



NATIONAL COMPUTER SECURITY CENTER

AD-A247 237



FINAL EVALUATION REPORT

International Business Machines Corporation

MVS/ESA Operating System

92-05776



17 September 1990

Approved for Public Release:
Distribution Unlimited

FINAL EVALUATION REPORT
International Business Machines Corporation
MVS/ESA

NATIONAL
COMPUTER SECURITY CENTER

9800 Savage Road
Fort George G. Meade
Maryland 20755-6000

17 September 1990



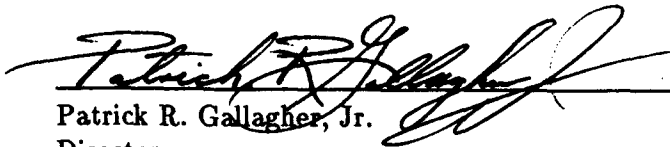
Report No. CSC-EPL-90/002
Library No. S235.899

Accession For	
NTIS Grant	<input checked="" type="checkbox"/>
ERIC Tab	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

FOREWORD

This publication, the Final Evaluation Report IBM MVS/ESA is being issued by the National Computer Security Center under the authority of and in accordance with DoD Directive 5215.1, "Computer Security Evaluation Center." The purpose of this report is to establish the candidate rating for the IBM MVS/ESA operating system. The requirements stated in this report are taken from the *Department of Defense Trusted Computer System Evaluation Criteria*, dated December 1985.

Approved:



Patrick R. Gallagher, Jr.
Director,
National Computer Security Center

17 September 1990

**Final Evaluation Report IBM MVS/ESA
ACKNOWLEDGMENTS**

ACKNOWLEDGMENTS

Team Members

Team members included the following individuals, who were provided by the following organizations:

Dana Nell Stigdon
Cynthia Grall
Kenneth B. Elliott, III
Jerzy W. Rub
Robin Oliver
James K. Goldston
Tom Anderson

National Computer Security Center
Fort Meade, MD

The Aerospace Corporation
El Segundo, CA

TABLE OF CONTENTS

Foreword	i
Acknowledgments	ii
Executive Summary	viii
1 Introduction	1
1.1 Evaluation Process Overview	1
1.2 Document Organization	2
1.3 MVS/ESA Background and History	3
2 Hardware Architecture	5
2.1 Introduction	5
2.2 Model 3090 Processors	6
2.2.1 Physical Partitioning	7
2.3 Model 4381 Processors	8
2.4 JES Complexes	9
2.5 Address Space Selection	10
2.6 Addressing Modes	11
2.7 Address Translation	12
2.8 Prefixing	16
2.9 Hardware Protection Mechanisms	16
2.9.1 Key-Controlled Protection	16
2.9.2 Page Protection	18
2.9.3 Low-Address Protection	18
2.10 Privileged and Semi-Privileged Instructions	18
2.11 Input/Output	25
2.11.1 The Channel Subsystem	25
2.11.2 Control Units and Devices	27
2.12 Interrupt Handling	30
3 Software Architecture	33
3.1 MVS/SP	34
3.1.1 Address Spaces	34
3.1.2 Dataspaces	37
3.1.3 Hiperspaces	38
3.1.4 Dispatcher	39
3.1.5 Inter-Address Space Communication	40

Final Evaluation Report IBM MVS/ESA
TABLE OF CONTENTS

3.1.6	Storage Managers	43
3.1.7	I/O Operations	44
3.1.8	Virtual Input/Output	48
3.1.9	Address Space Creation	49
3.1.10	System Initiation	51
3.1.11	Consoles	53
3.1.12	Exit Routines	54
3.1.13	System Management	55
3.1.14	Supervisor Calls (SVC)	59
3.1.15	Authorized Programs	61
3.2	Data Facility Product	62
3.2.1	Device Volumes and Labels	62
3.2.2	Volume Table of Contents	63
3.2.3	Catalogs	64
3.2.4	Data Sets	65
3.2.5	Data Set Types	65
3.2.6	Access Methods	68
3.2.7	Storage Management Subsystem	69
3.2.8	Other DFP Functions	70
3.3	Job Entry Subsystem	70
3.3.1	Job Control Language, Jobs, and Tasks	71
3.3.2	JES and MVS Communication	73
3.3.3	JES Data Sets	73
3.3.4	JES Complexes	75
3.3.5	Job Entry Subsystem 2	75
3.3.6	Job Entry Subsystem 3	80
3.4	ACF/VTAM	86
3.4.1	Network Addressable Units	87
3.4.2	Sessions	87
3.4.3	VTAM Application Program Interface	89
3.4.4	VTAM Terminal I/O Coordinator	90
3.4.5	Application Programs	90
3.5	TSO/E	91
3.5.1	TSO/E Logon Processing	91
3.5.2	The Terminal Monitor Program	92
3.5.3	User Attributes Data Set	93
3.5.4	TSO/E Interface with VTAM	93
3.5.5	Message Commands	95
3.6	Print Services Facility	95
3.6.1	Security Libraries	97
3.6.2	Security Resources	97
3.6.3	Mandatory Print Labeling	98

Final Evaluation Report IBM MVS/ESA
TABLE OF CONTENTS

3.6.4	PSF Interface to JES	98
3.7	Resource Access Control Facility	98
3.7.1	RACF Interface to MVS	99
3.7.2	RACF Data Base	100
3.7.3	RACF Manager	101
3.7.4	Recovery	102
3.7.5	RACF Users	102
3.7.6	RACF Groups	103
3.7.7	Resource Classes	103
3.7.8	User Properties	103
3.7.9	Profiles	107
3.7.10	Audit	108
3.7.11	RACF Commands	108
3.7.12	RACF Options	108
4	Protected Resources	111
4.1	Subjects	111
4.2	Objects	111
4.2.1	DASD Data Sets	112
4.2.2	SYSOUT Data Sets	112
4.2.3	Tape Volumes	112
4.2.4	TSO TPUT Messages	112
5	Protection Mechanisms	113
5.1	Identification and Authentication	113
5.1.1	Userid, Password, and Groupid	113
5.1.2	Mapping Subjects to Userids	114
5.2	Discretionary Access Control	115
5.2.1	Access Determination	116
5.2.2	Objects	118
5.2.3	Other Protected Resources	120
5.3	Mandatory Access Control	122
5.3.1	Access Control Policy	123
5.3.2	Tranquility	123
5.3.3	Subjects	124
5.3.4	Objects	124
5.4	Object Reuse	125
5.4.1	RACF ERASE Option	125
5.4.2	Controlling Reuse of Other Objects	126
5.4.3	Controlling Reuse of System Structures	127
5.4.4	Storage Reuse	129
5.5	Auditing	130
5.5.1	SMF Records and Data set Management	130

Final Evaluation Report IBM MVS/ESA
TABLE OF CONTENTS

5.5.2	Auditor	130
5.5.3	Logging	131
5.5.4	JES Complex Auditing	132
5.5.5	RACF Report Writer	132
5.5.6	Data Security Monitor	133
6	Other Assurances	135
6.1	Functional Testing	135
6.1.1	Software Testing	135
6.1.2	Hardware Testing	137
6.2	Security Model	137
6.3	System Modification Program	138
6.4	System Generation	138
6.5	System Integrity	139
6.5.1	On-Line Test Executive Program (OLTEP)	139
6.5.2	SYS1.LOGREC Error Recording Data Set	139
6.5.3	Processor Complex Exerciser	140
6.5.4	Built-In Diagnostics	140
7	Evaluation as a B1 System	141
7.1	Discretionary Access Control	141
7.2	Object Reuse	142
7.3	Labels	143
7.4	Label Integrity	144
7.5	Exportation of Labeled Information	145
7.6	Exportation to Multilevel Devices	145
7.7	Exportation to Single-Level Devices	146
7.8	Labeling Human-Readable Output	147
7.9	Mandatory Access Control	148
7.10	Identification and Authentication	149
7.11	Audit	150
7.12	System Architecture	151
7.13	System Integrity	152
7.14	Security Testing	153
7.15	Design Specification and Verification	154
7.16	Security Features User's Guide	155
7.17	Trusted Facility Manual	155
7.18	Test Documentation	158
7.19	Design Documentation	158
7.20	Additional Requirements	159
7.20.1	Discretionary Access Control	159
7.20.2	Trusted Facility Management	160

Final Evaluation Report IBM MVS/ESA
TABLE OF CONTENTS

8 Evaluator's Comments	161
A Evaluated Hardware Components	163
A.1 Processors	163
A.2 DASD Controllers	163
A.3 DASD Devices	163
A.4 Tape Controllers	163
A.5 Tape Devices	164
A.6 Printers	164
A.7 Terminals	164
A.8 Other Hardware	164
B Evaluated Software Components	165
B.1 TCB Software	165
B.2 Software Outside the TCB	171
C Acronyms	173

**Final Evaluation Report IBM MVS/ESA
EXECUTIVE SUMMARY**

Executive Summary

The security protection provided by the International Business Machines Corporation Multiple Virtual Storage/Enterprise Systems Architecture (MVS/ESA[®]) operating system with the additional required software products (see page 165 "Appendix B, Evaluated Software Components"), configured according to the most secure manner described in the Trusted Facility Manual, running on ESA/370[®] architecture machines (see page 163, "Appendix A, Evaluated Hardware Components") has been examined by the National Security Agency (NSA). The security features of MVS/ESA were evaluated against the requirements specified by the *DoD Trusted Computer System Evaluation Criteria* (the Criteria) dated December 1985.

The NSA evaluation team has determined that the highest class at which MVS/ESA satisfies all the specified requirements of the Criteria is Class B1. Therefore, the system using the specified hardware and consisting of MVS/SP[®] Version 3 Release 1.3, Job Entry Subsystem 2 Version 3 Release 1.3 or Job Entry Subsystem 3 Version 3 Release 1.3, Time Sharing Option/Extensions Version 2 Release 1.1, Advanced Communications Function for Virtual Telecommunications Access Method ESA Version 3 Release 3, Print Services Facility Version 1 Release 3.0, Data Facility Product Version 3 Release 1.1, and Resource Access Control Facility Version 1 Release 9 configured in the most secure manner described in the Trusted Facility Manual has a B1 rating.

A system that has been rated as a B division system provides a Trusted Computing Base (TCB) that preserves the integrity of sensitivity labels and uses them to enforce a set of mandatory access control rules. The system developer has provided the security policy model on which the TCB is based and furnished a specification of the TCB and evidence that the reference monitor concept has been implemented.

MVS/ESA is IBM's flagship operating system for IBM 3090 and IBM 4381 mainframe computers supporting the ESA/370 architecture. MVS/ESA adheres to this architecture taking advantage of the various protection mechanisms offered. MVS/ESA systems may execute on machines with up to six processors. The MVS/ESA systems may in turn be combined with other MVS/ESA systems, running the exact same copy of the operating system, to form JES complexes capable of supporting hundreds or thousands of users. MVS/ESA can be used in a wide variety of environments.

The MVS/ESA operating system is a general purpose time-sharing and batch system with several security features. The hardware provides address space separation and privilege isolation. The Resource Access Control Facility (RACF) product enhances certain security features providing a controlled access to system resources using access control lists, mandatory controlled access to resources using sensitivity labels, a default protection of these resources, an extensive auditing facility, and a role separation among privileged users.

[®]MVS/ESA is a registered trademark of the IBM Corporation

[®]ESA/370 is a registered trademark of the IBM Corporation

[®]MVS/SP is a registered trademark of the IBM Corporation

Chapter 1

Introduction

In April 1989 the National Security Agency (NSA) began design analysis of Multiple Virtual Storage/System Product (MVS/SP) Version 3 Release 1.3, Job Entry Subsystem 2 (JES2) Version 3 Release 1.3, Job Entry Subsystem 3 (JES3) Version 3 Release 1.3, Time Sharing Option Extensions (TSO/E) Version 2 Release 1.1, Advanced Communications Function for the Virtual Telecommunications Access Method (ACF/VTAM[®]) ESA Version 3 Release 3, Print Services Facility (PSF) Version 1 Release 3, Data Facility Product (DFP) Version 3 Release 1.1, and Resource Access Control Facility (RACF) Version 1 Release 9. All of these items are products of International Business Machines Corporation (IBM) and will collectively be referred to as MVS/ESA in this report. The objective of this evaluation was to rate the MVS/ESA system against the DoD Trusted Computer System Evaluation Criteria (TCSEC), and to place it on the Evaluated Products List (EPL) with a final rating. This report documents the results of the formal evaluation. This evaluation applies to the system as available from IBM in August 1990.

Material for this report was gathered by the NSA MVS/ESA evaluation team through documentation, interaction with system developers, testing, and use of the system.

1.1 Evaluation Process Overview

The Department of Defense Computer Security Center was established in January 1981 to encourage the widespread availability of trusted computer systems for use by facilities processing classified or other sensitive information. In August 1985 the name of the organization was changed to the National Computer Security Center. In order to assist in assessing the degree of trust one could place in a given computer system, the TCSEC was written. The TCSEC establishes specific requirements that a computer system must meet in order to achieve a predefined level of trustworthiness. The TCSEC levels are arranged hierarchically into four major divisions of protection, each with certain security-relevant characteristics. These divisions are in turn subdivided into classes. To determine the division and class at which all requirements are met by a system, the system must be evaluated against the TCSEC by an NSA evaluation team.

The NCSC supports the creation of secure computer products in varying stages of development from initial design to those that are commercially available. Preliminary to an evaluation, products must go through the Proposal Review Phase. This phase includes an assessment of the vendor's capability to create a secure system and complete the evaluation process. To support this assessment, a

[®]ACF/VTAM is a registered trademark of the MVS/ESA Corporation

Final Evaluation Report IBM MVS/ESA
CHAPTER 1. INTRODUCTION

Preliminary Technical Review (PTR) of the system is done by the NSA. This consists of a quick review of the current state of the system by a small but expert team and the creation of a short report on the state of the system. If a vendor passes the Proposal Review Phase, they will enter a support stage preliminary to evaluation. This support stage has two phases, the Vendor Assistance Phase (VAP) and the Design Analysis Phase (DAP). During VAP, the newly assigned team reviews design specifications and answers technical questions that the vendor may have about the ability of the design to meet the requirements. A product will stay in VAP until the vendor's design, design documentation, and other required evidence for the target TCSEC class are complete and the vendor is well into implementation. At that time, the support moves into DAP.

The primary thrust of DAP is an in-depth examination of a manufacturer's design for either a new trusted product or for security enhancements to an existing product. DAP is based on design documentation and information supplied by the industry source; it involves little "hands on" use of the system, but during this phase the vendor should virtually complete implementation of the product. DAP results in the production of an Initial Product Assessment Report (IPAR) by the NSA assessment team. The IPAR documents the team's understanding of the system based on the information presented by the vendor. Because the IPAR contains proprietary information and represents only a preliminary analysis by the NSA, distribution is restricted to the vendor and the NSA.

Products that have completed the support stage with the successful creation of the IPAR enter formal evaluation. Products entering formal evaluation must be complete security systems. In addition, the release being evaluated must not undergo any additional development. The formal evaluation is an analysis of the hardware and software components of a system, all system documentation, and a mapping of the security features and assurances to the TCSEC. The analysis performed during the formal evaluation requires "hands on" testing (i.e., functional testing and, if applicable, penetration testing). The formal evaluation results in the production of a final report and an Evaluated Products List entry. The final report is a summary of the evaluation and includes the EPL rating which indicates the final class at which the product satisfies all TCSEC requirements in terms of both features and assurances. The final report and EPL entry are made public.

1.2 Document Organization

This report consists of eight chapters and three appendices. Chapter 1 is an introduction. Chapters 2 and 3 are an overview of the hardware and software architectures respectively. Chapter 4 is a description of the protected resources in MVS/ESA. Chapter 5 is a description of the protection mechanisms available in MVS/ESA. Chapter 6 is a description of other assurances provided by IBM for MVS/ESA. Chapter 7 is a mapping of MVS/ESA protection mechanisms to the B1 requirements in the *DoD Trusted Computer System Evaluation Criteria*. Chapter 8 contains the evaluator comments. The three appendices identify the hardware and software components of an evaluated system, and the acronyms used throughout the report.

1.3 MVS/ESA Background and History

MVS has evolved from over 20 years of operating system design. Its development began in 1964 with the announcement of the System/360 computers. MVS was to be IBM's solution to incompatible system software for different computer systems. The six System/360 computers used a standard architecture and instruction set to provide compatibility for a wide range of computer systems.

The OS/360 was a single operating system designed to serve all System/360 computers. There were three versions of OS/360: Primary Control Program (PCP), Multiprogramming with a Fixed Number of Tasks (MFT), and Multiprogramming with a Variable Number of Tasks (MVT). The three versions were similar in that they shared the same Job Control Language (JCL), but differed in job scheduling and resource management. PCP was an operating system designed to control a single program at a time, whereas MFT and MVT offered multiprogramming.

The MFT used a single real address space which consisted of a nucleus residing in low storage and up to 52 fixed sized partitions. There were two types of partitions, system and problem state. User jobs executed in a problem state partition containing an active initiator and were assigned one of the four-bit protection keys (key 0 was used in system partitions). Since there were only 15 available keys, MFT could control a maximum of 15 multiprogrammed jobs.

In many ways, MVT was similar to MFT. MVT controlled job processing with JCL, used protection keys to assign jobs to a class, and contained many of the same system operations (e.g., a nucleus, Master Scheduler, and Link Pack Area) as MFT. However, MVT did not fix the sizes of its partitions but rather varied them dynamically based on allocation requests it received from job initiators.

In 1972 IBM announced virtual storage (VS) for many of their System/370 computers. The VS function could be incorporated into existing System/370 computers by adding Dynamic Address Translation (DAT) and Translation Lookaside Buffer (TLB) hardware, adding microcode (found on smaller systems), or bought outright on newly introduced machines. Although the hardware and its speed varied for each machine, the logical function of DAT did not change.

Based upon OS/360, IBM created three operating systems for the System/370 VS computers: Operating System/Virtual Storage 1 (OS/VS1), Operating System/Virtual Storage 2 Release 1 (OS/VS2 Rel 1), and OS/VS2 Rel 2, known as MVS. IBM also offers DOS/VS and Virtual Machine/370 for the System/370 computers, but these operating systems were developed from DOS and CP/67, respectively. OS/VS1 was the VS successor of MFT and offered fixed segments in a single-virtual 16M address space.

The two releases of OS/VS2 were significantly different. The OS/VS2 Rel 1 was the VS replacement for MVT. OS/VS2 Rel 1 was similar to MVT in that it dynamically allocated space to jobs, but different in that it had a 16M virtual address space and allocated space in 64K segments. This allocation scheme was used so that OS/VS2 Rel 1 could incorporate paging, but resulted in fragmentation of the virtual address space. A more advanced approach was used by OS/VS2 Rel 2 which has multiple virtual address spaces. OS/VS2 Rel 2, known as MVS, became available in

Final Evaluation Report IBM MVS/ESA
CHAPTER 1. INTRODUCTION

1974 and has undergone three product revisions since then. The first revision, MVS/SP 1, was announced in 1978 and was followed by MVS/XA, the extended architecture version, in 1982. The third revision, MVS/ESA (Enterprise Systems Architecture), was announced in 1988 and was made possible by a new architecture—Enterprise Systems Architecture/370.

In addition, IBM has developed programs which, when used with the operating system, serve and assist both the system and its users. Each one of these programs has a dedicated task. This report will discuss seven such programs: Data Facility Product (DFP), Job Entry Subsystem 2 (JES2), Job Entry Subsystem 3 (JES3), Advanced Communication Function for the Virtual Telecommunications Access Method (ACF/VTAM), Time Sharing Option/Extensions (TSO/E), Print Services Facility (PSF), and Resource Access Control Facility (RACF).

IBM has been developing MVS and its predecessor systems for the past 25 years and the most current version (MVS/SP Version 3 Release 1.3 and its corresponding products) has incorporated many of the capabilities, applications, and performance factors of former systems while expanding addressing capabilities, simplifying operations, and providing high performance storage and retrieval of data. MVS/ESA offers a level of compatibility among MVS systems and backward compatibility with former systems.

Chapter 2

Hardware Architecture

2.1 Introduction

The MVS/ESA system runs on machines with the Enterprise Systems Architecture/370 (ESA/370) architecture. Certain models have a vector processing capability, and these models are included in the evaluated configuration. The mainframes included in this evaluation are the 4381 and the 3090 series of machines (for specific model numbers, see page 163, "Evaluated Hardware Components"). These machines all support the Enterprise Systems Architecture mode of operation which is specified in the *IBM Enterprise Systems Architecture/370 Principles of Operation* manual and described briefly in the following section.

The system consists of main storage, optional expanded storage, one or more central processors (CPs), operator facilities, a channel subsystem, control units, and input/output (I/O) devices. Main storage, which is directly addressable by the CPs, provides for high-speed processing of data by the CPs and the channel subsystem. Each processor contains the sequencing and processing facilities for instruction execution, interruption action, timing functions, and other machine-related functions.

Each CP provides registers which are available to programs but do not have addressable representation in main storage. They include sixteen 32-bit access registers, the current Program Status Word (PSW), sixteen 32-bit general registers, four 64-bit floating-point registers, sixteen 32-bit control registers, the prefix register, and the registers for the clock comparator and the CP timer. The PSW is a 64-bit register containing the value of the instruction address, protection information, and interrupt status. Access registers, control registers, and the prefix register are described in subsequent sections.

The channel subsystem directs the flow of information between I/O devices and main storage. I/O devices are attached through control units to the channel subsystem via channel paths. Control units may be attached to the channel subsystem via more than one channel path, and an I/O device may be attached to more than one control unit.

The ESA/370 architecture provides two states of operation, supervisor and problem (user). The state is controlled by bit 15 in the PSW. When in supervisor state, the currently executing process may use privileged instructions. These instructions provide the current process with the ability to circumvent all security mechanisms. Therefore, all programs with the ability to run in supervisor state are part of the Trusted Computing Base (TCB). When in problem state, processes are restricted by the memory protection mechanisms and are not allowed to execute privileged instruc-

Final Evaluation Report IBM MVS/ESA
CHAPTER 2. HARDWARE ARCHITECTURE

tions. These restrictions cause problem-state processes to abide by the security policies enforced by the TCB. However, problem state programs can run with keys 0-7 (see page 16, "Key-Controlled Protection") and can run APF authorized (see page 61, "Authorized Programs") thus becoming part of the TCB.

Programs change from problem state to supervisor state only through Supervisor or Program Call instructions, or an interrupt. Both of these cause a new PSW to be loaded from a protected storage location which for SVCs is the low storage area (see page 18, "Low-Address Protection").

There are two families of machines that are included in the evaluated configuration: the 3090 family and the 4381 family. The following sections describe the specifics of the architectures of each family, and the differences in the models listed in Appendix A.

2.2 Model 3090 Processors

The models of the 3090 family that are in the evaluated configuration fall into two categories: those with one CP and those with more than one CP. Model numbers prefixed with a "1" (e.g., 120, 180) fall into the former category, while the 2X0, 3X0, 400, 500, and 600 refer to the 2, 3, 4, 5, and 6 processor models, respectively. Specific models have been further labeled with suffix letters E, S, J, and JH.

Models basic to this evaluation have an E suffix. Models S can be characterized by a shorter cycle time, a direct memory path from each processor to memory, larger cache and central storage, improved control communications, faster floating point operations, and an increased number of subchannels. These in turn can be further upgraded to models J and JH which feature shorter cycle times, increased cache memory, new chip technology, and more models. However, all models support the ESA architecture and the following discussion pertains equally well to all of them.

A 3090 uniprocessor has central storage, expanded storage, subchannels, a system control element, and a central processor. Subchannels and expanded storage are described in later sections in the overview.

The system control element (SCE) has the capability to control up to three processors. While the 1X0, some 2X0, and some 3X0 models only have one SCE, certain 2X0 and 3X0 models and the 400, 500, and 600 models have two SCEs, each of which controls from 1 to 3 CPs. When two SCEs are present, each SCE forms a side of the processor complex. These sides are referred to as side A and side B. Each SCE communicates directly with its own central and expanded storage, subchannels, and CPs, as well as with the other SCE (if another exists). Central storage is composed of up to four processor memory arrays (PMA), with a maximum of two being controlled by one SCE. All accesses to resources on the other side (i.e., a CP access to a resource that is controlled by the other SCE) will go through both SCEs, causing a slight delay. Expanded storage is composed of up to four expanded storage arrays, again with a maximum of two being controlled by one SCE. Expanded storage is a solid-state memory array that can only be addressed in 4-kilobyte

sections, called pages. Pages are transferred between expanded storage and central storage in a synchronous manner with respect to the CP. Expanded storage has no hardware-designed functions, but operating systems may use it as a paging device. The expanded storage is addressed through the use of special instructions. These instructions can only be issued by authorized programs, and in the evaluated system are only issued by the Real Storage Manager and the Auxiliary Storage Manager.

The hardware controller for the model 3090 machines is the Processor Control Element (PCE). On the 400, 500, and 600 series machines, and on certain 2X0 and 3X0 models, the PCE has two distinct sides which can either act as a primary-backup pair or a primary-primary pair. Each of these sides has a directly attached 3370 DASD unit, which is used to contain information about the system. When in primary-backup mode, this information is shadowed from the primary to the backup. When in primary-primary mode, no shadowing is performed and the two sides (comprised of SCEs, storage, CPs, etc.) are disjoint.

The PCE is a separate physical unit communicating with the 3090 through lines from PCE-based Logic Support Adaptors (LSA) to 3090 component-based Logic Support Stations (LSS). Each LSA controls an LSS. An LSS is contained in every module of the 3090 hardware (e.g., SCE, main memory).

It may come as no surprise that the PCE itself is subject to a monitoring function performed by another system. In this case, a specially tailored processor equipped with optical storage disk may be used to monitor the logic of the PCE. This service processor is enabled by the site to diagnose problems and optionally to establish a link with IBM diagnostics center.

The design of the 3090 allows growth in a number of areas. The system is designed so that there can be a maximum of 256 channels, 64 thousand devices, 16 processors, 2 gigabytes of central memory, and 4 gigabytes of expanded storage. At present, a maximally configured system contains 128 channels, 24 thousand devices, 6 processors, 512 megabytes of central storage, and 4 gigabytes of expanded storage.

2.2.1 Physical Partitioning

Several 3090 models (specifically, 250J and JH, 380J, and models E, S, and J of the 280, 400, 500, and 600) have the ability to be partitioned by the hardware into two distinct sections with no electronic communication between them. Hence, for example, a 400 could be partitioned so that it appeared to those running on it as two 200s, and likewise a 600 would appear as two 300s. Such a mode of operation is called physically partitioned (PP). When this is not in effect, the configuration is referred to as running in single image (SI) mode.

A machine can be put into PP mode (or taken from PP mode to SI mode) only from the system or service consoles, which are attached directly to the PCE (see page 53, "Hardware Consoles"). These are distinct from the operator's console, which is the console from which the operating system is controlled. The partitioning involves many components of the system. The system administrator

Final Evaluation Report IBM MVS/ESA
CHAPTER 2. HARDWARE ARCHITECTURE

selects the CPs, their associated central and expanded storages, channel subsystems, and possibly vector processors. Each side is also provided with its own console device.

When a machine is put into PP mode, the side of the machine to be partitioned must be taken offline. If this is not done, the partitioning operation will not be allowed. Next, the backup side of the PCE will be assigned as primary to the new side. After a power-on reset, the new side is available for use.

The separation is achieved by way of the LSSs. There are four bi-directional buses between the two sides: three between the two SCEs, and one between the two expanded storage controllers (each SCE has its own expanded storage controller, which is responsible for both banks of expanded storage arrays associated with that SCE). In addition to the buses, there are associated control and status lines. When the PP mode is activated, a bit in the LSS is zeroed. This bit is ANDed with all of the lines going between the two sides, including the data ready lines. This makes it appear to each side that the other side does not exist.

A bit in the LSA determines to which side of the PCE it is assigned. When powered up, all LSAs are assigned to one side, and the other side becomes the backup. This is determined by a switch on the front panel. When going to PP mode, the LSAs responsible for the LSSs on the offgoing side are reassigned to the (new) primary side. For instance, if A were the primary and B the backup, when going to PP mode, half of the LSAs would be assigned to the B side from the A side.

There is no way for the 3090 to instruct the PCE side controlling its LSSs to switch them on or off, and one side of the PCE (when in PP mode) cannot instruct the other side to switch its LSAs; this must be done by the side that owns the LSAs.

When the 3090 is physically partitioned into two separate machines, one side may run an evaluated system, while the other may run any kind of system or be idle. Both sides may run copies of the evaluated (MVS/ESA) system, but if DASD devices are shared, then both systems must adhere to the JES complex restrictions described in this chapter.

2.3 Model 4381 Processors

The 4381 series of processors are an evolution of the 4341 processors and are IBM's mid-range family of mainframes. There are both single (models 90E and 91E) and dual (model 92E) processor versions of this machine.

Each processor communicates with a common memory subsystem, while each processor communicates with its own channel subsystem. The two CPs can communicate through a maintenance subsystem. The maximum memory configuration on a 4381 machine is 64 megabytes, and each CP can communicate with a maximum of 12 channels. The channel subsystem is fully compatible with the 3090 channel subsystem, allowing complete interchange of peripherals. With a notable

exception of physical partitioning, which is not supported by the 4381s, the maintenance subsystem functions similarly to the 3090 PCE incorporating the control and service functions into one unit.

2.4 JES Complexes

In addition to the single machine configurations described above, configurations comprising more than one machine are also supported. These configurations are known as JES complexes, or simply as "complexes." In order for complexes to be considered as a valid configuration, some restrictions must be in force. A description of the complex itself and restrictions needed for a B1-evaluated system are provided below.

A complex can be composed of any number of machines (up to seven) running JES2 or JES3, but not both (see page 70, "Job Entry Subsystem"). Complexes included in the evaluated configuration will be composed only of those mainframes listed in Appendix A. All machines will be running instantiations of one copy of the TCB software, which is listed in Appendix B.

Additional hardware for complexes in the evaluated configuration includes Channel-to-Channel adaptors (CTCs), 3088s (described on page 27), and shared DASD. CTCs are devices that allow the JES running on one machine in a complex to communicate with the JES running on another machine in the complex. The CTC appears as an I/O device to each of the JESs, and is addressed as such. The shared DASD is needed for those data sets which must be accessed by more than one machine in the complex. The controller for a shared DASD mediates this sharing.

Machine IDs in the complex are defined to JES. A valid complex must have every machine in the complex defined to JES and to Global Resource Serialization, and all machines must use the same RACF data base (see page 100, "RACF Data Base"). Subsetting is not allowed. All shared data set use must be synchronized under control of the DASD on which these data sets reside. Synchronization mechanisms include RESERVE-DEQ (see page 58, "Serialization Managers") and GRS (see page 58, "Global Resource Serialization").

Interactive users may be able to dynamically select which machine in the complex they will log into. Terminals are defined via VTAM (see page 86, "ACF/Virtual Telecommunications Access Method") to one machine at Initial Program Load; however, VTAM may be set up to support TSO sessions on more than one machine. Batch jobs will be selected off the queue of ready batch jobs when an initiator is ready (see page 49, "Address Space Creation") regardless of the machine the initiator is on. Each of the phases of job execution (see page 78) can run on a different machine in the complex, but a given phase runs to completion on one machine.

In addition to the RACF data base, all JES spool data sets must be shared across the complex, as job phase scheduling is done on a complex-wide basis by the JESs. The checkpoint data set for job restart in case of failure must also be complex-unique. Data sets that cannot be shared among machines in the complex include each machine's paging data set, system log, and dump data set. Each machine in the complex also has its own copy of SMF (see page 57, "System Management

Final Evaluation Report IBM MVS/ESA
CHAPTER 2. HARDWARE ARCHITECTURE

Facilities”) and associated audit data set, and audit data is collected for each machine with the machine ID, time, and user included in the audit record. Facilities exist for merging the audit data sets before the audit reduction tool is used in order to provide a comprehensive audit trail (see page 132, “JES Complex Auditing”).

2.5 Address Space Selection

The MVS/ESA operating system, in combination with the ESA/370 architecture, supports virtual addressing for all address spaces existing on the system. The layout of the address space is discussed in the MVS Overview section of TCB Architecture (page 34). This section describes hardware management of address space switching. The mechanisms described here are used primarily for context switching and establishing new address spaces.

ESA/370 is a continued evolution of the System 370-XA Architecture. The significant characteristics of ESA/370 are the improved capability of handling address spaces and an introduction of data spaces. Programs execute only within address spaces, but they may access data in data spaces.

Each address space in the system is assigned an address space number (ASN). Previously, the system could support up to 65,536 different address spaces. Currently, the ASN is actually derived from an STOKEN (space token) which exists for every address space. An STOKEN is an 8-byte variable allowing for an increased number of address spaces (see page 37 for a complete discussion of STOKEN).

As in System 370-XA, a given processor can address two different address spaces without going through an address space translation process (as opposed to virtual address translation, described later). These two address spaces are called the primary and the secondary. In MVS/ESA, this mechanism is used to provide cross memory services (see page 40, “Cross Memory Services”). The ESA/370 architecture additionally provides a third type of address space: home.

An address space is defined by the segment table (see page 12, “Address Translation”) for that address space; the segment table addresses for the primary, secondary, and home address spaces are located in control registers 1, 7, and 13, respectively. Since the ESA architecture supports 31-bit virtual addressing, the size of each of the address spaces is about 2 gigabytes.

In order to swap an address space, the system uses a number of tables to perform a translation from the 16-bit address space number to the address of the segment table representing that address space. Two tables, the ASN first table and ASN second table, in conjunction with the ASN authority table, are used to perform this translation. Control register 14 contains the address of the ASN first table in bits 13 - 31. The first 10 bits of the ASN are an index into this table; the selected entry gives the address of the ASN second table entry. The final six bits of the ASN are an index into the ASN second table entry.

