local terminals or a wide area network (WAN). A CPDLP provides most of the functions of a CP 2000 communications processor but, unlike a CP 2000, can be connected directly to an I/O base in the host.

**communications processor local area network (CPLAN)**

In BNA Version 2, a cable and associated connections that can be used to hook the CP2000s at a site with the A Series hosts equipped with integrated communications processors (ICPs).

**compile**

To convert a program written in a source language, such as COBOL or ALGOL, to machine code that can be executed by a computer.

**compile time**

The time during which a compiler analyzes program text and generates an object code file.

**compiler**

A computer program that translates instructions written in a source language, such as COBOL or ALGOL, into machine-executable object code.

**COMS**

*See* Communications Management System.

**COMS (Full-Featured)**

A version of the Communications Management System (COMS) message control system (MCS) that provides full configuration capabilities through the COMS Utility. The COMS Utility enables the user to define and customize a COMS transaction processing environment, which provides additional features like transaction-based routing and database recovery. In addition, the user can track COMS statistics and use GEMCOS migration aids.

**COMS (Kernel)**

The transitional, temporary version of the Communications Management System (COMS) message control system (MCS). COMS creates a predefined configuration file that enables the user to use the window feature with the following three windows: a Menu-Assisted Resource Control (MARC) window with eight dialogues, a Command and Edit (CANDE) window with two dialogues, and a Generalized Message Control System (GEMCOS) window with one dialogue. Additionally, the user can communicate with remote-file programs.

**COMS Utility**

The Communications Management System (COMS) program that defines and maintains the specifications stored in the COMS configuration file.

**configuration**

(1) A description of selected hardware or software options or capabilities. (2) A set of hardware resources in an installed system. In particular, a configuration is a representation of the devices in the I/O subsystem and memory subsystem. (3) In the Interactive Datacomm Configurator (IDC), the logical representation of a data communications network.

**configuration file**

(1) A table that contains the configuration of a system. The configuration table is stored in the disk directory of the halt/load family. (2) For the SYSTEM/CONFIGURATOR utility, a file that lists and describes the hardware resources and selected software



information that make up a configuration. The configuration file can contain descriptions of several different hardware and software configurations for a system. (3) In X.25, a file that is constructed by the user and that contains static information for the specified data terminal equipment (DTE) addresses. (4) In the Communications Management System (COMS), a file that contains descriptions of the tables defined through the COMS Utility program. These tables contain information on message routing, security, dynamic program control, and synchronized recovery. This file is also referred to as the COMS CFILE.

**CP 2000**

*See* CP 2000 communications processor.

**CP 2000 communications processor**

A data communications processor (DCP) that provides communications interfaces to local area networks (LANs) and wide area networks (WANs), including BNA Version 2 and Transmission Control Protocol/Internet Protocol (TCP/IP) networks. The CP 2000 also provides connections to terminals controlled by BNA Version 2 software.

**CPDLP**

*See* communications processor data link processor.

**CPLAN**

*See* communications processor local area network.

**CPU**

*See* central processing unit.

**cross-reference (xref)**

A table that associates program identifiers with a list of sequence numbers of lines on which the identifiers appear. This information is most often generated by a compiler.

**cross-reference target (xref target)**

In the Editor, an identifier established as the item for which the cross-referencing commands are to display information.

**D****DASDL**

*See* Data and Structure Definition Language.

**database**

An integrated, centralized system of data files and program utilities designed to support an application. The data sets and associated index structures are defined by a single description. Ideally, all the permanent data pertinent to a particular application resides in a single database. The database is considered a global entity that several applications can access and update concurrently.

**Data and Structure Definition Language (DASDL)**

In Data Management System II (DMSII), the language used to describe a database logically and physically, and to specify criteria to ensure the integrity of data stored in

the database. DASDL is the source language that is input to the DASDL compiler, which creates or updates the database description file from the input.

### **data comm**

*See* data communications.

### **data comm mode**

(1) A mode of operation at an operator display terminal (ODT) in which interaction is controlled by the Communications Management System (COMS). The operator can use this mode to access COMS windows such as the Menu-Assisted Resource Control (MARC) interface or the Command and Edit (CANDE) message control system (MCS).  
(2) A method of operating at an operator display terminal (ODT) in which interaction is controlled by the Communications Management System (COMS). Data comm mode enables the operator to perform system functions without knowing the command syntax. An ODT in data comm mode has access to the Menu-Assisted Resource Control (MARC) interface, the Command and Edit (CANDE) message control system (MCS), and other available COMS windows, and can be used as a remote terminal.

### **data communications (data comm)**

The transfer of data between a data source and a data sink (two computers, or a computer and a terminal) by way of one or more data links, according to appropriate protocols.

### **Data Communications ALGOL (DCALGOL)**

A Unisys language based on ALGOL that contains extensions for writing message control system (MCS) programs and other specialized system programs.

### **data communications data link processor (DCDLP)**

A data communications processor (DCP) that combines the functions of a network support processor (NSP) and a line support processor (LSP) into one physical data link processor (DLP) and supports up to four lines of communication.

### **data descriptor**

An item in a stack that is used to point to a data area outside the stack, such as an array or a file.

### **data independence**

In data management, the property that establishes the ability to change the structural format of a database without requiring changes to or recompilation of unaffected application programs that use the database.

### **data link processor (DLP)**

A processor that serves as the system interface to a specific peripheral device, controller, or communications network.

### **Data Management ALGOL (DMALGOL)**

A Unisys language based on ALGOL that contains extensions for writing Data Management System II (DMSII) software and other specialized system programs.

### **Data Management System II (DMSII)**

A specialized system software package used to describe a database and maintain the relationships among the data elements in the database.

**Data Transfer System (DTS)**

A Unisys product that enables data transfer between workstations or personal computers (PCs) and Unisys mainframe computers. DTS also provides virtual disk and virtual printer capabilities.

**DCALGOL**

*See* Data Communications ALGOL.

**DCDLP**

*See* data communications data link processor.

**descriptor**

A computer word of a particular format that is used to reference data segments and code segments in memory or on a disk.

**device**

Any piece of I/O hardware, such as a data link processor (DLP) or a peripheral.

**directory name**

A name used to refer to a group of files whose file names are identical to the directory name, except that the file names have at least one additional node following the directory name.

**disk**

A random-access data storage device consisting of one or more circular platters that contain information recorded in concentric circular paths called tracks. Data on a disk are accessed by movable read/write heads. Some disks are removable.

**distributed systems service (DSS)**

One of a collection of services provided on Unisys hosts to support communications across multihost networks. DSSs can be services such as file handling, station transfer, and mail transfer.

**DLP**

*See* data link processor.

**DMALGOL**

*See* Data Management ALGOL.

**DMSII**

*See* Data Management System II.

**DSS**

*See* distributed systems service.

**DTS**

*See* Data Transfer System.

## E

### EMS

See Entry and Medium Systems (EMS).

### Entry and Medium Systems (EMS)

A designation referring to the Micro A and A 1 through A 10 systems.

### environmental software

Software products that provide productivity-enhancing environments for users. Examples are database environments such as the InfoExec family of products, applications development tools such as the Screen Design Facility Plus (SDF Plus), and participatory workstation environments such as the Workstation Query Facility (WQF).

### ERGO

See Extended Retrieval with Graphic Output.

### Extended Retrieval with Graphic Output (ERGO)

A program for Data Management System II (DMSII) database and file access, and for report generation.

## F

### family

(1) One or more disks logically grouped and treated as a single entity by the system. Each family has a name, and all disks in the family must have been entered into the family with the RC (Reconfigure Disk) system command. (2) The name of the disk or disk pack on which a physical file is located. (3) See *also* process family.

### file attribute

An element that describes a characteristic of a file and provides information the system needs to handle the file. Examples of file attributes are the file title, record size, number of areas, and date of creation. For disk files, permanent file attribute values are stored in the disk file header.

### File Transfer, Access, and Management (FTAM)

The standard developed by the International Standards Organization (ISO) for file exchange and management across an Open Systems Interconnection (OSI) network. FTAM systems can access file attributes (for example, password information) and the contents of files (including individual records, as well as entire files).

### firmware

The microcode stored in a device and used to control its operation.

### FORTRAN

Formula Translation. A high-level, structured programming language intended primarily for scientific use.

### FORTRAN77

A version of the FORTRAN language that is compatible with the ANSI X3.9-1978 standard.

**FTAM**

See File Transfer, Access, and Management (FTAM)

**G**

**GEMCOS**

See Generalized Message Control System.

**Generalized Message Control System (GEMCOS)**

A message control system (MCS) developed for online systems. GEMCOS is transaction oriented.

**H**

**halt/load**

A system-initialization procedure that temporarily halts the system and loads the master control program (MCP) from a disk to main memory.

**halt/load unit**

The disk pack that contains the currently operative master control program (MCP) object code file.

**HDU**

See host data unit.

**host data unit (HDU)**

The A 12 and A 15 system host interface to the I/O subsystem. An HDU is configured with up to three host-dependent ports (HDPs), each of which supports two message level interface (MLI) cables.

**Host Services**

A collection of services that are provided across multihost networks of A Series, B 1000 Series, CP9500, and V Series systems. The services can include file transfer, job transfer, station transfer, logical I/O, remote tasking, and remote command entry. *Contrast with* DSS Management and Network Services.

**HYPERchannel hardware**

The collection of hardware from Network Systems Corporation that facilitates the networking of computer systems from different manufacturers.

**I**

**I/O**

Input/output. An operation in which the system reads data from or writes data to a file on a peripheral device such as a disk drive.

**I/O processor (IOP)**

A specialized processor for moving data between system memory and the I/O subsystem.

## Glossary

---

### **I/O subsystem**

The hardware and software that manage all transfers of information between the operating system and peripheral devices.

### **ICP**

See integrated communications processor.

### **IDC**

See Interactive Datacomm Configurator.

### **IMG**

See Interactive Menugraph Generator.

### **indexed sequential-access method (ISAM)**

A method that provides efficient, flexible random access to records identified by keys stored in an index.

### **InfoExec**

Information Executive. The name of a family of Unisys products that define, maintain, retrieve, and update databases.

### **InfoGuard**

The Unisys security-enhancement software for A Series systems. InfoGuard provides such features as password management, selective logging and auditing, tape volume security, and simplified system-security configuration.

### **Information Executive (InfoExec)**

See InfoExec.

### **INFOVIEW II**

A window-oriented applications support environment for the BTOS family of workstations, the Unisys personal computer (PC) family of workstations, and workstations compatible with an IBM PC microcomputer. The INFOVIEW II product serves as an application building platform for distributed applications and enables concurrent access to host-based and local workstation applications.

### **initialization**

(1) The process of starting a program and giving starting values to variables. (2) A procedure that makes a system or subsystem available for its intended use. Important phases of initialization are the recognition of the physical environment, the identification of the available resources, and the establishment of the interface with the user. System initialization occurs as part of a halt/load.

### **integrated communications processor (ICP)**

In BNA Version 2, a data link processor (DLP) that handles a significant portion of communications overhead. The ICP also provides a connection to the communications processor local area network (CPLAN). At least one ICP for each A Series host is needed to connect the host to a BNA Version 2 network.

### **Interactive Datacomm Configurator (IDC)**

A Unisys interactive, menu-driven utility that enables the user to create, interrogate, and modify data communications network configurations.

**Interactive Menugraph Generator (IMG)**

A software tool for the design and modification of Menu-Assisted Resource Control (MARC), the Interactive Datacomm Configurator (IDC), and IMG screen menus and forms.

**Interactive Productivity (InterPro)**

*See InterPro.*

**Interactive Query Facility (IQF)**

A menu-driven member of the InfoExec family of products that is used interactively to generate reports from, and to browse and update, a Semantic Information Manager (SIM) database and files.

**International Standards Organization (ISO)**

A division of the United Nations under which the Consultative Committee on International Telegraphy and Telephony (CCITT) operates. The ISO was established in 1947 to promote the development of standards to facilitate international trade, and to develop mutual cooperation in areas of intellectual, scientific, technological, and economic activity.

**internationalization**

The software, firmware, and hardware features that enable users to customize products or to accommodate the needs of a specific language, culture, or line of business.

**InterPro**

Interactive Productivity. A family of Unisys software facilities used to create new products and enhance existing products.

**intersystem control (ISC)**

A direct hardware connection that enables data transfer between independent systems. The components that make up an ISC connection are a hub and its attached host control (HC) units. The HC unit type used to connect an I/O channel to a hub depends on the type of machine, specifically the I/O subsystem protocol.

**intrinsic**

A system-supplied program routine for common mathematical and other operations that is loaded onto the system separately. An intrinsic can be invoked by the operating system or user programs.

**IOP**

*See I/O processor.*

**IQF**

*See Interactive Query Facility.*

**ISAM**

*See indexed sequential-access method.*

**ISC**

*See intersystem control.*

## Glossary

---

### ISO

See International Standards Organization.

## J

### job

An independent process. The job of a particular task is the independent process that is the eldest ancestor of that task.

### job queue

A structure in the system software that stores a list of jobs that have been compiled and are waiting to be initiated.

### job summary

A file, produced after a job completes execution, that lists information such as the tasks initiated by the job, the beginning and ending times for each task, and the termination information for each task.

## L

### LAN

See local area network.

### Large Systems (LS)

Refers to systems that interface with a host data unit (HDU) or a resource management module (RMM).

### library

(1) A collection of one or more named routines or library objects that are stored in a file and can be accessed by other programs. (2) A program that exports objects for use by user programs. (3) (VDP) A collection of related files.

### library maintenance

A master control program (MCP) procedure that copies disk files to and from disk, tape, and compact disk (CD). Library maintenance is invoked by the Work Flow Language (WFL) statements *ADD*, *COPY*, and *ARCHIVE*.

### LINC II

See Logic and Information Network Compiler II.

### line adapter

The device that performs physical line control, including byte assembly and disassembly, data buffering, data synchronization, and modem and autocal equipment control.

### line support processor (LSP)

The data communications subsystem processor that manages communication with the host and initiates processes that control the input of messages to and the output of messages from data communications lines.



**local area network (LAN)**

A network that enables high-speed communications among various devices within a relatively small area.

**Logic and Information Network Compiler II (LINC II)**

A fourth-generation software development tool that enables programmers to create application programs at less cost and up to 10 times faster than by conventional programming methods.

**LS**

*See* Large Systems (LS).

**LSP**

*See* line support processor.

**M**

**maintenance mode**

A mode of operation at a system control terminal (SCT) that enables interaction with the maintenance subsystem. The operator can use this mode to initiate hardware diagnostics routines or to reinitialize the system.

**maintenance processor**

The main hardware component of the maintenance subsystem. Depending on the A Series system model you are using, the maintenance processor may be referred to by any of several names, including the following: the user interface processor (UIP), the system control processor (SCP), or the system maintenance processor (SMP).

**maintenance subsystem**

The software and hardware that serve as the interface between the user and the hardware. This software can initialize or configure the hardware, and halt or initialize system software. The subsystem is also used in diagnosing hardware and software problems.

**MARC**

*See* Menu-Assisted Resource Control.

**master control program (MCP)**

The central program of the A Series operating system. The term applies to any master control program that Unisys may release for A Series systems.

**Master Control Program/Advanced Systems (MCP/AS)**

The version of the master control program (MCP) that supports the Actual Segment Descriptor (ASD) memory architecture. As of the Mark 3.9 system software release, MCP/AS is the only master control program available on A Series.

**MCP**

*See* master control program.

**MCP/AS**

*See* Master Control Program/Advanced Systems.

## Glossary

---

### **MCS**

*See* message control system.

### **media**

The plural of medium.

### **medium**

A device used to store data. Examples are a disk pack or a magnetic tape (MT).

### **memory**

A temporary storage area where data and programs are placed while they are being processed.

### **memory subsystem**

One or more memory control units and one or more memory storage units, plus a memory interface that can be part of the control unit. The memory subsystem handles all transfers of data between main memory and the main processor.

### **Menu-Assisted Resource Control (MARC)**

A menu-driven interface to A Series systems that also enables direct entry of commands.

### **message control system (MCS)**

(1) A program that controls the flow of messages between terminals, application programs, and the operating system. MCS functions can include message routing, access control, audit and recovery, system management, and message formatting. (2) In X.25, a program that acts as an interface between the application and the Network Definition Language II (NDLII) and handles such functions as routing, security, auditing, and changes in the network.

### **message level interface (MLI)**

The interface between the host system, the I/O subsystem, and the data communications subsystem.

### **message level interface processor (MLIP)**

*See* I/O processor, Entry and Medium Systems (EMS).

### **microcode**

A sequence of elementary instructions that corresponds to a specific computer operator. For example, a sequence of microcode is executed for each A Series system operator.

### **Mirrored Disk**

A software feature on Unisys A Series systems that enables from two to four disks to be maintained as a mirrored set; that is, as exact copies of each other. The Mirrored Disk feature increases system availability and data integrity, decreases the possibility of data loss due to equipment or media malfunction, and can improve I/O throughput on disk subsystems that experience a high ratio of reads versus writes.

### **MLI**

*See* message level interface.

**MLIP**

An acronym for the obsolete term message level interface processor; *see* I/O processor, Entry and Medium Systems (EMS).

**MLS**

*See* MultiLingual System.

**modem**

A device placed between a computer system and a telephone line to permit transmission of digital pulses. Modems permit computers to communicate with other computers, terminals, and printers over communication lines. The term *modem* is derived from *modulator-demodulator*.

**MS-DOS**

Microsoft Disk Operating System. The single-tasking operating system for the IBM PC and compatible microcomputers. The MS-DOS operating system is repackaged and sold as the PC-DOS™ operating system by International Business Machines Corporation. The two operating systems are nearly identical.

**MultiLingual system (MLS)**

An environment that can process information using the standards and functional requirements of different localities, cultures, or lines of business. The processing of information depends on the conversion, languages, and conventions that are defined for the system. For example, output messages, online help text, menus, and screens can be developed and accessed in different natural languages, such as English, French, or Japanese.

**multiprocessing**

A state in which two or more processors in the same system run under the control of a single operating system.

**multiprogramming**

(1) A technique for handling multiple routines or programs concurrently by overlapping or interleaving their execution, enabling more than one program to timeshare machine components. (2) The ability of a single computer system to execute many processes concurrently.

**N****NAU**

*See* Network Administrative Utility.

**NCF**

*See* network control facility.

**NDLII**

*See* Network Definition Language II.

---

PC-DOS is a trademark of International Business Machines Corporation.

### **Network Administrative Utility (NAU)**

In BNA Version 2, a host software component that enables the configuration of the network, as a whole, to be specified interactively. The NAU generates a network configuration and status (NCS) database and network initialization files for individual BNA Version 2 nodes.

### **network control facility (NCF)**

A comprehensive software system for controlling and monitoring BNA Version 2 networks.

### **Network Definition Language II (NDLII)**

The Unisys language used to describe the physical, logical, and functional characteristics of the data communications subsystem to network support processors (NSPs), line support processors (LSPs), and data communications data link processors (DCDLPs).

### **network support processor (NSP)**

A data communications subsystem processor that controls the interface between a host system and the data communications peripherals. The NSP executes the code generated by the Network Definition Language II (NDLII) compiler for line control and editor procedures. An NSP can also control line support processors (LSPs).

### **NEWP**

A structured, high-level programming language used in developing some Unisys system software. Based on the ALGOL language, NEWP contains facilities necessary for the operating system to interact with A Series hardware.

### **NSP**

*See* network support processor.

## **O**

### **object code**

The instructions in machine code that are created as a result of compiling source code.

### **object code file**

A file produced by a compiler when a program is compiled successfully. The file contains instructions in machine-executable object code.

### **OCM**

*See* Operations Control Manager.

### **ODT**

*See* operator display terminal.

### **ODT mode**

- (1) A mode of operation at a system control terminal (SCT) that provides all the capabilities that would normally be available at an operator display terminal (ODT).
- (2) The method of operating at a system control terminal (SCT) that enables interaction with the CONTROLLER by using system control commands, otherwise known as system commands. ODT mode requires the operator to know the syntax of system commands.

**ODTDLP**

See operator display terminal data link processor.

**Operations Control Manager (OCM)**

A menu-driven member of the InfoExec family of products that is used to maintain databases.

**operator display terminal (ODT)**

(1) A terminal or other device that is connected to the system in such a way that it can communicate directly with the operating system. The ODT allows operations personnel to accomplish system operations functions through either of two operating modes: system command mode or data comm mode. (2) The name given to the system control terminal (SCT) when it is used as an ODT.

**operator display terminal data link processor (ODTDLP)**

A processor that serves as the controller of an operator display terminal (ODT) and provides the I/O interface between the system and the ODT.

**P****Pascal**

A high-level programming language developed by Niklaus Wirth, based on the block structuring nature of ALGOL 60 and the data structuring innovations of C.A.R. Hoare. Pascal is a general-purpose language.

**password**

A character string associated with a usercode or accesscode in the USERDATAFILE, and used to identify legitimate users of the system. When logging on to a message control system (MCS), a user must supply a usercode and a password.

**PC**

See personal computer.

**PDN**

an acronym for the obsolete term public data network; *see* packet-switched data network.

**peripheral**

A device used for input, output, or file storage. Examples are magnetic tape drives, disk drives, printers, or operator display terminals (ODTs).

**personal computer (PC)**

A microcomputer that is compatible with the Unisys Personal Workstation<sup>2</sup>, IBM PC, PC/XT, Personal Computer AT, or Personal System/2 microcomputer.

**Personal Workstation<sup>2</sup>**

A Unisys line of microcomputers that is software compatible with the IBM Personal System/2 line of computers.

## Glossary

---

### PL/I

Programming Language I. A high-level, structured programming language designed primarily for scientific and commercial use.

### primitive system command

Any system command that begins with two question marks (??). Primitive system commands bypass the system command handler and can be helpful when the system does not respond to nonprimitive system commands.

### Print System (PrintS)

A Unisys software product used to control when, where, and how printer backup files are printed on A Series systems.

### PrintS

See Print System.

### priority

(1) A characteristic associated with a process that determines its precedence in the use of system resources. A process with higher priority executes more quickly than it would if it had lower priority. (2) In X.25, the sequence in which various entries and tasks are processed; the sequence is determined by the analysis of action codes and other priority-real-time systems.

### private file

A file with a SECURITYTYPE attribute specified as PRIVATE. Only users logged on to a privileged usercode or to the usercode under which the file is stored can access a private file.

### process

(1) The execution of a program or of a procedure that was initiated. The process has its own process stack and process information block (PIB). It also has a code segment dictionary, which can be shared with other processes that are executions of the same program or procedure. (2) A software application; that is, any activity or systematic sequence of operations that produces a specified result. (3) In the Advanced Data Dictionary System (ADDS), a structure that models a logical view of relationships between different parts of a system.

### process family

A group of processes consisting of a single job and any tasks that are descendants of that job.

### process stack

A memory structure that stores information about the current state of the execution of a procedure. The process stack includes activation records for all blocks that the process has entered and not yet exited.

### program

(1) A specification of the sequence of computational steps that solve a computational problem. The steps are written (coded) in a particular programming language. (2) An object code file.

**Program Agent**

A program that is capable of issuing BNA commands and receiving responses and reports through a subport.

**protocol**

A formal set of rules governing the format, timing, sequencing, and error control of exchanged messages on a data network.

**public file**

A file with a SECURITYTYPE attribute specified as PUBLIC. Users who are logged on to any usercode can access a public file by specifying the (<usercode>)<file name> form for the file title.

**PW<sup>2</sup>**

See Personal Workstation<sup>2</sup>.

**R****RDB**

See Remote Database Backup.

**record**

(1) A group of logically related items of data in a file that are treated as a unit. (2) The data read from or written to a file in one execution of a read or write statement in a program. (3) In Reporter III, a collection of related data items, groups, or both. The term *record* is used to mean a record, list element, or member, depending on the term that is appropriate for the type of data structure under consideration.

**Remote Database Backup (RDB)**

A disaster recovery facility that provides a means of describing and maintaining a system of duplicate databases on multiple remote A Series systems. The primary database is update capable, and the remote database, also known as the secondary database, is inquiry capable.

**remote device**

An I/O unit or other piece of equipment that is physically removed from the computer but connected by a communications line.

**remote file**

A file with the KIND attribute specified as REMOTE. A remote file enables object programs to communicate interactively with a terminal.

**remote ODT**

A data comm station that provides the functions of an operator display terminal (ODT) running in system command mode, except that the remote ODT does not accept primitive system commands. An operator can use the REMOTESPO (Activate REMOTESPO) system command to activate a data comm station as a remote ODT.

**Remote Print System (ReprintS)**

A Unisys software system that controls the routing and printing of backup files at remote (data comm) destinations and on BNA networks.

## Glossary

---

### **remote support mode**

On A 1 through A 6 systems, a mode of operation at a system control terminal (SCT). In this mode, the operator can enable the remote support interface, which allows Unisys support personnel to perform hardware diagnostics tests and system monitoring through a phone modem.

### **ReprintS**

*See Remote Print System.*

### **RPG**

Report Program Generator. A high-level, commercially oriented programming language used most frequently to produce reports based on information derived from data files.

### **run time**

The time during which an object code file or user interface system (UIS) is executed.

## **S**

### **Screen Design Facility Plus (SDF Plus)**

A Unisys product used for creating user interface systems (UISs) for online, transaction-based application systems.

### **SCT**

*See system control terminal.*

### **SDF Plus**

*See Screen Design Facility Plus.*

### **segment**

(1) A contiguous region of memory that is referred to by a descriptor and stores code or data for use by a process. (2) On microcomputers using the INTEL® 8088, 8086, and 80X86 line of microprocessors, the first part of a two-part memory address. The segment is a 16-bit number that is first multiplied by decimal 16 (hex 10) and then added to another 16-bit number, the offset, to obtain a 20-bit memory address.

### **Semantic Information Manager (SIM)**

(1) The basis of the InfoExec environment. SIM is a database management system used to describe and maintain associations among data by means of subclass-superclass relationships and linking attributes. (2) In the Data Transfer System (DTS), a Unisys software product (SIM kernel) that DTS uses to extract data from host files.

### **SIM**

*See Semantic Information Manager.*

### **SMFII**

*See System Management Facility II.*



**SNA**

*See* Systems Network Architecture.

**source code**

An instruction or a set of instructions written in a programming language. Source code must be translated (compiled) to object code before the program can be executed.

**SSF**

*See* system software facility.

**system command mode**

A mode of operation at an operator display terminal (ODT) in which operations personnel can enter system commands and view automatic display mode (ADM) displays.

**system control terminal (SCT)**

(1) A terminal or other device that is connected to the system in such a way that it can communicate directly with the maintenance processor. An SCT can operate in maintenance mode or in operator display terminal (ODT) mode. On some systems, the SCT also provides a remote support mode. (2) A terminal used to enter information. An SCT can be used three ways: as an operator display terminal (ODT) to interface with the operating system, as a maintenance display terminal (MDT) to interface with the maintenance subsystem, or as a remote display terminal (RDT) to interface with remote support. The windows providing these uses are available once the automatic initialization sequence has finished.

**system log file**

A file that contains a record of events for a particular system.

**System Management Facility II (SMFII)**

A software system that monitors and provides data on four aspects of system performance: hardware, software, workload characterization, and system utilization.

**system software**

The master control program (MCP) and all other object code files necessary for system operation.

**system software facility (SSF)**

The system software programs essential for system operation.