The ASN second table entries are made up of a number of fields: the address for the authority table, the length of the authority table, an index into the authority table, an address for the segment table for this address space, and linkage table information which goes in control register 5.

The authority table is made up of groups of four 2-bit entries, with the first bit designated as the primary authority (authority to use the address space as a primary address space) and the second bit designated as the secondary authority (authority to use the address space as a secondary address space). Control register 4 contains a 14-bit index into the authority table. Bits 14 and 15 indicate which 2-bit section of the designated authority table entry to use in determining the authority. If the selected bit is 1, then the ASN is authorized; otherwise, it is not authorized. As in all cases where an address space translation encounters a disallowed condition, an interrupt is taken, and the translation and subsequent switch is not completed. The authority table is also used in the extended-authorization process, as part of access-register translation.

2.6 Addressing Modes

The ESA/370 architecture supports four addressing modes. All four modes are controlled by the value of bits 16 and 17 in the current PSW. The essence of two of these (primary-space mode and secondary-space mode) has been given in the discussion of address spaces. ESA/370 augmented those two modes with two new ones: access-register mode and home-space mode. Access-register mode enables the use of access registers in determining addresses for data. These registers contain addresses of segment-table designations. Values of these registers can be set by a program running in problem state. By means of these registers, programs can have concurrent access to sixteen, 2-gigabyte spaces. Furthermore, since these registers actually point to entries within access lists, the total amount of addressable data exceeds 8 terabytes. General purpose registers are subsequently used to identify locations within a space.

An access list, called DU-AL, is built for each dispatchable unit (task and service request). A list can contain up to 256 entries. This list is available to all programs (tasks) represented by and executing within the address space of the current task. The MVS/ESA associates another list, called PASN-AL, with the primary address space. This list is available to all programs executing within the address space which can include operating system code. By convention, the architecture reserves the first three entries in a list to point to the primary space, the secondary space, and the home space. Typically, a user address space has these three entries all set to primary when running a problem state user program.

Home-space mode is used by the control program to efficiently resume execution in its address space when, for example, processing an interrupt and when the control program cannot guarantee that the primary or secondary spaces still point to the interrupt processing logic.

2.7 Address Translation

In order for a virtual address to be converted to an address in real memory, the ESA/370 architecture provides the dynamic address translation (DAT) mechanism. A virtual address space is divided into segments and pages; there are 2048 segments, each with 256 4-kilobyte pages.

To translate a virtual address to a real address, the segment table designation (STD) is obtained from control register 1 (for the primary segment table), 7 (for the secondary), 13 (for the home), or an access register (for the access-register mode). Bits 1-11 of the virtual address form the index into the segment table origin (STO) for the desired page table origin (PTO) address. Bits 12-19 are the offset for the page table entry in the page table, which designates a page frame real address. The final bits (20-31) of the virtual address give the offset of the desired address in the page frame. Each of the page table entries has an invalid bit, used to detect page faults, and a protection bit, which is described below. Figure 2.1 presents a simplified diagram of the dynamic address translation.

Another process takes place when performing access-register translation (ART). This process occurs before DAT can be performed. The access-list designation (ALD), based on control register 2 (for DU-ALs) and control register 5 (for PASN-ALs), is used to locate the access-list origin for the currently executing task or address space. The access list entry token (ALET, which is in the specified access register) can have two special values. A zero will always result in the ART using control register 1 and the primary STD. A one will always result in the ART using control register 7 and the secondary STD. For all other values, an access list entry (ALE) number taken from the ALET is used to compute the offset into the access list. The entry thus found contains three important fields: the private bit, the access list entry authorization index, and the ASN-second-table entry.

The private bit (P), when off, allows access to any task or address space. The access list entry authorization index (ALEAX) is compared against the control-program assigned extended authorization index (EAX) found in control register 8. If they are equal, the ASN-second-table entry (ASTE) address is used to extract the segment table designation (STD) which is then supplied to DAT. Figure 2.2 presents a simplified diagram of ART when ALET is 0 or 1, or when ALEAX and EAX are equal.

If the ALEAX and the EAX do not match, the ASTE and the EAX are used to extract the authority table entry associated with the address space specified in the ALE. If that entry allows access, the ASTE address is again used to extract the STD which is then passed to DAT. In this way, programs executing with dissimilar EAXs (and potentially out of different address spaces) can all use a single ALE to access a particular address space. Figure 2.3 presents a simplified diagram of ART when ALEAX and EAX are not equal, and the authority table (AT) allows access.

Each CP has associated with it a translation-lookaside buffer (TLB). The purpose of this buffer is to speed up DAT. The size and implementation of the TLB is model dependent. When a virtual address is translated (using DAT) to a real address, all the information used in this translation such as all the applicable translation-table entries, current translation parameters, address-space

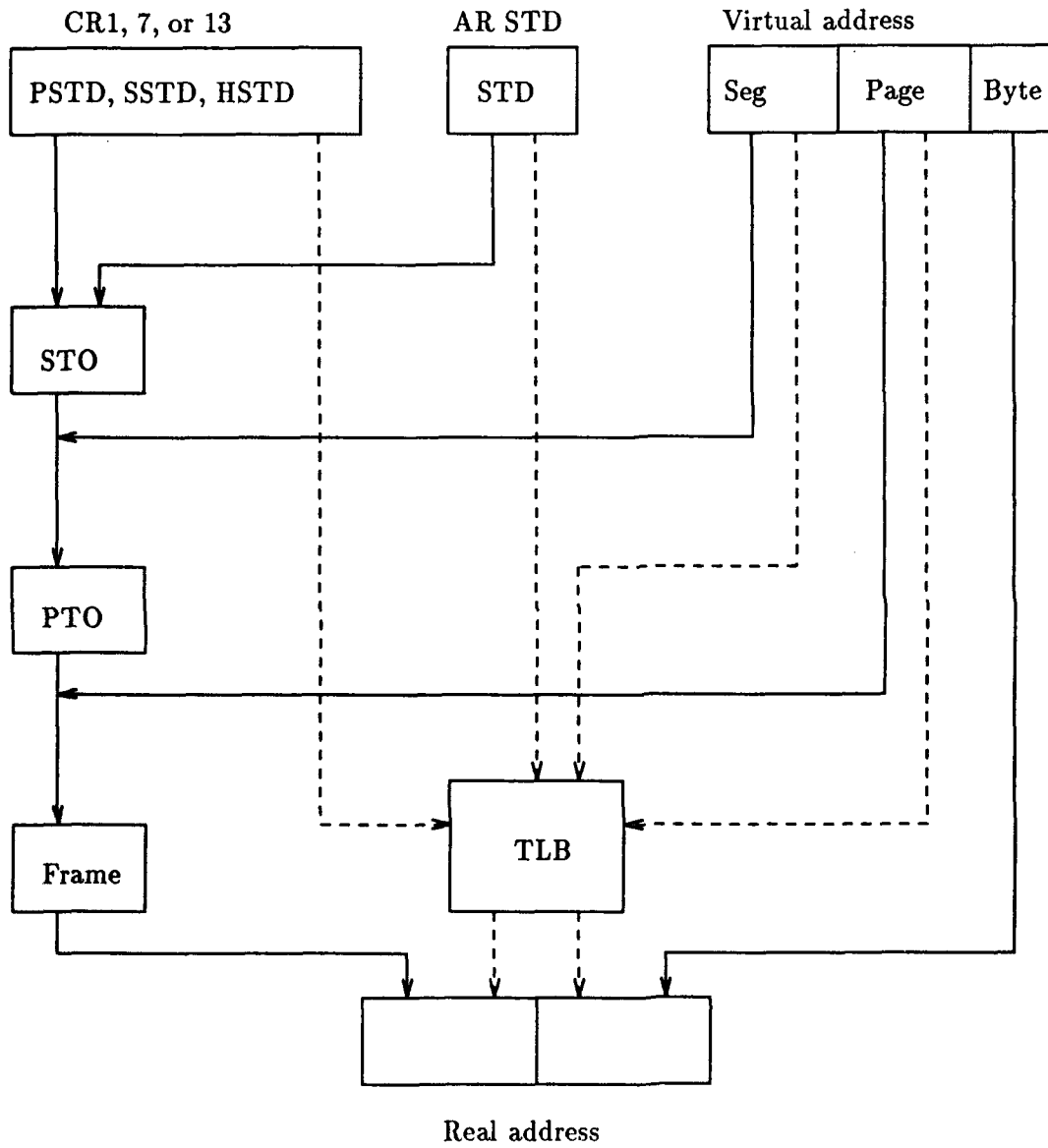


Figure 2.1. Dynamic Address Translation

Final Evaluation Report IBM MVS/ESA
 CHAPTER 2. HARDWARE ARCHITECTURE

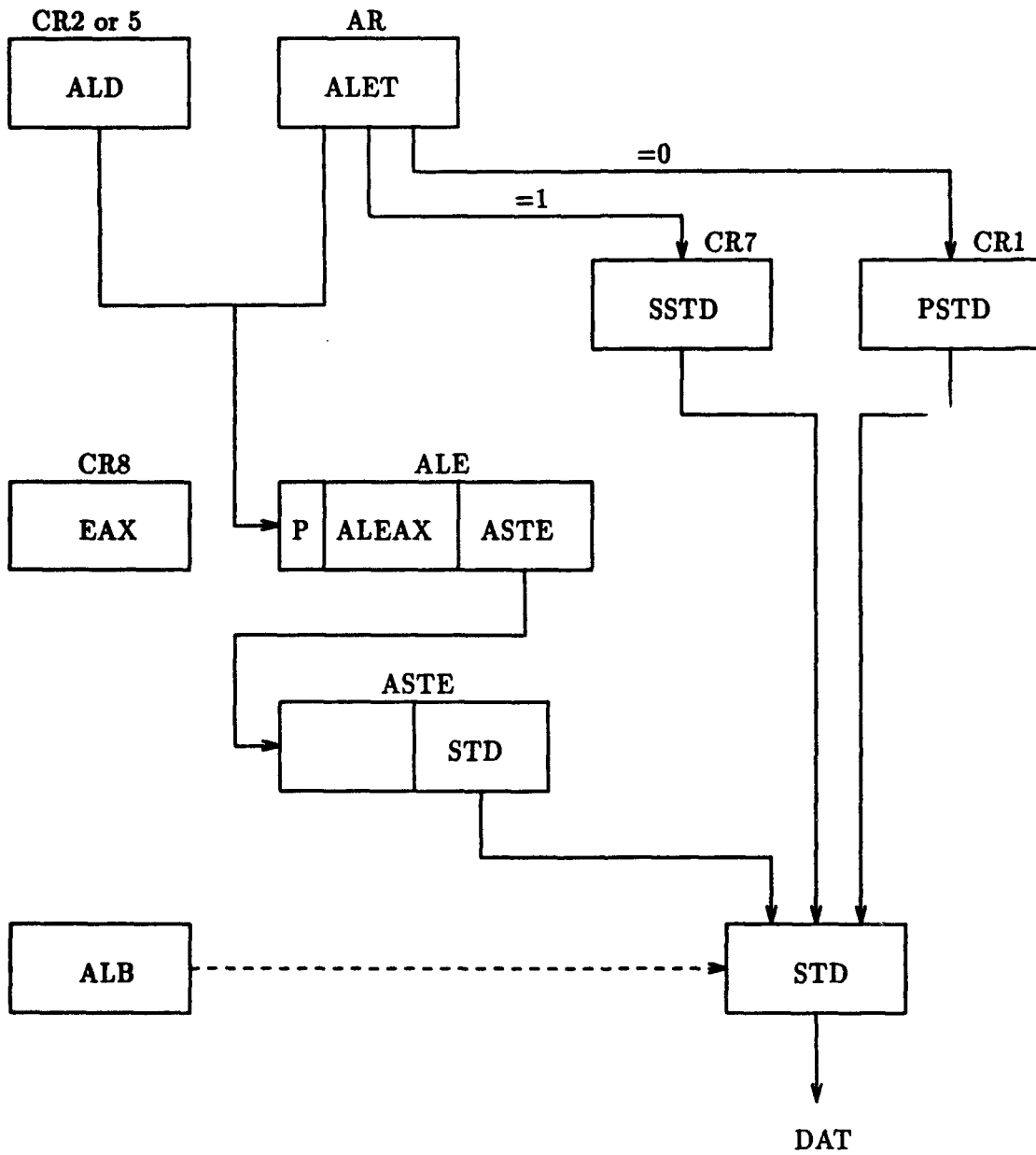


Figure 2.2. Access Register Translation, Case 1

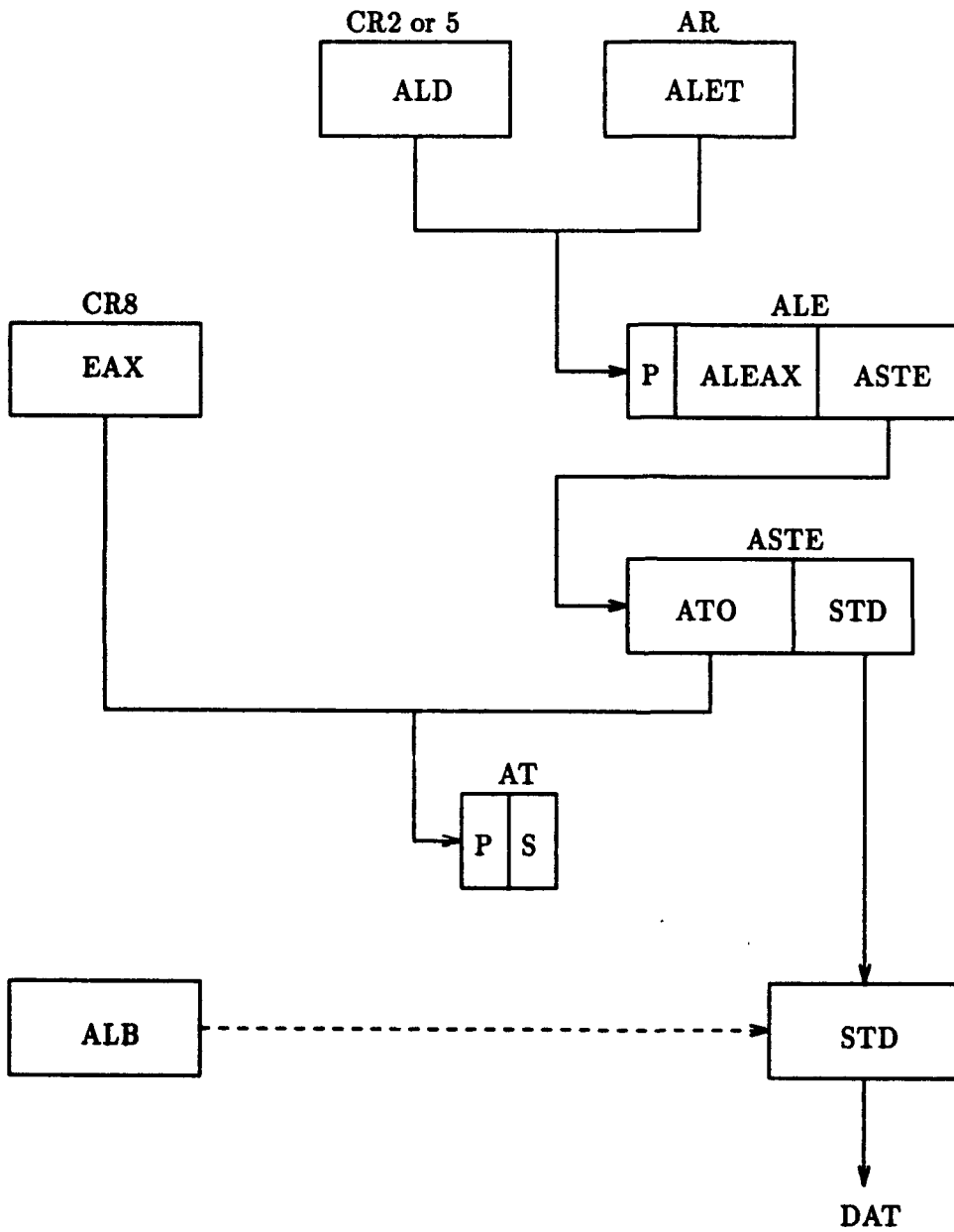


Figure 2.3. Access Register Translation, Case 2

Final Evaluation Report IBM MVS/ESA
CHAPTER 2. HARDWARE ARCHITECTURE

control bits, and the contents of the access registers, and including protection information such as the page-protection bit from the page table entry, is conceptually placed in this buffer. When the page is again referenced, the TLB can be used to locate the page in a much faster manner than with the normal DAT.

To further enhance performance, the ART mechanism employs for each CP special storage called ART-lookaside buffer (ALB). This buffer contains access-list designations and information specified in access lists, ASN second tables, and authority tables. Rules for placing information in the TLB and the ALB are complex, model-dependent, and are treated fully in the *IBM Enterprise Systems Architecture/370 Principles of Operation*.

2.8 Prefixing

In multi-CP systems, the different CPs sometimes need to access common low storage (address locations 0-4095) values (e.g., interrupt vectors) at the same time. To make this operation more efficient, the ESA/370 architecture includes the concept of a prefixed save area, which is the first 4 kilobytes of real storage. In the following discussion, absolute addresses are those addresses that are assigned to each physical memory location. Real addresses are those addresses which are used to access memory (i.e., a result of virtual address translation), and correspond exactly to absolute addresses except as outlined below.

Each processor has a prefix register, which is added to real addresses 0 through 4095 to obtain the absolute address. This allows a processor to access its own prefixed save area. Except for these two blocks of real addresses, all other real addresses correspond to the absolute addresses.

2.9 Hardware Protection Mechanisms

The ESA/370 architecture and the MVS/ESA operating system provide three separate protection mechanisms for data in main storage: key-controlled protection, page protection, and low-storage protection. In order for a process to access any location in storage, the checks from all three of these mechanisms must allow the access. These checks are performed after address translation (i.e., they are based on physical memory locations).

2.9.1 Key-Controlled Protection

Key-controlled protection provides protection against unauthorized storing, or unauthorized storing and fetching for locations in main memory. Memory locations cannot be fetch protected only (i.e., cannot be protected from reading while allowing writing). Each 4-kilobyte page of real storage has associated with it a 7-bit storage protect key composed of four access control bits (the protect key),

a fetch-protect bit, a reference bit, and a change bit. These keys, while associated with every page of real storage, are not a part of addressable storage, and are set with privileged instructions (see page 18, "Privileged and Semi-Privileged Instructions").

Whenever a program attempts to access a storage location, a comparison is made between the protect key on the page in storage and an access key that the program possesses. This access key is located in bits 8-11 of the PSW for programs running on a CP, and in the operation request block (see page 47, "I/O Supervisor") for channel programs. If the keys match, then the requested access is allowed. If the keys do not match, a protection exception interrupt occurs and the instruction is terminated. This mechanism is applied to both supervisor state and problem state programs.

There are, however, two exceptions to these rules. Read access can be obtained when the fetch-protect bit is not set, even if the access key and the protect key do not match. In addition, all accesses are allowed if the access key of the requesting program is key 0.

There are several instances when key-controlled protection is not active. They are:

- Processing a (hardware) interrupt.
- Fetching page table entries for DAT or ASN translation.
- Tracing program execution.
- During a store-status function (used to store register and timer values to low storage).
- During initial program load and CPU logout.
- During system console operator functions.

The following are the key assignments for the system. Those marked with an asterisk (*) are used in products found in the evaluated configuration.

- Key 0*: used for the MVS/ESA control program.
- Key 1*: used for the Job Scheduler and Job Entry Subsystems.
- Key 2: used for VSPC (virtual storage personal computing).
- Key 3,4: reserved.
- Key 5*: used for data management (DFP).
- Key 6*: used for TCAM/VTAM (terminal access to system).
- Key 7: used for IMS.
- Key 8: the normal user key. All users running in virtual address spaces run with key 8.
- Keys 9-15: used when a user process must run in V=R mode (see page 34, "Address Spaces").

2.9.2 Page Protection

Page protection controls the storing of data in virtual memory. This is done by means of bit 22 in each page table entry. If the bit is set, the page is read-only. If the bit is not set, read and write access are both allowed. Attempted writes to page protected portions of memory cause a protection exception interrupt to occur and the write does not take place. The page tables themselves are protected by the key-controlled protection mechanism.

2.9.3 Low-Address Protection

The low-address protection mechanism prevents code from modifying certain critical locations in low-storage which contain information for the processing of exceptions and interrupts. If the low-address-protection-control bit (bit 3 in control register 0) is set, low address protection is in effect and any attempted write access to the protected memory location will cause a protection exception interrupt to occur and the instruction to be terminated. Low-address protection does not apply to the CP or the channels when they are processing an interrupt (e.g., when storing the old PSW).

2.10 Privileged and Semi-Privileged Instructions

There are two types of security-relevant machine instructions for the ESA/370 architecture: privileged and semi-privileged. Privileged instructions are those instructions that can only be issued while the CP is in supervisor state. Semi-privileged instructions can be issued from either state, but have other restrictions applying to their use in the problem state. The I/O instructions are also privileged.

The following list contains the privileged instruction mnemonics and a brief description of each.

- DIAGNOSE This instruction has no mnemonic. Its arguments are a code that has different meaning to different models in the product line.
- ISKE Insert Storage Key Extended. Put the storage key for the designated block of real storage into a general register.
- IPTE Invalidate Page Table Entry. Invalidate the designated page table entry, and clear associated entries in the TLBs of all CPs.

2.10. PRIVILEGED AND SEMI-PRIVILEGED INSTRUCTIONS

LASP	Load Address Space Parameters. This instruction loads various values used in address space manipulations. The four operations performed by this instruction include primary ASN translation, secondary ASN translation, secondary ASN authorization, and control-register loading (can affect registers 1, 3, 4, 5, and 7).
LCTL	Load Control. This instruction controls loading of any contiguous block of control registers.
LPSW	Load PSW. Replaces the current PSW with the specified PSW.
LRA	Load Real Address. Loads the real address corresponding to the designated virtual address into the specified general register. A forced DAT is performed using a segment-table designation.
LURA	Load Using Real Address. Loads the real address without dynamic address translation.
PALB	Purge AR-Lookaside Buffer. Purges the access register translation lookaside buffer for the issuing CP.
PTLB	Purge TLB. This instruction purges the TLB of all entries for the issuing CP.
RRBE	Reset Reference Bit Extended. This instruction sets the reference bit in the storage key for the designated page to zero.
SCK	Set Clock. This instruction sets the Time-of-Day (TOD) clock to the specified value, and stops the clock. Starting the clock is dependent upon a bit in control register 0 being set.
SCKC	Set Clock Comparator. Set the clock comparator to the designated value.
SPT	Set CP Timer. Set the value of the CP timer to the designated value.

Final Evaluation Report IBM MVS/ESA
CHAPTER 2. HARDWARE ARCHITECTURE

SPX	Set Prefix. This instruction sets the prefix register of the issuing CP to the designated value, and has the side effect of clearing that CP's TLB.
SSKE	Set Storage Key Extended. This instruction sets the value of the storage key of the designated page to the value specified.
SSM	Set System Mask. The system mask refers to bits 0-7 of the PSW; this instruction sets those bits to the designated value.
SIGP	Signal Processor. Send an 8-bit order code, and possibly a 32-bit parameter to the designated CP.
SIE	Start Interpretive Execution. The CP is placed in the interpretive-execution mode and performs the functions of the interpreted machine.
STCKC	Store Clock Comparator. This instruction reads the clock comparator, and stores the value at the designated address.
STCTL	Store Control. This instruction reads the values of the designated contiguous set of control registers, and stores these values at the designated address.
STAP	Store CP Address. Every CP in a multi-processor configuration is identified by an address. This instruction stores the address for the issuing CP at the address specified.
STIDP	Store CP ID. The ID of an individual CP consists of a version code, ID number, and a model number. This information is stored at the designated address.
STPT	Store CP Timer. This instruction reads the value of the CP timer into the designated address.
STPX	Store Prefix. This instruction reads the value of the issuing CP's prefix register into the address specified.

2.10. PRIVILEGED AND SEMI-PRIVILEGED INSTRUCTIONS

STNSM	Store Then AND System Mask. This instruction saves bits 0-7 of the PSW in a specified address, logically ANDs bits 0-7 of the PSW with the second operand, and stores the result back to bits 0-7 in the PSW.
STOSM	Store Then OR System Mask. Same as above, except a logical OR is performed.
STURA	Store Using Real Address. Store without using dynamic address translation.
TB	Test Block. Tests the usability of the locations and the storage key associated with the designated 4-kilobyte block (page) of storage.
TPROT	Test Protection. This instruction tests the designated address for any protection exception that would occur with the access key given in the second operand.
TRACE	Trace. This instruction is used to form trace entries when tracing is turned on.

The following list gives the privileged I/O instruction mnemonics and a brief description of each.

CSCH	Clear Subchannel. This instruction clears the designated subchannel, and signals the channel subsystem (asynchronously) to perform a clear function at the associated devices.
HSCH	Halt Subchannel. This instruction signals the channel subsystem to terminate the current start function at the designated subchannel and associated devices.
MSCH	Modify Subchannel. This instruction causes the information in the subchannel information block to be placed into the appropriate program-modifiable fields of the subchannel itself. These fields influence clear, halt, resume, and start functions of the subchannel, as well as certain I/O support functions.
RCHP	Reset Channel Path. This instruction signals the channel-path-reset facility to perform a reset on the designated channel path.

Final Evaluation Report IBM MVS/ESA
CHAPTER 2. HARDWARE ARCHITECTURE

RSCH	Resume Subchannel. This instructions signals the channel subsystem to resume operations at the designated subchannel.
SAL	Set Address Limit. This instruction passes an address limit to the address-limit-checking facility for use in checking addresses for an out-of-bounds condition.
SCHM	Set Channel Monitor. This instruction sets monitoring modes of the channel subsystem as active or inactive, depending on the contents of general register 1.
SSCH	Start Subchannel. This instruction places the contents of the Operations Request Block (see page 47, "I/O Supervisor") in the subchannel and signals the subchannel to perform the start function.
STCPS	Store Channel Path Status. This instruction places up to 256 bits of information which reflects the active channel paths for that subchannel into the designated location.
STCRW	Store Channel Report Word. This instruction places a channel report word, which contains information about conditions affecting the channel subsystem (e.g., malfunction), into the designated location.
STSCH	Store Subchannel. This instruction places control and status information from the designated subchannel into a subchannel information block.
TPI	Test Pending Interruption. This instruction stores the code for a pending interruption at the subchannel in the designated location, and clears the interrupt request.
TSCH	Test Subchannel. This instruction stores control and status information from the designated subchannel into an interrupt request block.

The following table lists the mnemonic, function and privilege(s) needed for the semi-privileged instructions.

2.10. PRIVILEGED AND SEMI-PRIVILEGED INSTRUCTIONS

- EPAR** Extract Primary ASN. The primary ASN is placed in the designated general purpose register. Bit 4 of control register 0 (the extraction authority control) must be 1 in the problem state.
- ESAR** Extract Secondary ASN. The secondary ASN is placed in the designated general purpose register. Bit 4 of control register 0 (the extraction authority control) must be 1 in the problem state.
- IAC** Insert Address Space Control. Bit 16 of the PSW designates which of the address spaces (primary or secondary) will be used for DAT. This instruction places the bit in the designated general purpose register. Bit 4 of control register 0 (the extraction authority control) must be 1 in the problem state.
- IPK** Insert PSW Key. This instruction inserts the PSW access key (bits 8-11) into the designated general purpose register. Bit 4 of control register 0 (the extraction authority control) must be 1 in the problem state.
- IVSK** Insert Virtual Storage Key. This instruction puts the storage key for the location designated by the virtual address into the specified general purpose register. Bit 4 of control register 0 (the extraction authority control) must be 1 in the problem state.
- MVCP** Move to Primary and Move to Secondary. **MVCS** moves the data at the specified address in the secondary address space to the designated address in the primary address space. **MVCS** moves the data from the primary address space to the secondary address space. The operand in the secondary address space is verified using the key value specified in the instruction. In problem state, the bit in the PSW key mask (in control register 3) corresponding to the specified key must be one or access is not allowed. The operand in the primary address space is verified using the key value in the PSW. Also, bit 5 of control register 0 (secondary space control) must be set.

Final Evaluation Report IBM MVS/ESA
CHAPTER 2. HARDWARE ARCHITECTURE

- MVCDK Move with Destination Key and Move with Source
MVCSK Key. The accesses to the destination or source location are performed using the specified key. In problem state, the bit in the PSW key mask (in control register 3) corresponding to the specified key must be one or access is not allowed. The other operand is verified using the key value in the PSW.
- MVCK Move with Key. This instruction performs a move from one memory address to another. The memory address of the location the data is being moved from is checked with a key that is specified in the instruction; the memory location that the data is being moved to is checked against the key value in the PSW. In problem state, the bit in the PSW key mask (in control register 3) corresponding to the specified key must be one or access is not allowed.
- PC Program Call. This instruction is used to transfer execution to another location in either the current address space or a different address space. Transfer is controlled by the entry table. The entry table contains a field which, when the CP is in problem state, is checked against the current PSW-key mask to assure authorization for the program making the call. Various control bits must also be set before this instruction succeeds.
- PT Program Transfer. This instruction is similar in effect to PC above, but takes values for the various registers needed for a context switch from the two operand registers. If a switch from problem state to supervisor state (changing bit 15 of the PSW from one to zero) occurs, an exception results. For space switching forms of this request, the authorization codes (described above) must match for successful execution.
- SAC Set Address Space Control. This instruction sets the address space (primary, secondary, access register, or home) to be used. Bit 5 of control register 0 must be set, and DAT must be on for this instruction to successfully execute.

- SPKA** Set PSW Key from Address. Bits 8-11 of the PSW are replaced by bits 24-27 of the designated address. In problem state, the bit in the PSW key mask (in control register 3) corresponding to the specified key must be one or the PSW is not changed.
- SSAR** Set Secondary ASR. This instruction sets the secondary address space to the address space designated by the ASR given in the operand. Bit 12 of control register 14 must be set, and DAT must be on, for this instruction to successfully execute in problem state.

2.11 Input/Output

The I/O in ESA/370 architecture machines is performed at the hardware level by the channel subsystem, the control units, and the I/O devices. A description of a full I/O operation, including the use of Channel Command Words (CCWs), is presented later (page 44, "I/O Operations"). In addition to this "traditional" I/O method, the system also supports Virtual I/O, which is described in the MVS discussion (page 48, "Virtual Input/Output").

2.11.1 The Channel Subsystem

The channel subsystem directs communication between the I/O devices and main storage. Figure 2.4 depicts a sample hardware arrangement: the CUs represent control units, and the circles represent devices. The channel subsystem mediates communications between the I/O devices and storage, freeing the CP so that I/O and data processing can take place concurrently. The subsystem is composed of different "subchannels." There can be up to 64 thousand subchannels, and each subchannel is uniquely associated with a single device. A device, however, may be associated with a number of different subchannels. A subchannel is the addressable unit designated by a program in initiating I/O operations.

Subchannels are the logical abstraction of the hardware Channel Control Elements (CCE) and the actual channels. There can be up to two CCEs per machine, depending on the model. A CCE can control up to 16 Channel Elements, each of which has four physical channels. A maximally configured 3090 model 600 can contain 128 channels. Technically, a channel subsystem is one CCE with its associated channels, but in machines with two CCEs that are not partitioned, the two channel subsystems coordinate activity and appear to the control program as one dynamic channel subsystem. The CCE is the physical unit that implements the subchannel, holding all information necessary to define that subchannel.

Final Evaluation Report IBM MVS/ESA
CHAPTER 2. HARDWARE ARCHITECTURE

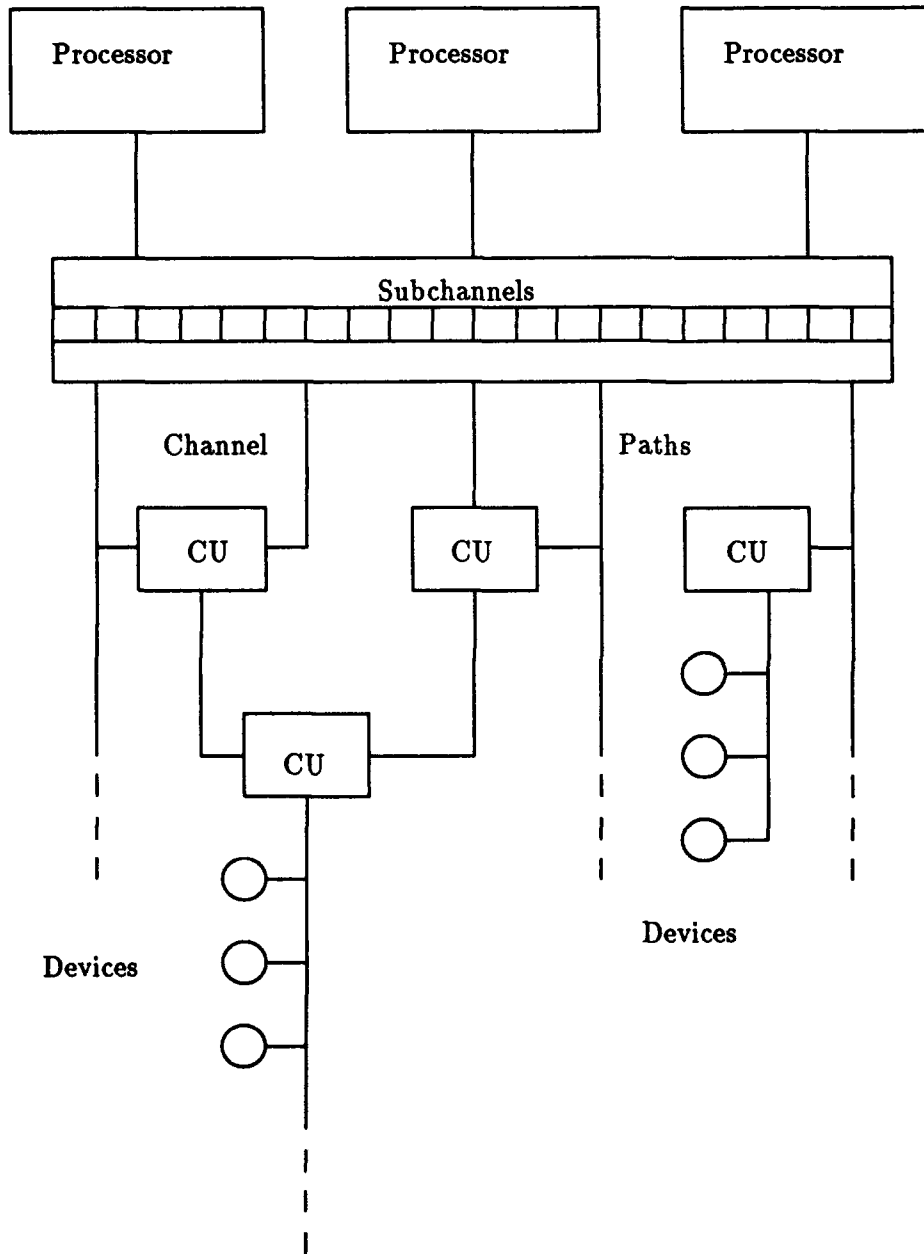


Figure 2.4. Channel and Device Configuration

There are two types of channels: byte multiplexer and block multiplexer. Byte multiplexer channels can be shared by many devices, or dedicated to one device. For 4381 machines, there can be a maximum of two byte multiplexer channels. On the 3090 series, the 400 and 600 models can have up to eight channels configured as byte multiplexers, while other models can only have a maximum of four.