**Systems Network Architecture (SNA)**

(VDP) In IBM computer systems, the description of the logical structure, formats, protocols, and operational sequences for transmitting information units through networks and controlling the configuration of networks.

**T**

**TADS**

*See* Test and Debug System.

**tape drive**

An I/O peripheral device that stores data on reels or cartridges of magnetic tape (MT).

## Glossary

---

### **task**

(1) A dependent process. (2) Any process, whether dependent or independent.

### **TCP/IP**

See Transmission Control Protocol/Internet Protocol.

### **Test and Debug System (TADS)**

A Unisys interactive tool for testing and debugging programs and libraries. TADS enables the programmer to monitor and control the execution of the software under testing and examine the data at any given point during program execution.

### **track**

A circular path on the surface of a disk used to store data.

### **transaction**

(1) The transfer of one message from a terminal or host program to a receiving host program, the processing carried out by the receiving host program, and the return of an answer to the sender. (2) In data management, a sequence of operations grouped by a user program because the operations constitute a single logical change to the database. (3) In the Screen Design Facility Plus (SDF Plus), the structure that performs the transfer of the message.

### **Transmission Control Protocol/Internet Protocol (TCP/IP)**

(1) A family of protocols that were originally developed for use in a Department of Defense network, and which have been widely adopted as standard protocols for multivendor networks. The applications protocols typically supported by TCP/IP are File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), and Telnet.

(2) A Unisys A Series product that provides telecommunications capabilities between A Series, 1100 Series, 5000 Series, and other computing systems. The A Series TCP/IP product is designed to interoperate with other TCP/IP systems in a multivendor network using industry and military standard protocols. The A Series TCP/IP product provides file transfer, station transfer, and (with the Mail System) mail transfer.

## **U**

### **UCF**

See User Communication Form.

### **UIP**

See user interface processor.

### **unit**

(1) A peripheral device such as a disk drive or a tape drive. (2) In COBOL, a module of mass storage, the dimensions of which are determined by each implementor.

### **User Communication Form (UCF)**

A form used by Unisys customers to report problems and express comments about Unisys products to support organizations.

**user interface processor (UIP)**

A processor serving two functions: it can be a system control subsystem (SCS) that performs a maintenance subsystem self-test before loading and transferring control to the maintenance subsystem software; or it can be a controller for the operator display terminal (ODT) when the ODT is in ODT mode and is communicating with the operating system.

**usercode**

An identification code used to establish user identity and control security, and to provide for segregation of files. Usercodes can be applied to every task, job, session, and file on the system. A valid usercode is identified by an entry in the USERDATAFILE.

**utility**

A system software program that performs commonly used functions.

**V****virtual memory**

A technique that treats disk storage space as an extension of main memory, giving the appearance of a larger main memory than actually exists.

**volume**

The medium of a mass storage device such as a disk, disk pack, or tape reel. The term *volume* is not restricted to the volume library on a cataloging system or the volume directory on a system with tape volume security. For example, on the BTOS family of workstations, the hard disk is a volume, and each floppy disk is a volume. When a volume is initialized, it is assigned a volume name and an optional password.

**W****WAN**

See wide area network.

**WFL**

See Work Flow Language.

**WFL job**

(1) A Work Flow Language (WFL) program, or the execution of such a program. (2) A collection of Work Flow Language (WFL) statements that enable the user to run programs or tasks.

**wide area network (WAN)**

A network that enables communications among various devices spread over a large area (for example, devices located in different cities).

**window**

(1) In the Communications Management System (COMS) architecture, the concept that enables a number of program environments to be operated independently and simultaneously at one station. One of the program environments can be viewed while the others continue to operate. (2) In the Editor, a conceptual aperture through which

any screen-sized group of lines in a work file can be viewed. The window is said to move forward in the file if lines of higher sequence numbers are being displayed, and backward if lines of lower sequence numbers are being displayed. (3) In the Forms Manager, a rectangular area in a form. (4) In data communications, a flow control mechanism, the size of which is equal to the number of frames, packets, or messages that can be sent from a transmitter to a receiver before any reverse acknowledgment is required. At the data terminal equipment (DTE)/data circuit terminating equipment (DCE) interface of a logical channel, a window is the maximum number of consecutive packets that can be transmitted. (5) A portion of a screen that has been allocated to display the contents of a specified area of memory.

### **window dialogue**

In the Communications Management System (COMS), a particular access to a program environment through a COMS window at a station. The exact number of window dialogues allowed at one time for a given window depends on the constraints established through the COMS Utility. No more than eight window dialogues are allowed at one time for any window. Each dialogue has an identifying number within its window.

### **word**

(1) A unit of computer memory. On A Series systems, a word consists of 48 bits used for storage plus tag bits used to indicate how the word is interpreted. (2) In COBOL, a character-string of not more than 30 characters that forms a user-defined word, a system-name, or a reserved word. (3) In Reporter III, a combination of not more than 30 characters that can consist of the alphabetic characters A through Z, the numeric characters 0 through 9, and the hyphen (-). A word must contain at least one alphabetic character and cannot begin or end with a hyphen.

### **Work Flow Language (WFL)**

A Unisys language used for constructing jobs that compile or run programs on A Series systems. WFL includes variables, expressions, and flow-of-control statements that offer the programmer a wide range of capabilities with regard to task control.

### **Workstation Query Facility (WQF)**

A menu-driven and window-driven member of the InfoExec family of products. WQF is used at a workstation to interactively generate reports from and to browse and update a Semantic Information Manager (SIM) database, and to view portions of its schema.

### **WQF**

*See* Workstation Query Facility.

## **X**

### **X.25**

A recommendation by the Consultative Committee on International Telegraphy and Telephony (CCITT) that specifies the interface between user data terminal equipment (DTE) and packet switching data circuit terminating equipment (DCE).

# Bibliography

- A 5 System Operations Guide* (form 2036059). Unisys Corporation.
- A 12 System Operating Guide* (form 3950 8882). Unisys Corporation.
- A 15 System Operating Guide* (form 1203718). Unisys Corporation.
- A 16 System Operations and Programming Reference Manual* (form 3950 8833). Unisys Corporation.
- A 17 System Console Operations Guide* (form 1201258). Unisys Corporation.
- A Series A 1–A 6 Systems Software Installation Guide* (form 8600 1468). Unisys Corporation. Formerly *A Series A 1/A 4 Software Installation Guide*, the *A Series System Software Installation Guide, Volume 1: A 2/A 3*, the *A Series System Software Installation Guide, Volume 1: A 3 Advanced*, the *A Series System Software Installation Guide, Volume 1: A 3 Dual-Processor*, the *A Series A 4 Model FX Software Installation Guide*, and the *A Series A 5/A 6 Software Installation Guide*.
- A Series ALGOL Programming Reference Manual, Volume 1: Basic Implementation* (form 8600 0098) Unisys Corporation.
- A Series ALGOL Programming Reference Manual, Volume 2: Product Interfaces* (form 8600 0734). Unisys Corporation.
- A Series ALGOL Test and Debug System (TADS) Programming Guide* (form 1169539). Unisys Corporation.
- A Series APLB Programming Reference Manual* (form 1203643). Unisys Corporation.
- A Series BASIC Programming Reference Manual* (form 1203650). Unisys Corporation.
- A Series Binder Programming Reference Manual* (form 8600 0304). Unisys Corporation.
- A Series BNA Version 1 Operations Guide* (form 8600 0783). Unisys Corporation.
- A Series BNA Version 1 Program Agent Programming Guide* (form 1169653). Unisys Corporation.
- A Series BNA Version 2 Network Software Implementation Planning Guide* (form 3787 5408). Unisys Corporation.
- A Series BNA Version 2 Operations Guide* (form 1222720). Unisys Corporation.

## Bibliography

---

- A Series BNA Version 2 User Program Agent Programming Guide* (form 1222723). Unisys Corporation.
- A Series Burroughs Network Architecture (BNA) Version 2 Installation Guide* (form 1169778). Unisys Corporation. (Discontinued)
- A Series Burroughs Network Architecture (BNA) Version 2 Operations Reference Handbook* (form 1182367). Unisys Corporation.
- A Series C Programming Reference Manual* (form 3950 8775). Unisys Corporation.
- A Series CANDE Configuration Reference Manual* (form 8600 1344). Unisys Corporation.
- A Series CANDE Operations Reference Manual* (form 8600 1500). Unisys Corporation.
- A Series Capacity Management Facility.Disk Operations Guide* (form 1221358). Unisys Corporation.
- A Series Capacity Management Facility.Disk Operations Training Guide* (form 1221050). Unisys Corporation.
- A Series COBOL ANSI-68 Programming Reference Manual* (form 8600 0320). Unisys Corporation.
- A Series COBOL ANSI-74 Programming Reference Manual, Volume 1: Basic Implementation* (form 8600 0296). Unisys Corporation.
- A Series COBOL ANSI-74 Programming Reference Manual, Volume 2: Product Interfaces* (form 8600 0130). Unisys Corporation.
- A Series COBOL ANSI-74 Test and Debug System (TADS) Programming Guide* (form 1169901). Unisys Corporation.
- A Series COBOL ANSI-85 Programming Reference Manual, Volume 1: Basic Implementation* (form 8600 1518). Unisys Corporation.
- A Series COBOL ANSI-85 Programming Reference Manual, Volume 2: Product Interfaces* (form 8600 1526). Unisys Corporation.
- A Series Communications Management System (COMS) Capabilities Manual* (form 8600 0627). Unisys Corporation.
- A Series Communications Management System (COMS) Configuration Guide* (form 8600 0312). Unisys Corporation.
- A Series Communications Management System (COMS) Documents Index* (form 8600 0635). Unisys Corporation.
- A Series Communications Management System (COMS) Migration Guide* (form 8600 1567). Unisys Corporation.

- A Series Communications Management System (COMS) Operations Guide* (form 8600 0833). Unisys Corporation.
- A Series Communications Management System (COMS) Programming Guide* (form 8600 0650). Unisys Corporation.
- A Series CP 2000 BNA Version 2 Implementation Guide* (form 1222727). Unisys Corporation.
- A Series CP 2000 SNA PUT5 Implementation Guide* (form 1225440). Unisys Corporation.
- A Series CP 2000 SNA PUT5 Operations Guide* (form 1225441). Unisys Corporation.
- A Series CP 2000/CPDLP SNA PUT2 Implementation Guide* (form 1225443). Unisys Corporation.
- A Series CP 2000/CPDLP SNA PUT2 Operations Guide* (form 1225444). Unisys Corporation.
- A Series CPDLP BNA Version 2 Implementation Guide* (form 1222731). Unisys Corporation.
- A Series Data Base Data Transfer (DBT) Utility User's Guide* (form 1180585). Unisys Corporation.
- A Series Data Comm Processor (DCP) to Network Support Processor (NSP) Configuration Conversion Operations Guide* (form 1182193). Unisys Corporation.
- A Series Data Communications Protocols Installation and Implementation Guide* (form 8600 0486). Unisys Corporation.
- A Series Data Management Functional Overview* (form 8600 0239). Unisys Corporation.
- A Series DCALGOL Programming Reference Manual* (form 8600 0841). Unisys Corporation.
- A Series DiagnosticMCS Reference Manual* (form 1169596). Unisys Corporation.
- A Series Disk Cache Module Installation and Operations Guide* (form 3950 8874). Unisys Corporation.
- A Series Disk Subsystem Administration and Operations Guide* (form 8600 0668). Unisys Corporation.
- A Series Distributed Systems Service (DSS) Operations Guide* (form 8600 0122). Unisys Corporation.
- A Series DMALGOL Programming Reference Manual* (form 8600 0874). Unisys Corporation.

## Bibliography

---

- A Series DMSII Application Program Interfaces Programming Guide* (form 5044225). Unisys Corporation. Formerly *A Series DMSII User Language Interface Programming Guide*.
- A Series DMSII Data and Structure Definition Language (DASDL) Programming Reference Manual* (form 8600 0213). Unisys Corporation.
- A Series DMSII Documents Index* (form 8600 1443). Unisys Corporation.
- A Series DMSII Interpretive Interface Programming Reference Manual* (form 8600 0155). Unisys Corporation.
- A Series DMSII Technical Overview* (form 5044191). Unisys Corporation.
- A Series DMSII Transaction Processing System (TPS) Programming Guide* (form 1164043). Unisys Corporation.
- A Series DMSII Utilities Operations Guide* (form 8600 0759). Unisys Corporation.
- A Series Documentation Library Overview* (form 8600 0361). Unisys Corporation.
- A Series Editor Operations Guide* (form 8600 0551). Unisys Corporation.
- A Series Extended Retrieval with Graphic Output (ERGO) Operations Guide* (form 8600 0205). Unisys Corporation.
- A Series File Attributes Programming Reference Manual* (form 8600 0064). Unisys Corporation. Formerly *A Series I/O Subsystem Programming Reference Manual*.
- A Series FORTRAN Programming Reference Manual* (form 1222691). Unisys Corporation.
- A Series FORTRAN77 Programming Reference Manual* (form 3950 8759). Unisys Corporation.
- A Series FORTRAN77 Test and Debug System (TADS) Programming Guide* (form 1222667). Unisys Corporation.
- A Series Generalized Message Control System (GEMCOS) Format Generator Operations Guide* (form 1185139). Unisys Corporation.
- A Series Generalized Message Control System (GEMCOS) Operations Guide* (form 8600 0676). Unisys Corporation.
- A Series GETSTATUS/SETSTATUS Programming Reference Manual* (form 8600 0346). Unisys Corporation.
- A Series Help Utility Operations Guide* (form 8600 0510). Unisys Corporation.
- A Series I/O Subsystem Programming Guide* (form 8600 0056). Unisys Corporation. Formerly *A Series I/O Subsystem Programming Reference Manual*.



- A Series InfoExec Advanced Data Dictionary System (ADDS) Operations Guide* (form 8600 0197). Unisys Corporation.
- A Series InfoExec Capabilities Manual* (form 8600 0254). Unisys Corporation.
- A Series InfoExec DMS.View Operations Guide* (form 8600 0775). Unisys Corporation.
- A Series InfoExec Documents Index* (form 8600 0791). Unisys Corporation.
- A Series InfoExec Interactive Query Facility (IQF) Operations Guide* (form 8600 0767). Unisys Corporation.
- A Series InfoExec LINC.View Operations Guide* (form 8600 0858). Unisys Corporation.
- A Series InfoExec Semantic Information Manager (SIM) Object Definition Language (ODL) Programming Guide* (form 8600 0189). Unisys Corporation.
- A Series InfoExec Semantic Information Manager (SIM) Object Manipulation Language (OML) Programming Guide* (form 8600 0163). Unisys Corporation.
- A Series InfoExec Semantic Information Manager (SIM) Programming Guide* (form 1195104). Unisys Corporation.
- A Series InfoExec Semantic Information Manager (SIM) Technical Overview* (form 8600 1674). Unisys Corporation.
- A Series InfoExec Structured Query Language Database (SQLDB) Programming Guide* (form 8600 0049). Unisys Corporation.
- A Series Integrated Communications Processor (ICP) BNA Version 2 Implementation Guide* (form 1225370). Unisys Corporation.
- A Series Integrated Communications Processor (ICP1) Memory Dump Analyzer Operations Reference Manual* (form 1182383). Unisys Corporation.
- A Series Intelligent Distributed Editor (IDE) Operations Guide* (form 1210960). Unisys Corporation.
- A Series Interactive Datacomm Configurator (IDC) Operations Guide* (form 1169810). Unisys Corporation.
- A Series Interactive Menugraph Generator (IMG) Operations Guide* (form 8600 0411). Unisys Corporation.
- A Series KEYEDIOII Programming Reference Manual* (form 8600 0684). Unisys Corporation.
- A Series LINC II Installation & Configuration Guide* (form 3943 4469). Unisys Corporation.

## Bibliography

---

- A Series Mail System Installation and Administration Guide* (form 8600 1559). Unisys Corporation.
- A Series Mail System Operations Guide* (form 8600 0247). Unisys Corporation.
- A Series Mark 3.9 Software Release Capabilities Overview* (form 8600 0015). Unisys Corporation.
- A Series Memory Subsystem Administration and Operations Guide* (form 1169836). Unisys Corporation.
- A Series Menu-Assisted Resource Control (MARC) Operations Guide* (form 8600 0403). Unisys Corporation.
- A Series Message Handling System (OSI-MHS) Capabilities Overview* (form 3987 7147). Unisys Corporation.
- A Series Message Handling System (OSI-MHS) Installation and Administration Guide* (form 3987 7162). Unisys Corporation.
- A Series Message Handling System (OSI-MHS) Operations Guide* (form 3987 7154). Unisys Corporation.
- A Series Message Handling System (OSI-MHS) Programming Reference Manual* (form 3987 7154). Unisys Corporation.
- A Series Message Translation Utility (MSGTRANS) Operations Guide* (form 8600 0106). Unisys Corporation. Formerly *A Series Message Translation Utility Operations Guide*.
- A Series Micro A A Series Memory Expansion Installation Instructions* (form 8600 1088). Unisys Corporation.
- A Series Micro A Data Communications Host Adapter 1 Board Installation Instructions* (form 8600 1096). Unisys Corporation.
- A Series Micro A Data Communications Host Adapter 2 Board Installation Instructions* (form 8600 1351). Unisys Corporation.
- A Series Micro A Ergonomic Video Display (EVD) Installation Instructions* (form 8600 1104). Unisys Corporation.
- A Series Micro A Full-Height Universal Disk Drive Expansion Cabinet Installation Instructions* (form 8600 1716). Unisys Corporation.
- A Series Micro A MAICP1 Board Installation Instructions* (form 8600 1112). Unisys Corporation.
- A Series Micro A MA150T Tape Drive Installation Instructions* (form 8600 1138). Unisys Corporation.

- A Series Micro A MA280D Disk Drive Installation Instructions* (form 8600 1146).  
Unisys Corporation.
- A Series Micro A MA280HD Disk Drive Installation Instructions* (form 8600 1377).  
Unisys Corporation.
- A Series Micro A MA560D Disk Drive Installation Instructions* (form 8600 1369).  
Unisys Corporation.
- A Series Micro A Mouse Input Device Installation Instructions* (form 8600 1120).  
Unisys Corporation.
- A Series Micro A OS/2 Version 1.1 Upgrade Kit Installation Instructions*  
(form 8600 1070). Unisys Corporation.
- A Series Micro A SCSI Converter Cabinet Installation Instructions* (form 8600 1385).  
Unisys Corporation.
- A Series Micro A SCSI Host Adapter Board Installation Instructions* (form 8600 1062).  
Unisys Corporation.
- A Series Micro A System Installation Guide* (form 5044100). Unisys Corporation.
- A Series Micro A Workstation Processor Adapter 1 Board Installation Instructions*  
(form 8600 1708). Unisys Corporation.
- A Series Micro A 800 OS/2 System Installation Guide* (form 8600 1054). Unisys  
Corporation.
- A Series Micro A 825 OS/2 System Installation Guide* (form 8600 0700). Unisys  
Corporation.
- A Series Micro A 825 UNIX System Installation Guide* (form 8600 0718). Unisys  
Corporation.
- A Series Model S System Software Installation Guide* (form 5044423). Unisys  
Corporation.
- A Series MultiLingual System (MLS) Administration, Operations, and Programming  
Guide* (form 8600 0288). Unisys Corporation.
- A Series Network Administrative Utility (NAU) Operations Guide* (form 1222706).  
Unisys Corporation.
- A Series Network Control Facility (NCF) Implementation Guide* (form 1222671). Unisys  
Corporation.
- A Series Network Control Facility (NCF) Operations Guide* (form 1222672). Unisys  
Corporation.
- A Series Network Definition Language II (NDLII) Programming Reference Manual*  
(form 1169604). Unisys Corporation.

## Bibliography

---

- A Series NEWP Programming Reference Manual* (form 5044233). Unisys Corporation.
- A Series On-Line Data Entry System (ODESY) Installation Reference Manual* (form 5023682). Unisys Corporation.
- A Series Operating System Installation Guide* (form 8600 1021). Unisys Corporation.
- A Series OSI Implementation Guide* (form 3787 5374). Unisys Corporation.
- A Series Pascal Programming Reference Manual, Volume 1: Basic Implementation* (form 8600 0080). Unisys Corporation.
- A Series Pascal Programming Reference Manual, Volume 2: Product Interfaces* (form 8600 1294). Unisys Corporation.
- A Series PC-Based Model S System Software Installation Guide* (form 8600 1534). Unisys Corporation.
- A Series PC/MARC Installation and Operations Guide* (form 8600 0817). Unisys Corporation.
- A Series Physical I/O Technical Overview* (form 1169943). Unisys Corporation.
- A Series PL/I Reference Manual* (form 1169620). Unisys Corporation.
- A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide* (form 8600 1039). Unisys Corporation.
- A Series Printing Utilities Operations Guide* (form 8600 0692). Unisys Corporation.
- A Series Railroad Diagram Reference Card* (form 5044266). Unisys Corporation.
- A Series Remote Job Entry (RJE) Operations Reference Manual* (form 1169828). Unisys Corporation. Formerly *A Series Remote Job Entry (RJE) Reference Manual*.
- A Series Remote Job Entry/Binary Synchronous Communication (RJE/BSC) Configuration and Operations Guide* (form 8600 0866). Unisys Corporation.
- A Series Report Program Generator (RPG) Programming Reference Manual, Volume 1: Basic Implementation* (form 8600 0544). Unisys Corporation.
- A Series Report Program Generator (RPG) Programming Reference Manual, Volume 2: Product Interfaces* (form 8600 0742). Unisys Corporation.
- A Series Report Program Generator (RPG) Programming Template* (form 1195302). Unisys Corporation.
- A Series Screen Design Facility (SDF) Capabilities Manual* (form 1180437). Unisys Corporation.

- A Series Screen Design Facility (SDF) Operations and Programming Guide* (form 1185295). Unisys Corporation.
- A Series Screen Design Facility Plus (SDF Plus) Capabilities Manual* (form 8600 0270). Unisys Corporation.
- A Series Screen Design Facility Plus (SDF Plus) Installation and Operations Guide* (form 8600 0262). Unisys Corporation.
- A Series Screen Design Facility Plus (SDF Plus) Technical Overview* (form 8600 0072). Unisys Corporation.
- A Series Security Administration Guide* (form 8600 0973). Unisys Corporation.
- A Series Security Features Operations and Programming Guide* (form 8600 0528). Unisys Corporation.
- A Series SNA LU6.2 Service Manager Implementation and Operations Guide* (form 3787 5457). Unisys Corporation.
- A Series SNA LU6.2 Service Manager Programming Guide* (form 3787 5465). Unisys Corporation.
- A Series SNA LU6.2 Service Manager Programming Reference Manual* (form 3787 5473). Unisys Corporation.
- A Series SNA RJE Implementation and Operations Guide* (form 1221406). Unisys Corporation.
- A Series SNA 3270 Terminal Emulator Implementation and Administration Guide* (form 1221404). Unisys Corporation.
- A Series SNA 3270 Terminal Emulator Operations and Programming Guide* (form 1221405). Unisys Corporation.
- A Series Software Documents Master Index* (form 8600 1542). Unisys Corporation.
- A Series Software Release Installation Guide* (form 8600 0981). Unisys Corporation.
- A Series SORT Language Programming Reference Manual* (form 1169794). Unisys Corporation.
- A Series System Administration Guide* (form 8600 0437). Unisys Corporation.
- A Series System Architecture Reference Manual, Volume 2* (form 5014954). Unisys Corporation.
- A Series System Assistant Programming and Operations Guide* (form 8600 0825). Unisys Corporation.
- A Series System Commands Operations Reference Manual* (form 8600 0395). Unisys Corporation.

## Bibliography

---

- A Series System Configuration Guide* (form 8600 0445). Unisys Corporation.
- A Series System Management Facility II (SMFII) Query Operations Guide* (form 7831 1867). Unisys Corporation.
- A Series System Management Facility II (SMFII) Resource Management Operations Reference Manual* (form 7831 1628). Unisys Corporation.
- A Series System Messages Support Reference Manual* (form 8600 0429). Unisys Corporation.
- A Series System Operations Guide* (form 8600 0387). Unisys Corporation.
- A Series System Software Documentation* (CD-ROM Library) (form 4127 4564). Unisys Corporation.
- A Series System Software Installation Guide, Volume 1: A 9* (form 1169695). Unisys Corporation.
- A Series System Software Installation Guide, Volume 1: A 10* (form 1169935). Unisys Corporation.
- A Series System Software Installation Guide, Volume 2: System Initialization* (form 1170263). Unisys Corporation.
- A Series System Software Support Reference Manual* (form 8600 0478). Unisys Corporation.
- A Series System Software Utilities Operations Reference Manual* (form 8600 0460). Unisys Corporation.
- A Series Systems Functional Overview* (form 8600 0353). Unisys Corporation.
- A Series SYSTEMSTATUS Programming Reference Manual* (form 8600 0452). Unisys Corporation.
- A Series Task Attributes Programming Reference Manual* (form 8600 0502). Unisys Corporation. Formerly *A Series Work Flow Administration and Programming Guide*.
- A Series Task Management Programming Guide* (form 8600 0494). Unisys Corporation. Formerly *A Series Work Flow Administration and Programming Guide*.
- A Series Transmission Control Protocol/Internet Protocol (TCP/IP) Implementation Guide* (form 1221328). Unisys Corporation.
- A Series Work Flow Language (WFL) Programming Reference Manual* (form 8600 1047). Unisys Corporation.
- A Series X.25 MCS Operations and Programming Reference Manual* (form 8600 0577). Unisys Corporation.

- A Series/V Series LINC II Administration & Operations Guide* (form 3943 4451).  
Unisys Corporation.
- A Series/V Series LINC II Customizer Programming Guide* (form 3943 4378). Unisys  
Corporation.
- A Series/V Series LINC II Development Operations Guide* (form 3943 4444). Unisys  
Corporation.
- A Series/V Series LINC II Network Implementation Guide* (form 1220320). Unisys  
Corporation.
- A Series/V Series LINC II Programming Reference Handbook* (form 3943 4360). Unisys  
Corporation.
- A Series/V Series LINC II Programming Reference Manual* (form 3943 4485). Unisys  
Corporation.
- A Series/V Series LINC II Workstation Operations Guide* (form 3943 4436). Unisys  
Corporation.
- A Series/V Series System Software Documentation CD-ROM Library Operations Guide*  
(form 4127 0117). Unisys Corporation.
- American National Dictionary for Information Processing Systems* (technical report).  
American National Standards Committee X3, Information Processing Systems.  
Washington, DC: Computer and Business Equipment Manufacturers Association  
(CBEMA), 1982.
- B20 Systems Executive WRITEone Word Processing Reference Manual* (form 1162997).  
Unisys Corporation.
- B 1000 Series to A Series Progression Guide* (form 8600 0619). Unisys Corporation.
- B 5000/B 6000/B 7000 Series CANDE Reference Card* (form 5014533). Unisys  
Corporation.
- B 5000/B 6000/B 7000 Series Editor Reference Card* (form 5014392). Unisys  
Corporation.
- B 5000/B 6000/B 7000 Series GEMCOS Reference Card* (form 1154499). Unisys  
Corporation.
- B 9290-30 Intelligent Laser Printing System Font User's Reference Manual*  
(form 1154994). Unisys Corporation.
- B 9290-30 Intelligent Laser Printing System Operator's Guide* (form 1154986). Unisys  
Corporation.
- BNA Version 2 Network Attributes Data Dictionary* (form 3787 5382). Unisys  
Corporation.