Block multiplexer channels are slightly different for 4381 machines and 3090 machines. For 4381s, there are two modes for block multiplexers: block multiplexer mode and selector mode. Block multiplexer mode allows more than one high-speed I/O device to use the channel, while selector mode restricts the channel to one I/O device until the I/O operation is complete. 3090 block multiplexer channels also have two modes: interlocked and data streaming. Both modes are similar to the 4381 selector mode, in that the device and channel are connected throughout the life of the I/O operation. Interlocked mode uses interlocking data transfer signals between the channel and the control unit, while the data streaming mode does not. Consequently the data streaming mode is faster than the interlock mode (4.5 megabytes per second, as opposed to 3 megabytes per second).

The channel subsystem contains the necessary logic and storage to direct the I/O between the device and main storage. To mediate the transfer, the subsystem uses control units, which lie between the subsystem and the device, and channel paths to these control units. An I/O device can be connected to up to eight control units; the control units in turn are connected via channel paths to the subsystem. Depending on the model, there can be more than one channel path to each of the control units. The maximum number of different channel paths that can be addressed by the control unit is 256. The control unit is responsible for converting the generic request issued by the subchannel to a request that can be recognized by the device. When an I/O request is given, an available channel path is chosen. This means that two successive reads from the same data set need not take the same channel path to the device, although the subchannel number would be identical for both reads.

The hardware, through the 3088 Multisystem Channel Communication Unit, allows for direct connection of channels from one machine to another. The 3088 is a stand-alone I/O controller designed to interconnect processor channels. It is an improvement over the CTCs in that one 3088 has the capability to completely interconnect eight machines. There are three models of the 3088: Model A1 interconnects up to two channels, Model 1 interconnects up to four channels, and Model 2 up to eight. All three models support data streaming mode and block multiplex mode. Internally, the device performs dynamic channel connections as channels become available (not busy). This is so because all the channels (up to eight) use one of only two available high speed data buses.

2.11.2 Control Units and Devices

Many different devices are supported in the evaluated configuration. The specific model numbers are listed in Appendix A, but general architectural details of the "intelligent" control units and

Final Evaluation Report IBM MVS/ESA
CHAPTER 2. HARDWARE ARCHITECTURE

devices are discussed in this section. The control units are identified to the system via 2-byte control unit addresses. The system uses these addresses as I/O addresses.

DASD Controllers

There are two main models of DASD controllers in the evaluated configuration: the 3880 and the 3990. The 3880 is the older of the two models. The model 3880-3 can have two paths to a DASD, meaning it can support two "strings" (strings of DASD are described below). The 3880-23 is identical to the 3880-3, except that it contains a cache to buffer data coming off of the drive. The 3880-21 is only used for paging and must be connected to the 3350 model DASD. Since the 3880 has two paths, two of these controllers can be cross connected with two strings of DASD, allowing two paths to the same string from different control units. This redundancy increases the reliability of the system.

The 3990-1 is a two path version of this model line, while the -2 and -3 support four paths to DASD. In addition, the 3990-3 can have a cache of up to 256 megabytes, which decreases the average access time from 22-32 ms to 3-5 ms. All of the 3990 controllers can be cross-configured as detailed in the 3880 discussion.

DASD Disks

The 3380 DASD devices in the evaluated system are divided into 2 families: the D/E family, capable of supporting two paths per string; and the J/K family, capable of supporting four paths per string (with the 3990 controller). Each DASD has two disks per cabinet. D and J type 3380s can store 1.25 gigabytes per disk, while the E stores 2.5 gigabytes and the K stores 3.75 gigabytes per disk. The J/K family also has a shorter seek time than the D/E family.

A string of DASD consists of a head of string device (designated by the letter A), and "dumb" devices (designated by the letter B). A head of string contains the intelligence needed to perform disk operations and error detection and correction logic for itself and all B devices in the string. There is a maximum of three B devices per A device, and the strings must be all of one type (i.e., all D/E or all J/K type). There also exists a DASD which does not require a controller; this is designated by the letter C.

The 3390 DASD devices all support four paths per string. A string of DASD consists of a head of string device (again designated by the letter A), and "dumb" devices (designated by the letter B). A head of string of a 3390 performs functions similar to that of the head of string of a 3380. There can be up to seven B devices per A device. Both 3390s and 3380s can be strung off an A device.

A devices have either two or four disks per cabinet, while Bs have either two, four, or six disks. Each disk contains two devices (two actuators and two volumes). Data densities vary from 1.89 to

3.75 gigabytes per device. A fully configured string of 3390 contains 32 devices and 60.5 gigabytes of data.

The 3350 DASD devices are an older model of disks dating back more than a decade. As such their performance specifications are far less than those of the 3390s. However, like the 3380s, the 3350s are channel attached through the 3880. There are three types of 3350s: models A2, B2, and C2. A2 and C2 contain a controller and two disks. B2 contains only two disks.

The 3350s are configured in strings with a controller (an A2 device) and up to eight disk devices. An alternate controller (a C2 device) can be added to the end of the string. Only one controller is active at any given time, and a manual switch over is required to utilize the other. All the models are also available with fixed heads for use as fast paging devices. These are designated as A2F, B2F, and C2F.

Magnetic Tape Controllers and Devices

The evaluated line of tape devices is not as extensive as the DASD line. There are two controllers: the 3480-A11 and the 3480-A22, which control the 3480-B11 and the 3480-B22, respectively. The 3480 devices utilize 18 track, 38,000 BPI, 200 megabytes capacity cassettes.

Tape drives 3490 models A01, A02, B04, D31, and D32 are architecturally equivalent to the 3480s. Packaging has been improved.

The 3422 tape subsystem features a conventional 10.5 inch reel device. There are two models of the 3422. Model A01 contains a controller and one drive. Up to seven additional drives (model B01) can be attached to the controller. The 3422 subsystem attaches to the channel subsystem. The connection can operate in the data streaming mode. Two channel attachment is possible providing redundancy or flexibility of sharing the subsystem between two machines.

Terminal Controllers and Terminals

The 3274 control units manage the transfer of data to and from terminals attached to them. The 3274s are attached directly to channels of the host data processing system. The control units are customized at the site to support a selected configuration. The 3174 control units are newer versions of 3274 possessing similar characteristics.

Physically, the 3274 and the 3174 control units are small, floor-standing devices. Each one is capable of controlling of up to 32 locally attached display stations (terminals). The major differences between models are in the amount of memory the units possess and the attachment method. Only controllers utilizing the local attachment are evaluated. This includes 3274 Models 41A and 41D, and 3174 Models 1L and 11L. All these models connect to the system through a byte or a block multiplexer channel, or through a selector channel. All models partition their storage among the terminals they are controlling.

Final Evaluation Report IBM MVS/ESA
CHAPTER 2. HARDWARE ARCHITECTURE

The 3178 and the 3278 display stations are IBM's monochrome terminals. Both offer a variety of screen formats and feature many display characteristics. The 3179 and the 3279 display stations are the color equivalent of the 3178 and the 3278. Again, various models are offered. All the terminals attach to the control units (3174 and 3274) via terminal adaptors.

Printers

There are three printers capable of supporting system-labeled output: the 3825, the 3827, and the 3835. All three have the following characteristics: laser printing mechanism, all points addressable mode of operation, one or two channel connection to the mainframe, built-in control unit, 2-4 megabytes of local storage, and some form of an operator console. Printers with two channel paths can be connected to more than one system in a complex. All three disallow retrieval of data located in the printer.

The 3825 console consists of a touch-sensitive display. It displays device status and messages, and allows for job cancellation for the normal mode of operation. The authorized mode is used for printer configuration, setting date and time, defining paper forms, and performing certain diagnostics. The customer engineer mode is used for hardware diagnostics.

The 3827 and 3835 consoles consist of a display, a keyboard, and a diskette drive. Their functions are similar to that of the 3825. Printer configurations can be stored and retrieved from the diskette.

2.12 Interrupt Handling

ESA/370 machines support six different types of machine interrupts: external, machine check, I/O, program, restart, and supervisor call (SVC). Whenever an interrupt is encountered, the current PSW is saved and a new PSW is loaded from a known memory location (the specific memory location is dependent upon the type of interrupt encountered). There are accompanying interruption codes for many of the interrupts, and in some cases there are also addresses for a parameter block.

The new PSW points to a first level interrupt handler (FLIH). The function of the handler depends upon the interrupt generated. The FLIH will save status associated with the job, then check any needed authorities (in the case of SVCs), and pass control to the appropriate control program after enabling interrupts. After the control routine has performed the action required by the interrupt, either control is returned to the routine that was executing when the interrupt occurred (non-preemptive), or the system dispatcher gets control (see page 39, "Dispatcher") and the highest priority ready unit of work is dispatched. All service request blocks (SRBs) are non-preemptive, and task control blocks (TCBs) are non-preemptive if the SVC associated with the control block is non-preemptive. These units of work (SRBs and TCBs) are further described on page 39.

External interrupts are generated by conditions from either inside or outside the system. Interrupts from outside the system may reach the CP only via hardware connections. Examples of events

Final Evaluation Report IBM MVS/ESA
2.12. INTERRUPT HANDLING

causing external interrupts are the operator pushing the Interrupt key at the console, a CP in a multi-CP configuration losing power, and the Time of Day clock being in an error or non-operational state. Interrupts from inside the system are generated when clock events occur.

Machine check interrupts are generated when equipment malfunction is recognized by the system itself. Various codes indicate the severity and location of the component that caused the interrupt.

I/O interrupts are generated by I/O devices requesting service from the CP. In order for I/O interrupts to be serviced, the CP must have enabled interrupts for that device.

Program interrupts are generated when an execution of a program attempts to perform some action that the system does not allow. Examples of the types of actions that can cause program interrupts are attempts to access unauthorized memory locations, attempts to execute privileged instructions, and arithmetic errors (divide by zero, etc.).

Restart interrupts are generated when the operator requests a system restart. In a multi-CP environment a restart interrupt can also be initiated by one CP sending a SIGP instruction to another CP.

Supervisor Call interrupts are generated when a SVC instruction is executed. Bits 8-15 of the SVC instruction contain the number (0-139) of the SVC that the program is requesting (see page 59, "SVCs in MVS/ESA").

All external, I/O, and machine check interrupts are maskable, as are some program check interrupts. All others are non-maskable, and try to execute immediately. Priority for interrupts is: SVC, program check, repressible machine check (irrepressible machine checks execute immediately, as they usually signal a catastrophic failure), external, I/O, and restart.

**Final Evaluation Report IBM MVS/ESA
CHAPTER 2. HARDWARE ARCHITECTURE**

This page intentionally left blank

Chapter 3

Software Architecture

The system under evaluation contains a number of TCB components, of which MVS/SP is only a part. The components under evaluation are MVS/SP, JES2, JES3, DFP, ACF/VTAM, TSO/E, PSF, and RACF. Specific information on the release numbers of these components can be found on page 165, "Evaluated Software Components."

The MVS/SP operating system is a very large program. It is the main controller for the system under evaluation, and has many different modules. It is the controller for the rest of the products.

Job Entry Subsystem 2 (JES2) and Job Entry Subsystem 3 (JES3) prepare jobs to be executed by MVS/SP by obtaining the resources necessary to execute the job. After completion of the job, JES releases those resources back to the system and prepares output, if any. A BI system may include either JES2 or JES3.

The Data Facility Product (DFP) handles all I/O processing for MVS/SP and its components. DFP is invoked by all users, including MVS routines, to perform data set functions (e.g., open, close, allocate, copy, erase, etc.). It calls RACF to validate access authority and the I/O supervisor in MVS to create the channel programs to perform the physical reads and writes.

Access Communications Function for the Virtual Telecommunications Access Method (ACF/VTAM) is the Virtual Telecommunications Access Method that is used to communicate via terminals with the host hardware.

Time Sharing Option Extensions (TSO/E) provides users with an interactive command environment that interfaces directly with MVS/SP and its subsystems. Users can also submit batch jobs from the TSO/E session.

The Print Services Facility (PSF) is the IBM print driver program that supports page printers attached to MVS/SP. PSF provides the capability to print security labels on hardcopy output and to audit any attempt by a user to override this labeling.

The Resource Access Control Facility (RACF) provides security functions for MVS/ESA. RACF provides identification and authentication, discretionary and mandatory access control, and accountability mechanisms.

These eight products are described in detail in the following sections.

3.1 MVS/SP

The MVS/ESA Operating System contains many large modules. The following subsection describes the important and security relevant modules, as well as important data structures and facilities. It will also discuss concepts and mechanisms new to MVS/SP for the ESA/370 architecture.

3.1.1 Address Spaces

As the name implies, the key to the system is many different virtual address spaces. The following description of address spaces also introduces some very important functional components. Conceptually, an MVS/ESA address space consists of 2 gigabytes of virtual storage. An MVS/ESA address space contains the system prefixed save area (PSA), private areas, and common areas. Each user has an entire address space and thus has access to all three kinds of areas. MVS/ESA effectively isolates one address space from another by means of segment and page tables previously described (see page 12, "Address Translation"). Users can share programs and data through the common areas of the address space. Since MVS/ESA has been made compatible with 24-bit MVS, the layout of the virtual address space is somewhat unusual (see Fig. 3.1).

The virtual address space is, for convenience, divided into many different areas of storage called subpools. Pages in a given subpool have several common characteristics, including the location of the subpool in common, private, extended common, or extended private storage, the protect key that the subpool can be accessed with, whether the data in the subpool is fetch protected, when and how the subpool is freed, and whether the subpool is pageable or fixed in real storage. Subpools allow users to free all data at once, even though they may have been allocated at different times earlier in the job. There are 256 subpools: subpools 0-127 are allocated for use by general users; subpool 128 is for compatibility with OS/VS1 programs; subpools 129-202, 206-212, and 216-222 are undefined; the rest are allocated for system storage (SQA, LSQA, ELSQA, etc.).

The PSA contains critical information about both the MVS/ESA operating system and the processor(s). It includes fixed storage locations for interrupt handling, register save areas for system routines, and pointers to critical control blocks as described earlier.

The private area contains modules and data not shared by other address spaces. It consists of five subsections: system region, user region, LSQA, SWA, and AUK. The system region is used by system functions performing work for an address space. These system functions run under the region control task (RCT), which is the highest priority task in each address space and plays a key role when an address space must be swapped in or out.

The user region and the extended user region are the subsections of the private area in which user programs run. There are two types of user regions: virtual (V=V) and real (V=R). Virtual user regions are pageable and swappable. These are the regions most often used in a timesharing system. Real regions occur only below the 16 megabyte line, and are non-pageable and non-swappable. Although DAT is used, the virtual address always corresponds to the real address.

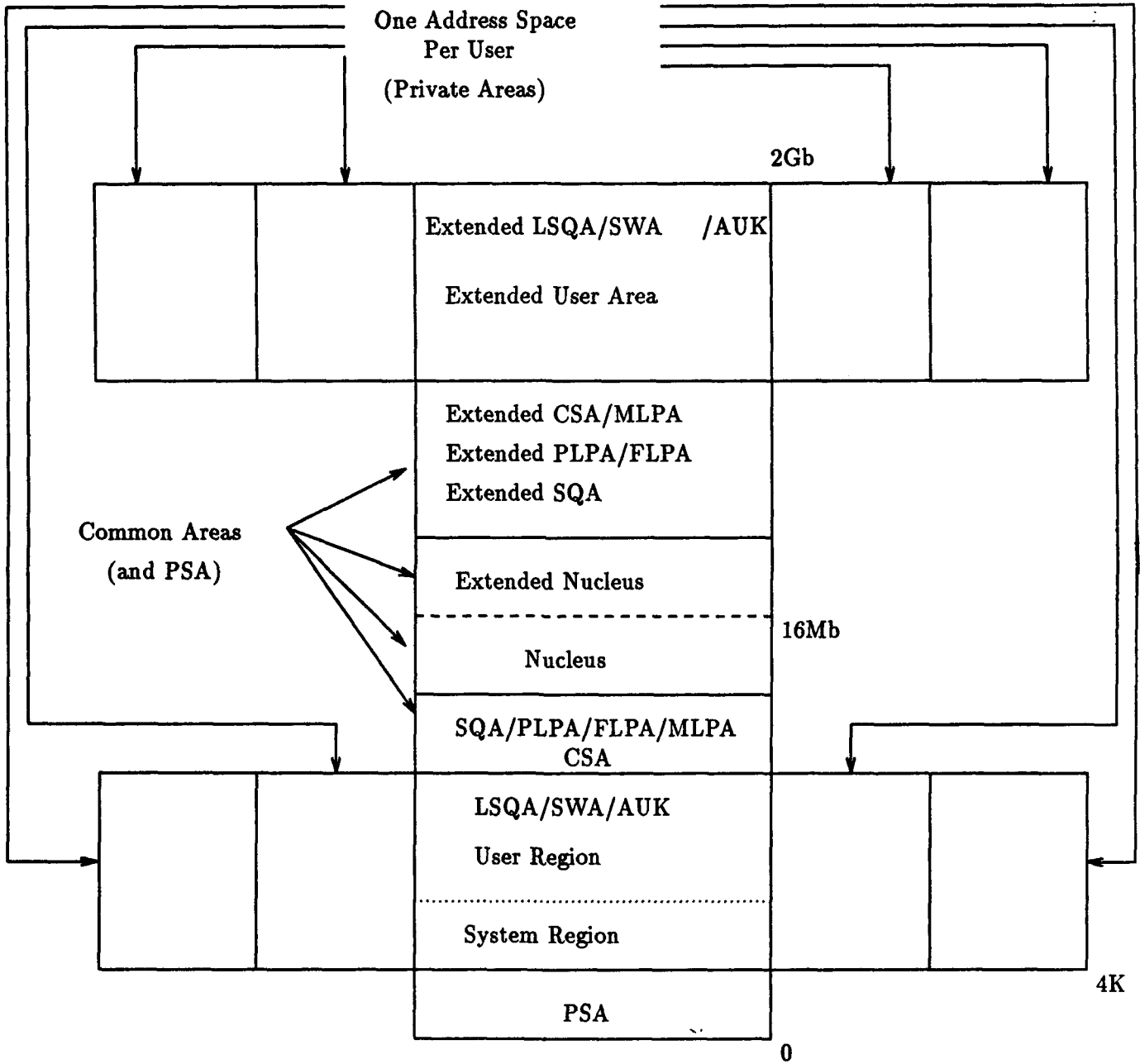


Figure 3.1. MVS/ESA Virtual Address Space Layout

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

The system administrator, at system generation time, determines the size of the real region. After initial program load (IPL), whenever a user specifies the size of the address space (an optional parameter), one of three actions can occur. If the requested space is greater than the space specified at sysgen, then the user job step will not run. If the requested space is not available (i.e., other users have already allocated it), the job step will wait until the region or key becomes available; the job step is then run. If there is enough space, then the user is assigned a key from 9-15, the next sequential key number that is not being used, and execution begins in that region. If all keys are being used, then the job step will wait for a key. When users are not executing in this region, the system uses it as a fast paging area. When it is required, the pages are migrated to the expanded store.

The authorized user key (AUK) area and the extended authorized user key area of the private region contain system data relating to a specific user. Protected user control blocks reside in this area. AUK also contains data for the LNKST lookaside (LLA), which is an in-storage directory of all of the modules that are in SYS1.LINKLIB. The LLA address space provides a cross-memory search routine used to speed up searches of SYS1.LINKLIB by other components of the system. The scheduler work area (SWA) and the extended scheduler work area contain the control blocks that exist from job step initiation to job step termination. These contain the internal (interpreted) form of the Job Control Language (JCL) statements that accompany a job. The local system queue area (LSQA) and the extended local system queue area contain tables and queues that are unique to a particular address space such as the user's segment table and private area page tables. LSQA also contains all the control blocks that the RCT requires. LSQA is swappable but not pageable.

The common area holds system information, such as program code, control blocks, tables, and data areas. The common storage area (CSA) and the extended common storage area are addressable by all active programs, and they are used by all swapped-in users for inter-address space communication. CSA contains some fixed and some pageable system and user data areas. CSA cannot, however, be directly addressed by user programs; system programs access this area on behalf of the user. The pageable link pack area (PLPA) and the extended pageable link pack area contain MVS/ESA control program functions (SVC routines), access methods, other read-only system programs, and selected user programs. PLPA is pageable but no physical page-outs occur since PLPA provides read-only modules. The fixed link pack area (FLPA) and the extended fixed link pack area are fixed in storage. FLPA consists of modules which could be in PLPA but because of fast response requirements are fixed instead.

The modified link pack area (MLPA) and the extended modified link pack area can be used for reentrant modules from selected system or user libraries. MLPA exists for the duration of the active MVS/ESA system but it is not saved from one MVS/ESA start-up to another. The system queue area (SQA) and the extended system queue area contain tables and queues that relate to the entire system, as well as address space control blocks. For example, the page tables that define the system area and the common area reside in SQA.

The nucleus and the extended nucleus hold the resident part of the MVS/ESA control program. In addition they contain the page frame table entries, Data Extent Blocks (DEB) for the sys-

tem libraries, recovery management support routines, and unit control blocks (UCBs) for the I/O devices. While most of the nucleus executes with the DAT enabled, certain recovery processing routines within the nucleus execute with the DAT disabled.

3.1.2 Dataspaces

The ESA/370 architecture described previously (see page 12) was implemented to provide users with enormous amounts of new virtual storage space. A dataspace is an addressable block of that virtual storage that can range from 4K to 2G bytes. The value of a dataspace is that data in one space can be moved to and from another space, while the instructions are being fetched from an address space. Note that execution of instructions in a dataspace is not possible; one is not able to branch to, jump to, transfer to or call routines in a dataspace. Data in a dataspace behaves exactly like data in an address space, in that it is directly addressed by the ESA/370 instruction set, and is paged to and from real memory. A program must be running in access-register mode (AR mode) to gain access to a dataspace.

Dataspaces are created by calling the DSPSERV macro. A user will specify a number of parameters for the dataspace, and DSPSERV will return an STOKEN. An STOKEN is an 8-byte value that identifies the dataspace and is guaranteed to be unique throughout an IPL. The user will then use the ALESERV macro to obtain an Access List Entry Token (ALET) for the dataspace. The ALESERV macro is the privileged macro which puts a new entry onto the access list for the user (the DU-AL). The user then has to load a base address for the dataspace in a general register, the ALET in the corresponding access register, and use unprivileged 370 instructions to manipulate data in that data space. Note that the base address can be any address in the dataspace, and merely provides a starting point for subsequent data accesses in that dataspace.

Dataspaces can be of two types, called "SCOPE=SINGLE" and "SCOPE=ALL" dataspace. SCOPE=SINGLE dataspace are used to hold information that cannot be directly shared by programs running in different address spaces. If a SCOPE=SINGLE dataspace is on a PASN-AL, then it can be shared by any programs in that address space. If a SCOPE=SINGLE dataspace is on a DU-AL, it can only be shared by programs whose home address space is the same as the dataspace's owner. A SCOPE=ALL dataspace can be used by programs in the same address space as the dataspace's owner and in other address spaces, and must be non-swappable. A problem state user cannot create a SCOPE=ALL dataspace.

Dataspaces can be shared by unprivileged programs, providing the restrictions outlined above are met. This can be accomplished when the calling program specifies "ALCOPY=YES" on the ATTACH macro. The effect is that the caller's DU-AL is copied to the callee's DU-AL. The ALETs for the dataspace to be shared must also be passed.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

3.1.3 Hiperspaces

A hiperspace is similar to a dataspace in that it can be up to 2G bytes of virtual storage that a program can access. However, there are three fundamental differences between hiperspaces and dataspaces:

- Data in a dataspace can only be accessed in AR mode, while data in a hiperspace can be accessed in either primary or AR modes.
- Data in a hiperspace cannot be accessed directly—they must first be moved to the address space.
- Data in a dataspace are backed in the same manner as other virtual data in the address space, while hiperspace data are backed by expanded and auxiliary storage.

A hiperspace uses the expanded storage of the machine as the primary storage area for its data. Data can be moved into and out of the address space by using the HSPSERV macro. Data are moved in 4K blocks only. When a hiperspace is created by an unprivileged user, the data in the hiperspace are backed by expanded storage, and if expanded storage is full, then they are written to auxiliary storage.

A hiperspace is created in exactly the same way as a dataspace. Only authorized programs can create a hiperspace which is backed by expanded storage only. When such a hiperspace is filled up, data will be thrown away. Sharing of hiperspaces is identical to sharing of dataspaces.

Hiperbatch

Hiperbatch is an alternate facility for sharing VSAM and QSAM data sets (for more on VSAM and QSAM data sets, see page 68). The Data Lookaside Facility (DLF) is called to create the hiperspace where the data sets will reside, and to maintain the hiperspaces between jobs and job steps (when they would normally be destroyed). Hiperbatch is responsible for putting the VSAM and QSAM data sets into the hiperspaces, and for calling the DLF routines which manage the hiperspaces. Hiperbatch will receive from DLF the name, the STOKEN and the ALET for the hiperspace. DLF will add the hiperspace to the PASN-AL for the requesting address space (i.e., the user who requested the space, *not* hiperbatch). The hiperspace created is expanded storage only (no backing in auxiliary storage). Access checking for the data set in the hiperspace is done by hiperbatch when the user requests to access the data set. The access check is done by RACF in the same fashion as for any other data set access request.

3.1.4 Dispatcher

The dispatcher for MVS/ESA is responsible for removing a "unit of work" from a queue and handing it to a processor for execution. Units of work, when ready to execute, are called dispatchable units. The dispatcher will dispatch the highest priority unit of work that is available. Units of work are represented in three ways on the system: as special exits, as Service Request Blocks (SRB), and as Task Control Blocks (TCB). There are many queues of different types of SRBs, while there is only one queue of TCBs per address space.

Special exits are extremely high priority service routines that must run as soon as they become ready. These are branched to directly by the dispatcher, instead of being handled in the normal manner. For example, when one processor crashes, a special exit is invoked by another processor on the same system in an attempt to perform alternate CP recovery on the failed CP.

A non-preemptive unit of work is one which can be interrupted, but must receive control when the interrupt is done. A preemptive unit of work will not receive control; in this case, the dispatcher is invoked and will determine the next available unit of work. SRBs are non-preemptive requests for service from or for a particular address space. SRBs can be created only by a key 0 program running in the supervisor state. SRBs can run either in the address space in which they were created, or in another address space.

TCBs are control blocks for the "normal" work done on the system, e.g., user tasks, most system tasks, and started jobs. All tasks ATTACHED by a task are its subtasks. They are chained in its subtask chain. Each task has its own subtask chain. All tasks in an address space are also chained in a TCB chain for the address space. This chain is used for dispatching and is in priority order. As opposed to SRBs, TCBs are preemptive, except for those that are controlling non-preemptive SVCs.

A task is merely a unit of work that is represented by a TCB. An address space can contain many concurrent tasks (and even SRBs). A task can be in one of four states:

- Active: A task in this state is running on a CP.
- Ready: A task that is ready to run but not currently dispatched on a CP.
- Waiting: A task in this state has a non-zero wait count, and cannot be dispatched until the wait count is zero. A "post" to the event the task is waiting on will reduce the wait count by one, and is generated by an active task issuing a POST macro.
- Non-dispatchable: A task stopped by the system for some reason. Examples of tasks in this state are those waiting for a suspend lock or for page fault processing to complete.

All tasks associated with an address space are in a list of TCBs (see Fig. 3.2). A header block called the Address Space Control Block (ASCB) resides in the SQA. The ASCB points to the Address Space Control Block Extension (ASXB), which is located in the LSQA. The ASXB is what points

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

to the head of the chain of TCBs. The first TCB in every address space is the RCT. The three types of address spaces that are associated in some manner with users of the system are the batch address spaces, the started procedure address spaces, and the TSO/E user address spaces.

In each of these address spaces, the RCT points to the DUMP TCB, which in turn points to the started task control (STC) TCB. At this point, different TCBs are chained. For batch address spaces, the STC TCB points to an initiator, which in turn points to batch TCBs (representing units of work requested by the user who started the batch job). For started procedure address spaces, the STC TCB points to started tasks. For TSO/E users, the STC TCB points to a terminal monitor program; TCBs representing commands run by the TSO/E user are then pointed to by the terminal monitor program TCB.

3.1.5 Inter-Address Space Communication

There are several different methods for two address spaces to communicate in MVS/ESA. The method chosen is based on whether the service is authorized, and how much data needs to be moved on the call. Using cross-memory services in the MVS/XA system, only 256 bytes of data could be moved at one time between two address spaces, and all data that an address space operated on had to be in that address space. With the advent of AR mode, these constraints are lifted, so proliferation of "service" address spaces should increase.

The communication mechanisms discussed below are cross memory services (the basic inter-address space communication mechanism), the sub-system interface (used for system routines to communicate with other system routines or MVS itself through a well defined interface), and service request blocks (used to schedule a service on the dispatching queue).

Cross Memory Services

As previously mentioned, MVS/ESA provides each user with a unique address space and maintains the distinction between the code and data belonging to each address space. MVS/ESA also includes cross memory services that permit a single user to access other address spaces when necessary.

Cross memory allows programs to pass control to programs in other address spaces and to move data from one address space to another. Because a program using cross memory capabilities can directly access programs and data in the private area of another address space, cross memory can reduce the amount of common area needed in the virtual address spaces in the system.

Cross memory services execute out of the PC/AUTH address space. They create and manage the data structures that support the program call (PC) instruction and allow control of the cross memory authorization structure. These services are used by supervisor state or PSW key 0-7 callers, usually subsystems, to set up the environment for controlling cross memory access to programs and data. This restriction excludes non-authorized users from directly utilizing these services. A PC

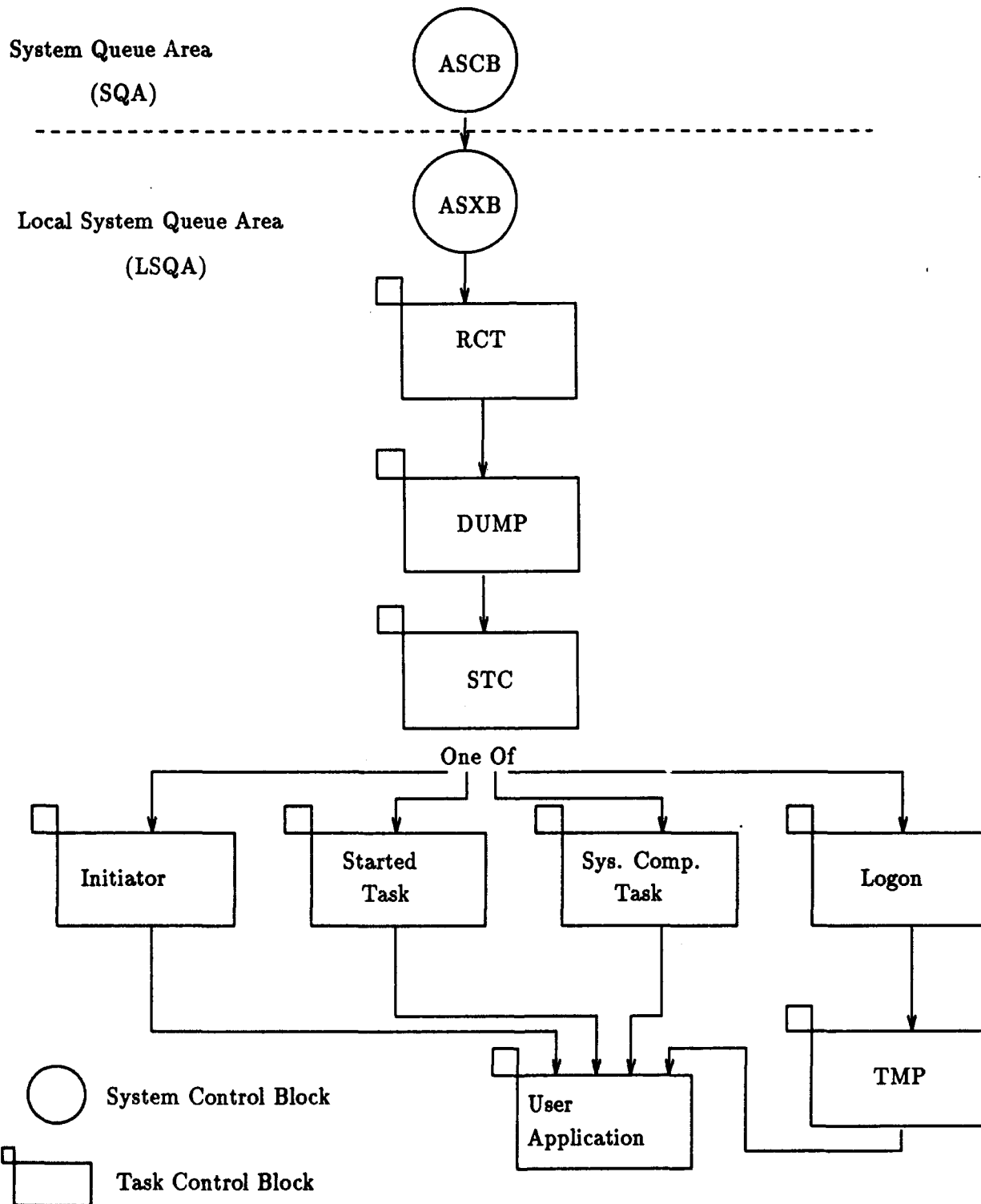


Figure 3.2. Task Headers for Subjects in MVS/ESA

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

instruction will cause a switch to the address space designated by control register 3, using the authorization table mechanism described on page 11.

PC/AUTH services are used by MVS/ESA itself. Whenever an address space is created (see also page 49, "Address Space Creation"), PC/AUTH initializes its address space second table entry (ASTE) and chains the new address space to the system linkage table (SLT) and the system authorization table (SAT). The new address space is unauthorized to issue the program transfer (PT) instruction to another address space or the set secondary address space number (SSAR) instruction. It has access only to those global PC services that are available to all address spaces via the SLT.

When a task or an address space terminates, the PC/AUTH resource manager gets control. If a cross memory resource owning task or address space is terminating, PC/AUTH-related resources are recovered.

Finally, PC/AUTH provides services that allow authorized programs (supervisor state or key 0-7) to build and manipulate entry tables and linkage tables used for housekeeping while cross memory functions are performed.

Sub-System Interface (SSI)

The sub-system interface is more a signalling mechanism than a cross address space function. That is, a program can call SSI passing a function code and SSI will call programs that have been identified to be called for that function code. For a given function code, 0, 1 or more programs may be called. The caller does not know about the called routines. Both the caller and the called routine execute in the same address space. There is no mode or key switching done on the call. Usually the called routine is provided by a subsystem (e.g., JES2, JES3), and it will use other services to communicate cross memory from the caller's address space to the subsystem's address space. The data passed on the SSI call could be in either private or common storage and may vary depending on the function code specified.

Service Request Blocks

Another method for authorized programs to request services is for the requesting program to schedule a Service Request Block (SRB). SRBs are described in detail on page 39. An SRB will cause the asynchronous execution of the service, using the CSA for passing data.

3.1.6 Storage Managers

There are three types of storage in the MVS/ESA system, each of which has its own storage manager: the real storage manager (RSM), the auxiliary storage manager (ASM), and the virtual storage manager (VSM). These three components are discussed in this section.

Real Storage Manager

Real storage refers to the main memory of the system, which can vary according to the model of the processor that is running MVS/ESA. RSM manages the real storage and the expanded storage on the system. MVS/ESA uses the expanded storage region as a solid-state paging device which conceptually lies between real storage and auxiliary storage.

Functions of RSM include handling segment and page faults, providing paging services such as fixing and freeing pages in main memory (i.e., making the page unpageable and pageable, respectively), paging storage out (in conjunction with ASM), and releasing and loading pages. Other services provided by RSM are virtual I/O and virtual fetch services, real storage reconfiguration, V=R allocation, address space creation, swapping, storage and key error handling, page migration, and virtual data access services.

Real storage is divided into 4K blocks called frames. In addition to managing which frames are currently in the configuration of the system, RSM controls which frames are occupied by the pages themselves. If an error occurs when accessing a frame, one of three actions will be performed. If the error is an uncorrectable storage error, the frame is placed offline. If the error is an uncorrectable key error, the key is refreshed; if this is unsuccessful, the frame is taken offline. For correctable errors, an appropriate service request block (see page 39, "Dispatcher") is scheduled.

Page stealing is a process which is managed by RSM. The system resources manager (see page 55, "System Resources Manager") tells RSM which pages to steal by providing information on the address spaces which are candidates for having their pages stolen. This is done by selecting the least recently used page frame. RSM will then take that page frame for use in a different address space, thus reducing the working set for the address space the page was stolen from.

Page migration refers to the movement of pages from extended storage to auxiliary storage. This will happen either because there is a shortage of available frames in extended storage, or because the extended store is being configured off-line. In the first case, RSM invokes the purge migration and LRU migration routines. In the second case, RSM invokes the reconfiguration migration routines.

The RSM also manages the swapping of address spaces into and out of real storage. If a machine has extended storage, there are four swap functions that RSM can perform: swapping to and from auxiliary storage, and swapping to and from extended storage. The System Resources Manager decides which address spaces are to be swapped, and tells that address space's region control task

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

(see page 51, "System Initialization"). RCT then calls RSM to make the actual swap. An address space on extended storage can also be migrated (by RSM) to auxiliary storage.

Auxiliary Storage Manager

The ASM manages storage that is not in the main memory; for example, disk storage as it relates to processes is managed by ASM. The specific types of data that are managed by ASM include the content of page data sets, swap data sets, and VIO data sets. The ASM units are called slots, as opposed to pages for virtual memory, and frames for real memory. The ASM primarily interacts with RSM for paging operations, and for swapping address spaces to and from main memory.

Since ASM must access secondary storage, it goes through Input/Output Services (IOS; see page 47). It does not use EXCP, but instead is an IOS driver, responsible for creating its own channel programs for accessing the data it needs.

Virtual Storage Manager

Since virtual storage is used by every user on the system, a manager is needed to supervise virtual storage operations. The VSM is responsible for managing the subpools, which includes keeping track of which subpools are allocated and how much free space is in each allocated subpool. VSM also maintains an accounting of each address space's storage, services 10 external macro instructions, and supports virtual storage operations during system initialization.

Two important macros are GETMAIN and FREEMAIN, which are used to allocate and deallocate virtual storage for a process. GETMAIN cannot be used in AR mode, since it is an SVC and will cause an interrupt. In AR mode, a program will use the STORAGE macro to obtain and free space. Another way of obtaining space without causing an interrupt is to use the CPOOL macro, which will obtain a pool of storage "cells" which the program can use. There are macros which are responsible for writing to and reading from cells.

Other macro functions include obtaining information about which areas of storage are allocated and which are free, verifying which locations are allocated to certain virtual storage areas (such as LSQA), and operations for manipulating cells, which are small areas of virtual storage. The VSM also is responsible for obtaining the storage protection key from the unit of work and passing it to RSM for checks against the key on the page frame.

3.1.7 I/O Operations

In MVS/ESA there is usually more than one way to accomplish a given task, and I/O is no exception. A program that wishes to interact with the channel subsystem has at its disposal a number of different methods to accomplish this. However, all of the methods will eventually rely

on the Input/Output Supervisor (IOS) to perform the actual I/O operation. The following sections will describe the various interfaces to the IOS, a typical unprivileged user's use of I/O, and the IOS itself.

IOS Drivers

An IOS driver is a service or program that uses IOS routines directly. Only authorized programs can be IOS drivers. IOS drivers include: EXCP (discussed below), ASM, GRS, VSAM, VTAM, and JES3. An IOS driver is responsible for building and passing a channel program to IOS, and ensuring that the program passed will not access any data that the originator of that channel program is not permitted to access.

All the access methods which are described beginning on page 68, with the exception of VSAM, use EXCP to communicate with IOS. An IBM supplied access method will build a channel program on behalf of the user's application, and then call EXCP to perform the I/O. As described next, an unprivileged user can use one of these access methods, or use EXCP directly. JES2 also uses EXCP, unlike JES3, which is an IOS driver. The distinction between the two types of I/O requesters (access methods and IOS drivers) is shown in Figure 3.3.

Typical I/O Operation

I/O operations actually begin when a user program issues an OPEN instruction. The user program provides a data control block (DCB) to the OPEN macro. The macro will fill the DCB with information such as the device and data set information (from the job file control block), specific information from the data set control block, and addresses for the access method routines (see page 68, "Access Methods") to be used for this data set.

The OPEN routine also builds a data extent block (DEB), which contains pointers to the DCB, a structure called the unit control block (UCB, which contains device specific control information), and appendage routines (user supplied exits, for example). Appendages can gain control anywhere in the I/O cycle, possibly changing the channel program. Appendages can only be established by authorized routines from authorized libraries (see page 61, "Authorized Programs") and can only be loaded by OPEN for authorized users.

Thereafter, whenever the user program issues an I/O instruction (e.g., GET, PUT, READ, WRITE) the access method routines get control. These access method routines build an I/O block (IOB), an event control block (ECB), and the necessary channel program. The IOB contains pointers to the DCB, the ECB and the channel program. The ECB contains information about the status of the I/O so that the user program can access its contents to see the results. The access method then issues an Execute Channel Program (EXCP) instruction to pass control to the EXCP processor.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

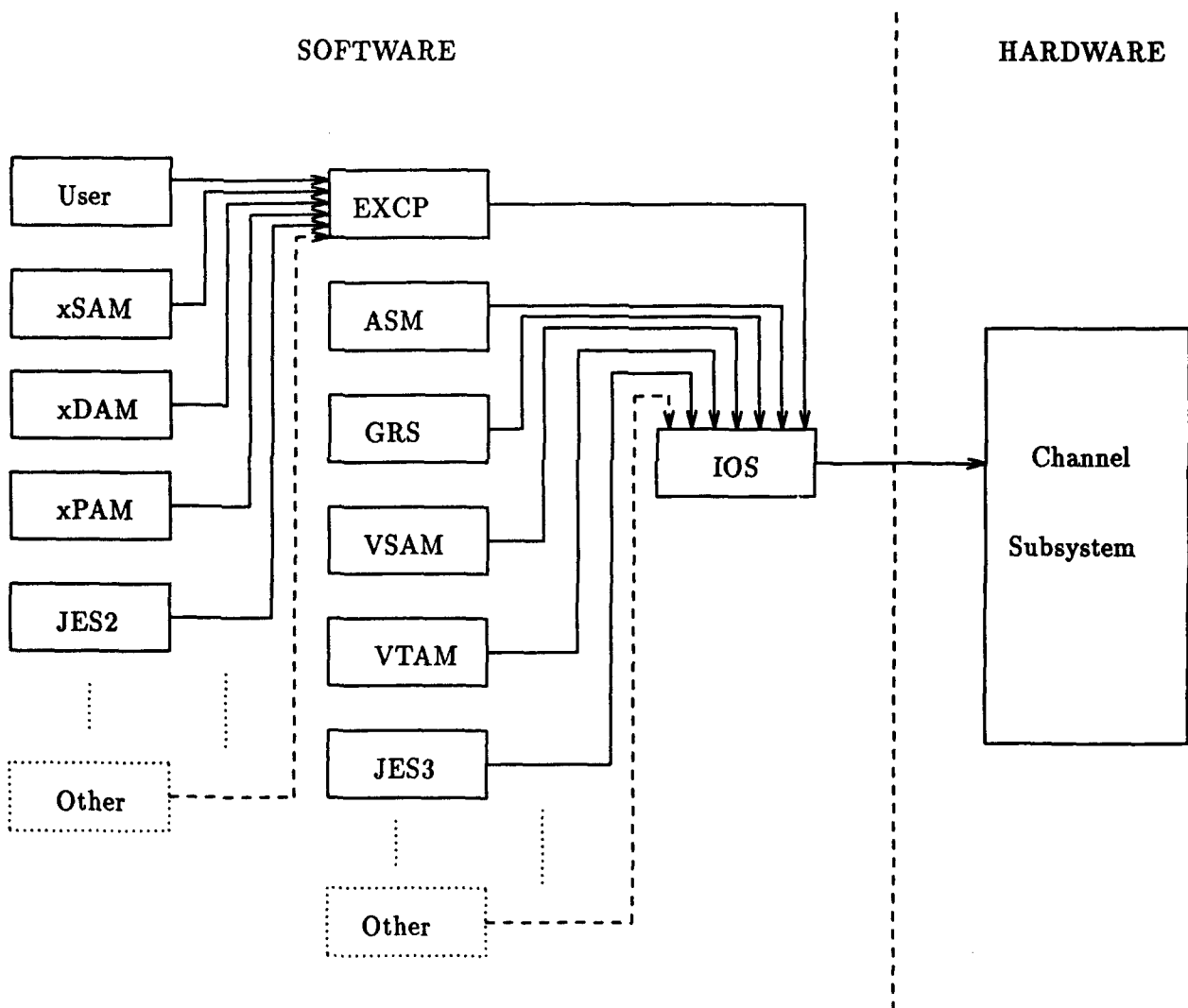


Figure 3.3. I/O Requestors in MVS/ESA

If desired, a user may choose not to utilize any of the standard access methods but rather communicate directly with the EXCP processor. This of course means that the user must fill in all of the information that the EXCP processor expects. Before proceeding, the EXCP processor will perform address checks by obtaining valid address ranges from the DEB. The DEB is created by the OPEN macro, which every user must call in making an access (even those writing their own access methods); thus the address ranges are never directly specified by the user.

EXCP is an IOS driver consisting of three parts: the front end, which prepares the request; exit processing, which monitors and handles interrupts; and the back end, which handles cleanup functions and status returns. The front end of the EXCP processor first creates an I/O Supervisor Block (IOSB). The IOSB contains the address of the UCB and the address of the channel program that is translated by EXCP.

The channel program is copied from the user space to system space so that it cannot be tampered with, and then translated (i.e., put into a form that the IOS can understand), converting the virtual storage addresses to real storage addresses. The front end also fixes the I/O buffers that are used to hold the incoming or outgoing data because the channel subsystem operates with real addresses. Another function of the front end involves prepending a track or cylinder interrupt routine to every channel program. The purpose of this prepended portion is to ensure that when a new track or cylinder is beginning to be read, the user is authorized to read that track or cylinder. Before a new track or cylinder can be read, the prepended portion of the channel program causes an interrupt; this allows the EXCP exit processor to get control. The exit processor will check the DEB to make sure that the new track or cylinder is readable (writable) by the user, and then either fail or proceed, depending on the outcome of the decision. The EXCP processor then issues a STARTIO macro to pass control on to the I/O Supervisor.

When EXCP issues the STARTIO macro, control returns to either the access method or the user program, depending on the type of macro used to initiate the I/O. The access method (or the user program) will then wait on the ECB, which means that it will wait until status is posted to the ECB before continuing.

The IOS will communicate with the subchannel, passing interrupts back to the EXCP exit processor for resolution. After IOS is through with the operation, control is passed back to the EXCP back end. The EXCP back end posts the results of the I/O in the ECB. The access method or user program can test the contents of the ECB to determine the outcome of the I/O. The back end also "unfixes" the storage buffers used for this I/O.

I/O Supervisor

Since all accesses to the channel subsystem must go through IOS, it is important to describe how IOS operates. This section details a typical IOS operation. The IOS chooses the subchannel to be used and creates an Operation Request Block (ORB). The ORB contains the necessary information for the channel subsystem to perform the I/O request. IOS then invokes the channel subsystem by

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

issuing the Start Subchannel (SSCH) instruction with the address of the ORB being passed as an argument.

The channel subsystem fetches the first Channel Command Word (CCW) to ensure it passes certain validity checks. There are two types of CCWs: format 0 and format 1. Although both formats contain the same types of information (command code, flags, byte count, and data address), format 1 fields are arranged differently in order to accommodate 31-bit addresses. Assuming the first CCW passes the validity tests, a channel path is established. Command codes from the CCWs are then sent down this path to the device to initiate the physical I/O. If a path cannot be found, the operation remains pending until one opens up.

The channel subsystem supervises the execution of the channel program, transfers the data, updates the relevant control blocks, and posts I/O interrupts when necessary. EXCP monitors these interrupts and takes the appropriate actions. For instance, EXCP could pass control to a certain appendage specified in the DEB. After the channel subsystem has completed the request, IOS regains control. It will examine the status found in the IOSB, and either initiate error processing or return control to the EXCP back end.

3.1.8 Virtual Input/Output

Virtual Input/Output (VIO) is a method of performing I/O for temporary data sets that eliminates the time-consuming transfer of data using the channel subsystem. Temporary data sets are the only type of data set that can use VIO, and only for the duration of the job which creates them. VIO uses the system paging routines for data transfer. VIO data sets cannot be shared between users because the control blocks for a given VIO data set reside in a user's address space, and cannot be accessed by any other user.

VIO moves data from the channel program's data transfer buffer to an area in the user's address space known as a window. The size of a window is commensurate with the size of a track on the device specified on the data set definition statement.

When the system is initialized, certain I/O unit names are assigned for VIO. Thereafter, when VIO is to be used, the user can specify the unit name, and the system will create a temporary data set that has a system-generated name. When EXCP is translating the channel program, it will invoke VIO instead of IOS for performing the data transfer.

When transferring data, VIO uses a long move instruction to move the data between the window and the buffer. If the window becomes "full" (i.e., a track boundary is crossed), VIO will write the contents of the window to a page data set, and then sever the connection between the two. Thus when the write begins again (e.g., the second track is being written), a page fault will occur, and blank page frames will be associated with the window.

When reading data back in, VIO locates the necessary pages in auxiliary storage (if the pages are not currently in the window), and then sets the page table entries of the window to point to those

pages that contain the desired data. The page table entry invalid bits are set, causing the resulting page fault to bring in the desired pages. Since RSM (see page 43, "Real Storage Manager") tries to keep these pages in real memory as long as possible, there is a good chance that no physical I/O will be done.

3.1.9 Address Space Creation

There are four ways for address spaces in MVS/ESA to be created: use of the LOGON, START, or MOUNT commands, or use of internal routines to create a system component address space. System component address spaces are created directly by MVS/SP using the ASCRE macro. Most of the system component address spaces are created at MVS/ESA initialization.

User address space creation revolves around three address spaces: master scheduler, user, and JES (JES2 or JES3). Address spaces are not created when a batch job is submitted to the system. Instead, the job will run in an initiator's address space (described later). Initially, after receiving a START, MOUNT, or LOGON command, the master scheduler invokes the address space creation routine which assigns an address space identifier, creates control blocks, and requests concurrence from the system resources manager (SRM). The master scheduler then either releases control blocks or invokes VSM to assign virtual storage, sets up addressability, builds LSQA, and creates RCT control blocks. At this point execution resumes in the (newly created) user's address space.

The region control task builds control blocks and invokes the started task control (STC). The STC determines which command is being processed and builds in-storage JCL text for the job. The execution now transfers to the JES address space where JES reads the job, scans the JCL and writes it to spool, invokes the converter to transform the job to internal text, queues the job, assigns a job identifier and passes it to the initiator. The initiator routine is a part of the STC within a user's address space. It requests JES to prepare the job for execution. JES—in its address space—invokes the interpreter to build control blocks from internal text, and passes control back to the initiator in the user's address space. The initiator then invokes the allocation routines and finally attaches the appropriate program/processor. The START command results in the execution of a specified program, the MOUNT command invokes the MOUNT command processor, and the LOGON command activates the terminal monitor program (TMP). TMP controls the interchange of user commands with TSO/E.

Initiators also are used by JES to run job steps that JES has prepared for execution. The initiator will request work from JES according to the job class (see page 72, "Job Classes"), and the job that JES passes to the initiator will then run in the initiator's address space. The initiator is responsible for cleaning up all control blocks associated with the jobs it runs. An initiator can be assigned one or more job classes.

The region control task (RCT) is the highest priority task in an address space. Its functions are: preparing an address space to be swapped out, preparing an address space for execution after a swap-in, and ensuring proper scheduling of a user attention exit. When SRM determines that an

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

address space should be swapped out, the RCT sets all tasks under the RCT as non-dispatchable, purges all its I/O requests, copies the saved functional recovery routine stacks from the SQA to the LSQA, breaks active addressing binds from other address spaces, and finally calls the RSM swap-out routine to initiate the swap-out.

When the address space is swapped in, the RCT common processing invokes the restore routine which prepares the address space, reschedules purged I/O requests, and sets all tasks under the RCT as dispatchable. When a user requests an attention exit, RCT routines ensure that it is properly scheduled and executed.

The started task control (STC) routines oversee the initialization of system component address spaces and the processing of START, MOUNT, and LOGON commands. The initialization could be for either a limited or a full function address space. A limited function address space cannot allocate data sets, read JCL procedures from the system procedure library, allocate a SYSOUT file, or use system services running in cross memory mode. A full function address space does not have these restrictions. All system component address spaces are limited function address spaces except the dumping services address space (DUMPSRV) and the system management facility (SMF) address space.

The STC routines perform five major functions: obtain the region in which STC will run, determine which command was specified, build internal JCL text for the command task, build the control blocks required for initiator/terminator processing, and free those control blocks after the initiator/terminator terminates the command task.

The purpose of the initiator/terminator is to make all the necessary preparations for the execution of a job step or job task. In order to accomplish this, the initiator performs the following functions: obtains storage for a task, initializes the control blocks for a task, assigns properties to a task, oversees the allocation of data sets and devices for a task, opens any required catalogs and libraries for a task, and attaches (spawns) the task.

When a task has completed execution, the terminator performs the following functions: deletes the control blocks no longer needed, deletes the RACF accessor environment (see page 102), oversees the freeing of data sets and devices used by the task, and detaches (kills) the task. When an entire job is complete, the initiator clears and deletes the control blocks and data areas used as well as the storage space occupied.

The initiator provides the above functions in four situations: completing master scheduler initialization; starting a subsystem (such as JES); processing a START, MOUNT, or LOGON command; and initiating a normal job. In the first three situations, the initiator is used as a subroutine to initiate a single job. When that job is completed, the initiator subroutine returns to its caller. In the last case, the initiator is a task created as a result of a START command; i.e., the initiator is a started task. This initiator can, in turn, attach another task. When that attached task completes, the initiator requests another job from the job entry subsystem. JES returns to the initiator with either another job or an indicator to stop processing. In a typical configuration there may be several such initiators, each executing a batch job.

3.1.10 System Initiation

The initialization process consists of loading the nucleus, initializing system resources and resource managers, initializing system component address spaces, and initializing the primary Job Entry Subsystem (see page 70, "Job Entry Subsystem"). The process is divided into three phases: initial program load (IPL), nucleus initialization program (NIP), and master scheduler initialization.

System initialization begins when the system operator initializes the hardware. This is done by invoking the initial microprogram load (IML) to start the processors, by mounting the necessary disk and tape volumes, and finally by requesting the LOAD function. This function activates the IPL control program which in turn utilizes IPL resource initialization modules (IRIMs). The IPL program clears real storage, prepares an environment in which the IRIMs can execute, controls the loading and deleting of the IRIMs, and provides basic service routines for this phase of initialization.

The first IRIM loads the nucleus while another builds the DAT-off to DAT-on linkage table used to establish addressability between entries in the DAT-off nucleus in real storage and entries in the DAT-on nucleus in virtual storage. Other IRIMs initialize or reserve storage for many system component control blocks, work areas, and programs. The IRIMs also begin to initialize the private area of the master scheduler address space, which is the first address space to be created. The VSM IRIM reserves storage in SQA for the system tables and queues. It also sets up extended LSQA with tables and queues to be used by the master scheduler. An RSM IRIM initializes a segment table whose entries are the addresses of page tables for the common area of virtual storage. This common segment becomes a part of the master scheduler address space segment table. Another RSM IRIM initializes the tables that identify how the frames of real storage are assigned. This table resides in the read-write extended nucleus. Yet another IRIM builds the PSA.

The NIP processing is the second phase of the initialization process which utilizes resource initialization modules (RIMs) in establishing the master scheduler address space. This process is completed when the common segment table is copied from the master scheduler's private area into SQA for all address spaces to use. VSM and ASM RIMs allocate virtual storage in the common area for CSA, SQA, and LPA. The IOS RIMs perform device initialization by building the unit control blocks (UCBs) and the installed channel path table. Building UCBs requires initializing the channel subsystem, testing the availability of a device, testing the accessibility of a device, and then checking for duplicate volumes. The master catalog, used to locate cataloged data sets and other catalogs, is then initialized, as is ASM. Page and swap data sets are also opened and initialized.

During this phase, the program call/authorization (PC/AUTH), system trace (TRACE), global resource serialization (GRS), RACF, and DUMPSRV address spaces are initialized. The PC/AUTH routines initialize all the cross-memory tables needed to establish communication with other address spaces. Other system component address spaces use PC/AUTH services to create and initialize their own cross-memory tables. TRACE provides for tracing of system events, installation-defined events, and component events (i.e., certain events that occur in component address spaces). GRS serializes the use of local and global serially reusable resources (see page 58, "Global Resource Serialization").

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

Initializing the master scheduler is the final phase of the system initialization process. In this phase routines required by system-initiated cancel, SWA management, and resource management are loaded. Control blocks needed to invoke the initiator are created and initialized. Then the master scheduler base initialization routine initializes the subsystem interface, the communications task, and initial TSO/E addresses. It performs master trace initialization, sets the time-of-day clocks, and attaches the initiator to start the master scheduler. This initiator is not one of the JES initiators discussed on page 49, but rather a task which is used in a similar fashion to start the master scheduler.

The initiator allocates the required data sets and internal reader data sets, which will be later used to pass JCL from system routines to JES, and invokes RACINIT to establish the security environment. Finally, the initiator attaches master scheduler region initialization as the job step task, thereby activating the master scheduler. The master scheduler accepts system commands and activates their processing. The START JES command starts the initialization of the job entry subsystem.

While initializing the JES, the master scheduler creates an address space for JES. This is accomplished via the address space create routine which builds LSQA in the private area and initializes segment tables and page tables to represent the new address space. Then the routine builds task control blocks for a region control task (RCT) and places the address space control block (ASCB) on the dispatching queue. When the JES address space becomes active, the first task dispatched is the RCT. After the RCT is initialized, it attaches the STC to initiate JES. The STC does so by building job scheduler control blocks in the SWA while the initiator allocates the required data sets. Finally, the initiator attaches the primary JES and MVS/ESA begins accepting jobs.

Before TSO/E logons can be accepted, VTAM and TCAS must be initialized by the operator via a START command. Both VTAM and TCAS operate in their own address spaces. After these two address spaces are initialized, a TSO/E logon can be accepted by the system.

This completes the system initialization process. During this phase five new address spaces have been created: CONSOLE, ALLOCAS, SMF (see page 57, "System Management Facility"), JES2 or JES3 (see page 70, "Job Processing"), and LNKLIST (the LNKLIST lookaside address space; see page 36). From this point on, user address spaces may be created.

Although most of the allocation of control blocks is done in the user's own address space, the ALLOCAS address space contains the control blocks used by the unit allocation status recording module. In addition, the allocation address space initialization routine and the display allocation tables manager both execute in the ALLOCAS address space.

The CONSOLE address space contains the communications tasks. These tasks are primarily used for communications between a user and a system console or a TSO/E monitoring device. The Write to Operator, Write to Operator with Reply, and Delete Operator Message macros are used to perform the communication. The issuer of the macro can specify the console to which the message is directed.

3.1.11 Consoles

In the MVS/ESA system there are two types of consoles. Hardware consoles are those consoles used in support of and in configuration of the hardware of the system. Operator consoles (Multiple Console Support (MCS) and JES3 consoles) are used to control and monitor MVS/ESA operations. This section discusses each type of console.

Hardware Consoles

3090 Hardware Consoles Consoles which attach directly to the PCE are different from MCS consoles. It is also important to distinguish between "displays" and "consoles." The displays are the physical terminals (CRTs) which attach to the PCE. There are four display ports—two service displays and two system displays. Logical consoles are sets of functions that can be performed. There are many logical consoles, each of which can be associated with the same display. Logical consoles are the system, service, program mode, system monitor, service monitor, data bank access, and remote consoles. Physical displays can only be assigned one console at a time, and some consoles can only be assigned to one display at a time. The two most important consoles are the system and the service consoles.

A minimum console configuration for a 3090 machine is a service display and one system display for each SCE. The system console has two access levels. Level 2 (the lowest access level) gives the operator access to configuration, control, and monitoring functions. Level 1 gives access to level 2 functions, as well as recovery functions.

The service console also has two levels. Level 2 is used for diagnostics, while level 1 is used for PCE address and channel manipulation, as well as level 2 functions.

The service display is used to perform very low-level functions, such as the initial microcode load (IML), and patching of microcode into the machine. The system displays are the displays that generally are used in configuring and IPLing the system.

The service displays can be located a maximum of 10 feet from the PCE (co-axial cable hookup), and the system displays can be located up to 1500 feet away. Both displays are considered part of the physical hardware of the system, and are treated as such.

4381 Hardware Consoles The 4381 has a separate maintenance subsystem, but this subsystem is not physically separate from the rest of the machine, as the PCE is in a 3090 complex. The service console is attached directly to the maintenance subsystem, is used to IML and IPL the system, and can also be used for diagnostics and maintenance functions.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

MVS/SP Operator Consoles

MCS consoles MCS consoles are attached via a normal channel to the system. There is one master MCS console on the system, and optionally up to 98 other MCS consoles. There is also a provision for switching the master console from one device to an alternate device (defined in the PARMLIB CONSOLxx member). This switching is triggered by an external interrupt from the system console, by a VARY MSTCONS command, or by an uncorrectable I/O error on a console. The master console is either defined in the PARMLIB member, or if not defined there, it defaults to the first full function (i.e., has the ability to perform all console functions) console defined in the PARMLIB member. All MCS consoles execute in the CONSOLE address space. These consoles control MVS/ESA functions, and display messages relating to MVS/ESA performance, user requests, etc. The subchannels that the consoles are attached to are defined in a data set. This data set is read at IPL, and the changes to the consoles are put into effect at that time. In order to add a new console, the data set must be edited, and the system re-IPLed.

MCS consoles can be (and in the evaluated configuration must be) used to issue operator commands for other TCB products, such as JES2, JES3 and TSO/E.

JES3 Consoles JES3 supports three types of consoles: JES3/MCS, JES3 with MVS command capability, and JES3 consoles only. A JES3/MCS console configuration allows all JES3 commands (i.e., the functions of a JES3 local console) to be issued from an MVS console of the type described above. This is the mode in which all JES3 consoles on an evaluated system are to be run. A JES3 console with MVS command capability is a JES3 local console which can execute MVS commands via the "SEND" command. A JES3 local console can issue JES3 commands, but not MVS commands. It is a separate physical device apart from the MVS console. These last two types of JES3 consoles are not allowed in the evaluated configuration.

3.1.12 Exit Routines

An exit is a defined point in a system program where that program calls another program. At some exit points, IBM supplies a program which performs default processing when an exit is reached in a system program. At other exit points, the system will check to see if an installation-defined exit routine exists; if so, the system goes on to execute that routine. At each exit point, an installation-written routine can be supplied for the system to call. For example, an exit is supplied in JES to allow the installation to supply its own algorithm for searching the job queue for jobs to be processed. Installation-written exit routines are not allowed in the evaluated configuration.

3.1.13 System Management

Managing the work in a system such as MVS/ESA is a complex task, and there exist many different modules which perform system management functions. In addition to the process dispatcher previously described, four major system management modules are described below: the System Resources Manager (SRM), System Authorization Facility (SAF), System Management Facilities (SMF), and the serialization managers.

System Resources Manager

The SRM is responsible for controlling and monitoring system resources in order to optimize performance and gain maximum utilization of those resources. It is responsible for providing swapping and page stealing decisions to RSM, dispatching priorities, memory utilization analysis, and inhibiting or calling for increased creation of address spaces. SRM is divided into three parts: SRM control, the workload manager, and the resource manager.

The SRM control has two functions: scheduling other SRM routines to run as needed, and making swapping decisions. Conceptually, there are four different types of swapping that SRM control dictates. RSM, which actually swaps the address spaces (with the help of ASM), is invoked by SRM control. The four types are unilateral swap-out, ENQ exchange, exchange swap, and unilateral swap-in. Swapping decisions made by SRM control are based on input from the other two parts of SRM (workload manager and resource manager).

The workload manager subsection of SRM has three functions: monitoring the rates at which address spaces are consuming resources, providing swapping recommendations to SRM control as requested, and collecting data on resource utilization for measurement tools (such as the Resource Management Facility).

The resource manager subsection of SRM monitors and manages four areas: storage management, I/O management, processor management, and resource monitoring.

The storage manager takes action when one of the following shortages is detected: free page frames in real storage, pageable frames in real storage, available slots in auxiliary storage, and space on the SQA (which means the SQA may expand into the CSA). If a shortage of free page frames is detected, SRM initiates page stealing until the shortage is alleviated. If a shortage of space on the SQA is detected, SRM disables new address space creation until the SQA has sufficient storage restored as a result of old address spaces terminating. The actions taken on detection of shortages of auxiliary slots and pageable frames are the same. SRM will reduce the workload by disallowing the creation of new address spaces and delaying any newly created address spaces from executing. SRM also swaps out the top user of the scarce resource, and notifies the operator of the identity of that user.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

The I/O manager makes recommendations to SRM for swapping decisions. It also determines which device to allocate when more than one of the type requested by the user is available.

The processor manager controls the dispatching priorities of address spaces, prevents the swap-out of users ENQed on a resource that other users are waiting for, and it also makes swapping recommendations to SRM control.

The resource monitoring function makes recommendations for adjusting the number of address spaces currently in memory based on the rate at which those address spaces are consuming resources.