## Bibliography

---

- BNA Version 2 Network Capabilities Overview* (form 3787 5440). Unisys Corporation.
- BNA Version 2 Network Encoded Messages Programming Reference Manual, Volumes 1 and 2* (form 3787 7529). Unisys Corporation.
- BNA Version 2 Library Overview Reference Card* (form 3787 5416). Unisys Corporation.
- BNA Version 2 Master Index Reference Handbook* (form 3787 5390). Unisys Corporation.
- BNA Version 2 Memory Dump Analyzer Status Codes Support Reference Manual* (form 1222729). Unisys Corporation.
- BNA Version 2 Operations Reference Manual, Volumes 1 and 2: Commands and Inquiries* (form 3787 7511). Unisys Corporation.
- BNA Version 2 Operations Reference Manual, Volume 3: Reports, Log Messages, and Error Messages* (form 3787 7552). Unisys Corporation.
- BTOS Business Graphics Package (BGP) Operating and Programming Guide* (form 1188133). Unisys Corporation.
- BTOS Data Transfer System (BTOS DTS) Operations and Programming Guide* (form 1195229). Unisys Corporation.
- BTOS Multiplan Operations Guide* (form 5022197). Unisys Corporation.
- Burroughs Network Architecture (BNA) Architectural Description Operating and Programming Reference Manual, Volume 1* (form 1132172). Unisys Corporation.
- Capacity Management Facility Snapshot (CMF.Snapshot) End Use Guide* (form 7830 7378). Unisys Corporation.
- Communications Processor Local Area Network (CPLAN) Planning Guide* (form 1225371). Unisys Corporation.
- CP 2000 Communications Processor Capabilities Overview* (form 1182391). Unisys Corporation.
- CP 2000 SNA PUT2/PUT5 and CPDLP PUT2 User Program Interface (UPI) Programming Guide* (form 1203585). Unisys Corporation.
- CP 2000 SNA PUT5 Implementation and Operations Guide* (form 1208360). Unisys Corporation.
- CP 2000 SNA PUT5 Operations Reference Manual* (form 1225442). Unisys Corporation.
- CP 2000/CPDLP Custom Protocol Programming Guide* (form 1222730). Unisys Corporation.



- CP 2000 SNA PUT5 Operations Reference Manual* (form 1225442). Unisys Corporation.
- CP 2000/CPDLP SNA PUT2 Operations Reference Manual* (form 1225445). Unisys Corporation.
- Department of Defense Trusted Computer System Evaluation Criteria* DOD 5200.28-STD. Fort George G. Meade, Maryland: National Computer Security Center (formerly DoD Computer Security Center), 1985.
- Dictionary of Computing*. Frank J. Galland (ed.). New York: John Wiley & Sons, 1982.
- DOS Data Transfer System (DOS DTS) Operations and Programming Guide* (form 1185345). Unisys Corporation.
- LINC II Executive Overview* (form 4124 1233). Unisys Corporation. Formerly *LINC II Capabilities Overview*.
- LINC II Reference Card* (form 1220324). Unisys Corporation.
- LINC II Run Time Transfer Operations Guide* (form 3943 4428). Unisys Corporation.
- LINC II Technical Overview* (form 4124 1241). Unisys Corporation.
- Pascal User Manual and Report*. K. Jensen and N. Wirth. New York: Springer-Verlag, 1978.
- Personal Workstation Data Transfer System (DTS) Installation and Configuration Guide* (form 8600 0890). Unisys Corporation.
- Personal Workstation Data Transfer System (DTS) Operations Guide, Volume 1: Simple Transfers* (form 8600 0908). Unisys Corporation.
- Personal Workstation Data Transfer System (DTS) Operations Guide, Volume 2: Advanced Operations* (form 8600 0916). Unisys Corporation.
- Personal Workstation Data Transfer System (DTS) System Messages Support Reference Manual* (form 8600 0924). Unisys Corporation.
- Personal Workstation<sup>2</sup> InfoExec Semantic Information Manager (SIM) Design Assistant (SDA) Operations Guide* (form 8600 0585). Unisys Corporation.
- REPORTER III Report Language Operations Guide* (form 1177185). Unisys Corporation.
- REPORTER III Vocabulary Language Operations Guide* (form 1177177). Unisys Corporation.
- "Revised Report on the Algorithmic Language ALGOL 60." *Communications of the ACM, Vol. 6, No. 1* (January 1963): pp 1-23.

## Bibliography

---

*Standard Pascal, ANSI/IEEE, 770X3.97-1983.* Institute of Electrical and Electronics Engineers and American National Standards Institute. 1983.

*System/36 to A Series RAMP Migration Aid Operations Guide* (form 8600 1450). Unisys Corporation.

*Vocabulary for Data Processing, Telecommunications, and Office Systems* (form GC10-1699-6). Poughkeepsie, New York: International Business Machines Corporation, 1981.

*Workstations Extended Graphics Terminal (EGT) Operations Guide* (form 1203692). Unisys Corporation.

*Workstations InfoExec Workstation Query Facility (WQF) Operations Guide* (form 1185279). Unisys Corporation.

*Workstations INFOVIEW II Operations Guide* (form 1210978). Unisys Corporation.

*Workstations INFOVIEW II Programming Reference Manual* (form 1210895). Unisys Corporation.

# Index

## A

- ACCESS command, in MAKEUSER utility, 9-19
- access file, duplicating, 7-9
- access, in SIM security, 9-14
- accesscode, 9-12
- accesscode password, 9-12
- accounting applications, 1-3
- Activity Reporting System (BARS), A-1
- actual segment descriptor (ASD) memory, 2-14
- AD (Access Duplicate) system command, 7-9
- ADD command, 7-5
- ADDS, (See Advanced Data Dictionary System (ADDS))
- ADM, (See automatic display mode (ADM))
- administration, (See system administration)
- Advanced Data Dictionary System (ADDS), 3-16
  - and application development, 5-10
  - and DMSII databases, 3-14
  - and DTS, 4-12
  - and IQF, 3-11
  - and SDF Plus, 5-19
  - and security, 9-14
  - and SIM databases, 3-9
- Advanced Police Information Management System (APLIMS), 1-3
- agenda, in COMS, 4-3
- ALGOL, 5-6
  - purpose, 5-6
- Alumni Development, 1-2
- APLB, A-1
- APLIMS, (See Advanced Police Information Management System (APLIMS))
- Application Services, in TCP/IP, 8-11
- application software, 2-7
  - business applications, 1-2
  - compatibility
    - across software releases, 11-3
    - across system models, 2-21
  - development by Unisys consultants, 10-2
  - Application Support Center (ASC), 1-2
  - architecture, (See system architecture)
- Archive Manager, in SDF Plus, 5-18
- ARPANET network, 8-10
- ASC, (See Application Support Center (ASC))
- ASD (Actual Segment Descriptor) system command, 7-4
- ASD memory, (See actual segment descriptor (ASD) memory)
- ASD table size, 7-4
- assembly language, 5-1
  - and A Series hardware design, 2-10
- auditing for security, 9-16
- automatic display mode (ADM), 6-4
- AVAILMIN memory management parameter, 7-4

## B

- B 1000 Series
  - and BNA Version 1, 8-28
  - progression to A Series, 10-3
  - support of progression tools, A-2
- B 21, B 25, B 26, B 27, B 28, B 38 systems, (See BTOS workstations)
- backup facilities for files, 7-6
- backup file, in printing, 7-17
- Backup Processor utility, 7-20
- BAMCS, (See Business and Manufacturing Control System (BAMCS))
- banking applications, 1-3
- BARS, (See Activity Reporting System (BARS))
- BASIC, A-1
- BASIC Random file format, 4-11
- BASIC Sequential file format, 4-11
- BCT Translator Filter, 10-3

## Index

---

- BDLC, (See Data Link Control (BDLC))
  - BDP, (See Unisys Distribution Partner (BDP))
  - billing, process, 7-3
  - BILLINGSUPPORT library, 7-3
  - Binder, 5-15
    - and Message Translation Utility (MSGTRANS), 5-11
  - BNA Print Routing, 7-19
  - BNA Version 1, 8-27
    - and LINK family, 8-3
    - and Mail System, 8-13
    - applications programming features, 5-17
    - coexistence with BNA Version 2, 8-22
    - connections to non-A Series systems, 8-28
    - hardware components, 8-30
    - Host Services, 8-28
    - operations, 8-32
    - support status, A-2
  - BNA Version 1/BNA Version 2 migration, 8-22
  - BNA Version 2, 8-15
    - and LINK family, 8-3
    - and Mail System, 8-13
    - and TCP/IP, 8-11
    - applications programming features, 5-17
    - centralized operations, 8-20
    - coexistence with BNA Version 1, 8-22
    - custom line protocols, 8-21
    - custom protocols, 8-21
    - debugging tools, 8-21
    - decentralized operations, 8-19
    - hardware components, 8-15
      - multihost networks, 8-17
      - single-host networks, 8-15
    - Host Services, 8-15, 8-28
    - network definition, 8-19
    - SNA compatibility, 8-5
  - BRT Translator Filter, 10-3
  - BTOS II operating system, 4-11
  - BTOS operating system
    - and DTS level 3.0, 4-11
    - and DTS level 4.1, 4-11
    - and DTS level 5.0, 4-12
  - BTOS workstations, vi
    - and CMF.Snapshot, 9-2
    - and DTS, 4-11
    - and ERGO, 3-15
    - and IDE, 5-20
    - and INFOVIEW II, 4-9
    - and MAPPER, 5-5
    - and WQF, 3-11
  - Buildscreens, in SMFII, 9-8
  - Business and Manufacturing Control System (BAMCS), 1-3
  - business applications
    - standard, 1-2
  - B1000COPY, 10-3
- ## C
- C, 5-7
  - CANDE, (See Command and Edit (CANDE))
  - capacity, 2-15
  - Capacity Management Facility.Disk (CMF.Disk), 7-15
  - Capacity Management Facility.Snapshot (CMF.Snapshot), 9-1
  - capacity planning studies, by Unisys consultants, 10-3
  - CASTS, (See Computerized Accounting Student Terminal System (CASTS))
  - catalog file, duplicating, 7-9
  - CATALOGING, 7-7
    - and Mirrored Disk, 7-10
  - CD-ROM Library, 12-1
  - central processing module (CPM), 2-2
  - central processing unit (CPU), 2-2
    - upgrades, 2-18
  - central processor subsystem, 2-2
  - CHANGE command, 7-5
  - class, in SIM databases, 3-5
  - classes, education, 10-1
  - CM (Change MCP) system command, 7-9
  - CMF.Disk, (See Capacity Management Facility.Disk (CMF.Disk))
  - CMF.Snapshot, (See Capacity Management Facility.Snapshot (CMF.Snapshot))
  - CMF.Snapshot Workstation Component, 9-2
  - CMSCOPY, 10-3
  - COBOL(68), A-2
  - COBOL74, 5-7
    - and ADDS, 5-11
    - progression from B 1000 Series COBOL68, 10-3
    - progression from B 1000 Series COBOL74, 10-3
  - COBOL85, 5-7
  - code
    - nonmodifiable, 2-12
    - reentrance, 2-12
    - segmentation, 2-12

- Command and Edit (CANDE), 4-8
    - file maintenance statements, 7-5
    - recovery, 4-8
  - command mode, in MARC, 6-3
  - command transfer, in Host Services, 8-28
  - commercial applications, 1-3
  - COMMLINK, 6-6
  - Communications Management System (COMS), 4-1
    - and INFOVIEW II, 4-10
    - and MARC, 6-1
    - and SDF Plus, 5-19
    - and work flow management, 7-1
    - flexible configuration, 4-2
    - migration from GEMCOS, 4-7
    - security, 9-15
    - Statistics facility, 9-9
    - synchronized recovery, 4-6
    - Utility, 4-2
    - versions
      - Full-Featured, 4-7
      - Kernel, 4-7
    - windows, 4-2
      - automatic program initiation, 4-3
      - database recovery, 4-6
      - processing item, 4-4
      - transaction routing, 4-3
  - communications processor data link processor (CPDLP), 8-17
    - and SNA, 8-5
  - communications processor local area network (CPLAN), 8-16
  - communications processor 2000 (CP 2000)
    - interface to LINK family, 8-3
  - compatibility
    - across software releases, 11-3
    - among A Series models
      - application software, 2-21
      - hardware upgrades, 2-18
      - operating systems, 2-20
      - operations procedures, 2-21
  - Computerized Accounting Student Terminal System (CASTS), 1-2
  - COMS, (See Communications Management System (COMS))
  - COMS Statistics facility, 9-9
  - COMS Utility, 4-2
  - concurrency control, in databases, 3-20
    - and LINC II, 5-3
  - configuration, 7-20
    - application program independence of, 2-15
    - operating system independence of, 2-15
    - configuration file, 7-20
    - configuration utility, 7-21
  - COPY command, 7-5
    - and tape file names, 7-16
  - Court Management System, 1-3
  - CP 2000, (See CP 2000 communications processor)
  - CP 2000 communications processor, 8-16
    - and TCP/IP, 8-11
    - Memory Dump Analyzer, 8-22
    - SNA compatibility, 8-5
  - CP 2000 Memory Dump Analyzer, 8-22
  - CPDLP, (See communications processor data link processor (CPDLP))
  - CPLAN, (See communications processor local area network (CPLAN))
  - CPM, (See central processing module (CPM))
  - CPU, (See central processing unit (CPU))
  - CP9500 systems, and BNA Version 1, 8-28
  - critical support object tape, 11-1
  - CU (Core Usage) system command, 9-4
  - CUBE, B-1
  - CUSTOMCARE customer services, 10-1
    - education, 10-1
    - professional services, 10-2
    - software support, 10-4
  - customer education, 10-1
  - C2 security, 9-19
- D**
- D-notes, 11-4
  - DAM, (See direct access method (DAM))
  - DASDL, (See Data and Structure Definition Language (DASDL), Data and Structure Definition Language (DASDL))
  - Data and Structure Definition Language (DASDL), 3-14, 3-18
  - Data Base Transfer (DBT) utility, 10-3
  - data comm, (See data communications)
  - data comm mode
    - and MARC availability, 6-1
  - data communications, 8-1
    - BNA Version 1 networks, 8-27
    - coexistence with BNA Version 2, 8-22
    - connections to non-A Series systems, 8-28
    - hardware components, 8-30
    - Host Services, 8-28

## Index

---

- operations, 8-32
- BNA Version 2 networks, 8-15
  - centralized operations, 8-20
  - coexistence with BNA Version 1, 8-22
  - configuration, 8-19
  - custom line protocols, 8-21
  - debugging tools, 8-21
  - decentralized operations, 8-19
  - hardware components, 8-15
  - Host Services, 8-15, 8-28
- DCDLP and NSP/LSP networks, 8-23
  - configuration through LINC II, 5-2
  - configuration tools, 8-25
  - custom line protocols, 8-26
  - debugging tools, 8-26
  - hardware components, 8-24
  - traffic monitoring, 8-27
- open networks
  - HYPERchannel and NETEX, 8-1
  - LINK family, 8-2
  - OSI, 8-8
  - SNA, 8-5
  - TCP/IP, 8-10
  - X/25, 8-13
- Data Communications ALGOL (DCALGOL), 5-7
  - automating operator tasks, 6-7
  - security restrictions, 9-16
  - use in system monitoring, 9-9
- data communications data link processor (DCDLP), 8-24
  - and SNA 3270, 8-8
  - and SNA/RJE, 8-8
- data communications data link processor II (DCDLPII), 8-24
- data communications host adapter (DCHA), 8-24
- data definition languages (DDLs), 3-18
- data file security, 9-13
- data interchange format (DIF) file format, 4-11
  - and WQF, 3-12
- Data Link Control (BDLC) protocol, 8-26
  - station, 8-31
- data link processor (DLP), 2-4
  - performance, 2-19
- Data Management ALGOL (DMALGOL), 5-8
- Data Management System II (DMSII) database, 3-12
  - creation and reorganization through ADDS and DASDL, 3-14
  - through LINC II, 5-2
- database structures, 3-12
- operations, 3-18
  - backup, auditing, and recovery, 3-19
  - integrity certification, 3-21
  - performance monitoring, 3-22
  - structure analysis, 3-21
- query and update facilities, 3-15
  - DMS.View, 3-16
  - DMSII interpretive interface, 5-10
  - Extended Retrieval with Graphic Output (ERGO), 3-15
  - host language interface, 3-16, 5-10
- security, 9-14
- Data Transfer System (DTS), 4-10
  - and ERGO, 3-15
  - and IQF, 3-10
  - release level 3.0, 4-11
  - release level 4.1, 4-11
  - release level 5.0, 4-12
- data transfer unit (DTU), 2-5
  - and Hardware Disk Cache, 7-12
- data-valued attribute (DVA), 3-5
- database management, 3-1
  - backup
    - remote, 3-20
  - concurrency control, 3-20
  - DMSII databases, 3-12
    - creation and reorganization, 3-14
    - database structures, 3-12
    - query and update facilities, 3-15
  - InfoExec and DMSII environments, 3-2
  - operations, 3-18
    - backup, auditing, and recovery, 3-19
    - integrity certification, 3-21
    - performance monitoring, 3-22
    - structure analysis, 3-21
  - progression to new releases, 11-3
  - security, 9-14
  - SIM databases, 3-4
    - creation and reorganization, 3-9
    - data structures, 3-5
    - query and update facilities, 3-9
  - SQLDB databases, 3-3
- DATACOMINFO file, 8-25
- Datamapper utility, 9-8
- DBANALYZER utility, 3-21
- dBASE III file format, 4-11
- DBCERTIFICATION utility, 3-21
- DBT, (See Data Base Transfer (DBT) utility)
- DBTGEN, 10-3

- DCALGOL, (See Data Communications ALGOL (DCALGOL))
- DCAUDITOR utility, 8-27
- DCDLP, (See data communications data link processor (DCDLP))
- DCDLPDUMPANALYZER utility, 8-27
- DCDLPPII, (See data communications data link processor II (DCDLPPII))
- DCHA, (See data communications host adapter (DCHA))
- DCKEYIN function, in DCALGOL, 9-10
- DCSTATUS utility, 8-27
- DD (Directory Duplicate) system command, 7-9
- DDL, (See data definition languages (DDLs))
- DDN, (See Defense Data Network (DDN))
- Defense Data Network (DDN)  
and OSI, 8-9  
and TCP/IP, 8-11
- deimplementation warnings, 11-3
- descriptors, 2-13
- Diagnostic MCS, A-2
- Dialog Manager, in DTS, 4-12
- DIF, (See data interchange format (DIF) file format)
- direct access method (DAM), 4-11
- Direct HY file, 8-2
- directive facility, in MARC, 6-3
- directory, 7-5
- disk drive, 7-8  
performance, 2-19
- disk pack, 7-8
- disk subsystem  
family, 7-8  
file  
backup facilities, 7-6  
maintenance, 7-5  
Memory Disk, 7-13  
Mirrored Disk feature, 7-9  
protection against failures, 7-9  
reliability, 7-9  
scrubbing for security, 9-17
- disk unit, 7-8
- distributed systems software, in OSI, 8-10
- distribution applications, 1-3
- DLP, (See data link processor (DLP))
- DMALGOL, (See Data Management ALGOL (DMALGOL))
- DMMONITOR utility, 3-22  
for data management, 3-22
- DMS.View utility, 3-16
- DMSII, (See Data Management System II (DMSII) database)
- DMSII environment, 3-2
- DMSII Inquiry, A-2
- DMSII interpretive interface, 5-10
- DMUTILITY program, 3-20
- DO files, in CANDE, 4-8
- documentation, 12-1  
distribution, 12-2  
online, 12-2  
printed, 12-2  
temporary, 12-3
- DOCUMENTS tape, 11-4
- DOS Random file format, 4-11
- DOS Terminal Emulator (DTE), 4-12
- DTE, (See DOS Terminal Emulator (DTE))
- DTS, (See Data Transfer System (DTS))
- DTU, (See data transfer unit (DTU))
- DU (Disk Utilization) system command, 9-4
- DUMPALL utility, 7-6
- DUMPANALYZER utility, 5-20
- duplicate system files, 7-9  
MCP/AS code file  
performance, 7-15  
reliability, 7-9
- DVA, (See data-valued attribute (DVA))
- ## E
- EDCDLP, (See enhanced data communications data link processor (EDCDLP))
- Editor, 5-21
- education applications, 1-2
- education classes, 10-1
- EMS, (See Entry and Medium Systems (EMS))
- enhanced data communications data link processor (EDCDLP), 8-24
- entity-valued attribute (EVA), 3-5
- entity, in SIM databases, 3-5
- Entry and Medium Systems (EMS), 2-1
- environmental software, 2-7
- ERGO, (See Extended Retrieval with Graphic Output (ERGO))
- Ethernet network, 8-9, 8-11
- EVA, (See entity-valued attribute (EVA))
- Executive LINC, 1-3
- Executive WRITEone, 4-11
- extended attributes, in SIM databases, 3-8

## Index

---

Extended Retrieval with Graphic Output  
(ERGO), 3-15, A-2  
and COMS Statistics, 9-9

## F

FACTOR memory management parameter,  
7-4

family

of disks, 7-8

of processes, 7-3

FCN, (See Field Change Notice (FCN))

Field Change Notice (FCN), 11-4

file

accessing across a BNA network, 8-28

attribute, 5-13

backups, 7-6

equation, 5-13

maintenance, 7-5

CATALOGING, 7-7

DUMPALL utility, 7-6

security, 9-13

transfer across a BNA network, 8-29

File Maintenance Utility (FMU), 9-8

File Transfer Protocol (FTP), 8-11

File Transfer, Access, and Management  
(FTAM), 8-10

FILECOPY utility, 7-7

FILEDATA utility, 2-8

financial applications, 1-2

firmware updates, 11-4

first-generation language, 5-1

flashes, 12-3

FMU, (See File Maintenance Utility (FMU))

Forms Editor, in SDF Plus, 5-18

Forms Processor, in SDF Plus, 5-18

FORTRAN

ANSI-66 version, A-2

ANSI-77 version, 5-8

FORTRAN77, (See FORTRAN)

fourth-generation language

definition, 5-1

LINC II, 5-2

MAPPER, 5-5

FTAM, (See File Transfer, Access, and  
Management (FTAM))

FTP, (See File Transfer Protocol (FTP))

functional subsystems, 2-1

## G

GC (Group Configuration) system command,  
7-21

GEMCOS, (See Generalized Message  
Control System (GEMCOS))

generalization hierarchies, in SIM databases,  
3-7

Generalized Message Control System  
(GEMCOS), 4-7

support status, A-2

GETSTATUS function, in DCALGOL, 9-10

GFS, (See Global Financial System (GFS))

Global Financial System (GFS), 1-2

Global Wholesale Banking (GWB), 1-3

grants, in SQLDB security, 9-14

growth factor, 2-16

growth paths

application compatibility, 2-21

hardware compatibility, 2-18

operating system compatibility, 2-20

operations compatibility, 2-21

performance measurements, 2-15

guard file, 9-13

GUARDFILE utility, 9-13

GWB, (See Global Wholesale Banking  
(GWB))

## H

hardware

basic components, 2-1

input/output (I/O) subsystem, 2-4

maintenance subsystem, 2-5

memory subsystem, 2-3

processor subsystem, 2-2

compatibility, 2-18

I/O subsystem upgrades, 2-19

mainframe upgrades, 2-18

compiler-oriented, 2-10

data communications

advanced networks, 8-15

basic multihost networks, 8-30

basic single-host networks, 8-24

functional subsystems, 2-1

microcode

processor, 2-3

updates, 11-4

reliability monitoring, 9-1

updating, 11-4



- HASP, (See Houston Automatic Spooling Program (HASP))
- HDU, (See host data unit (HDU))
- help book, 6-4
- help text, 12-2
- Help Utility
- and IDC, 8-26
  - and MARC, 6-4
  - multilingual capabilities, 6-10
- HMS, (See Hospital Management System (HMS))
- hospital applications, 1-3
- Hospital Management System (HMS), 1-3
- host data unit (HDU), 2-4
- systems, 2-4
- host language interfaces, 5-10
- Host Services, 8-28
- command transfer, 8-28
  - file access, 8-28
  - file transfer, 8-29
  - interprocess communication, 8-29
  - job transfer, 8-29
  - logging, 8-30
  - print routing, 8-29
  - remote tasking, 8-29
  - security, 8-30
  - station transfer, 8-30
- Houston Automatic Spooling Program (HASP), 8-5
- HYDLP, (See HYPERchannel data link processor (HYDLP))
- HYPERchannel, 8-1
- Direct HY file interface, 8-2
  - HYPERchannel link, 8-1
  - NETEX, 8-2
- HYPERchannel data link processor (HYDLP), 8-2
- I**
- I/O subsystem, (See input/output (I/O) subsystem)
- I/O upgrades, 2-19
- IBM compatibility, 8-5
- IBM PC
- and DTS, 4-11
  - and IDE, 5-20
  - and INFOVIEW II, 4-9
  - and MAPPER, 5-5
  - and WQF, 3-11
- RJE/BSC, 8-5
- SNA, 8-5
- ICP, (See integrated communications processor (ICP))
- ICP Memory Dump Analyzer, 8-22
- IDC, (See Interactive Datacomm Configurator (IDC))
- IDE, (See Intelligent Distributed Editor (IDE))
- IGSDA, (See InfoGuard System Data Access (IGSDA) support library)
- IMG, (See Interactive Menugraph Generator (IMG))
- INDEPENDENTTRANS option, in databases, 3-20
- indexed sequential-access method (ISAM), 4-11
- and DTS, 4-11
- Individual Kit, 11-1
- industrial applications, 1-3
- InfoExec, 3-2
- Advanced Data Dictionary System (ADDS), 3-9, 3-14
  - Interactive Query Facility (IQF), 3-10
  - Operations Control Manager (OCM), 3-18
  - Semantic Information Manager (SIM), 3-4
  - Workstation Query Facility (WQF), 3-11
- InfoGuard, 9-17
- C2 security, 9-19
  - password aging, 9-18
  - preventing usercode abuse, 9-18
  - security administrator status, 9-17
  - security message workstation, 9-19
  - selective logging, 9-19
  - selective logging features, 9-19
  - simplified security administration, 9-17
  - tape security, 9-18
- InfoGuard System Data Access (IGSDA) support library, 9-19
- INFOVIEW II, 4-9
- and IDE, 5-21
  - and MAPPER, 5-5
  - and SDF Plus, 5-19
  - and WQF, 3-12
  - CMF.Snapshot, 9-2
- inherited attributes, in SIM databases, 3-8
- initialization, system, 7-24
- input/output (I/O) subsystem, 2-4
- management, 7-5
  - disk subsystem, 7-8
  - file maintenance, 7-5
  - print system, 7-17
  - tape subsystem, 7-16