System Authorization Facility

SAF provides an interface to RACF (see page 99, "Resource Access Control Facility") and/or to a user supplied processing routine by using a system service called the MVS router. The MVS router is the focal point and common system interface for all products providing resource control. The MVS router is always present in the MVS operating system. The MVS router provides two exit points (see page 54) for security products. The first will invoke an installation-written security processing routine if one exists. The second invokes RACF if it exists. Installation-written security processing routines or systems, if installed, will invalidate the rating of the system. The RACF must always be installed.

The resource managing subsystems (e.g., JES, DFP) invoke the SAF MVS router by issuing the RACROUTE macro instruction. A parameter telling which RACF macro to process and the associated parameters needed are passed via RACROUTE to the MVS router. The MVS router then calls the MVS router exit, which returns to the MVS router with a return code indicating whether or not to invoke RACF. In an evaluated system, RACF is always invoked when security processing cannot be completed by SAF. After invoking RACF, the MVS router converts the RACF return and reason codes and passes them to the caller via RACROUTE. The RACROUTE return code indicates that either the requested security function was completed successfully, the requested security function was not processed (possibly because RACF is not active), or the requested security function was processed and failed.

The RACROUTE macro instruction is used to access the RACF functions provided by RACDEF, RACINIT, RACXTRT, RACLST, RACHECK, and FRACHECK. Authorized programs may issue the RACF functions directly but RACROUTE is the preferred method.

For this version of MVS/SP, SAF actually provides function. It propagates security labels and userids, and builds default UTOKENs and ACEEs. SAF is used to extract a UTOKEN from an ACEE. SAF is also used by privileged callers to modify fields within an existing RTOKEN or UTOKEN (see page 122 for more on RTOKENs and UTOKENs).

System Management Facilities

The master scheduler task attaches SMF during IPL. SMF runs in its own address space which contains SMF control blocks and buffers. SMF routines collect data, provide for user-supplied data collection routines, and record the collected data in an SMF data set. The following components contain SMF data collection routines and exits for user-supplied data collection routines:

- The interpreter (JES2 and JES3)
- The initiator/terminator (MVS/SP)
- The command processor (TSO/E, MVS/SP, RACF)
- The timer supervisor (MVS/SP)
- All storage managers (MVS/SP)
- The system resource manager (MVS/SP)
- All access methods
- JES2 and JES3
- PSF
- TSO/E
- RACF

The components listed above build records as they collect data. Once a complete SMF record is built, the component issues SVC 83 to transfer the records to an SMF buffer. SVC 83 then schedules a service request for SMF routines to write data to an SMF data set, and when necessary initiates a switch to another SMF data set.

Up to 36 SMF data sets may be allocated. These data sets are RACF protected. The primary data set is the one used for recording. The secondary data sets are used when the primary is full. Each data set is a VSAM data set that resides on a single volume, is cataloged, and not extendible. Each SMF data set should be created before the first IPL that starts SMF recording. If there are no available data sets when SMF is initialized, SMF will start recording and storing records in buffers until a data set is made available to SMF. For a discussion on the specific types of records SMF keeps and processes, see page 130.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

Serialization Managers

MVS/ESA provides three methods of serializing resources: the ENQ-DEQ discipline, system provided locks, and Global Resource Serialization (GRS). The mechanisms are useful for both single CP and multiple CP machines, as well as multiple machine (JES complex) configurations.

ENQ-DEQ

The ENQ macro is performed on a name of a resource; this name is the means by which the operating system recognizes the resource. Users can ENQ exclusively, or indicate that they are willing to share. If the user ENQs exclusively, then that user must wait until all current ENQs on the resource have been DEQed. A shared ENQ capability enables users to allow others to access the resource. Both reads and writes are allowed for a shared ENQ. In both ENQ cases, the ENQ is released by the DEQ macro. Whereas the ENQ macro locks a resource, the RESERVE macro locks a DASD volume. The RESERVE lock is released by the DEQ macro. The lock is a bit on the controller for the volume, and only locks other systems in a complex out of that volume (i.e., other tasks in the system, on the same or other CPs, can still access the volume).

Locks

MVS/ESA also provides locks, a mechanism to exclusively acquire system resources. There are two categories of locks: global locks, which are used on resources related to more than one address space, and local locks, which reserve resources local to a single address space. A global or local lock can be one of two types: spin or suspend. Locks are only obtained by units of work in key 0 and supervisor state.

Each lock type has its own lock manager composed of routines that control the use and behavior of the locks. The lock types are differentiated by the action that the processor attempting to lock the resource takes when the resource is already locked or otherwise unavailable.

A spin lock causes the requester to poll the lock until the lock is released, at which time the requester gains the lock on the resource. A suspend lock, on the other hand, causes the requesting unit of work to be suspended until the lock is released; this will free the processor that the requester was running on for other work. The lock manager is responsible for waking up the suspended process when the lock becomes available.

When a lock mechanism exists on a system, it is possible for two processes to each hold locks on a resource the other process requests, resulting in a deadlock. To prevent this situation, a hierarchy of locks has been established so that a process can only request (and obtain) locks that are higher in the hierarchy than all locks it currently holds.

Global Resource Serialization

Global Resource Serialization (GRS) is a mechanism which allows many different machines in a complex (see page 9, "JES Complexes") to serialize on a specific resource. Before the advent of GRS, the RESERVE instruction was used, which effectively locked the entire volume the resource

was on. GRS allows serialization on specific resources on the volume. GRS uses the channel-to-channel adaptors to communicate among the machines in the complex in reserving the resources. GRS includes many different mechanisms that enable one to detail the state of the system with respect to the resources affected by GRS. These mechanisms include ways to dump the GRS control structures, format the GRS data, scan the resource information directly, and display the resource contention information.

In order to use GRS, the user must specify a scope of "SYSTEMS" on the ENQ instruction. In order to provide compatibility with older programs, there are three exits that GRS invokes. The first exit is called the inclusion exit and is invoked when a scope of SYSTEM is specified on an ENQ or DEQ. An associated list, called the SYSTEM inclusion list, contains names of resources that should be serialized via GRS. If the name of the resource is on this list and not on the SYSTEMS exclusion list (described next), GRS will be used to serialize the resource. Otherwise, local serialization is performed.

The second exit is the SYSTEMS exclusion exit, which uses an associated SYSTEMS exclusion list. If a resource is specified on this list and the SYSTEMS scope is used on the ENQ or the DEQ, the scope will be changed by GRS so that local serialization is performed.

The third exit is the RESERVE conversion exit, which also has an associated resource name list. If the resource name is specified on the SYSTEMS exclusion list, an ENQ with SYSTEM scope will be issued, and the RESERVE instruction will be issued. If not on the exclusion list and on the RESERVE resource name list, the resource will be ENQed with a SYSTEMS scope, and the RESERVE will be suppressed by GRS. If the resource is neither on the exclusion list nor the resource name list, that resource will still be ENQed with a SYSTEMS scope, but the RESERVE instruction will be issued.

3.1.14 Supervisor Calls (SVC)

There are 139 different Supervisor Calls in MVS/ESA, which are divided into five types: 1, 2, 3, 4, and 6. A type 5 SVC designates a spot in the SVC table that is "reserved," as user sites have the ability to add their own SVCs. A type 5 slot may be filled in with a user SVC of one of the other five types. However, such extensions are not part of the evaluated system.

All SVCs run in the supervisor state, and hold protection key 0. The use of some SVCs is also APF authorized (see page 61, "Authorized Programs"); these are summarized below. An SVC can hold and acquire locks (see page 58, "Locks"); the majority of these routines are entered with locks specified in the SVC table. An SVC can call another SVC, if the caller does not hold a lock.

The code for types 1, 2, and 6 resides in the nucleus, while the code for types 3 and 4 resides in the Link Pack Area (see page 34, "Address Spaces"). Type 1 SVC routines are always entered with the LOCAL lock, even if they are not specified in the SVC table. Type 3 SVCs consist of one load module, while type 4 SVCs consist of more than one load module. Type 3 and 4 SVCs must fix their pages in real storage to avoid disabled page faults; i.e., page faults occurring after the SVC

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

has acquired a disabled lock (any lock other than LOCAL, CMS, or CMSEQDQ). A type 6 SVC must always run with interrupts disabled, and must not enable them during its execution. Type 6 SVCs also may not be suspended for a lock request.

The SVC table resides in the system area, and contains eight bytes of information regarding each SVC. This information includes an entry point for the routine, the type and function code (authorized or not), locks to be acquired before entering the routine, and the addressing mode indication (24-bit or 31-bit).

The following is a list and short description of all of the SVCs that are restricted to invokers who are in supervisor state, keys 0-7, or are running APF authorized.

SVC 15: ERREXCP macro This is a type 1 SVC, which holds the LOCAL, IOSUCB, and IOSCAT locks. Its function is to post an error status to the ECB on a return from IOS.

SVC 32: (no macro name) This is a type 4 SVC, which acquires a LOCAL lock. Its function is to do initial allocation of space on DASD.

SVC 39: LABEL macro This is a type 3 SVC which holds no locks. Its function is to write the label to a tape.

SVC 52: RESTART macro This is a type 4 SVC which holds LOCAL, CMS, and SALLOC locks. Its function is to restart a job from a checkpoint data set under the Checkpoint/Restart facility of MVS.

SVC 59: OLTEP macro This is a type 3 SVC which holds LOCAL and CMS locks. Its function is described on page 139.

SVC 76: (no macro name) This is a type 3 SVC which holds no locks. Its function is to format and write out various hardware and selected software errors to SYS1.LOGREC.

SVC 83: SMFWTM or SMFEWTM macro This is a type 3 SVC and holds no locks. Its function is to complete and write the information contained in all SMF records out to the SMF data set.

SVC 85: DDRSWAP macro This is a type 3 SVC which holds the LOCAL lock. Its function is to interface with a user-written routine that manages non-standard labeled tapes; the SVC verifies these tapes, and determines if it needs to be re-positioned.

SVC 86: ATLAS macro This is a type 4 SVC which holds no locks. It is a disk recovery system, in which the module represented by this SVC is the first load module. The SVC's primary responsibilities involve determining if the error is correctable, and also acquiring information about the disk volume the error is on (such as the number of available alternate tracks).

SVC 104: TOPCTL macro This is a type 4 SVC which holds no locks. It is used exclusively as an interface to TCAM. TCAM is not a part of the evaluated configuration, so this SVC is never called in an evaluated system.

SVC 107: MODESET macro This is a type 6 SVC, which holds no locks. Its function is to change the mode of the system by altering the information in the old PSW (i.e., the one that the program which invoked the SVC was using). The information that can be affected is the state (supervisor or problem), the PSW protection key, or the key mask.

SVC 123: PURGEDQ macro This is a type 2 SVC which holds the DISP lock. Its function is to purge Service Request Blocks from the service manager queues, ensuring that suspended SRBs have completed their processing.

SVC 126: MSS Interface This is a type 3 SVC which holds the LOCAL and CMS locks. The function of the SVC is to issue the TESTAUTH macro to verify the specified user is authorized.

RACF SVCS: 130-133 These are type 3 SVCs which hold no locks. Their functions are described on page 99.

3.1.15 Authorized Programs

Although RACF contains privileges which can be used by certain subjects, MVS/ESA recognizes a class of programs which are "authorized". A program is authorized if one of the following conditions is true: the program is in supervisor state, the program runs with a system key (keys 0-7), or the program is APF authorized.

The Authorized Program Facility (APF) enables certain programs to have special privileges. In order to gain these privileges, an APF authorized program must fulfill two conditions: it must be loaded from an authorized library, and it must be linked with AC=1 (authorization code).

Authorized libraries are specified in a configuration data set at Initial Program Load. They include:

- SYS1.LINKLIB
- SYS1.SVCLIB
- SYS1.LPALIB

IEAAPFxx is the name of the member of SYS1.PARMLIB which lists all APF-authorized data sets in the system.

Three tables also specify authorized commands:

IKJEFTNS Specifies those commands that cannot be issued in the background.

IKJEFTE2 Specifies those commands to be ATTACHed authorized.

IKJEFTE8 Specifies those programs to be ATTACHed authorized.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

Anyone can link a program with AC=1; however, the program is not authorized until it is put into one of the authorized libraries. When an authorized program calls an unauthorized program, the authorized program ABENDs. In addition, when an authorized program is dispatched for a TSO/E (foreground and background) address space, all other tasks in that address space are marked non-dispatchable until the authorized program is finished.

3.2 Data Facility Product

The MVS/Data Facility Product (DFP) performs storage management, data management, program management, and device management for IBM processors which implement the ESA System 370 architecture. DFP is the link between the processor and the storage devices, and as such does most of the reading and writing of data to storage devices. All programs running in MVS depend on the management functions and services of DFP. DFP provides these services in support of MVS/ESA objects, which are described later.

3.2.1 Device Volumes and Labels

Device volumes are the physical containers of data. Physical disk packs may contain one direct access volume while tape cartridges contain one (logical) tape volume per cartridge.

Direct Access Disk Volumes

Direct access storage device (DASD) volumes (disk volumes) are used to store executable programs, including the operating system itself. DASD storage is also used for data and for temporary working storage. A volume table of contents (VTOC) is used to account for allocated and available space on the volume.

Each DASD volume is identified by a unique volume label that is stored at track 0 of cylinder 0. Additional volume labels may follow the standard volume label, but these are not processed by the operating system. Each DASD volume is initialized by a utility program before being used on the system. The initialization program generates the volume label and builds the table of contents.

Although DASD devices differ in appearance, capacity, and speed, they are similar in data recording, data format, and programming. Each recording surface of each volume is divided into many concentric tracks. Information is recorded on all DASD volumes in a standard format. In addition to device data, each track contains a track descriptor record and data records. The track descriptor record contains: the location of the record by cylinder, head, and record numbers; its key length (for keyed or indexed data sets, or 0 if keys are not used); and its data length. A track overflow option allows a block of data that does not fit on the track to be partially written on that track and

continued on the next (adjacent) track. This adjacent track must be allocated to the same logical grouping (called data set); otherwise the write fails.

Magnetic Tape Volumes

Magnetic tape volumes are used to store data mainly for archival purposes. Magnetic tape volumes in the evaluated configuration include tape reels and tape cartridges. Within a B1 system, IBM standard or ANSI (American National Standards Institute) labels must always be used to identify the volumes.

IBM standard tape labels consist of unique volume labels and groups of data set labels. The volume label, the first record on the tape, identifies the volume and its owner. The data set groups precede (header labels) and follow (trailer labels) each data set on the volume, and identify and describe the data set. IBM standard labeled tapes contain a tapemark between the data set and its header and trailer labels, and a double tapemark after the last trailer label.

ISO/ANSI labels are similar to the formats of IBM standard labels. However, whereas ISO labeled tapes are coded in the International Standard Code for Information Interchange (SCII) and ANSI labeled tapes are coded in the equivalent American National Standard Code for Information Interchange (ASCII), IBM labeled tapes are coded either in the extended binary-coded-decimal interchange code (EBCDIC) or in binary coded decimal (BCD). ISO/ANSI labeled tapes contain a tapemark between the data set and its header and trailer labels, and a double tapemark after the last trailer label.

Data sets (files) are placed on magnetic tape immediately following the header label (of a labeled tape) and are written sequentially from the beginning of the tape. The integrity of the tape volume label is maintained because the TVTOC (explained below) is not physically stored on the tape. Authorized user(s) are given full access to the tape volume. Therefore, within a B1 system, all data sets residing on one volume will have the same mandatory access controls as the first data set on the tape, which is enforced by MVS/SP. The user should ensure that all data sets on a particular tape have the same discretionary access controls as the system does not enforce different discretionary controls per data set.

3.2.2 Volume Table of Contents

Each data set stored on a volume has its name, location, organization, and control information stored in the tape volume table of contents (TVTOC) for magnetic tape volumes, and in the volume table of contents (VTOC) for DASD volumes. In addition, each data set entry in both tables contains a discrete profile flag, creation date, and the last reference date. The TVTOCs are stored in the RACF data base while the VTOCs are stored on the disk volumes they represent.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

The DASD storage management routines control allocation of space on DASD volumes through the VTOC of that volume, and through the VTOC index if one exists. The VTOC resides in a single extent (as a contiguous sequential data set) anywhere on the volume after cylinder 0, track 0. Its address is located in the standard label of that volume. The VTOC is composed of data set control blocks (DSCBs) that correspond either to a data set currently residing on the volume, or to contiguous, unassigned tracks on the volume. DSCBs for data sets describe their characteristics. DSCBs for contiguous, unassigned tracks indicate their starting location and length.

To locate a data set by name, DFP must read each DSCB sequentially in the VTOC until it determines the location of the data set. In order to avoid lengthy sequential searches, an Indexed VTOC is optionally available. An Indexed VTOC consists of a VTOC and a VTOC index. The VTOC index is a physical-sequential data set residing on the same volume as the VTOC. It contains an entry for each data set on the volume and a pointer to the DSCB in the VTOC that describes the data set. This allows direct access to the correct DSCB, avoiding sequential searches of the VTOC. The index also manages free space information. Its name has the form of SYS1.VTOCIX.Vnnnnnn where nnnnnn represent the volume name.

3.2.3 Catalogs

In order to facilitate data set storage and retrieval, MVS/ESA, through a specialized DFP service, provides for cataloging of data sets using the integrated catalog facility. In the evaluated configuration, all user accessible data sets must be cataloged. The catalog structure consists of a master catalog and user catalogs. There is one master catalog on each MVS/ESA system, and it is allocated at system generation time. This catalog is a virtual storage access method (VSAM) key-sequenced data set containing volume security information, data set ownership and security information, and other information for VSAM and non-VSAM data sets. (VSAM and non-VSAM data sets are described below.) The master catalog contains pointers to user catalogs. The number of user catalogs is defined by the installation. User catalogs, in turn, contain pointers to the respective VSAM volume data sets or to VTOCs residing on the target volumes which ultimately point to the VSAM and non-VSAM data sets, respectively.

A system address space, called the catalog address space or CAS, contains most catalog modules and control blocks instead of their residing on commonly addressable storage. This reduces common virtual storage requirements and enhances protection by making them unaddressable by unprivileged programs.

The DFP provides several functions for manipulating catalogs. These include creating a catalog, converting a catalog, defining objects in a catalog, modifying a catalog, deleting catalog entries, and copying, merging, splitting, backing up and listing a catalog. The catalogs are only indirectly accessible to unprivileged users.

3.2.4 Data Sets

A data set (file) is a collection of logically related data records that are stored on a volume. Data sets can reside on DASD volumes or on tape volumes. DFP supplies the system and the users with a wide range of data set and volume manipulation functions.

All data sets are allocated in increments of zero or more tracks, although the user may optionally specify cylinders or blocks as units of allocation. Space is allocated based on two quantities: primary and secondary. The primary quantity is the first extent (block) allocated. For partitioned data sets, it also includes the space necessary to hold the data set directory. The secondary quantity specifies the number of additional tracks, cylinders, or blocks to be allocated, if more space is needed. A direct data set is composed of only the primary space. All other data sets, except for VSAM data sets, normally comprise up to 16 extents on each volume. The first volume on which a multi-volume data set resides typically comprises one primary extent and up to 15 secondary extents. Subsequent volumes on which a non-VSAM data set resides contain up to 16 secondary extents. On each volume on which a VSAM data set resides, the data set may comprise one extent and up to 122 secondary extents. A user accesses data sets that reside on a DASD volume. However, DASD volumes may not be accessed as an entity within the system.

All data sets cataloged within one catalog must have unique names. This restriction holds for all the data sets recognized only by name (specified without the volume identification) and all the data sets residing on a given volume. (When referring to a cataloged data set, a user needs only to specify the data set name which must be unique to the catalog. When referring to an uncataloged data set, a user must specify the volume holding the data set. That data set must be unique to that volume.) A data set name is one or more simple names joined together with periods. The first name is called the high level qualifier and it is often identical to the userid of a user owning the data set. Each simple name consists of from one to eight characters, the first of which must be alphabetic. The length of the data set name cannot exceed 44 characters.

In summary, the following is the MVS/ESA storage hierarchy. Bytes of data are grouped to form records of either fixed or variable length. Records make up blocks which make up tracks. A data set occupies at least one track. Data sets are stored on volumes and may also span volumes. A data set location, organization, protection, and some VSAM data set names are found in a VTOC, while most data set names, ownership, additional protection information, and volume location reside in a catalog. The master catalog contains information about all other catalogs and volumes.

3.2.5 Data Set Types

MVS/ESA recognizes four types of data sets: temporary data sets, spool data sets, VSAM data sets, and non-VSAM data sets. Some temporary data sets are implemented using Virtual I/O (VIO) and appear as extensions to user address spaces. They are system-owned, not shared, and never cataloged. They are allocated to jobs only for the duration of a job's existence. Other temporary data sets are non-VSAM data sets kept under strict control of DFP. VSAM data sets

Final Evaluation Report IBM MVS/ESA

CHAPTER 3. SOFTWARE ARCHITECTURE

are used by the system primarily for paging and swapping data sets and in implementing volume catalogs. They are also used by applications where the user supplies most of the access method details. Non-VSAM data sets make up a large majority of data sets used by users to store and share data.

VSAM Data Set Organization

VSAM data sets can be organized in one of three ways, each with a different set of characteristics. In a key-sequenced data set, records are loaded in key sequence. Each record must have a key, and the ordering of the records is determined by the numeric value of the keys. New records are added in key sequence. In an entry-sequenced data set, records are loaded in sequential order as they are entered. New records are added at the end of the data set. In a relative record data set, records are loaded according to a relative record number that can be assigned either by VSAM or by the user program. VSAM-numbered records are added at the end of the data set; user-numbered records can be added in relative record number sequence.

The format of a VSAM data set record is different from that of other data sets. All VSAM data set records are stored in control intervals. A control interval is a continuous segment of DASD storage. With key-sequenced data sets, the user can gain access to a record by specifying its key or its relative byte address. With entry-sequenced data sets, the user can gain access to a record only in the order the records were added to the data set. Finally, with a relative record data set, the user can gain access to a record only by specifying its relative record location.

Non-VSAM Data Set Organization

Direct access devices may contain four types of non-VSAM data sets: direct data sets, sequential data sets, indexed sequential data sets, and partitioned data sets.

Direct Data Sets There are two types of addresses that can be used to store and retrieve data in a direct data set: actual addresses and relative addresses. The actual address of a record contains a 1-byte binary number specifying the relative location of an entry in a data extent block (DEB). The DEB is created by the system when the data set is opened. Each extent entry describes a set of consecutive tracks allocated for the data set. The actual address further contains three 2-byte binary numbers specifying the cell, cylinder, and head number for the record (its track address). A 1-byte binary number specifying the relative block number on the track is also a part of the actual address.

There are two kinds of relative addresses each of which may use an actual key: relative block addresses and relative track addresses. The relative block address is a 3-byte binary number that describes the position of the block relative to the first block of the data set. This data set can be allocated with noncontiguous sets of blocks without affecting the relative block address.

The relative track address contains a 2-byte binary number specifying the position of the track relative to the first track allocated for the data set. The track position for the first track is 0. Allocation of noncontiguous sets of tracks does not affect the relative track address. This address also contains a 1-byte binary number specifying the number of the block relative to the first block on the track previously specified. This number for the first block of data on a track is 1.

In addition to the relative track or the relative block address, the address of the virtual storage location containing the record key may be specified. The system then computes the actual track address and searches for the record with the correct key.

Sequential Data Sets Sequential data sets may reside on both DASD devices and tape devices. They have the same limited set of characteristics: records are written sequentially from the beginning of the data set; new records are added to the end of the data set (extending the data set); records may be updated but they cannot be deleted or their length modified. Information must be searched from the beginning of the data set.

Indexed Sequential Data Sets Indexed sequential data sets offer many advantages over the sequential data sets. The data set can be read or written sequentially, individual records can be processed in any order, records can be deleted, and new records can be added.

The records in an indexed sequential data set are arranged according to a collating sequence by a key field in each record. Each block of records is preceded by a key field that corresponds to the key of the last record in the block.

Indexed sequential data sets reside on DASD devices. They may occupy three different areas. The prime area contains data records and related track indexes. The overflow area contains records that overflow from the prime area when new data records are added. The index area contains master and cylinder indexes associated with the data set. It exists for a data set that has a prime area occupying more than one cylinder.

Partitioned Data Sets Partitioned data sets can only be stored on DASD devices. A partitioned data set is divided into sequentially organized members, each composed of one or more records. Each member has a unique name stored in a directory that is a part of the data set. The records of a given member are written or retrieved sequentially. The individual members can be added or deleted as required. However, deleted space is not reused until the entire partitioned data set is copied or compressed.

The directory, located at the beginning of the data set, is made up of 256-byte records containing an entry for each member. Each directory entry contains the member name and the starting location of the member within the data set. The directory entries are arranged by name in alphanumeric collating sequence. The starting location of each member is recorded by the system as a relative track address (from the beginning of the data set). If there is not sufficient space available in the

**Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE**

directory for an additional entry, or not enough space available within the data set for an additional member, or no room on the volume for additional extents, no new members can be stored.

3.2.6 Access Methods

Corresponding to the data set organizations, there are four classes of access methods. An access method is a system service which a user may invoke to access data stored in a data set. JES uses its own routines in handling the spooled data sets. The remaining three types of data sets—VSAM, non-VSAM, and temporary—are handled with the virtual storage access method and the conventional access methods, respectively.

VSAM

The virtual storage access method is an access method used to organize system and user data and to maintain information about that data in a catalog. The VSAM performs catalog management and record management.

The VSAM is specifically designed to take advantage of virtual storage, and is used to access disk data; it runs in virtual storage and uses virtual storage to buffer I/O operations. The VSAM does not use the EXCP processor, but employs its versions of queued and basic access techniques, allowing it to process the three types of data sets previously described.

Information is requested from or supplied to VSAM data management in logical records. The VSAM uses control intervals to contain records. Whenever a record is retrieved from direct access storage, the entire control interval containing the record is read into a VSAM I/O buffer in virtual storage. From the VSAM buffer, the desired record is transferred to a user-defined buffer or work area in that user's address space.

A control interval is a continuous area of DASD storage that VSAM uses to store data records and control information that describes the records. The control intervals in a VSAM data set are grouped together into contiguous areas of DASD storage called control areas. A VSAM data set is actually composed of one or more control areas. The maximum size of a control area can vary between one track and one cylinder of DASD storage.

Non-VSAM

There are two techniques a program can use to access the records in a non-VSAM data set: the queued access technique or the basic access technique. The queued access technique is used when the sequence in which records are to be read or written is known to the access method. The system can anticipate which records are needed and make them available through buffering. The access

method does not return control to the program utilizing this technique until the requested I/O operation has been completed.

The basic access technique is used when no assumption can be made about the sequence in which records are to be processed. The basic technique allows access to any records in the data set. No grouping of records takes place, and no anticipation of future I/O requests occurs. The program utilizing this access technique must test for the completion of the I/O operation because the access method returns control to the program before the I/O operation is completed.

Conventional Access Methods

Conventional access methods move data to and from non-VSAM data sets. These access methods are identified by the technique they employ and the type of data organization to which they apply. MVS/ESA supports the six types of conventional access methods.

Data Set Organization	Access Methods	
	Basic Technique	Queued Technique
Sequential	BSAM	QSAM
Partitioned	BPAM, BSAM	QSAM
Indexed Sequential	BISAM	QISAM
Direct	BDAM	

VIO temporary data sets may be accessed with the BPAM, BSAM, QSAM, or BDAM access methods, or via the EXCP macro instruction.

3.2.7 Storage Management Subsystem

The Storage Management Subsystem (SMS) portion of DFP is the first attempt IBM has made toward system-managed (vs. user-managed) storage. It simplifies the management and use of external storage resources by providing a device-independent way to request data set services. SMS provides facilities that allow the storage administrator to define space, availability, and performance services and data definition attributes based on user specified needs. The administrator can control the assignment of those services and attributes to specific data sets when they are created.

The SMS manages an installation's storage according to the currently active storage management policy. This policy is defined by the installation in a source control data set. The collection of information found in this data set is referred to as an SMS configuration. The SMS configuration is composed of a set of SMS constructs, automatic class selection routines, and the base configuration.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

Constructs

Constructs are lists of traits and characteristics that are assigned to data sets and volumes. They control the allocation, performance, and availability of the data that is associated with a particular construct. An SMS configuration can contain four types of constructs: storage group, management class, storage class, and data class. The storage group allows a defined list of volumes to be managed as if they were one large, single volume. The management class allows for the definition of levels of migration, backup, and retention services. Storage class allows the definition of performance and device availability requirements, and the data class allows the definition of allocation defaults.

3.2.8 Other DFP Functions

As mentioned in the introduction to this section, DFP also provides program management and system support functions. The linkage editor combines previously compiled or assembled object modules into a program ready to be loaded and executed. It also allows users to edit program modules and selectively replace sections within the program. Program fetch is a mechanism for reading a load module into virtual storage and relocating any address constants in the module. The loader combines the basic editing functions of the linkage editor and the loading functions of program fetch into one step. It loads for execution object modules produced by language translators, and load modules produced by the linkage editor.

The checkpoint/restart facilities gather and record information about the status of a job and its related control blocks to allow a restart, should one become necessary. Execution resumes at the beginning of a job step (step restart) or from a place within a job step (checkpoint restart). Operators control step restarts. User application programs are responsible for taking checkpoints for possible future restarts.

DFP also provides a set of general purpose data set utility programs for copying, merging, loading, unloading, reblocking, comparing, updating, and printing data sets. System utility programs are also supplied. They provide the means to label magnetic tapes, locate and assign alternate tracks on a disk, rebuild defective records, list partitioned data set directories, and modify system control data.

3.3 Job Entry Subsystem

The Job Entry Subsystem (JES) is used to manage jobs before and after execution. The function of JES is to screen jobs before admission to the system, organize the necessary resources for job processing (by spooling, performing conversion/interpretation, and SYSOUT processing), and to handle job termination. This section defines some terms relating to JES and describes the functions of JES.

Two kinds of job entry subsystems are available for use with MVS: JES2 and JES3. The basic functions of reading and spooling job input, converting JCL, selecting jobs from the JES job queue, spooling and writing jobs, and purging jobs is accomplished by the "main task" or "main" for JES2 and JES3, respectively. These "JES mains" (for our discussion) perform their work through a set of programs called JES processors. IBM uses the term "JES processor" to denote an MVS/ESA system with either JES2 or JES3 and "processors" to denote the set of programs that accomplish JES job processing. Care will be exercised to ensure that the context provides clarification on the proper reference.

A fundamental difference between JES2 and JES3 is the manner in which they manage multiple JES processor environments (called JES complexes, see page 75). The JES2 concept is one of decentralized control and a common job queue—each JES2 processor operates independently of the others, but each may add jobs to or select jobs from the common job queue. JES3, in contrast, exercises centralized control over all processors. JES3 also uses a common job queue, but all jobs reach the queue through one central JES3 processor that selects and distributes work to the other processors.

Certain functions that JES2 can utilize are not allowed in a B1 environment. These include Network Job Entry and Remote Job Entry. Only certain PSF-supported printers are allowed (see page 95, "Print Services Facility"). No user exits are allowed. Daughter task elements (DTEs) and processor control elements may not be modified (see page 76). The ability to enter system level commands through the input job stream must be revoked.

For JES3, in addition to the above, automation consoles and JES3-managed local consoles are not allowed because there is no ability to support identification and authentication at these consoles. Four tape utilities are not allowed. They are: tape-to-tape, tape-to-print, tape dump, and tape label. Bulk data transfer is also not supported in this B1 evaluation. Statements that support this capability must be removed from the initialization stream.

This section will first provide information that is common to both JES2 and JES3: JCL, communication with MVS/ESA, spool data sets, and complexes. The structure and internals of each JES will then be covered separately.

3.3.1 Job Control Language, Jobs, and Tasks

The Job Control Language (JCL) for JES consists of statements which define a unit of work to be done for a job. (JES2 and JES3 JCL control statements can be embedded in the JCL and they are incompatible with each other.) A job is a unit of work given to an initiator, and all the job steps from that job will execute in that initiator (see page 49). A job step is the smallest executable portion of a job, and it is made up of one or more JCL statements required to execute a program. Job steps execute sequentially. A new MVS task is created for every job step in a job.

Three important JCL statements are JOB, DD, and EXEC. The JOB statement marks the beginning of a job, tells the system how to process the job, assigns a name to the job, and contains

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

information such as password, userid, and groupid (see example below). The DD statement is used to identify a data set and to specify the input and output resources needed for the data set. Information placed on the DD statement includes the number of copies, the access method, the destination, the name, and the volume serial number of the data set. The EXEC statement marks the beginning of each job step in the job. This statement is used to identify the program, cataloged procedure, or in-stream procedure that is to be executed. Some items found on the EXEC statement include the program name, the length of time the step may execute, and the type of storage required for the step. Other JCL statements are used for marking the end of a job and specifying options for printing data sets. An example of a JCL program follows:

```
//MYJOB      JOB      ,'JOHN SMITH',TIME=(4,30),  
//          USER=JSMITH,PASSWORD=ASDF  
//STEP1     EXEC     PGM=CREDIT  
//CHARGE    DD      DSN=CHARGE.FILE,DISP=SHR  
//STATEMNT  DD      SYSOUT=A, DCB=(LRECL=132,  
//          BLKSIZE=1320,RECFM=FB)  
//          *****  
//STEP2     EXEC     PGM=DEBIT  
//MASTER    DD      DSN=MASTER.FILE,DISP=SHR  
//PRINT     DD      SYSOUT=*  
//
```

In the above example, which is neither JES2 or JES3 specific, there are two job steps in the job MYJOB: STEP1 and STEP2, with their corresponding DD statements. The job was submitted by userid JSMITH with a password of ASDF. The first step, STEP1, executes the program named CREDIT and uses one data set as input and a SYSOUT data set for output. The second step, STEP2, executes program DEBIT, uses one data set as input, and uses SYSOUT as its output data set.

Jobs of similar characteristics and processing requirements can be assigned to a job class. There are 36 job classes, A-Z and 0-9 (the names have no inherent meaning). A job class is assigned to one or more initiators, and a job can only be started on an initiator if the job's class is assigned to the initiator. The job class is specified on the JOB statement or, if not specified, assigned based on the device through which the job entered JES. For example, if an initiator is assigned job class H and job class H has been defined by the installation as jobs with high I/O requirements, then a user can specify job class H on the JOB statement for an I/O intensive job.

3.3.2 JES and MVS Communication

Subsystem Interface

The subsystem interface (SSI) is the interface that provides communication between MVS and JES. MVS functions issue a macro to invoke JES. The calling routine uses the subsystem option block (SSOB), which contains a function code that defines the request being made and the address of the subsystem identification block (SSIB). The SSIB identifies the subsystem to which the request is to go (in this case JES) and the required processing. The calling routine uses the macro to pass the SSIB and SSOB to JES. There are about 40 SSI functions supported by JES3 and a lesser number supported by JES2.

Functional Subsystem Interface

JES uses a functional subsystem (FSS) to provide certain JES functions (e.g., an FSS to do output service). The functional subsystem interface (FSI) exists between JES and FSS to provide control of and data set services for the FSS. As an extension to JES processing, FSS processing takes place in an address space separate from the JES address space. Specific functional processing (e.g., device processing) that JES would normally perform within its own address space can be accomplished by the FSS or a number of FSSs; the FSS address space isolates this processing from JES.

3.3.3 JES Data Sets

Spool Data Sets

Spool data sets are collections of direct access volumes that contain the jobs that JES processes and the various job-related control blocks, such as the job control table (JCT) and input/output table (IOT).

JES uses spool data sets to hold a job and its associated information between various JES phases. The spool data set is a sequential data set residing on a DASD. A spool data set is contained within a spool volume, and the spool volume is defined to JES. There are different types of spool data sets, for example SYSIN, SYSOUT, and dump. A SYSIN data set contains data entered into the system input stream with JCL statements. During the output phase the jobs are queued to the SYSOUT data set for that job on the spool device. Application programs on both JESs use BSAM and QSAM (see page 68) to read SYSIN and write SYSOUT. JES3 uses its own access methods to mediate access to spool data sets called JES3 spool access method and user spool access method (see page 85). A dump data set is a backup of the spool data set. In JES complexes, different copies of JES use the same spool data sets.

Final Evaluation Report IBM MVS/ESA CHAPTER 3. SOFTWARE ARCHITECTURE

JES creates and manages several data sets for a job. Some of these are JES system or private data sets; others contain SYSIN and SYSOUT data for jobs in the system. The data sets contain job elements necessary for scheduling, executing, converter/interpreter (C/I) and set up processing, main DSP (Dynamic Support Programs) processing, and purge processing. Standard names are reserved for these data sets in order that a RACF profile may be built for them.

Other JES System Data Sets

The JES uses two other system data sets. The JES checkpoint data set, which contains information to allow communication among the processors of a complex, and the JES spool offload data set (known as the dump data set for JES3), which contains copies of spool data.

JESNEWS Data Set

JESNEWS is a text data set that is printed with job output. JESNEWS is commonly used to provide system users with information on such events as scheduled down time, changes/additions to the system and other general system and or user information. The JESNEWS data set is printed by the JES print processor immediately following the JES information at the beginning of the output separator page for JES2 and immediately after the output for JES3. The JESNEWS data set is represented by a JES2 started task (STC) and by JESNEWS control blocks in JES3. The data set is shared among all processors in a complex; when one processor updates the data set, all processors automatically pick up the updated data set.

In JES2, a job can replace or delete JESNEWS. In JES3, a job can replace, extend, or delete JESNEWS. Both JESs recognize the creation/deletion of JESNEWS and anchor the data set to the internal JESNEWS control blocks instead of to the job that changes JESNEWS.

In JES2, JESNEWS is changed by running a job that specifies a program name of JESNEWS on a SYSOUT DD statement and writes data into the data set. JESNEWS is deleted by creating a null data set. That is, by never opening the data set or opening the data set and closing the data set without writing any data into the data set. To change JESNEWS, a user must be authorized through a profile in the RACF OPERCMDS class.

In JES3, JESNEWS is changed by specifying a “//* PROCESS JESNEWS” JCL statement followed by parameters indicating the change in the job’s JCL. To change JESNEWS, the user must have update authority through a JESNEWS profile in the JESSPOOL class. In addition, the user must be authorized through a profile in the OPERCMDS profile.

3.3.4 JES Complexes

The collection of physically connected processors (executing on physically connected machines or sides) is called a JES complex. In a JES complex the job queues are shared. This allows a job's phases to be performed on any machine in the complex. For example, one machine may convert a job's JCL and place the job on the spool, while another machine may retrieve the job from the spool and execute it. Each JES phase will complete on only one machine. In a complex the queuing structure resides in the checkpoint data set.

A JES2 complex consists of from 2-7 MVS/JES2 systems supporting an installation's jobs that are located on the spool, while a JES3 complex can support up to 8 MVS/JES3 systems.

3.3.5 Job Entry Subsystem 2

A single JES2 configuration consists of one MVS/JES2 system supporting an installation's jobs. These jobs are located on a common spool. This section describes the way JES2 works, including a discussion of JES2 structure, the various phases of JES2 job processing, and the execution batch monitor function of JES2. Finally, a brief note on JES2 access methods is included.

JES2 Structure

The JES2 operates in its own address space. The JES2 address space is created as part of system initialization. JES2 consists of four load modules: HASPSSSM, HASPINIT, HASPFSSM, and HASJES20. HASPSSSM may reside in either the link pack area (LPA) or be directly loaded into common storage. HASJES20 resides in the private area of the JES2 address space. The HASPINIT load module is loaded into the JES2 private area during initialization and deleted at the end of the JES2 initialization process. The load modules are characterized as follows:

- *HASPSSSM*: MVS interfaces directly with HASPSSSM through the SSI to provide job scheduling, data management (SYSIN and SYSOUT), functional subsystem connect and disconnect, and other subsystem functions and operator communication functions.
- *HASPINIT*: HASPINIT resides in the JES2 private area during JES2 initialization. The initialization routine administrator (IRA) is entered first; it performs some preliminary initialization and invokes other JES2 initialization modules. When the other JES2 initialization modules have been called, IRA completes the initialization.
- *HASPFSSM*: HASPFSSM is made up of a single load module. It is loaded into the private area of the FSS address space during connect processing. HASPFSSM contains service routines that support the function subsystem interface.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

- *HASJES20*: The HASJES20 module is made up of source modules that perform JES2 main task and subtask processing. The JES2 main task provides the basic job processing functions mentioned in the JES Introduction. The subtasks provide services to the JES2 main task, which may involve MVS waits and, as such, are performed in subtasks, because the JES2 main task is not permitted to enter an MVS wait until all currently outstanding JES2 functions are completed.

Processor control elements (PCEs) are pointers used by the HASP communication table (HCT) to dispatch processors when an event is posted or resource is released. The HCT uses a queue of PCEs to control the dispatching of processors. When a processor is eligible for dispatching, its PCE is on a dispatcher queue called the \$READY queue. The \$READY queue can be in the following states:

- *Active Processor PCE*: The currently active processor's PCE is addressed by a field in the HCT. Processors are dispatched in FIFO order.
- *Empty \$READY Queue*: No PCEs are ready to be dispatched and the \$READY queue is empty.
- *Waiting Processing PCE*: When a processor is waiting on an event, it is ineligible for dispatching; its PCE is on one of the several wait queues depending on what the processor is waiting for. If the processor is waiting for a resource, its PCE is chained to the designated resource wait queue; if the processor is waiting for a specific event, its PCE is queued to itself.

In addition to waiting for a JES2 event or resource, a processor might be waiting for an MVS post of an event control block (ECB). The address of this PCE and the address of the specific PCE are contained in the extended ECB (EXCB). A processor is dispatched when the ECB is posted or when the event or resource is released.

The dispatcher, running under the JES2 main task, is responsible for giving control to individual JES2 processors. When a JES2 processor gets control, it runs until it relinquishes control via the \$WAIT macro. The \$WAIT macro causes control to be passed to the dispatcher, which gives control to the next available and ready JES2 processor.

Processors become ready when they are posted. This post can be a specific post, signaling the completion of an event, or a general resource post, indicating that a resource is available. Posting occurs within the JES2 main task (via \$POST) or from a JES2 subtask or other address space (via \$\$POST). JES2 subtasks and programs (running in other user address spaces) use the \$\$POST macro to inform JES2 of particular events. The \$\$POST macro causes a HASPSSM interface routine to cross memory post the JES2 main task, which can dispatch another processor.

JES2 Job Processing

During each job processing stage, control blocks and JES2-managed data sets are created and/or used to represent a job. The major JES2 job-related control blocks are:

- The JQE (Job Queue Element), used to represent a job being processed or awaiting processing by JES2. The JQE contains several queues of job output elements (JOEs), which describe current system output requests.
- The JCT (Job Control Table), used to store information about job characteristics. The JCT also contains the offset in the job queue table for the corresponding JQE.
- The IOT (Input/Output Table), used to represent the spool space allocated to the job. The JCT points to the IOT, and the IOT points to the first JCL block.
- The PDDB (Peripheral Data Definition Block), in the IOT used to represent a JES2-managed data set for the job. A PDDB is assigned to each SYSOUT data set, console message, and operating system message pertaining to a job.

The spool data set is used to store job-related control blocks (except the JQE and JOEs) and the JES2-managed data sets associated with jobs (JCL, SYSIN, converter/interpreter text, and SYSOUT). The purpose of storing job information on the spool data set is to optimize storage use and facilitate multi-access spool processing. The checkpoint data set is used to store JQEs and JOEs. The purpose of the checkpoint data set is to facilitate JES2 warm, quick, and hot restarts.

JES2 divides job processing into six phases: reading jobs into the system, converting jobs to internal form, selecting jobs for execution, preparing jobs' output for printing, placing jobs' output on the hardcopy queue, and purging jobs from the system. More detailed descriptions of these phases follow.

Input Input streams can come from a magnetic tape or a direct access device. JES2 can also receive input streams from internal readers. The input streams are data sets that other programs can use to submit jobs, control statements, and commands. Any job can use an internal reader to pass input streams to JES2. There can be up to 255 internal readers on a system, and the number of internal readers affects the number of jobs JES2 can receive simultaneously. A job ID is assigned to each input stream as it comes in. The job's JCL, optional JES2 control statements, and input data are placed into spool data sets.

Conversion JES2 invokes the MVS converter to scan each job's JCL for syntax errors. The MVS converter converts the JCL to internal text, which JES2 writes to spool. JES2 queues a job with correct JCL to the execution queue by its priority. For a job with JCL errors, JES2 bypasses the execution stage and queues the JCL together with appropriate diagnostic messages to the output queue for direct use by the output stage.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

Execution An eligible initiator uses the SSI to request a job from JES2 so that the job can be run. JES2 selects a job for execution from the execution queue. JES2 reads input from spool and writes output to it during the execution of the processing program. When the job completes execution, the job termination portion of MVS informs JES2 (via the SSI). JES2 then places the job in an output queue to await processing of its output. The job queue resides in the JES address space in main memory, and a copy is kept in the checkpoint data set. The job queue contains the following:

- Jobs waiting to run
- Jobs currently running
- Jobs waiting for their output to be produced
- Jobs having their output produced
- Jobs waiting to be purged from the system

To process the jobs on the job queue, JES2 waits for an initiator to ask for a job. JES2 keeps track of what job class or job classes are assigned to the initiator and in what order the job classes should be searched for a job. Jobs are selected by priority and then passed to the initiator. After JES2 selects the job from available jobs for the initiator and passes the job to it, the initiator invokes the interpreter to build control blocks from the Converter/Interpreter text that the converter created for the job. The interpreter builds these control blocks in the scheduler work area (SWA) of the initiator's address space. The initiator then allocates the I/O devices specified in the JCL for the first step of the job. This allocation ensures that the devices are available before the job step starts running. The initiator then starts the program requested in the EXEC statement.

Output JES2 takes a particular job, after it completes execution, and processes all of its SYSOUT data sets by creating JOEs and placing them in the job output table (JOT). JES2 builds JOEs based on the output characteristics of the SYSOUT data sets for the job.

Hardcopy JES2 selects JOEs for processing from the JOT according to work specified for the device (its priority). The selected output can be in a number of states: output from the job to be processed locally, output from the job to be processed at a remote location (not allowed in a B1 configuration), the job output itself just passing through this JES2 system and which must be transmitted to another JES2 system in an NJE network (also not allowed in a B1 configuration), or output from the job to be processed by a functional subsystem.

The use of priority for output is an installation's choice; as a default, priority is not used. If priority is used, a user can specify the level (1-255) at which the SYSOUT data set enters the output queue. If the priority is not specified on the JOB statement, an output priority is calculated based on the

number of lines of print and the number of pages. The job's print data sets are queued to the hardcopy queue.

Purge After all processing for a job is finished, JES releases all spool space belonging to the job and notifies the operator that the job has been purged. The operator can use the Message Processing Facility to suppress these messages at the console. The messages are then stored in a log data set instead of being sent to the console.

Because spool space is a reusable resource, the following information is used when jobs are purged from the system. JES2 prevents the same space from being allocated to more than one job at a time by using the master track group bit map (MTGBM). The BLOB (Bunch of Loose Old Bits) is a JES2 control block used to record addresses of available track groups. The BLOB consists of track group blocks (TGBs), each of which represents one track group. The BLOB is refilled periodically from the MTGBM. The track group allocation entries (TGAEs) are in the job's IOT and contain the address of the track groups assigned to the job, and the JCT spools allowed mask contains a bit for every possible spool volume.

Execution Batch Monitor

When all initiators are busy, throughput of certain jobs might fall below normal expectations. To help in these situations, JES2 performs an additional scheduling function that attempts to reduce the time required to schedule jobs. This scheduling function is the execution batch monitor (XBM).

XBM is an extension of normal JES2 job scheduling that helps to increase throughput by reducing the JCL required for certain types of jobs. Jobs eligible for XBM are of relatively short duration, particularly single-step jobs that have common device setup requirements and that are run frequently.

XBM support is significantly simplified, such that it is only implemented in JES2 input services. In subsequent processing, XBM input streams appear to the rest of JES2 and MVS components as normal batch jobs. XBM jobs are often referred to as joblets because they do not require all the other JCL usually needed to get a job through the system.

The JES2 input service processing was changed to place the following in the JCL input data for each XBM joblet:

- The generated JCL for the EXEC statement with the XBM procedure name
- The generated SYSIN DD statement

This allows the user's JCL to remain unchanged.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

The JOB statement, along with all the data statements supplied by the user, are still placed in the SYSIN data set. This allows the XBMs to remain unchanged. However, monitors that are expected to remain active over the life of more than one joblet now see only one joblet. This version of XBM processes each job individually, and does not support the additional read. This is the method currently used for stopping an unneeded monitor.

JES2 Access Methods

The JES2 uses conventional access methods for determining access control. See page 68 for a discussion on QSAM and BSAM.

3.3.6 Job Entry Subsystem 3

The JES3 installations can have from one to eight processors. If an installation has only one processor, that processor is called the global. If there are two or more processors, one processor becomes the global, and all others become locals.

The JES3 services the job processing requirements of all JES3 processors in a complex. JES3 refers to the processors as mains. Each main represents a single instance of JES3. Each instance of JES3 in the complex is identified by the MAINPROC initialization statement. The global processor is called the global main, while the other instances of JES3 are called local mains.

A local main participates in JES3 processing by performing some of the JES3 functions, as determined by the global main. The parts of the JES3 program that are executed on the local main handle the interface between MVS and the JES3 address space on the local main and the interface between jobs executing on the local main and the global main's services. Functions performed on local mains can also be performed on the global; however, the reverse is not true.

This section describes the operation of JES3, beginning with a discussion of JES3 structure. The various phases of JES3 job processing is addressed next. The function that handles JES3 spool data management is covered. Finally, JES3 access methods are addressed.

JES3 Structure

There are three functional areas of JES3: initialization, JES3 processing, and JES3 support processing. Initialization consists of routines that prepare JES3 to process jobs by creating the necessary control blocks and establishing channel-to-channel communication between the global and local processors.

The JES3 processing consists of four stages. Job input management deals with receiving jobs, preparing jobs for processing and the actual processing of the JES3 job through the scheduling done by the JSS. Job resource management deals with scheduling devices required by a particular

job. Job scheduling deals with how jobs are selected for processing within a JES3 complex. Job output and termination deals with delivery of the results of JES3 job processing and termination of job processing.

The JES3 support processing consists of four areas of peripheral JES3 processing. Complex management provides the options available to the operator during processing and handles the JES3 job queue and data areas. Spool data management controls space allocation and access to JES3 data on direct access storage. JES3 communication is that part of JES support processing that governs communication within JES3 and between JES3 and MVS. Remote processing examines all facets of remote workstation and job network processing (neither of which is allowable in this B1 evaluation).

JES3 Job Processing

The JES3 job processing is divided into four areas: job input management, job resource management, job scheduling, and job output and termination. Each will be covered in turn.

Job Input Management The first part of JES3 job processing is called job input management and contains three functions:

- JES3 input routines that create scheduler elements, process control statements, and add jobs to the job queue.
- Converter/interpreter (C/I) routines that determine what resources a job requires during execution (available devices, volumes, data sets, etc.), and then convert these data into internal control blocks.
- The job segment scheduler which controls the flow of work within the JES3 by scheduling job scheduling elements.

The modules that provide this service control the processing at the beginning of a typical MVS job. Input routines create scheduler elements that represent jobs to JES3, process control statements, and add jobs to the job queue. Input service accepts and queues all jobs entering the JES3 system. The global processor reads the job into the system from a TSO SUBMIT command and the internal reader. This service reads from the input source and adds jobs to the job queue. Input processing occurs in two phases: reader processing and control statement processing. The reader phase reads jobs from any of the sources mentioned above and places the jobs on spool in batches for later processing. For each reader batch, an input service job is created.

Control statement processing begins as each batch job produced by the reader phase is read from the spool. Under the ISDRVR DSP, the JES3 control blocks are modified with the information

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

retrieved from the job's control statements. The control blocks are written to spool and later used by JES3 functions in determining processing requirements for the job and its output data.

The C/I is the first scheduler element for every standard job. After a job passes through this first segment of processing, JES3 determines what resources the job will require during execution. C/I routines provide input to the main device scheduler (MDS) routines by determining available devices, volumes, and data sets. These service routines process the job's JCL to create control blocks for setup and also prevent jobs with JCL errors from continuing in the system. The C/I section of setup processing is further divided into three phases: MVS converter/interpreter processing, prescan processing, and postscan processing.

The first two phases can occur in either the JES3 address space on the global processor or in the C/I functional subsystem address space on either the local or the global processor.

The JES3 invokes the MVS C/I function to process the JCL statements in a job stream. Converter processing converts the JCL to C/I internal text and flushes out jobs with JCL errors. If the converter detects any JCL syntax errors or logic errors, the job is printed and purged (and the MAIN scheduler element is bypassed). If there are no errors, the interpreter routines then convert the C/I text to scheduler work area (SWA) control blocks. The interpreter routines build all SWA blocks above the 16 megabyte line in virtual storage in the JES3 global and C/I FSS address spaces.

The prescan phase uses the SWA control blocks to build tables used for postscan processing. Prescan builds the intermediate job summary table, the locate request table, and the job volume table; these specify required resources for the job. Prescan also writes SWA control blocks to spool for later use when the job executes.

Postscan accomplishes the following:

- Invokes the locate routine, which involves catalog management.
- Constructs the job summary table (JST), which contains information about the devices and data sets needed by the job.
- Constructs the final version of the job control table (JCT), which holds the volume serial number of every volume needed during MVS execution for this job.

Three functional areas make up JES3 scheduling:

- Management of job numbers via routines that provide central services for the allocation and control of job numbers.
- Methods that provide access to the job queue elements (JQEs) and JCT.
- Determining when jobs will run by scheduling job segments in the JSS.

Job Resource Management The second part of JES3 processing is called job resource management, which provides for the effective use of system resources. The JES3 resource management, commonly referred to as setup, ensures the operative use of non-sharable mountable volumes, eliminates operator intervention during job execution, and performs data set serialization. It oversees specific types of pre-execution job setup and generally prepares all necessary resources to process the job. The main device scheduler routines use resource tables and allocation algorithms to satisfy a job's requirements through the allocation of volumes and devices, and, if necessary, the serialization of data sets.

JES3 setup processing is defined by JES3 initialization statement parameters, JCL control statements, and JES3 operator commands. Locate routines are invoked by the postscan phase of converter/interpreter processing, where a determination is made as to which processor will perform the locate function. Main device scheduling (MDS) represents the second phase of setup processing. The converter/interpreter routines construct a job summary table (JST) that lists required data sets and devices, and a job volume table (JVT) that describes the volumes the MDS routines will fetch and allocate. Verify processing checks the volume serial number after a volume is mounted and passes the results back to MDS. The verify function is invoked when it is notified by the allocate phase of MDS, the verification phase of MDS, and the breakdown phase of MDS. When verify processing is invoked by the breakdown phase of MDS, it unloads a volume.

Dynamic allocation allows the user to allocate the devices, volumes, and data sets to the jobs after they have started to execute. Users can delay specifying device, volume, and data set resources until execution of jobs. To use dynamic allocation, TSO users use the TSO ALLOCATE command which invokes an MVS DYNALLOC macro or include an SVC 99 instruction in their program. MVS uses the SSI to forward the request to JES3.

Job Scheduling The third part of JES3 job processing is called job scheduling, and is the group of routines that govern where and when MVS execution of a job occurs. Job scheduling involves the routines invoked by the MAIN Dynamic Support Programs (DSPs), which are represented by the MAIN scheduler elements of the job control table entry, and therefore constitute the focal point of the overall JES3 process. Through the job scheduling routines, the global processor communicates with all other processors. There are four categories of modules in job scheduling: main service processing, generalized main scheduling (GMS), deadline scheduling, and dependent job control.

Main service selects a job to be processed by MVS initiators. Main service selects a job for execution using the job selection algorithms established at JES3 initialization. The MAINPROC, SELECT, CLASS, and GROUP initialization statements control the key variables in the job scheduling and job execution process.

These routines will schedule a job after it is placed on the selection queue, control the workload, and maximize system throughput. GMS provides a framework for establishing priority between job classes within groups and between groups.

Deadline scheduling and dependent job control are two ways of altering the job selection process.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

Job Output And Termination The last segment of JES3 job processing examines the final phase of any JES3 job. Output service routines operate in various phases to process SYSOUT data sets destined for print or punch devices, TSO users, internal readers, external writers, and writer functional subsystems. The three phases of output service run in different address spaces. Purge processing removes the job structure and records data. The three phases of the output service function are (1) queueing of output, (2) scheduling of output, and (3) printing and punching of output. Phases 1 and 2 occur in the JES3 global address space on the global processor. Phase 3 can run in the global address space under the global primary task, the global auxiliary task, or the functional subsystem.

Purge processing represents the last scheduler element of any JES3 job; that is, the last processing step for any job. It releases the resources used during the job and uses the System Management Facility (SMF) to record statistics. Purge processing consists of the following steps:

- JES3 releases all spool space assigned to the job and updates resident control blocks.
- JES3 writes SMF record type 25 which contains device allocation information.
- JES3 writes SMF record type 26 which contains final job accounting information.
- JES3 informs the operator that the job has been purged.
- JES3 deletes the job control table entry for the job.

JES3 Support Processing

Complex Management The JES3 complex management governs communications with the system operator and also produces various utility functions. Console service routines provide communication between the operator and JES3. The JES3 inquire/modify routines provide the ability to obtain information about the processing status of a given job or JES3 function and allow modification of parameters that affect one or more jobs and JES3.

Utilities handle many types of special operator services and enhance the capabilities provided by inquiry functions. The remaining functions are JES3 Monitoring Facility, general routines, and abnormal program termination.

Spool Data Management The spool data management function handles the JES3 spool data and refers to that set of DASDs especially set aside for JES3. Spool data management routines control the allocation, access, and deallocation of space on these designated DASDs.

The JES3 records several types of data on the spool data sets. These include (1) information (originally taken from the initialization statements) necessary to initialize JES3 in the global and local processors, (2) JES3 control blocks that define the scheduling and operational characteristics of the jobs, and (3) SYSIN (DD * or DD DATA) data sets and SYSOUT data sets for jobs.

Spool data management can be divided into three topics:

- Spool space management, which deals with the allocation of space on JES3 DASD.
- The access methods: JES3 spool access method (JSAM), the user spool access method (USAM), and the block spooler.
- Spool I/O scheduling.

JES3 Access Methods

The JES3 uses two techniques for recording spooled data: single record files (SRFs) and multirecord files (MRFs). Control blocks are recorded as SRFs. This means that a control block is recorded as a single buffer image (a block of data on a spool volume). While most of the control blocks will fit into a single buffer image, some (due to their variable lengths and multiple parts) may extend across multiple buffers. When this occurs, the buffers are chained together into chained single record files. MRFs consist of JCL, SYSIN, and SYSOUT recorded as data records, placed into spool buffers, chained together, and written to spool.

The recording and retrieval of spool data are the responsibilities of three access methods. They are:

- JSAM, used by JES3 modules to read and write SRFs and MRFs.
- USAM, used by programs in the user address spaces to read SYSIN data and write SYSOUT data.
- The block spooler, used to read and write spool data in a writer FSS address space.

JES3 Spool Access Method The JSAM is a collection of routines invoked by macro calls to acquire spool space, create spool files, read and write records, and to purge SRFs and MRFs when they are no longer needed. JSAM buffer pool management procures space for SRFs and MRFs in JES3 private storage.

The single track table is a section of spool space used exclusively for JES3 control blocks not associated with a single job, such as control blocks used to track JES3 functions and to save status. The allocation mechanism of the single track table is by record as opposed to track group, in contrast to the rest of JES3 spool space allocation.

User Spool Access Method The USAM provides user access to SYSIN data and the creation of SYSOUT data sets. The MVS data management access method macros (BSAM and QSAM) pass control to JES3 by the compatibility interface to allocate spool space for output data sets.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

The USAM thereby provides for the opening, closing, reading, and writing of data sets to JES3 spool.

The USAM buffer pool management involves user buffer pools. The user storage buffer pools are inside the user address space and consist of either one page of virtual storage for a SYSIN data set or multiple pages for SYSOUT data sets. The contents of user storage buffers are not transmitted directly to or from spool volumes, but rather are moved to or from the USAM protected buffer pools that reside in the common save areas (CSA) or in a JES3 auxiliary address space (AUX).

The Block Spooler This module is invoked by the writer functional subsystem spool input routine to read blocks of SYSOUT data from spool and to read and write writer checkpoint records that are used by FSS writers. The block spooler reads data one track at a time (when possible) into buffers in the USAM buffer pool.

3.4 ACF/VTAM

Advanced Communications Function for the Virtual Telecommunications Access Method (ACF/VTAM also referred to as VTAM) provides the data communication capability used by the system to interconnect the main processor and certain devices in a data communication network. The components within a data communication network include the processor (either a 3090 or a 4381), cluster controllers (model 3174 and 3274) and terminals (e.g., 3178 and 3179). Components must be defined to VTAM by a system administrator before they may become active. Defining a component causes a symbolic name, a local address, the identity of an initial application program (optional), the local address of the controlling physical unit, and physical characteristics (e.g., screen size, line length) to be stored in the SYS1.VTAMLST data set.

The term network used in this section refers only to the data communication network mentioned above. A VTAM network in the evaluated configuration can only have components listed in Appendix A, "Evaluated Hardware Components." Including any other networking hardware or connecting the evaluated system to another system invalidates the rating.

The ACF/VTAM's functions are starting and stopping the network, changing the network configuration, assisting in establishing communications in the network, and sending and receiving messages. The VTAM allows a VTAM application program (e.g., TSO/E) to communicate with terminals using symbolic names. Any program which uses VTAM to send and receive data between it and a terminal (or another program) is a VTAM application program. The use of symbolic names by an application program reduces the complexity of the application program. These symbolic names allow a VTAM application program to be unaware of how a terminal is attached to the communication network (directly or through a controller), where the terminal is located (network address), or if any intermediate devices exist. If a VTAM application program is non-APF-authorized, VTAM calls RACF to see if it can communicate with the application.

The SYS1.VTAMLST is a partitioned data set, with each member identifying a major node. A major node is a set of resources that can be independently activated or deactivated as a group. Examples of major nodes are a set of VTAM application programs a user can log onto during the day, a set of VTAM application programs a user can log onto in the evening, and a set of local terminals and their controller. Changes to SYS1.VTAMLST become effective each time an inactive major node is activated. Major nodes are made up of minor nodes. A minor node is a uniquely defined resource in VTAM (e.g., a physical unit or a logical unit). A table in the VTAM address space called the Resource Definition Table contains entries for all minor nodes and application programs. These entries contain information including minor node name, application name, and whether the entry is active or inactive.

3.4.1 Network Addressable Units

The VTAM data communication network includes components which can transmit or receive data. These components are known as network addressable units (NAUs). An NAU can be either a device, a program, or a portion of ACF/VTAM. There are three types of network addressable units: the system service control point (SSCP), physical units (PUs), and logical units (LUs).

An SSCP is a component of VTAM which runs within the host machine and manages the network. The SSCP takes part in network start up and shutdown, initiating and terminating communication between NAUs, and network recovery. There is exactly one SSCP per machine (e.g., 3090 or 4381).

A PU is the microcode within a physical device which controls the device, or for the PU in the host machine, it is VTAM code. Each physical device is controlled by a PU, although this PU need not be unique for each physical device. Every system and cluster controller contains a PU. The controlling PU of a terminal may either be within the terminal or within the controlling unit (i.e., cluster controller or processor) to which the terminal is attached. A PU is known to VTAM by its channel unit address (see page 25, "Channel Subsystem").

An LU is programming or built-in logic associated with either a terminal or an application program. The VTAM considers an LU as the source of a request coming into the network (i.e., from either an application program or a terminal). Figure 3.4 depicts a sample VTAM data communication network and identifies its NAUs.

3.4.2 Sessions

Before communication between two NAUs occurs, a "session" must be established. A session is the logical connection established between two NAUs. There are four types of sessions that can be established: SSCP-PU sessions, SSCP-LU sessions, LU-LU sessions, and SSCP-SSCP sessions. SSCP-PU, SSCP-LU, and SSCP-SSCP sessions are called support sessions.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

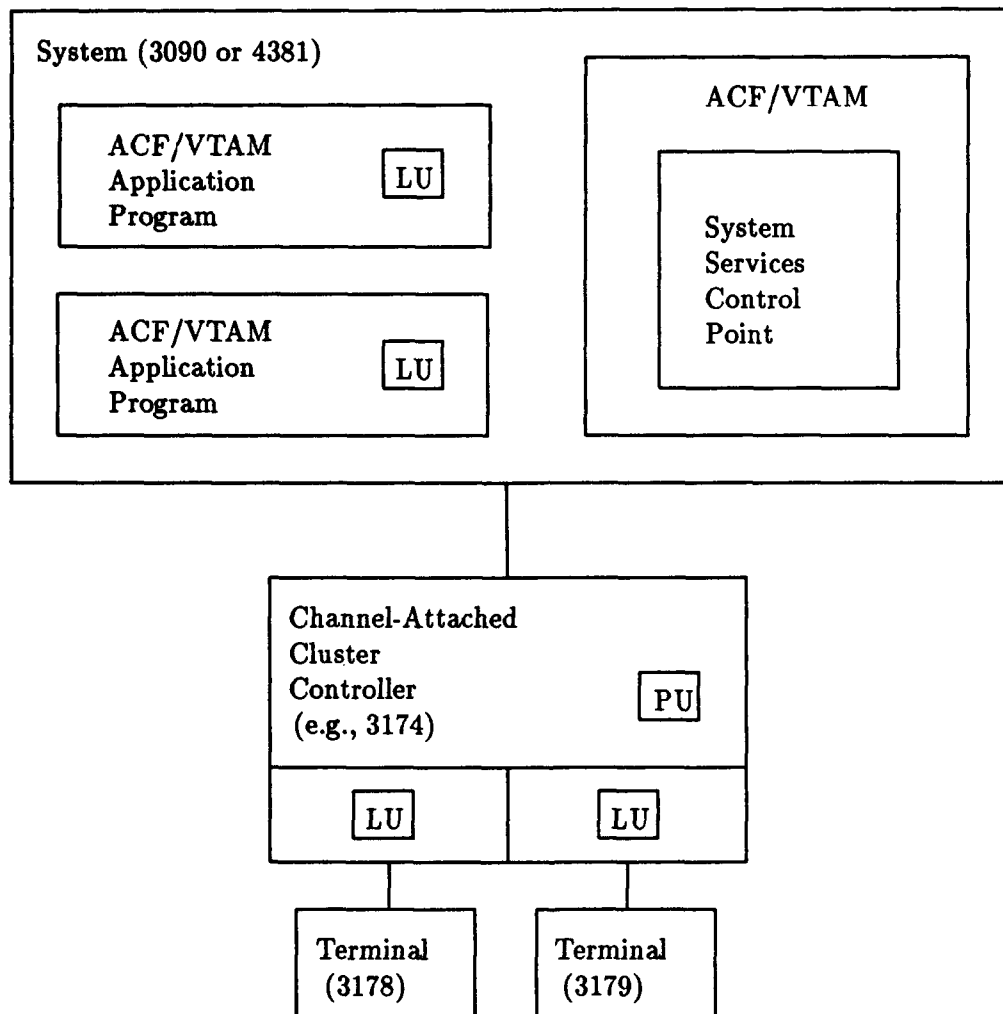


Figure 3.4. Example VTAM Network

An SSCP-PU session is established for each PU when the network is initiated. These sessions can only be established if the resource has been defined to VTAM and remains active until a PU is no longer in service (e.g., powered down). After an SSCP-PU session is established, SSCP-LU sessions are established for each LU controlled by the PU.