## Index

---

- programming language features, 5-13
  - sharing between separate hosts, 7-24
- input/output processor (IOP), 2-4
  - performance, 2-19
- INSTALL (Install Software) system
  - command, 11-2
- integrated communications processor (ICP), 8-16
  - debugging, 8-22
- Intelligent Distributed Editor (IDE), 5-20
- Intelligent Peripheral Interface (IPI), 2-5
- Interactive Datacomm Configurator (IDC), 8-25
  - multilingual capabilities, 6-10
- Interactive Menugraph Generator (IMG), 6-3
  - and IDC, 8-26
- Interactive Query Facility (IQF), 3-10
- INTERACTIVEXREF utility, 5-22
- internationalization, 1-8, 6-7
- InterPro, 1-7
- interprocess communications (IPC), 5-16
  - across a BNA network, 8-29
- inverse attribute, in SIM databases, 3-7
- IOP, (See input/output processor (IOP))
- IPC, (See interprocess communications (IPC))
- IPI, (See Intelligent Peripheral Interface (IPI))
- IQF, (See Interactive Query Facility (IQF))
- ISAM, (See indexed sequential-access method (ISAM))

## J

- JCL, (See Job Control Language (JCL))
- JES, (See Job Entry System (JES))
- job, 7-2
  - history monitoring, 9-1
- Job Control Language (JCL), 8-5
- Job Entry System (JES), 8-5
- job queues, 7-3
- job summary, 5-23
  - customizing, 9-5
- job transfer, across a BNA network, 8-29

## K

- KEYEDIO, 5-14, A-2
- KEYEDIOII, 5-14

- and DTS, 4-11
- kit release, 11-1

## L

- languages
  - natural, (See MultiLingual System (MLS))
  - programming, (See programming languages)
- Large Systems (LS), 2-1
- legal applications, 1-3
- library
  - in program development, 5-14
  - in system architecture, 2-12
- library maintenance, 7-5
  - tape file format, 7-6
  - tape file names, 7-16
- LINC II, (See Logic and Information Network Compiler II (LINC II))
- LINC.View utility, 5-4
- line of business applications
  - customized, 10-2
- line support processor (LSP), 8-24
- LINK Family Data/Voice Network Exchange, 8-2
- LISTNOTES program, 11-4
- local alias usercode, 8-30
- local area network (LAN)
  - BNA Version 1, 8-31
  - BNA Version 2, 8-16
  - Ethernet, 8-9, 8-11
  - HYPERchannel, 8-1
  - OSI, 8-9
  - TCP/IP, 8-11
- log-on requirements, 9-11
- LOGANALYZER utility, 9-5
  - and security auditing, 9-16
- LOGCONSOLIDATOR utility, 9-7
  - and security auditing, 9-16
- LOGGER utility, 9-5
  - and security auditing, 9-16
  - for billing, 7-3
- logging
  - of BNA activity, 8-30
  - security auditing, 9-16
    - InfoGuard enhancements, 9-19
    - system summary log, 9-5
- LOGGING (Logging Options) system
  - command, 9-19

Logic and Information Network Compiler II  
(LINC II), 5-2  
 B 1000 progression, 10-3  
 logical database, 9-14  
 LOGSELECT usercode attribute, 9-19  
 Lotus 1-2-3, 3-12, 4-11  
 LS, (See Large Systems (LS))  
 LSP, (See line support processor (LSP))  
 LTTABLEGEN utility, 7-20

## M

machine language, 5-1  
 Macros, in the Editor, 5-21  
 magnetic tape, (See tape)  
 Mail System, 8-13  
 maintenance mode, 2-6  
 maintenance subsystem, 2-5  
 MAKEUSER utility, 9-12  
 ACCESS command, 9-19  
 LOGSELECT usercode attribute, 9-19  
 MAPPER, 5-5  
 MAPPER Connect, 5-5  
 MARC, (See Menu-Assisted Resource Control (MARC))  
 Marketing Associate agreement, 1-3  
 master control program (MCP), 2-7  
 Master Control Program/Advanced Systems (MCP/AS)  
 duplicate code files, 7-9, 7-15  
 MCP/AS, (See Master Control Program/Advanced Systems (MCP/AS))  
 MCS, (See message control system (MCS))  
 medical applications, 1-3  
 memory subsystem, 2-3  
 management, 7-4  
 ASD memory, 7-4  
 menu mode, in MARC, 6-2  
 Menu-Assisted Resource Control (MARC), 6-1  
 command mode, 6-3  
 customizing the interface, 6-3  
 functions, 6-1  
 menu and command mode, 6-2  
 menu mode, 6-2  
 multilingual capabilities, 6-10  
 security, 6-3  
 message control system (MCS), 4-1  
 in the System Software Facility, 2-8  
 message level interface processor (MLIP), 2-4  
 Message Translation Utility (MSGTRANS), 6-9  
 and application development, 5-11  
 and TADS, 5-20  
 microcode  
 processor, 2-3  
 updates, 11-4  
 migration, (See progression)  
 millions of instructions per second (MIPS), 2-15  
 MILNET network, 8-10  
 MINIMAL security class, 9-17  
 MIPS, (See millions of instructions per second (MIPS))  
 Mirrored Disk, 7-9  
 performance, 7-15  
 MLIP, (See message level interface processor (MLIP))  
 MLIP systems, 2-4  
 MLS, (See MultiLingual System (MLS))  
 MONITOR facility, in CMF.Snapshot, 9-3  
 monitoring the system, (See system monitoring)  
 MS-DOS  
 and DTS, 4-11, 4-12  
 and INFOVIEW II, 4-10  
 MSGTRANS, (See Message Translation Utility (MSGTRANS))  
 multilingual capabilities  
 and application development, 5-11  
 and IQF, 3-11  
 and LINC II, 5-3  
 and TADS, 5-20  
 MultiLingual System (MLS), 6-9  
 Multiplan  
 and DTS, 4-11  
 and WQF, 3-12  
 multiprocessing, 2-11  
 multiprogramming, 2-11

## N

native window, 4-10  
 NAU, (See Network Administrative Utility (NAU))  
 NCF, (See Network Control Facility (NCF))  
 NDII, (See Network Definition Language II (NDLII))  
 NDII post compiler (NPC), 8-26

## Index

---

NDLIANALYZER utility, 8-27  
NETEX, (See NETwork EXecutive (NETEX))  
Network Administrative Utility (NAU), 8-19  
Network Control Facility (NCF), 8-20  
Network Definition Language II (NDLII), 8-26  
NETwork EXecutive (NETEX), 8-2  
network support processor (NSP), 8-24  
networking, (See data communications)  
NEWP language, 5-8  
NPC, (See NDLII post compiler (NPC))  
NSP, (See network support processor (NSP))  
NSPDUMPANALYZER utility, 8-27

## O

object code files, security restrictions for, 9-13  
OCM, (See Operations Control Manager (OCM))  
ODESY, (See On-Line Data Entry System (ODESY))  
ODT, (See operator display terminal (ODT))  
ODT mode  
    and SCTs, 2-6  
    capabilities of, 6-4  
ODTDLP, (See operator display terminal data link processor (ODTDLP))  
office management applications, 1-3  
OFIS Link, 1-3  
OLAYGOAL memory management parameter, 7-4  
On-Line Data Entry System (ODESY)  
    progression from B 1000 Series, 10-3  
    support status, A-2  
online documentation, 12-2  
Online Reporter III, A-2  
Open Systems Interconnection (OSI), 8-8  
operating system  
    basic concepts, 2-9  
        ASD memory architecture, 2-14  
        code segmentation, 2-12  
        compiler-oriented hardware, 2-10  
        configuration-independent software, 2-14  
        descriptors, 2-13  
        libraries, 2-12  
        multiprogramming and multiprocessing, 2-11  
        reentrant code, 2-12  
        stack architecture, 2-10  
        virtual memory, 2-13  
    compatibility, 2-20  
    Master Control Program/Advanced Systems (MCP/AS), 2-7  
    operational interface, 6-1  
        automating operator tasks, 6-7  
        coexistence of menu and command interfaces, 6-5  
        command interface, 6-4  
        compatibility across A Series, 2-21  
        menu interface, 6-1  
        multilingual interfaces, 6-9  
            language selection, 6-10  
            menugraph translation, 6-10  
            message translation, 6-9  
            screen translation, 6-10  
    PCMARC  
        communications, 6-6  
        host component, 6-6  
        system configuration, 6-6  
        workstations, 6-6  
    security restrictions, 9-15  
    workstation, 6-6  
Operations Control Manager (OCM), 3-18  
operator display terminal (ODT), 2-4, 6-4  
    MARC and system command mode  
        coexistence, 6-5  
        system command mode, 6-4  
operator display terminal (ODT) commands, (See system commands)  
operator display terminal data link processor (ODTDLP), 2-4  
OS/2  
    and DTS, 4-12  
OSI, (See Open Systems Interconnection (OSI))

## P

P-notes, 11-4  
packet-switched data networks (PSDNs), 8-13  
    and OSI, 8-9  
    and TCP/IP, 8-11  
Pascal, 5-8  
password, 9-11  
    generation and aging, 9-18  
PATCH utility, 5-22  
PCMARC, 6-6

- COMMLINK, 6-6
  - terminal emulator, 6-6
- performance
  - data link processor (DLP), 2-19
  - database, 3-22
  - disk drives, 2-19
  - I/O, 2-19
  - I/O processor (IOP), 2-19
  - units for measuring, 2-15
  - upgrade paths, 2-18
- performance monitoring, (See system monitoring)
- performance, in SIM databases, 3-8
- peripherals, 2-5
- permission, in SIM security, 9-14
- personal computer (PC)
  - IBM, (See IBM compatibility)
  - Unisys, (See Unisys PC)
- Personal Workstation Data Transfer System (PW DTS), 4-12
- Personal Workstation 2 (PW2)
  - and DTS, 4-11
  - and INFOVIEW II, 4-9
- PL/I, A-2
- police applications, 1-3
- Polish notation, 2-11
- port file, 5-16
- PPP, (See Professional Project Practices (PPP))
- primitive system commands, 6-4
- Print System (PrintS), 7-17
  - BNA Print Routing, 7-19
  - LTTABLEGEN utility, 7-20
  - ReprintS, 7-19
  - spooling, 7-17
- PRINTBINDINFO utility, 5-16
- printer backup file, 7-17
- PrintS, (See Print System (PrintS))
- priority, 2-11
  - and work flow management, 7-2
- PRIORITY memory management parameter, 7-4
- private file, 9-13
- privileged status, 9-12
- privileges
  - of programs, 9-13
  - of stations, 9-12
  - of users, 9-11
- PRN, (See Professional Resource Network (PRN))
- PROBEMANAGER, 9-2
- process, 2-10
  - billing, 7-3
  - family, 7-3
  - management, 7-1
    - billing for resources used, 7-3
    - restarting processes, 7-3
    - work flow management, 7-1
  - restarting, 7-3
- process stack, 2-10
- processing item, in COMS, 4-4
- processor, (See central processing unit (CPU), input/output processor (IOP))
- product support information manuals, 12-3
- Professional Project Practices (PPP), 10-2
- Professional Resource Network (PRN), 1-3
- professional services, 10-2
- program
  - controlling access to, 9-13
  - privileges of, 9-13
- program development, 5-1
  - by Unisys consultants, 10-2
- fourth-generation, 5-2
  - LINC II, 5-2
  - MAPPER, 5-5
- third-generation, 5-6
  - development environment, 5-18
  - language extensions, 5-13
  - product interfaces, 5-10
  - programming languages, 5-6
- PROGRAMDUMP facility, 5-20
- programming languages
  - combining
    - at compile time, 5-15
    - at run time, 5-14
  - hardware support of, 2-10
  - standards, 5-6
- progression, 10-3
  - B 1000 Series to A Series, 10-3
  - BNA Version 1 to BNA Version 2, 8-22
  - GEMCOS to COMS, 4-7
  - to a new software release, 11-2
- protected input, 4-6
- PROTECTION file attribute, 7-26
- protocols, customized
  - for CP 2000s or CPDLPs, 8-21
  - for NSP/LSPs or EDCDLPs, 8-26
- PSDNs, (See packet-switched data networks (PSDNs))
- public file, 9-13
- public sector applications, 1-3
- Public Service, 10-4
- purgeless cache memory, 2-3

## Index

---

PUT2 Gateway, 8-6  
PUT5 Gateway, 8-6  
PW DTS, (See Personal Workstation Data Transfer System (PW DTS))  
PW2 DTS, (See Personal Workstation 2 (PW2))

## Q

QLA, (See Quad Line Adapter (QLA))  
Quad Line Adapter (QLA), 8-24  
QUERY, 9-7  
query facilities  
    for DMSII databases, 3-15  
    for SIM databases, 3-9

## R

ready queue, 2-11  
REAPPLYCOMPLETED option, in  
    databases, 3-20  
reconfiguring the system, 7-21  
reentrant code, 2-12  
referential integrity, in SIM databases, 3-7  
Relative Performance Measurement (RPM),  
    2-16  
release  
    hardware, 11-4  
    software, 11-1  
    support of old releases, 11-2  
Release Notes, 11-4  
remote job entry (RJE), (See Remote Job Entry/Binary Synchronous Communication (RJE/BSC), SNA Remote Job Entry (SNA/RJE))  
    and the A Series MCS, A-2  
    as a generic term, 8-5  
Remote Job Entry/Binary Synchronous Communication (RJE/BSC), 8-5  
Remote Print System (Reprints), 7-19  
Remote Support Center, 2-6  
remote support interface, 2-6  
remote support mode, 2-6  
remote tasking, 8-29  
REMOVE command, 7-5  
Report Program Generator (RPG), 5-9  
Reporter III, A-2  
reporting facilities  
    for DMSII databases, 3-15  
    for SIM databases, 3-9

Reprints, (See Remote Print System (Reprints))  
resource management module (RMM), 2-5  
resource utilization monitoring, 9-1  
response time, 2-15  
restarting processes, 7-3  
RESTRICT (Restrict Unit or Volume) system command  
    and programs, 9-14  
    and system command mode, 9-16  
Reverse Polish notation, 2-11  
RJE/BSC, (See Remote Job Entry/Binary Synchronous Communication (RJE/BSC))  
RLTABLEGEN utility, 7-16  
RMM, (See resource management module (RMM))  
RPG  
    progression from B 1000 Series RPG, 10-3  
RPM, (See Relative Performance Measurement (RPM))