An LU-LU session is established whenever one LU wishes to begin communications with another LU. Requests to start an LU-LU session may come from (1) the secondary LU or operator (this is the most common LU-LU connection as it is used for application logons), (2) the primary LU (an application program), (3) the VTAM operator, or (4) an application LU (an application LU in session with a terminal LU can pass that session to another application through a call to the SSCP). An LU associated with a terminal may have only one LU-LU session, while an LU associated with a VTAM application program may have many LU-LU sessions. Data transmission between LUs is performed by the VTAM SEND and RECEIVE commands.

Finally, an SSCP-SSCP session occurs only between two systems within a JES complex. Each processor contains one SSCP, which controls a set of PUs and LUs. The set of PUs and LUs (i.e., application programs and devices) controlled by an SSCP is known as a domain. An SSCP-SSCP session is used to communicate across domains. The SSCP-SSCP sessions allow an LU to transparently communicate with an LU in another domain. In order for resources from one domain to communicate with resources from another, the domains and the resources must be defined to each other. These definitions are put in major nodes called the Cross Domain Resource Manager node and the Cross Domain Resources node.

3.4.3 VTAM Application Program Interface

The ACF/VTAM consists of a VTAM address space and a VTAM application program interface. The VTAM address space is responsible for maintaining information describing the VTAM configuration (e.g., physical address of a terminal). The VTAM address space is also the address space that is used to create channel control programs that pass information to and from the terminals. The VTAM application program interface (API) is the part of VTAM that executes in each client address space. The VTAM API is scheduled as a task for each client and is used for communication between VTAM and the client.

The ACF/VTAM API provides four types of macros that allow application programs to request services. These four types of macros are declarative macros, manipulative macros, ACB-based macros, and RPL-based macros.

Declarative macros are used by an application program to build control blocks. These control blocks describe the application program (Access method Control Block—ACB), its exits (EXitLiST—EXLIST), its session communication requirements (Node Initialization Block—NIB), and its required parameters (Request Parameter List—RPL).

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

Manipulative macros access or modify fields within control blocks. These macros provide a more consistent and convenient method for manipulating control blocks than application-issued assembler instruction.

There are only two ACB-based macro instructions: OPEN and CLOSE. These macros are used to inform VTAM that the application program is beginning or ending its use of VTAM services. When an application program issues the OPEN ACB macro, it is allowed to access VTAM resources. If the application is non-APF-authorized, then VTAM calls RACF for an authority check (authority to open the ACB). Non-APF-authorized resources which need to use the services of VTAM must be defined to the VTAMAPPL class in RACF (see page 103, "Resource Classes").

The RPL-based macros are used to request session establishment, data transfer, and program operator control. Each of these macros accepts an RPL control block which contains parameters for the macro.

The VTAM API provides macros that allow an application program to send information via VTAM. The VTAM API copies data and macro parameters from the private area of an address space into CSA. Once in CSA the data and parameters are visible to the VTAM address space. The VTAM collects multiple requests for a controller before generating the channel program which is passed to IOS to perform the terminal I/O.

3.4.4 VTAM Terminal I/O Coordinator

The Time Sharing Option Extensions (TSO/E) interface to VTAM is more complex than the interface just described. The TSO/E is a program product that is older than VTAM, and the interface between TSO/E and VTAM is not directly compatible (see page 91, "TSO/E"). Therefore, an interface program was developed called Virtual Telecommunications I/O Coordinator (VTIOC), which is responsible for monitoring the information flowing between TSO/E and VTAM, and translating it into the form accepted by each program product (e.g., TSO/E TPUT and TGET macros are translated into VTAM SEND and RECEIVE macros).

3.4.5 Application Programs

As mentioned earlier, when a terminal is powered on, a session is established between the terminal and the VTAM application program specified when the terminal was defined to VTAM. If no VTAM application program is specified for a terminal, the user must instruct VTAM to connect the terminal to a VTAM application program. These VTAM application programs are by default any programs identified in the SYS1.VTAMLST data set. When TSO/E is the application program, the terminal control address space is connected to the terminal to handle logons. Every terminal that can be connected to TSO/E has a unique application program identity (e.g., TCAS00001, TCAS00002). In a B1 environment non-APF authorized VTAM applications are not allowed. This

is controlled by activating the VTAMAPPL resource class in RACF (see page 103, "Resource Classes").

When a terminal is powered on, VTAM establishes a session between an SSCP and the terminal's LU. If an application program is specified for the terminal, VTAM establishes an LU-LU session between the terminal and the application program.

3.5 TSO/E

Time Sharing Option Extensions (TSO/E) allows users to share computer time and resources, and it supplies services to process user commands and batch jobs interactively. The TSO/E's interface to the user is the Terminal Monitor Program (TMP), which is started in a user address space during logon and which communicates with a user through VTAM. The following paragraphs will describe TSO/E logon processing, TMP functions, the UADS data set, the TSO/E interface with VTAM, and message commands.

3.5.1 TSO/E Logon Processing

During system generation, TSO/E is defined to VTAM so that VTAM will recognize TSO/E and pass commands to it. When a user types the logon command at a terminal, the user is communicating with VTAM. The VTAM cross memory posts to the Terminal Control Address Space (TCAS).

The TCAS, as its name implies, executes within its own address space which is established during system IPL. The TCAS waits for cross memory posts from VTAM indicating a logon request at a terminal. TCAS then causes the master scheduler to create a new address space for the user logging on. This new address space requests a terminal session from TCAS, which in turn requests a terminal session from VTAM. The TCAS moves the terminal session to the new address space and terminates its session with the terminal. The new address space is executing a copy of TCAS and is referenced as TCASnnnnn (e.g., TCAS00001 or TCAS00002). TCAS continues to wait for signals from VTAM indicating logon requests. Note that the user's terminal is always connected to VTAM and that VTAM transparently passes information between TSO/E and the terminal (see below, "TSO/E Interface with VTAM").

At this point user identification and authentication takes place by a call to RACF. If the user is allowed to log on, the user's logon procedure is retrieved and given to the new address space's STC. A logon procedure contains JCL that executes the required programs (e.g., TMP, see below) and allocates the required data sets needed to use TSO/E. The STC passes the JCL to a TSO internal reader in JES which in turn gives the job to an initiator. The STC links to the initiator with the logon procedure JCL, and the logon procedure is then started in the new address space. This is when a user sees the TSO/E prompt on the screen indicating that TMP is running. At this point

the user could stay in TMP or run the Interactive System Productivity Facility (ISPF) panels. The ISPF is an unauthorized program which provides users menus for executing commands.

3.5.2 The Terminal Monitor Program

The Terminal Monitor Program (TMP) is responsible for obtaining commands from the user, checking the validity of the commands, attaching the command processor, handling command processor abends, handling user attention requests, detaching the command processor, and monitoring for logoff.

During execution, the TMP uses TSO service routines to obtain user commands. The operations offered by service routines include performing I/O operations to terminals, searching input buffers, and allocating or freeing data sets. When service routines are operating for a task, they perform at the same task level as the task.

If a valid TSO/E command was entered, the TMP determines whether an operation is to be performed or a program/command processor is to be executed. Operations include displaying second level messages (e.g., help or the time) and performing no-ops (e.g., carriage return). Command processors are problem state or APF-authorized programs that perform TSO/E commands. Some command processors invoke system utilities (e.g., a compiler). Service routines are also used by the command processors.

Command processors are attached as tasks under the TMP to perform the TSO/E command. The TSO/E commands and operations normally run unauthorized. If the command/program or operation needs to run authorized, it must be included in the authorized command/program table, even if the program is already in an authorized library.

After a command processor completes normally, the TMP detaches the task.

When an abend occurs, if the error occurred in a program/command processor, control is passed to its recovery routine. If this recovery routine fails, then control is passed to the TMP recovery routine. Only when the recovery routine successfully completes will the command processor continue. If the error occurred within the TMP, then the TMP attempts recovery, displaying the TSO/E prompt if successful.

Another function of the TMP is to monitor attention requests from the user. The requests occur during command processing and are handled for both problem state and APF authorized command processing. When the attention key is entered, control is passed to the TMP exit routine and the TSO/E prompt is displayed as long as the program/command processor has not set up its own attention routine. For problem state processing, the exit routine scans for an operation. For example, the no-op operation causes the command to resume. If an attention occurred during APF authorized processing, then processing is terminated.

The TMP executes as a task until an operator issues a CANCEL command against the user or the user requests a logoff. At that point the TMP is terminated and returns control to the logon/logoff scheduler which terminates the session and returns control to the STC.

3.5.3 User Attributes Data Set

The TSO/E may use a partitioned data set containing user attributes for each TSO/E defined user. This data set, SYS1.UADS, regulates access to the system by maintaining information about the user identity, (TSO) password, account numbers, and procedure names unless the user is defined to RACF. When RACF is being used, then RACF is used for managing the password. With RACF version 1.9, TSO user attributes may be kept in the RACF data base instead of SYS1.UADS, and in a B1 environment TSO user attributes must be kept in the RACF data base (see page 100, "RACF Data Base" for a description of the RACF data base). This allows an installation to centralize TSO attributes with other user attributes.

During identification and authentication, TSO/E will invoke RACF to check the data base first to verify the user. If the userid is not found by RACF in the data base, TSO/E will then check SYS1.UADS. If the userid is not contained in either one, the logon attempt fails. If a user attempts to log onto TSO/E, and the user is defined to RACF but doesn't have a TSO/E segment in the RACF data base or any information in SYS1.UADS, then the user is denied access. If a userid is defined only in SYS1.UADS, a user would be logged onto TSO/E but would be unable to access resources defined to and protected by RACF.

If RACF is deactivated for maintenance, TSO/E reverts to the SYS1.UADS data set to check for authorized access to the system. A limited number of TSO/E accounts for system programmers and administrators are usually maintained in SYS1.UADS for this situation. In a B1 environment the TFM advises that one duplicate user be defined in SYS1.UADS for recovery purposes.

3.5.4 TSO/E Interface with VTAM

Data from TSO/E is passed to VTAM via VTAM Terminal I/O Coordinator (VTIOC) buffers (see Fig. 3.5). The VTIOC translates the I/O macros of TSO/E (e.g., TGET and TPUT) into VTAM SEND or RECEIVE macros. Then VTIOC communicates with the terminal through VTAM's Application Program Interface.

The TPUT and TGET are macros which provide simple I/O. They are the basic macros used by TSO/E to transmit and receive a line of data: TPUT transmits data from a VTAM I/O routine to a terminal; TGET obtains data from a terminal and passes it to a VTAM I/O routine. The TPUT and TGET macros are the interface to the terminal via VTAM. The information transmitted via TPUT or TGET is called a TPUT message.

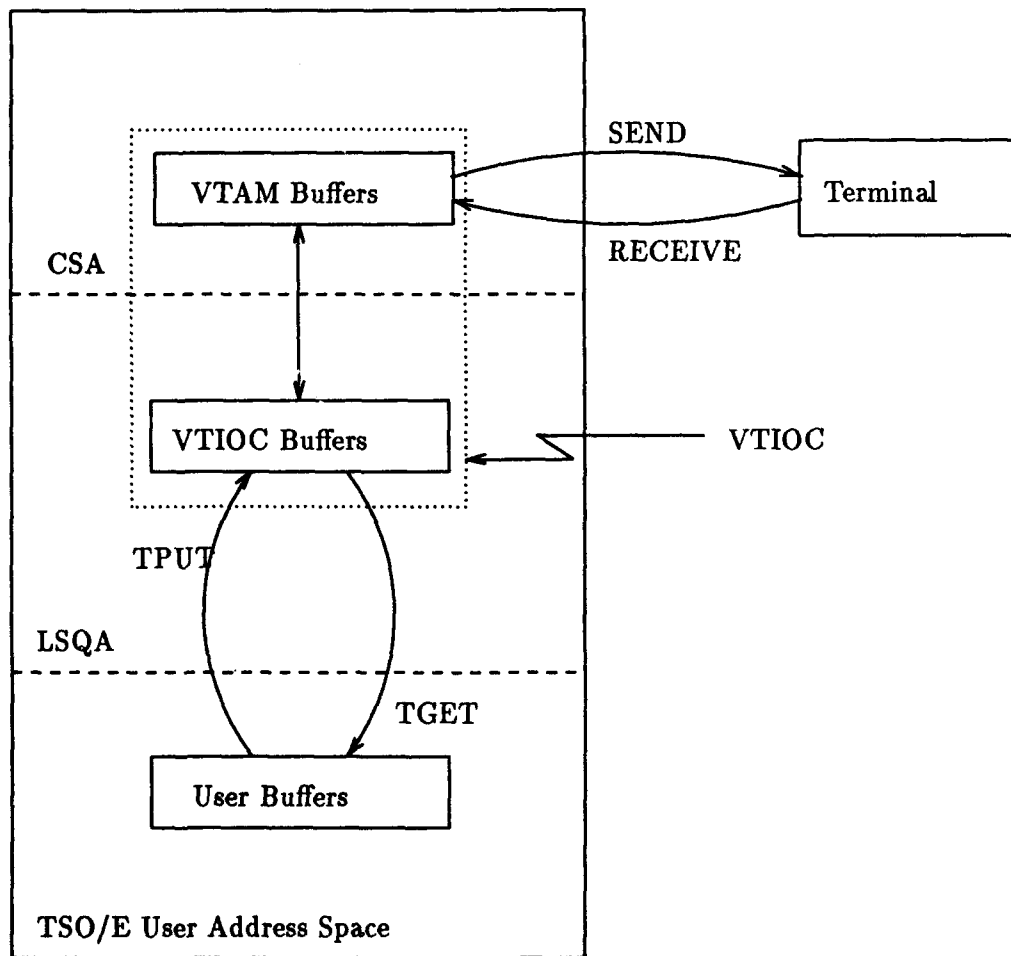


Figure 3.5. TSO/E Interface with VTAM

3.5.5 Message Commands

The TSO/E has several commands to send and receive messages. These commands are SEND, LISTBC, TRANSMIT, and RECEIVE. Messages are considered resources for the B1 environment, and mandatory controls are enforced for these commands. For more information on the mandatory controls enforced see page 122, "Mandatory Access Control."

A message for the SEND command can be up to 115 characters in length. The sender can specify one of three parameters: "now" (the default), "save," and "logon." "Now" will try to send the message immediately. If the receiver is not logged in, is not in the same MVS image, or is logged in at a lower security label than the message, the message is discarded. "Save" will store the message in the receiver's mail log, and the receiver must issue a LISTBC to read it. The mail log can be shared among MVS images as long as the DASD it is on is shared. "Logon" will attempt to perform "now" processing if the receiver is logged in; otherwise the "save" processing is performed.

LISTBC is used to read messages from the individual mail logs. LISTBC will not display a message if the message was sent at a higher security label than that at which the user is logged on. If a receiver will never be able to read a saved message because of the message's security label, the message is discarded.

A message or data set can be sent with the TRANSMIT command. The receiver must issue the RECEIVE command to get the message.

3.6 Print Services Facility

The Print Services Facility (PSF) provides the capability to print security labels on hardcopy output and to audit any attempt by a user to override this labeling. Labeled printed output is controlled through PSF and RACF. The PSF supports Advanced Function Printing (AFP) printers as system output devices for deferred printing under JES, as shown in Figure 3.6. When running in deferred-printing mode, PSF is the output writer that processes the spooled output from JES and sends a data stream to a page printer.

Some system output devices are managed by a JES output writer that operates entirely in the JES address space. PSF attached printers, however, are defined as functional subsystem applications (FSAs) that run under a separate address space called a PSF functional subsystem (FSS). A functional subsystem interface (FSI) maintains communication between the FSS and the JES global address space. A cataloged startup procedure in SYS1.PROCLIB specifies PSF initialization parameters and libraries that contain system and installation resources.

3.5.5 Message Commands

The TSO/E has several commands to send and receive messages. These commands are SEND, LISTBC, TRANSMIT, and RECEIVE. Messages are considered resources for the B1 environment, and mandatory controls are enforced for these commands. For more information on the mandatory controls enforced see page 122, "Mandatory Access Control."

A message for the SEND command can be up to 115 characters in length. The sender can specify one of three parameters: "now" (the default), "save," and "logon." "Now" will try to send the message immediately. If the receiver is not logged in, is not in the same MVS image, or is logged in at a lower security label than the message, the message is discarded. "Save" will store the message in the receiver's mail log, and the receiver must issue a LISTBC to read it. The mail log can be shared among MVS images as long as the DASD it is on is shared. "Logon" will attempt to perform "now" processing if the receiver is logged in; otherwise the "save" processing is performed.

LISTBC is used to read messages from the individual mail logs. LISTBC will not display a message if the message was sent at a higher security label than that at which the user is logged on. If a receiver will never be able to read a saved message because of the message's security label, the message is discarded.

A message or data set can be sent with the TRANSMIT command. The receiver must issue the RECEIVE command to get the message.

3.6 Print Services Facility

The Print Services Facility (PSF) provides the capability to print security labels on hardcopy output and to audit any attempt by a user to override this labeling. Labeled printed output is controlled through PSF and RACF. The PSF supports Advanced Function Printing (AFP) printers as system output devices for deferred printing under JES, as shown in Figure 3.6. When running in deferred-printing mode, PSF is the output writer that processes the spooled output from JES and sends a data stream to a page printer.

Some system output devices are managed by a JES output writer that operates entirely in the JES address space. PSF attached printers, however, are defined as functional subsystem applications (FSAs) that run under a separate address space called a PSF functional subsystem (FSS). A functional subsystem interface (FSI) maintains communication between the FSS and the JES global address space. A cataloged startup procedure in SYS1.PROCLIB specifies PSF initialization parameters and libraries that contain system and installation resources.

3.6.1 Security Libraries

The Security Definitions Library is a partitioned data set or a series of concatenated data sets defined by the system security administrator. It contains a member for each SECLABEL allowed in the system. Within each member is an entry for each paper size defining the system-defined UPA (User Printable Area) and security overlays for the paper size. If RACF class PSFMPL (PSF/Mandatory Print Labeling) is in effect (mandatory for a B1 system), PSF will issue a message to the system console if the Security Definitions Library is not defined. To support security labeling, there are three additional libraries that contain the various security resources. These libraries, which are not accessible to the user, are as follows:

- Security Font Library
- Security Overlay Library
- Security Page Segment Library

All of these libraries must be defined to RACF at the SYSHIGH security level. If security libraries have been specified on PRINTDEV, PSF will open them during initialization and close them during termination. When PSF is called to retrieve a resource, it will be passed an indication of whether or not the resource is a security resource. The PSF will always fetch security resources from the security libraries. The security libraries will not be searched for non-security resources. If PSF fails to find a security resource in the appropriate security library, an error will be returned to the caller and the job will go on hold.

3.6.2 Security Resources

There are three types of security resources: overlays, page segments, and fonts. Each type of resource is stored in a secure library, and it is managed differently from other resources stored on the system. Security resources are defined as those required for output page labeling and are loaded by PSF from the security resource libraries. Security overlays are specified in the security definitions file. The system security administrator defines security overlay(s) for a given page size and SECLABEL which are used to produce the security labeling on output pages. These security overlays may also include fonts and page segments. Since these fonts and page segments are required to produce the security labeling, they are also considered security resources.

If the printer detects an error processing a security overlay (or related font or page segment), a message will be issued to the system console, the print operation will be terminated, and the data set will be released to JES for holding with a PSF message ID identifying why the data set is being held.

3.6.3 Mandatory Print Labeling

Mandatory print labeling (MPL) is a function which establishes printed security labeling of separator pages for human readable output. The RACF class PSFMPL must be enabled before the printers are started.

MPL provides the following:

- **Separator Page Labeling** - PSF ensures that the separator pages (the job header and trailer) are always produced with appropriate security labeling. Separator pages can be turned off by the operator but cannot be overridden by the end user.
- **Data Page Labeling** - a security label that corresponds to the sensitivity level of the data set (document being processed) is placed on each data page. This security label cannot be subverted by users but can be turned off by RACF authorized users. It is also audited.
- **System Defined User-Printable Area** - a user-printable area is the area within the valid-printable area (VPA) where user-generated data can print without causing an exception condition. The VPA is the area within the physical page boundaries on which any data can be printed. Security labels are printed outside the UPA but within the VPA. User-generated data cannot be printed outside the UPA.
- **Auditing** - at the end of a data set print operation, PSF generates the System Management Facility (SMF) Type-6 record with relevant security and accounting information (see page 130, "Auditing").

3.6.4 PSF Interface to JES

When using JES, a user does not use PSF directly because output from an application is spooled to JES. The JES sets up, starts, and controls each output writer and the associated system output devices so printing is deferred until JES schedules the spooled output data set for PSF to print.

3.7 Resource Access Control Facility

The Resource Access Control Facility (RACF) controls user access to resources by verifying user identities, authorizing user access to resources, and recording and reporting events via System Management Facilities (SMF). Access controls for the TCB-controlled objects are described starting on page 122, "Mandatory Access Control" and page 115, "Discretionary Access Control." The information that RACF uses to control user access to resources is contained in profiles; these profiles are stored in the RACF data base. The RACF Manager routines handle all input/output to the RACF data base.

3.7.1 RACF Interface to MVS

The RACF executes in the address space of its sometimes implicit invoker. The default RACF installation places the RACF modules in SYS1.LINKLIB and SYS1.LPALIB. The RACF commands, RACF data base initialization program, data security monitor, and RACF utilities reside in an APF-authorized library. The RACF manager, the RACF router, RACF SVC processing routines, and RACF related exit routines run in key 0 and supervisor state, and reside in the link pack areas (LPA, FLPA, and MLPA). The system uses seven macro instructions that interact with RACF, five of which issue SVCs. When requesting services available prior to RACF release 1.9, these macros may be called by authorized programs directly, or through SAF (see page 56, "System Authorization Facility"). However, all functions new to release 1.9 are supported only through a SAF macro RACROUTE. Since system components execute using 31-bit addresses, they are forced into using RACROUTE as other RACF macros are limited to 24-bit addresses. When the instructions are issued through SAF (via RACROUTE), the MVS router passes control to the RACF router. The RACF router determines whether or not to call RACF for a particular request, and sets appropriate return and reason codes based on RACF processing.

Macros that issue SVCs are as follows:

- **RACINIT:** The TSO/E LOGON and batch job initiators issue the RACINIT macro instruction to request that RACF verify the identity of the user attempting to enter the system. If the user is RACF-defined, the RACF module that receives control verifies that the user's password, supplied group name (if any), security label (if any), and terminal authorization (if a TSO/E logon) are valid. If the user supplied the name of an application program, it also checks to see if the user has authorization to the application. If the user is authorized to enter the system, an Accessor Environment Element (ACEE) is built for the user. An ACEE is a control block specifying a user's authorizations throughout the life of the job. The ACEE resides in LSQA.
- **RACHECK:** The resource managers (system components such as DFP, TSO/E, etc.) and JES issue the RACHECK macro instruction to determine if a user has authority to access a RACF-protected resource. The RACF verifies that the user is authorized to access the resource. RACHECK may also be issued by non-APF-authorized programs.
- **RACDEF:** This macro instruction is issued by data management (DFP) to define an entity (e.g., data set) to RACF or to change or delete a volume's RACF description. If the automatic data set protection (ADSP) option has been enabled for the user or if the user specified RACF protection option, data management invokes discrete profile processing. Otherwise, generic profile processing (see page 107, "Resource Profiles") is used.
- **RACLIST:** This macro is provided for resource managers requiring high performance checking to request that in-storage profiles (see page 107, "Profiles") be constructed for resources defined by a given class descriptor (see page 103). The RACF, which is the actual issuer of

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

this macro, builds an in-storage profile for a resource from the information obtained from the resource profile and from all resource groups that define the resource.

- **RACXTRT:** This macro instruction is issued to manipulate fields in a user profile. Fields can be retrieved, replaced, and encrypted. The RACF obtains a work area and fills it with the appropriate user profile fields. Modified fields are written back to the profile. If RACXTRT specified TYPE=EXTRACT, the address of the work area is passed back to the caller. If RACXTRT specified TYPE=ENCRYPT, the password text passed to the module is encrypted and the result is returned to the area that held the clear text.

Macros that do not issue SVCs are listed below:

- **FRACHECK:** The RACF resource commands RDEFINE and RALTER issue the FRACHECK macro instruction to determine if a user has the authority to access a RACF-protected resource. FRACHECK verifies access authority for only those resources whose profiles have been brought into memory by the RACLIST macro. If the profile is not in memory, the RACHECK macro is called. The FRACHECK macro is used only by RACF.
- **RACSTAT:** The resource managers issue the RACSTAT macro instruction to determine the status of RACF (active/inactive) and the status of a given class (active/inactive).

3.7.2 RACF Data Base

The RACF data base holds all RACF access control information. The RACF uses the data base each time a RACF-defined user enters the system, each time a user wants to define access to a RACF-protected resource, and when a user accesses a RACF-protected resource. However, since portions of the data base may exist in main memory as tables or profiles, access to the data base residing on a disk does not always occur.

The RACF data base physically occupies a contiguous non-VSAM data set that resides on a DASD volume. It is made up of 1-kilobyte blocks and is cataloged. The RACF manager addresses these blocks by relative byte addresses. The RACF manager uses the EXCPVR macro (which calls EXCP) to read or write to the data base. When the system is IPLed, MVS opens and allocates the data base and updates the RACF control blocks with the physical location of the data base on the volume.

To reduce device contention and to minimize the number of resources made unavailable by the loss of one device, the logical RACF data base may be divided into multiple physical data bases (up to 90) and spread across several devices. In the case of multiple physical data bases, the master scheduler initialization routine, at IPL time, constructs an internal RACF control block—the data base descriptor table. The physical data base descriptor table resides in common storage area or extended common storage area. The RACF manager uses this table to maintain and process the data bases.

Final Evaluation Report IBM MVS/ESA
3.7. RESOURCE ACCESS CONTROL FACILITY

An active RACF data base is considered the master data base. It is the primary data base used for making access control decisions. The RACF also allows for backup data bases (see page 102, "Recovery").

The RACF data bases consist of the following types of records: header blocks, block availability mask (BAM) blocks, index blocks, templates, and profiles. A description of each type of record follows.

The header or index control block (ICB) is the first block in a RACF data base and provides a general description of the data base. It contains information such as the total number of BAM blocks in the data base and the tape and DASD volume protection options. The ICB has a relative byte address of 0. The RACF uses the ICB to locate the other blocks in a RACF data base. Each physical RACF data base has an ICB, but RACF uses only the ICB for the primary data set when determining the setting of options.

The BAM blocks determine the availability of all the blocks in a RACF data base. Each 1-kilobyte BAM block contains header information followed by block masks. Each bit in these block masks corresponds to a 256-byte segment within a RACF data base block.

Index blocks are used to locate profiles. The RACF uses a multiple level index to locate profiles in the RACF data base. All index searches begin with the highest level index block, whose relative byte address is contained in the ICB. At every level but the lowest, the first entry in a block that is equal to or alphabetically greater than the requested profile name is used to reach the next level index block. If no entry is greater than or equal to the profile name, the index search continues with the relative byte address pointed to by the last index entry in the block being searched.

The RACF supplies a template for each type of profile (group, user, data set, and general resource) and five unused templates that are reserved for future use. The templates contain a 14-byte definition for each field in the profile. This definition contains the field name, a set of five flags, and the field length. Each template also contains a number that corresponds to the type of profile it is describing.

Profiles contain descriptions of the attributes and authorities for every entity defined to RACF. The number in the entry type field identifies the type of profile and corresponds to the number of the template that maps this type of profile. Profiles will be further discussed on page 107.

The RACF data base can also be initialized in a restructured form. In this case, longer profile names are supported, data base records have been extended in length, templates include identity numbers for each field, and the BAM blocks map to 4K blocks.

3.7.3 RACF Manager

The RACF manager handles I/O to the RACF data base on behalf of the RACF commands and system macro instructions by using the EXCPVR macro. The RACF manager also performs

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

serialization and maintains the index structure and space allocation on the RACF data base. The RACF system SVCs (RACHECK, RACINIT, RACDEF, and RACLIST) and RACROUTE branch directly to the RACF manager, while RACF commands issued from a TSO/E session are conveyed to the RACF manager.

3.7.4 Recovery

The RACF allows the identification of backup RACF data bases that may be used in case of failure of the primary RACF data base. The backup data bases are allocated at the same time the primary data base is allocated, but only the primary data base is used while processing. When a backup data base becomes the primary, another data base is activated as backup. There are three backup options: all updates are duplicated on the backup data base, all updates except for statistics are duplicated on the backup data base, or no updates are duplicated on the backup data base.

The RACF data base recovery consists of two parts: the dynamic maintenance of a backup copy of the primary data base and the use of the RVARY command. The RVARY command is used to switch to the backup data base or to deactivate a specific physical data base to perform maintenance. When the RVARY command is issued by the system administrator, the system (MVS) operator must examine the userid to ensure that the issuer has the proper authority to enter the command. If so, the operator issues the password (if one has been defined using the SETROPTS command) or enters YES to allow RVARY to complete.

If all RACF data bases are deactivated, failsoft processing is in effect. For users that are already logged on, RACF uses whatever in-memory tables are still valid. If the user requests a profile that is not in a valid internal table, RACF prompts the operator to approve the request. If not already logged on, the only users that may log on are those who have userids in SYS1.UADS and know their UADS password. The RACF then requests the operator's concurrence each time the user requests access to a general resource or to a data set that does not start with the user's ID.

3.7.5 RACF Users

A RACF user is identified by an alphanumeric userid. A RACF group is a collection of users having common access requirements. A RACF group is identified by an alphanumeric groupid. Users must be connected to at least one group.

User's identification information is placed in an accessor environment element (ACEE). An ACEE is created (by RACINIT) after a user attempting to logon is authenticated. It is based on the information extracted from the user's profile. An ACEE is located in the system storage protected with the storage key of 0. It contains the following information: the current userid, current connect group, user attributes, group authorities, default universal access, name of started task, terminal identifier for terminal users, addresses to tables of various security information (e.g., list of groups of which this user is a member, model of protection to be applied to newly created data sets, list of

Final Evaluation Report IBM MVS/ESA
3.7. RESOURCE ACCESS CONTROL FACILITY

categories to which this user is allowed access), and a pointer to the UTOKEN. See the subsequent sections for descriptions of user attributes and authorities.

3.7.6 RACF Groups

In RACF a group is a set of users with the same access requirements. Groups are flexible and can be structured to reflect the organization of logical entities like departments or projects. When groups own other groups, the owning group is called the superior group. (The concept of ownership refers to the type of access initially associated with the creating or the controlling entity.) The privileges and restrictions a user has within a group apply only to the resources within the scope of the group. Resources within the scope of the group are:

- Resources owned by the group
- Resources owned by users who are owned by the group
- Resources owned by subgroups that are owned by the group

The scope of control of a group percolates from a group to its subgroups. The percolation stops when a subgroup is owned by a user, or by a group which is not its superior group. Figure 3.7 depicts a sample arrangement of groups and users.

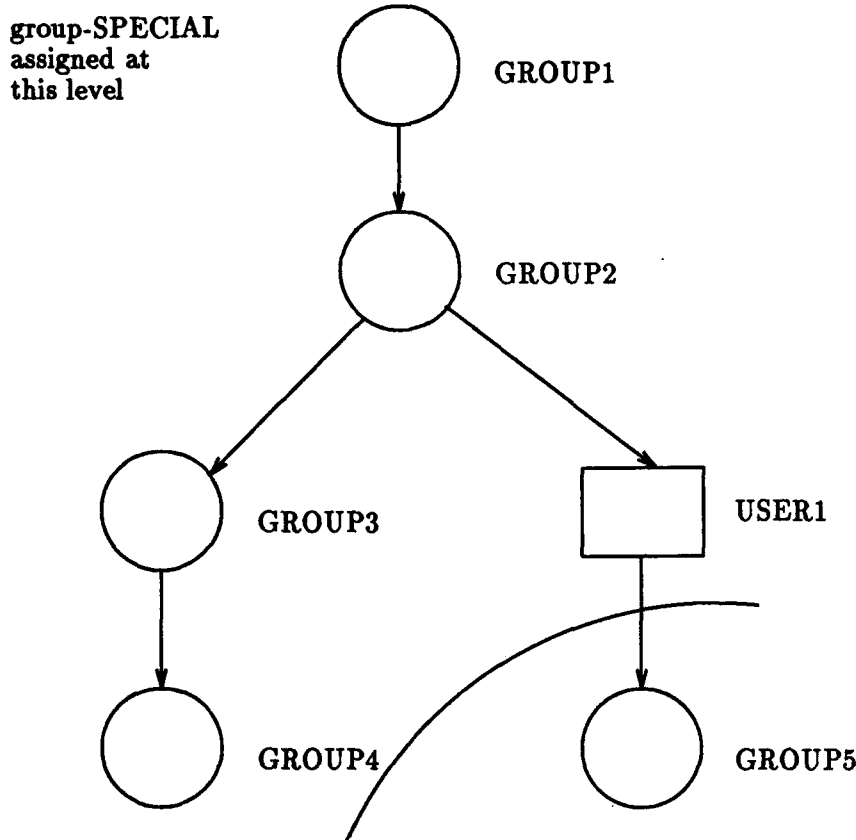
3.7.7 Resource Classes

Resources other than users, groups, and data sets are grouped into resource classes. A class is a necessary and convenient way of referring to resources with similar characteristics, or to operational attributes (privileges). Terminals, disk volumes, and fields in RACF profiles are just some of the predefined resource classes. Installations are also allowed to define their own resources and classes.

A class of resources can optionally be designated as a resource group class or a resource member class. For a resource group class, each user and group of users permitted access to that resource group is permitted access to all members of the resource group. Each resource group class created consists of other resource classes representing members of the group.