## S

SAM, (See sequential access method (SAM))  
SAMPLER, 9-7  
SCP, (See system control processor (SCP))  
Screen Design Facility (SDF)  
    migration from, 5-19  
    support of, A-2  
Screen Design Facility Plus (SDF Plus)  
    and ADDS, 5-11  
    and COMS programming, 5-13  
scrubbing, of disk or tape files, 9-17  
SCS, (See system control subsystem (SCS))  
SCT, (See system control terminal (SCT))  
SDA, (See Semantic Information Manager (SIM) Design Assistant (SDA))  
SDF Plus, (See Screen Design Facility Plus (SDF Plus))  
second-generation language, 5-1  
SECOPT screen and command, 9-17  
security administration, 9-10  
    access controls  
        for data files, 9-13  
        for databases, 9-14  
        for multiple host networks, 8-30  
        for programs, 9-13  
        for system commands, 9-15  
        for transactions, 9-15  
        in MAPPER, 5-6

- auditing, 9-16
- basic security features, 9-11
- computer viruses, 9-14
- C2 Security, 9-19
- in MARC, 6-3
- InfoGuard, 9-17
- logging analysis, 9-19
- password generation and aging, 9-18
- privileges
  - of programs, 9-13
  - of stations, 9-12
  - of users, 9-11
- security administrator status, 9-17
- security monitoring workstation, 9-19
- simplified security administration, 9-17
- tape security, 9-18
- usercodes, preventing abuse of, 9-18
- security administrator status, 9-17
- security class, 9-17
- SECURITY command, 7-5, 9-13
- security message workstation, 9-19
- Security Support Library, A-2
- SEGARRAYSTART memory management
  - parameter, 7-4
- segment dictionary, 2-13
- selective logging, 9-19
- Semantic Information Manager (SIM)
  - database, 3-4
  - creation and reorganization, 3-9
  - data structures, 3-5
  - operations, 3-18
    - backup, auditing, and recovery, 3-19
    - integrity certification, 3-21
    - performance monitoring, 3-22
    - structure analysis, 3-21
  - query and update facilities, 3-9
    - host language interface, 5-10
    - Interactive Query Facility (IQF), 3-10
    - Workstation Query Facility (WQF), 3-11
  - security, 9-14
- Semantic Information Manager (SIM) Design Assistant (SDA), 3-9
- Semantic Information Manager Object Definition Language, 3-18
- sequential access method (SAM), 4-11
- services, customer, (See CUSTOMCARE customer services)
- SETSTATUS function, in DCALGOL, 9-10
- SF (Set Factor) system command, 7-4
- SI, (See Simple Installation (SI))
- SIM, (See Semantic Information Manager (SIM) database)
- SIM Design Assistant (SDA), (See Semantic Information Manager (SIM) Design Assistant (SDA))
- SIM ODL, (See Semantic Information Manager Object Definition Language)
- Simple Installation (SI)
  - for installing a new system, 7-24
  - for progressing to a new release, 11-1
- Simple Mail Transfer Protocol (SMTP), 8-11
- SITEINPUT, 9-7
- SITEINPUT Buildscreens, 9-8
- SITESUPPORT library, 9-5
- SMFII, (See System Management Facility II (SMFII))
- SMS I, (See system maintenance station)
- SMS II, (See system maintenance station)
- SMS III, (See system maintenance station)
- SMTP, (See Simple Mail Transfer Protocol (SMTP))
- SNA, (See Systems Network Architecture (SNA))
- SNA Gateway, 8-6
- SNA Logical Unit Type 6.2 Service Manager, 8-7
- SNA LU6.2 Service Manager, 8-7
- SNA Remote Job Entry (SNA/RJE), 8-8
- SNA 3270 Terminal Emulator, 8-8
- SNA/RJE, (See SNA Remote Job Entry (SNA/RJE))
- SNAPSHOT facility, in CMF.Snapshot, 9-3
- snapshot performance analysis
  - by Unisys consultants, 10-3
  - using CMF.Snapshot, 9-1
- soft configuration, 7-20
  - configuration file, 7-20
  - FREE and ACQUIRE, 7-21
- software
  - application, 2-7
  - environmental, 2-7
  - system, 2-7
  - updating, 11-1
- software support, 10-4
- SORT, (See SORT language)
- SORT intrinsic, 5-17
- SORT language, 5-9, 5-17
- spooling, of printer output, 7-17
- SQL DDL, (See Structured Query Language Data Definition Language (SQL DDL))

## Index

---

- SQL DML, (See Structured Query Language Data Manipulation Language)
- SQLDB, (See Structured Query Language Database (SQLDB))
- SSF, (See System Software Facility (SSF))
- stack architecture, 2-10
- station privileges, 9-12
- station transfer, across a BNA network, 8-30
- statistics facility, in COMS, 9-9
- Structured Query Language Data Definition Language (SQL DDL), 3-18
- Structured Query Language Data Manipulation Language, 3-3
- Structured Query Language Database (SQLDB), 3-1, 3-3
  - database structure, 3-3
  - security, 9-14
- subclass, in SIM databases, 3-7
- SUMLOG, (See system summary log (SUMLOG))
- support
  - software, 10-4
    - support of old releases, 11-2
  - support release, 11-1
  - Support Service, 10-5
  - support services, (See CUSTOMCARE customer services)
- SUSPENDUSERCODE command, in MARC, 9-18
- SYLK, (See Symbolic Link (SYLK) file format)
- Symbolic Link (SYLK) file format, 4-11 and WQF, 3-12
- synchronized recovery, 4-6
- system administration, 7-1
  - I/O management, 7-5
  - memory management, 7-4
  - process management, 7-1
  - soft configuration, 7-20
  - system initialization, 7-24
- system architecture, 2-1
  - growth paths, 2-15
  - hardware components, 2-1
  - operating system concepts, 2-9
  - system software components, 2-7
- System Assistant, 6-6
- system commands
  - and ODT interface, 6-4
  - restrictions on use of, 9-15
- system control processor (SCP), 2-6
- system control subsystem (SCS), 2-5
- system control terminal (SCT), 2-5
- system directory, 7-9
- system initialization, 7-24
- system maintenance station, 2-6
- System Management Facility II (SMFII), 9-6
  - Datamapper, 9-8
  - FMU, 9-8
  - for billing, 7-3
  - LOGCONSOLIDATOR, 9-7
  - QUERY, 9-7
  - SAMPLER, 9-7
  - SITEINPUT, 9-7
  - SITEINPUT Buildscreens, 9-8
- system monitoring, 9-1
  - customized monitoring programs, 9-9
  - database statistics, 9-9
  - detailed performance analysis, 9-6
  - log analysis, 9-5
  - realtime performance displays
    - CMF.Snapshot, 9-1
    - SAMPLER, 9-7
  - snapshot performance reporting, 9-1
  - through system commands, 9-4
  - transaction statistics, 9-9
- system notes, 11-4
- system operations, (See operational interface)
- system performance, (See performance)
- System Services Control Point (SSCP), in SNA, 8-6
- system software, 2-7
  - basic components, 2-7
    - language compilers, 2-9
    - Master Control Program/Advanced Systems (MCP/AS), 2-7
    - message control systems (MCSs), 2-8
    - programming utilities, 2-9
    - ReprintS, 2-8
    - system utilities, 2-8
- System Software Facility (SSF), 2-7
- system summary log (SUMLOG), 9-5
  - InfoGuard enhancements, 9-19
  - security auditing, 9-16
- SYSTEM/BACKUP, 7-20
- SYSTEM/BCT Translator Filter, 10-3
- SYSTEM/BRT Translator Filter, 10-3
- SYSTEM/B1000COPY, 10-3
- SYSTEM/CMSCOPY, 10-3
- SYSTEM/DBTGEN, 10-3
- Systems Network Architecture (SNA), 8-5
  - LU6.2 Service Manager, 8-7
  - PUT2 Gateway, 8-6
  - PUT5 Gateway, 8-6



SNA 3270 Terminal Emulator, 8-8  
 SNA/RJE, 8-8  
 SYSTEMSTATUS function, in DCALGOL,  
 9-10  
 S2 security class, 9-17

## T

TADS, (See Test and Debug System  
 (TADS))  
 tag field, 2-12  
 tape  
   DUMPALL file format versus library  
     maintenance format, 7-6  
   management, 7-16  
   scrubbing, 9-17  
   security, 9-18  
   supported tape label types, 7-16  
 task, 7-2  
 task attribute, 5-16  
 task control processor (TCP), 2-5  
   and multiprogramming, 2-12  
 task equation, 5-16  
 TCP, (See task control processor (TCP))  
 TCP/IP, (See Transmission Control  
 Protocol/Internet Protocol (TCP/IP))  
 Technical Information Papers, 12-3  
 TELENET packet-switched data network,  
 8-9, 8-11  
 Telnet, 8-11  
 Test and Debug System (TADS), 5-20  
 third-generation programming, 5-6  
   definition, 5-1  
   development environment, 5-18  
     automated program compilation and  
     testing, 5-23  
   change control and cross-references,  
   5-22  
   interactive debugging environment,  
   5-20  
   terminal-based program development,  
   5-21  
   user interface generation, 5-18  
   workstation-based program  
   development, 5-20  
 language extensions, 5-13  
   communication between host systems,  
   5-17  
   compile-time sharing of program  
   modules, 5-15  
 I/O features, 5-13

interprocess communication, 5-16  
 run-time sharing of program modules,  
 5-14  
 SORT intrinsic, 5-17  
 task initiation and control, 5-16  
 languages, 5-6  
   ALGOL, 5-6  
   C, 5-7  
   COBOL74, 5-7  
   COBOL85, 5-7  
   DCALGOL, 5-7  
   DMALGOL, 5-8  
   FORTRAN77, 5-8  
   NEWP, 5-8  
   Pascal, 5-8  
   RPG, 5-9  
   SORT, 5-9  
   WFL, 5-9  
 product interfaces, 5-10  
   centralized data definitions, 5-10  
   COMS, 5-13  
   database interfaces, 5-10  
   multilingual interfaces, 5-11  
   transaction processing interfaces, 5-13  
 throughput, 2-15  
 Timeplex, 8-3  
 TPS, (See transaction processing system  
 (TPS))  
 transaction management, 4-1  
   and workstations, 4-9  
   on the mainframe, 4-1  
     file maintenance and program control,  
     4-8  
     generalized transaction control, 4-1  
     response time monitoring, 9-9  
     security, 9-15  
 transaction processing system (TPS), A-2  
 transaction trail, in COMS, 4-6  
 Transmission Control Protocol/Internet  
 Protocol (TCP/IP), 8-10  
 TYMNET packet-switched data network,  
 8-9, 8-11

## U

U (Utilization) system command, 9-4  
 UIP, (See user interface processor (UIP))  
   and SCTs, 2-6  
 UIS, (See user interface system (UIS))  
 Unisys Distribution Partner (BDP), 1-3  
 Unisys PC

## Index

---

- and DTS, 4-11
- and IDE, 5-20
- and INFOVIEW II, 4-9
- and MAPPER, 5-5
- and WQF, 3-11
- CMF.Snapshot, 9-2
- Unisys SureNet electronic bulletin board, 10-4
  - Public Service, 10-4
  - Support Service, 10-5
- Unisys SURETY Service Program, 10-4
  - service options, 10-5
  - Unisys SureNet Electronic Bulletin Board, 10-4
- Unisys Users Association (UUA), B-2
- updating
  - hardware, 11-4
  - software, 11-1
- upgrades, 2-18
  - central processing unit (CPU), 2-18
  - I/O, 2-19
  - mainframe, 2-18
- user groups, B-1
  - CUBE, B-1
  - Unisys Users Association (UUA), B-2
- user interface processor (UIP), 2-4
  - and ODTs, 2-4
- user interface system (UIS), 5-18
- user privileges, 9-11
- usercode, 9-11
  - local alias usercode, 8-30
  - monitoring for misuse, 9-18
- UUA, (See Unisys Users Association (UUA))
- U5000 Series, 8-9, 8-10

## V

### V Series

- and BNA Version 1, 8-28
- views, in SQLDB security, 9-14
- virtual disk, in DTS, 4-12
- virtual memory, 2-13
- virtual printer, in DTS, 4-12
- virus, computer, 9-14
- VOLUME statement, in WFL, 9-18
- Von Neumann architecture, 2-9

## W

- WFL, (See Work Flow Language (WFL))
- windows
  - in COMS, 4-2
  - in WQF, 3-11
- work file, 4-8
- Work Flow Language (WFL), 5-9
  - file maintenance statements, 7-5
  - in program development, 5-23
  - job queues, 7-3
  - job restarts, 7-3
  - system command mode, 6-4
  - tape file names, 7-16
  - used to automate operator tasks, 6-7
- work flow management, 7-1
- workload analysis, by Unisys consultants, 10-3
- workstation, 4-9
  - and ERGO, 3-15
  - and LINC II, 5-4
  - data transfer and reformatting, 4-10
  - database queries from, 3-11
  - windows into the mainframe environment, 4-9
- Workstation Query Facility (WQF), 3-11
- WQF, (See Workstation Query Facility (WQF))
- WRITEone, 4-11

## X

- X.25 message control system, 8-14
- X.25 network, 8-13
  - and OSI, 8-9
  - and TCP/IP, 8-11
- X.25 station group, 8-14
- XE520 system
  - and INFOVIEW II, 4-9
  - and WQF, 3-11
- XREFANALYZER utility, 5-22

- 1100 Series, 8-9, 8-10
- 2780 terminal, 8-5
- 3270 Terminal Emulator, 8-8
- 360/20 RJE system, 8-5
- 3780 terminal, 8-5





86000353-000