3.7.8 User Properties

A user's level of authorization to access system-protected resources is determined by a combination of four variables (for a description of how these variables are used to make access control decisions see page 122, "Mandatory Access Control," and page 115, "Discretionary Access Control"):



The sample shows the scope of control of an attribute assigned at the group-level. *GROUP1* owns *GROUP2*, *GROUP2* owns *GROUP3* and *USER1*, and so on. Attributes assigned to *GROUP1* apply to anything owned by *GROUP1*, and if *GROUP1* owns a group those attributes apply to what that group owns as well, and so on. In the figure a user is connected to *GROUP1* with the group-SPECIAL attribute. This allows that user to use the group-SPECIAL authority anywhere but in *GROUP5*. In other words, the connected user (and any user with the group-SPECIAL attribute in *GROUP1*) can access the profiles and resources owned by *GROUP1*, the profiles owned by *GROUP2* (*GROUP2* is owned by *GROUP1*), the profiles owned by *GROUP3* (*GROUP3* is owned by *GROUP2*), the profiles owned by *GROUP4* (*GROUP4* is owned by *GROUP3*), and the profiles owned by *USER1*. The connected user cannot access the profiles or resources in *GROUP5* because *GROUP5* is owned by a user. This is also true for a user owned by a user (for example, if *USER1* owned a user).

Figure 3.7. Scope of Control

Final Evaluation Report IBM MVS/ESA
3.7. RESOURCE ACCESS CONTROL FACILITY

- **User's attributes:** The security administrator can assign attributes to each RACF-defined user. Attributes determine the privileges and restrictions a user has on the system. Attributes are classified as either user-level attributes or group-level attributes.
- **User's group authorities:** The security administrator or group administrator can assign a group authority to each user of a group.
- **Security clearance of the user and the resource:** Each user and each resource has security information in its profile. This information consists of a security label which is used to refer to that information.
- **Access authority:** This authority determines to what extent the specified user or group can use the resource. The owner of a resource can grant or deny a user specific type of access to that resource.

Attributes

There are seven user attributes: SPECIAL, AUDITOR, OPERATIONS, CLAUTH, GRPACC, ADSP, and REVOKE. The attributes apply regardless of what group the user is in. The group attributes are group-SPECIAL, group-AUDITOR, and group-OPERATIONS, which apply only to the group (and the scope of control within that group) for which the user has the group attributes. A description of each attribute follows.

The SPECIAL attribute allows the user to issue all RACF commands and gives the user full control over the RACF profiles. This attribute can only be given by a user with the SPECIAL attribute. The group-SPECIAL attribute gives the user full control over the resources within the scope of a group only, i.e., the effects of RACF commands will only apply to resources within the scope of the group. A user with the SPECIAL attribute is usually designated as the security administrator.

The AUDITOR attribute gives the user the responsibility for auditing the security controls and the use of the system resources for the whole system. The user assigned the AUDITOR attribute can specify logging options on RACF commands, can list auditing options of profiles, and can control additional logging to the SMF data set. The user may also list the profile information available to the SPECIAL user. This attribute can only be assigned by a user with the SPECIAL attribute. The group-AUDITOR attribute restricts authority to the resources within the scope of the group.

The OPERATIONS attribute allows the user to perform maintenance functions like copying, reorganizing, cataloging, and scratching RACF-protected resources. The group-OPERATIONS attribute restricts authority to the resources within the scope of the group.

The CLAUTH (class-name authorization) attribute is given to users on a class-by-class basis, and it cannot be assigned at the group level. CLAUTH allows the user to define profiles in that class to RACF. The user with the CLAUTH(USER) attribute may also add new users to RACF if that user is the owner of or has JOIN authority to a group. This group will become the new user's

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

default group. The CLAUTH attribute also allows the user to define resources to be included in the assigned class.

The GRPACC (Group Access) attribute makes any group data set profiles the user defines to RACF automatically accessible to other users in the group if the user defining the profile is a member of that group. If assigned to the user, this attribute applies to all groups of which the user is a member, and if assigned at the group level, the attribute applies only to that group.

The ADSP (Automatic Data Set Protection) attribute causes every data set created by the user to have a discrete profile automatically created. If this attribute is assigned at the group level, ADSP is only in effect if the user is within the group.

The REVOKE attribute provides an authorized user with the capability to prevent another RACF user from entering the system. The authorized user can also place this kind of restriction at the group level in which case the user, thus restricted, cannot enter the system by connecting to that group or access the resources of that group. Using RACF commands, a future time can be set for REVOKE to occur or for REVOKE to be removed. The owner of a user's profile or the system administrator can assign the REVOKE attribute. A user with the REVOKE attribute can also specify how many consecutive logon attempts are permitted by RACF before the userid is revoked.

Group Authorities

There are four types of authorities: USE, CREATE, CONNECT, and JOIN. The group authority USE allows a user to access data sets within the group and to create RACF-protected data sets whose names begin with that user's userid.

The group authority CREATE allows a user to RACF-protect and control access to data sets within the scope of the group. The CREATE authority includes the privileges of the USE authority.

The CONNECT group authority includes the privileges of the USE and CREATE authorities. It also allows the user to connect users to a group, and the user may assign USE, CREATE or CONNECT group authorities to users in that group.

A user with the JOIN group authority can define new users (provided the user also has the CLAUTH attribute for the USER class) and groups to RACF and give those new users any level of group authority. The JOIN group authority allows a user possessing that authority to create new groups. The newly created group will become a subgroup of the group to which the user has JOIN authority. The JOIN authority includes the privileges of the USE, CREATE and CONNECT authorities.

3.7.9 Profiles

The information that RACF uses to control access to protected resources is contained in RACF profiles. Each profile is owned by a user or group. By default, the owner of a profile is the user who creates it. There are four types of profiles: user, group, data set, and general resource.

Profiles can always be found in the RACF data base. However, in order to enhance system performance, the system administrator can request that profiles pertaining to specified resource classes be copied to common storage where they can be found when making access control decisions. Such an in-storage profile must subsequently be manually refreshed whenever a change to the identical profile in the data base takes place.

User Profiles

A user profile defines an individual user. When a user is defined to RACF, a user profile is created in the RACF data base. A user profile consists of several fields containing the following information: user's identifier, encrypted password, user's attributes, name of the default group, flags indicating which activities should be logged, password change interval, password expiration date, minimum password length allowed, a specified number of previously used passwords, user's security categories, user's security level, user's default security label, user's current security label, name of a model profile to use when creating new data set profiles, TSO logon information, and default DFP information.

Group Profiles

A group profile defines a group of users. A group profile consists of several fields containing the following information: owner of this group, groups owned by this group, members (users) of the group, group authorities of each member, name of a model profile to use when creating new group-named data set profiles, and default DFP information for the group.

Data Set and General Resource Profiles

There are two forms of resource profiles: data set profiles and general resource profiles. Data set profiles define access to disk and tape data sets. General resource profiles define access to objects in the system where data resides (other than data sets), to objects in the system where data passes during data processing (such as terminals), and to functions by which users work with data (such as commands). The resources falling under the protection of general resource profiles correspond to the classes defined for them. A site can further define additional resource classes to be protected by RACF.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

Data set and general resource profiles can be further divided into discrete and generic profiles. A discrete profile defines the protection of a single object. That is, there is a single unique object protected by the profile. A generic profile defines the protection of a group of objects. The scope of protection provided by a generic profile encompasses more than one object, usually having similar names.

3.7.10 Audit

The RACF has the ability to audit events where user-resource interaction has been attempted. The actual access activities or variances from the expected use may be recorded. The RACF audits by passing records to System Management Facilities (SMF) routines which then write the information in SMF data sets. The RACF report writer is used to extract pertinent SMF records. For more information see page 130, "Auditing."

3.7.11 RACF Commands

The RACF commands are used to create, alter, list, or delete profiles and to define system-wide options. To successfully issue a command, a user must be defined to RACF with a sufficient level of authority. The RACF commands are executed in a TSO/E session by entering the commands directly or by using the RACF Interactive System Productivity Facility (ISPF) panels. All RACF command functions, except RVAR and RACFRW, have ISPF entry panels (menus) and associated help panels. The SETROPTS command is used to set system-wide RACF options such as to enable or disable the global access checking facility and to activate and control the scope of erase-on-scratch processing (see page 125). The RVAR command is used to deactivate and reactivate RACF and to switch from using the primary RACF data base to the alternate RACF data base. The RACFRW command is used to invoke the RACF Report Writer in generating a variety of security audit reports.

3.7.12 RACF Options

The evaluated system has 10 RACF installation options which must be active. They are invoked with the SETROPTS command. JES(BATCHALLRACF, XBMALLRACF) allows only the users authorized to RACF to execute batch jobs. ERASE(ALL) forces an overwrite with binary zeroes of any deleted or released DASD data set. PROTECTALL ensures that every data set about to be created or accessed is protected by a profile. SECLABELCONTROL prevents unauthorized users (those without the SPECIAL attribute) from changing a security label associated with a resource. CATDSN(FAILURES) prevents unauthorized users from accessing DFP-controlled data sets that are not cataloged or that are not system temporary data sets. GENERICOWNER allows users to create more specific profiles for resources that do not yet have discrete profiles. MLACTIVE(FAILURES) allows only the RACF-defined users to log on to the system and forces given

Final Evaluation Report IBM MVS/ESA
3.7. RESOURCE ACCESS CONTROL FACILITY

classes of profiles (user, data set, device, tape volume, terminal, and writer) to have security labels. **MLS(FAILURES)** prevents users from copying a data set with a certain security label to another data set with a lower security label. **MLSTABLE** prevents all users (authorized and unauthorized) from modifying the security label of a profile. **MLQUIET** is used by authorized users to prevent unauthorized users from logging on, starting new jobs, and accessing resources. It is generally used while making changes to labels, restoring the RACF data base, etc. **CLASSACT(TEMPDSN)** provides protection for temporary data sets.

Final Evaluation Report IBM MVS/ESA
CHAPTER 3. SOFTWARE ARCHITECTURE

This page intentionally left blank

Chapter 4

Protected Resources

4.1 Subjects

Subjects in MVS/ESA are tasks performing user and system functions. There are four types of subjects: console operators, started tasks, TSO/E users, and batch jobs. Subjects are identified by userids and associated groupids. The userid must be one to seven characters in length and it must start with an alphabetic or a national character (\$, @, and #). The groupid, like the userid, must start with an alphabetic or national character. The groupid is used to identify a group to RACF. Userids and groupids must be unique.

Console operators (users) are tasks providing an interface between the system operator terminal(s) and the system. An operational system has at least one operator console terminal and an operator. Console users are the only subjects sharing a common address space; each console user has a task within one communication address space.

Started tasks are tasks initiated by explicit commands taken from a system start-up data set or received from an operator. System components such as JES, VTAM, and TCAS are examples of started tasks. Started tasks commonly execute until either the system is stopped or the operator deletes them. Each started task is assigned its own address space for execution.

The TMP tasks represent logged-in TSO/E (interactive) users. Each TSO/E user is assigned its own address space. TSO/E users are provided with ways of sharing data, but unless they are authorized users, they cannot access and modify data in other address spaces. The users and their RACF-built ACEEs are collectively maintained throughout their interactive sessions.

Batch jobs can be created by each of the previously described subjects. Batch jobs are created by submitting jobs to a JES for background processing. Executing batch jobs can submit other batch jobs via the internal reader. A batch job executes in the address space of an initiator processing that job.

4.2 Objects

The MVS/ESA provides users with access to four kinds of objects: direct access storage device (DASD) data sets, SYSOUT data sets, tape volumes, and TSO TPUT messages. This section provides a definition of these objects.

4.2.1 DASD Data Sets

The DASD data sets (disk files) are the basic containers in which information is stored in this system. With respect to security, there are two types of data sets: virtual storage access method (VSAM) data sets, and non-VSAM data sets. Users may be authorized to control the contents and the access to both of these types of data sets. For further reading, see page 65.

4.2.2 SYSOUT Data Sets

The JES is the system function that spools and schedules output data streams. These data streams are designated as SYSOUT. SYSOUT is implemented as data sets which are opened and closed in the same manner as any other data set processed on a unit record device. Each job may be associated with many such spool data sets. For further information see page 70.

4.2.3 Tape Volumes

A tape volume represents a collection of one or more data sets stored on a magnetic tape. Users may be authorized to control the contents and the access to tape volumes. For further discussion see page 63.

4.2.4 TSO TPUT Messages

Terminal users may communicate with one another using the TPUT macro. Users prepare message text, but once that text is sent, no further user control over it is possible. The message receives the sender's security label. A user may choose not to receive TPUT messages from other users. For additional information see page 91.

Chapter 5

Protection Mechanisms

5.1 Identification and Authentication

The MVS/ESA provides identification and authentication through the use of RACF. Identification and authentication is applied to all subjects including MVS console operators. Any subject identified and verified by RACF is a RACF user. Once a user is verified an ACEE is created. A RACF user is placed in a default group unless another authorized group is specified at logon. To change the active group, a user must logoff and relogon specifying the new group.

5.1.1 Userid, Password, and Groupid

The RACF requires a userid, groupid, and password from a user attempting to logon. The MVS/ESA maintains and protects userids through all the steps of job/task execution. A user can be active in only one group at a time. The groupid identifies to RACF the group to which the user belongs.

A user is defined by a user profile which is created with the RACF ADDUSER command. The issuer of this command must have the SPECIAL attribute or must have the CLAUTH attribute for the USER class and meet one of the following conditions: be the owner of the default group specified in the command, have JOIN authority in the default group specified in the command, or the default group specified in the command must be within the scope of a group in which the issuer has the group-SPECIAL attribute. The owner of the user profile is specified by using the OWNER parameter of the ADDUSER command. If no owner is specified, the user creating the profile is defined as the owner.

The ADDGROUP command is used to create a group profile. To use the ADDGROUP command, the user must have the SPECIAL attribute, or have the group-SPECIAL attribute within the superior group, or be the owner of the superior group, or have JOIN authority in the superior group. The OWNER parameter of this command specifies the group or user to be assigned as the owner of the new group. If the OWNER parameter is not used, the user creating the new group profile is defined as the owner.

The RACF gives the system administrator flexibility in password management. This allows the administrator to control the minimum and maximum length of passwords, password lifetimes, how many consecutive password verification attempts RACF is to permit before it revokes a userid, and the authority to cause RACF to revoke the user's right to use the system if the userid has remained

Final Evaluation Report IBM MVS/ESA
CHAPTER 5. PROTECTION MECHANISMS

unused beyond a specified number of days.

There may also be further checks to limit logon such as terminalid, time of day, or day of week. A user may be limited to enter the system on certain days of the week and during certain hours of the day. A user can also be restricted to use specific terminals on certain days of the week and during certain hours each day if the terminalid is used.

5.1.2 Mapping Subjects to Userids

Each subject in MVS/ESA is initially associated with its userid in a different way. This section describes this association for each subject.

Console Operator

Console operators issue commands from a single address space called the communication task address space. This address space is a non-swappable system address space created during system initialization. It transfers messages from user programs and system routines to the operators at the consoles. The physical consoles are defined to the system at system initialization. Console operators are required to log on to the system.

Started Tasks

When an operator issues the START command, an address space is created for a started task. The START command includes the name of the procedure to be executed. Since no identification or authentication information is included with the START command, a userid/groupid replacement table is scanned for the procedure name.

The userid/groupid replacement table contains a procedure name, its associated userid and groupid, and flags. The table is created during RACF installation, and it resides in the link pack area.

If either userid or groupid is found for the procedure's name, this information is placed in the ACEE and is used by RACINIT to identify the user. If the procedure's name is not found, a generic entry is looked for. If no entry is found the default userid (*) and groupid (*) are placed in the ACEE and null authority is given to the started task. With the default userid and groupid, the started task can access RACF protected resources only if the universal access authority for the resource allows the access. If identification or authentication fails, the ACEE storage is freed and a non-zero return code is returned to the caller of the RACINIT SVC.

TSO/E Logon

A TSO/E logon is initiated with the TSO LOGON command. When a user logs onto TSO/E, TSO/E checks the TSO segment in the user's RACF profile for the user's authority to use TSO/E resources. If a user doesn't have a TSO segment, then TSO/E checks SYS1.UADS for the information to build a session for the user. TSO/E then issues the RACINIT SVC and passes the userid, groupid, and password information to RACF. If the user is identified and authenticated, RACF builds an ACEE for the user. If identification or authentication fails, the ACEE storage is freed and a non-zero return code is returned to the caller of the RACINIT SVC.

Batch Jobs

The JES propagates the current RACF userid when an already validated RACF user submits a batch job to JES via JES internal readers. The jobs are marked as password validated so that password validation is not performed. When the initiator processes the job, the propagated userid and the user's default group are used by RACF to create the ACEE.

Jobs submitted by a TSO user are automatically identified with that user and the default RACF group of that user. In this case a password is not required in the JCL JOB statement. If a user wishes to submit a job in a group other than the default group, the user must specify the userid, password and groupid when the job is submitted. However, if a job is submitted in a user's current group, only the groupid is required on the JOB statement. If a TSO user submits a job for another user, the userid and password (groupid is optional) must be present on the JOB statement.

To prevent unauthorized users from running batch jobs, a site can require all batch jobs to have RACF identification. This is accomplished with the RACF SETROPTS command option JES(BATCHALLRACF, XBMALLRACF). When this option is specified (mandatory for a B1 system), any batch job that does not have a RACF-defined user specified on the USER parameter of the JOB statement, or propagated security information associated with it, will fail.

If identification or authentication fails, the ACEE storage is freed and a non-zero return code is returned to the caller of the RACINIT SVC.

5.2 Discretionary Access Control

This section discusses the mechanisms MVS/ESA uses to provide discretionary access control. Checks that are made in determining a user's access to protected resources are identified. This section also addresses the modes of access to protected resources.

Discretionary access control is enforced in the system by both MVS/SP and the RACF program product. MVS/SP provides the abstraction and separation of subjects. The RACF maps every

Final Evaluation Report IBM MVS/ESA
CHAPTER 5. PROTECTION MECHANISMS

subject to a userid and groupid (see page 114, "Mapping Subjects to Userids") to identify the user attempting to gain access to protected resources.

5.2.1 Access Determination

Figure 5.1 provides a pictorial view of authorization checking for RACF-protected resources. With global profile checking active, RACF searches for profiles that match a protected resource by first searching the global access (GACC) table entry. If allowed access, all other checking is bypassed.

When examining profiles, RACF searches for discrete profiles first. If a discrete profile does not exist, RACF searches for a generic profile in order from most specific to least specific. Once a profile has been found, RACF performs DAC checks slightly differently for data sets than for general resources. The RACF begins access checking by examining the security label (SECLABEL). If SECLABEL checking fails, then RACF denies access. A successful security classification check does not grant access but rather allows further checking to continue. The remaining authorization checks can allow access, each of which will bypass further checking.

The owner of a resource always has the ability to change the discretionary components of the profile (and possibly give himself or herself access authority). For data sets, RACF checks to see if the high level qualifier of the data set matches the user accessing the data set, and if so, it automatically grants ALTER (see page 118) access to that data set (the user has the same access authorities as the owner). If the high level qualifier does not match the userid, then RACF will examine the data set profile. For spool data sets, RACF compares the userid and node of the requester with the userid and node of the creator of the spool data set (using the RTOKEN, see page 122). If the userids match access is granted. When a profile has been found for a general resource, RACF does not check for ownership of the resource, but begins with a check of the standard access list.

The access list in the profile is checked next. If the userid is found in the access list, success depends upon the access mode requested and the access mode specified in the access list. If the userid profile explicitly denies access, the conditional access list is checked. If the userid is not found in the access list, the groupid profile is checked. If the groupid profile explicitly denies access, the conditional access list is checked. If the groupid is not found in the access list, the universal access authority (UACC) is checked.

The UACC defines the mode of access permitted to users not explicitly named by the access list. If the requested access mode is not permitted by the UACC attributes (i.e., OPERATIONS, group-OPERATIONS) are examined. Conditional access lists are next examined. These check for specified conditions, e.g., user must be logged on at a specified terminal. Specific users and groups may be denied access just as in the standard access list check.

If all checks have passed, RACF determines whether the data set is temporary (with a system generated name), or permanent. If temporary, and CLASSACT(TEMPDSN) is active, RACF ensures that the data set was created by the current job, or that the user has the OPERATIONS attribute and will only be deleting the data set. If temporary, and neither condition is true, the

Final Evaluation Report IBM MVS/ESA
 5.2. DISCRETIONARY ACCESS CONTROL

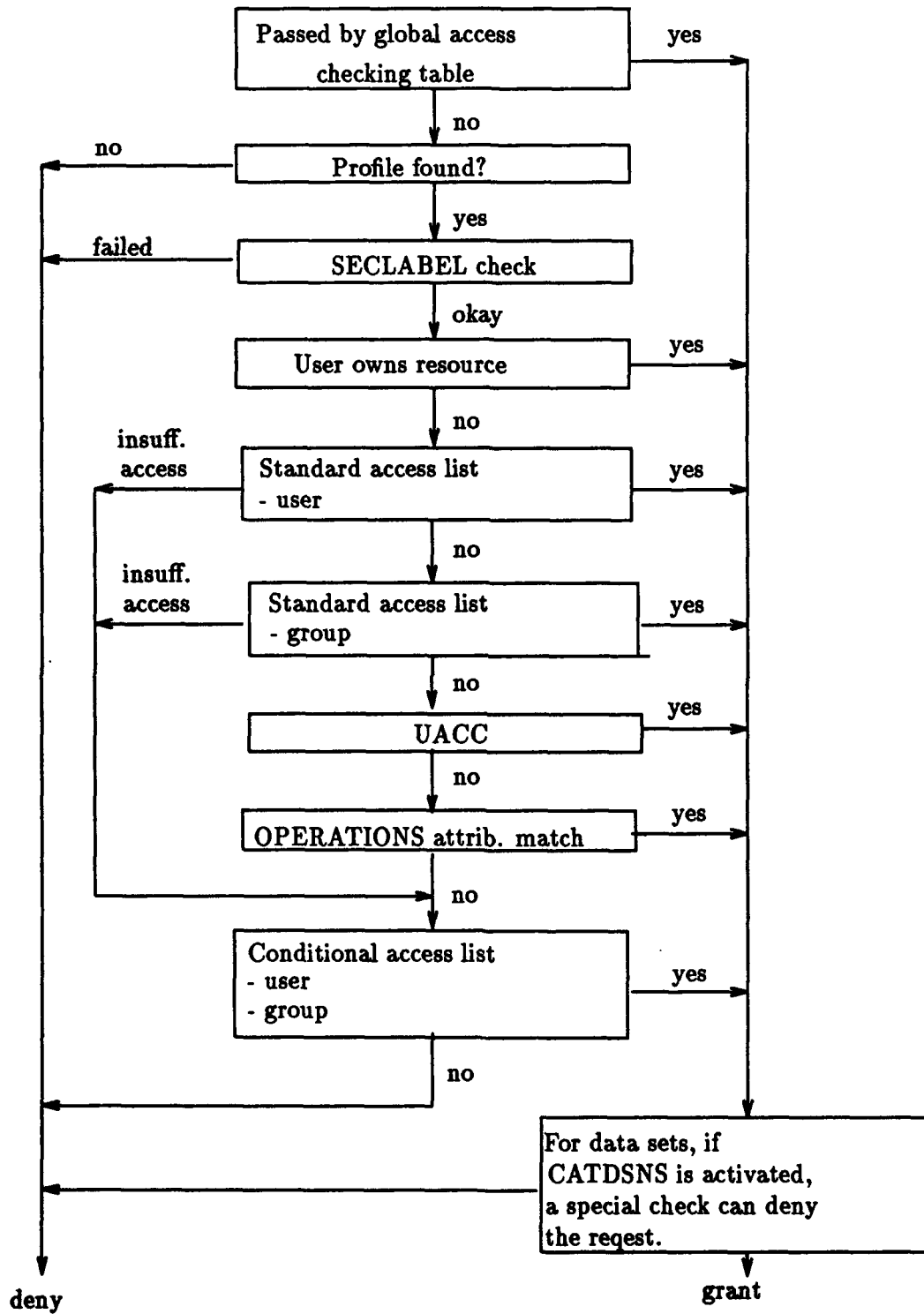


Figure 5.1. Resource Access Checks

Final Evaluation Report IBM MVS/ESA
CHAPTER 5. PROTECTION MECHANISMS

access is denied. If not temporary, then if CATDSNS is on, RACF ensures that the data set is cataloged, or was created in this job. If neither is true, the access is denied unless the user is SPECIAL.

Resource Access Authorities The system provides five RACF access authorities (i.e., access modes) used to access profile-protected resources: NONE, READ, UPDATE, CONTROL, and ALTER.

- **NONE** The specified user or group is not permitted to access the protected object.
- **READ** This access authority allows a user or a group access to the resource for the purpose of reading the object only.
- **UPDATE** This access authority allows read and write access to the resource.
- **CONTROL** This access authority varies depending upon the resource being protected and is described in the following discussion of each protected resource.
- **ALTER** This access authority is different for discrete and generic profiles. If ALTER authority is provided by a discrete profile, a user is allowed to control the discretionary information fields of the profile as well as to control the contents of the object. ALTER authority provided by a generic profile allows a user to control the contents and existence of the data set. Control over the contents of the profile requires that a user be the owner of the profile, have either the SPECIAL or group-SPECIAL attribute, or a userid that matches the high level qualifier of the profile. A high level qualifier is the portion of a data set or profile name that appears before the first period, having a maximum of eight characters.

5.2.2 Objects

This section discusses the objects protected by MVS/ESA. The objects are DASD data sets, SYSOUT data sets, Tape Volumes, TSO messages, and data set and resources profiles.

DASD Data Sets

All DASD data sets (both VSAM and non-VSAM) are protected via RACF. All DAC checking is done via the normal RACF DAC checking procedures. The PROTECTALL option (mandatory for B1) ensures that all DASD data sets are created with either a generic or a discrete profile covering them.

The RACF interprets the access authorities of NONE, READ, UPDATE and ALTER as described above for DASD data sets. The CONTROL access authority, however, is interpreted differently for VSAM and non-VSAM data sets. For non-VSAM data sets, the CONTROL access authority is

equivalent to UPDATE. For VSAM data sets, the CONTROL access authority permits a user to access (for both read and write) a VSAM data set's control interval (see page 66).

SYSOUT Data Sets

The JES provides per-job data sets (i.e., SYSIN, SYSOUT) whose information is stored within the JES address space. The SYSIN and SYSOUT data set of one job cannot be accessed by another job. These data sets are not shared between jobs.

The RACF can be used to provide additional access to SYSOUT data sets and JESNEWS so that users other than the owner of the data set can access the data set. A TSO user can access SYSOUT data sets created by jobs that run with the TSO user's userid. A TSO user can also be authorized to access another user's SYSOUT data sets via profiles in the JESSPOOL class. The TSO user can be authorized to read the data set or to print, delete, or change attributes of the data set. Only the job that created the SYSOUT data set can write into it. Access controls for SYSOUT data sets are enabled by activating the RACF JESSPOOL class.

Tape Volumes

Tape volumes are protected in the same manner as DASD data sets with one exception: the user is responsible for ensuring that all data sets on a volume have the same DAC controls.

While RACF attempts to provide discretionary access to the level of individual data sets on a tape volume, it cannot guarantee enforcement of this policy for more than one data set per tape. The RACF provides access controls to the granularity of a tape volume. That is, users granted access to a tape volume have access to all data on the tape volume.

Administrators are responsible for enabling RACF protection of tape volumes by making the TAPEVOL class active. Access checking is then performed whenever a tape volume is accessed (e.g., OPEN). RACF protects only defined volumes with an IBM standard or ANSI label. A tape volume is defined in two ways:

1. Issuing the RACF RDEFINE command without the TVTOC operand.
2. Issuing the RACDEF macro and the JCL PROTECT operand.

Only users with the CLAUTH attribute for the TAPEVOL class may define a tape volume or create a profile for a tape volume via RDEFINE. For tape volumes, RACF interprets the access authorities of NONE, READ, or UPDATE as described on page 118, "Resource Access Authorities." RACF issues a message to the operator to remove the write-enable ring if a tape volume is to be read-only. The access authority of CONTROL is equivalent to the access authority of UPDATE for tape volumes. The access authority of ALTER allows a user to overwrite the tape label and modify the

Final Evaluation Report IBM MVS/ESA
CHAPTER 5. PROTECTION MECHANISMS

tape volume profile. If a data set spans two or more tape volumes, RACF combines the tapes into a tape volume set and protects them with one TAPEVOL profile.

TSO Messages

The TSO messages pass between a terminal and a VTAM application program only after a session has been established. The VTAM ensures that data correctly passes between two LUs that are in session. Therefore, a VTAM application program may only send TSO messages to the terminal that has opened a session with the address space containing the application program.

A user can send any other user a message as long as mandatory access checking passes. The system administrator can control whether or not users are allowed to receive messages sent by the TSO/E SEND command by using the RDEFINE SMESSEGE command.

5.2.3 Other Protected Resources

This section discusses resources protected by MVS/ESA. Catalogs and VTOCs are accessed indirectly by a subject through the subject's access to an object. The Virtual I/O (VIO) data sets, temporary data sets, and spool data sets are protected such that non-privileged subjects other than the object owner have no access to the resource. Access to the remaining resources—data set and resource profiles, terminals, consoles, and printers—can be controlled through MVS/ESA.

Catalogs

The master catalog and user catalogs are VSAM key-sequenced data sets. An unprivileged user's access to these catalogs is mediated by DFP. Unprivileged users may read a catalog; only system key, supervisor state, or APF authorized programs may write to a catalog. Unprivileged users may indirectly write to a catalog through the creation and deletion of cataloged data sets. As described earlier (see page 118), RACF controls a user's access to data sets.

Volume Table Of Contents

The VTOC is a system maintained data structure stored as a data set. Unprivileged users may indirectly cause entries in a VTOC to be created or deleted during the creation or deletion of data sets. Only the system may write to VTOCs, and only through the use of APF authorized programs, key 0, or supervisor state.

VIO Data Sets

VIO temporary data sets are data sets that exist within an address space and are destroyed when the address space terminates. These data sets are protected from any accesses (by MVS) except by the job or session that created them.

Temporary Data Sets

Other types of temporary data sets may exist outside an address space. Normally, they are deleted when the job terminates. However, system failures can leave these data sets unprotected. This can be alleviated by activating the RACF TEMPDSN class. Temporary data sets may only be shared among tasks within the same address space (i.e., within the same job). System-generated temporary data sets can be protected by using the naming conventions table to modify the name that RACF will use to look like a permanent name.

Spool Data Sets

The following are spool data sets: SYSIN and SYSOUT, JESNEWS, SYSLOG, and trace data sets (for JES2). RACF protection is implemented through JES by defining parameters in the JESSPOOL class.

Data Set and Resource Profiles

Each data set and resource profile defined to RACF requires a RACF-defined user or group as the owner of the profile. The owner has full control over the discretionary information in the profile, including the access lists for any data set that the profile protects.

If the owner of the profile is a group, users with the group-SPECIAL in that group have full control over the profile. Ownership of the profiles is assigned when the profiles are defined to RACF.

Terminals

If the TERMINAL class is active (optional for a B1 system), RACF controls access to terminals. Profiles are used to specify which users, or groups, may use a terminal. The VTAM-specified symbolic for a terminal is used as the RACF terminal id for controlling access. Two access authorities are valid for terminals, READ and NONE. Users with NONE access to a terminal may not access the system through that terminal.

Final Evaluation Report IBM MVS/ESA
CHAPTER 5. PROTECTION MECHANISMS

Consoles

If the CONSOLE class is active (mandatory for a B1 system) RACF controls access to consoles. Access checking is similar to that of terminals; however, time or day-of-week restrictions cannot be specified.

Printers

If the WRITER class is active (optional for a B1 system), RACF controls access to printers. The printer is protected by a security label and, optionally, a UACC. The WRITER class is activated by SETROPTS CLASSACT(WRITER).

5.3 Mandatory Access Control

Mandatory access control (MAC) is enforced by both MVS/ESA and RACF. The MAC checking is performed by RACF while MVS/ESA ensures label integrity. The MAC check takes place before any discretionary checks.

The RACF uses a security label (SECLABEL) to perform the MAC check. The security administrator (using RACF commands) defines the representations of up to 254 hierarchical security levels. Each security level is represented by an installation defined name (1-44 characters) and an internal number (1 to 254). The hierarchy is determined by the numerical representation of the number. The security administrator also defines the representations of categories. Categories are represented by an installation defined name (1-44 characters). Approximately 32,000 bytes of storage are allocated for categories where each character in a category name uses 1 byte. The number of actual categories allowed is determined by the lengths of the category names. The maximum number of categories possible is 11,761. After security levels and categories have been defined, the security administrator must define the representations of SECLABELs. A SECLABEL contains one security level and zero or more security categories. These definitions are kept in RACF profiles which are stored in the RACF data base. Each SECLABEL is represented by an installation chosen name (1-8 characters).

There are three SECLABELs which are already defined when the system is IPLed. These are SYSHIGH, SYSLOW, and SYSNONE. These labels cannot be changed directly; they are affected indirectly by the addition or deletion of security levels and categories. The SYSHIGH SECLABEL contains the numerically highest security level and all the categories on the system. The SYSLOW and SYSNONE SECLABELs contain the lowest security level on the system and no categories. The SYSNONE SECLABEL allows all users to write into an object with that label, even if the user does not have write access to SYSLOW information. For example, all users need to write to the catalog so that data sets can be cataloged. If SYSLOW was the catalog's security level, only users with write access to SYSLOW objects would be able to write to the catalog.

The following is an example of how these labels are defined. Suppose a security administrator defined three security levels (the number in parentheses is the hierarchical number representing the level): UNCLASSIFIED(10), SECRET(30), and TOP_SECRET(20); three categories: ABC, LMN, and XYZ; and three SECLABELS: ACCESS1 (UNCLASSIFIED, LMN), ACCESS2 (SECRET, ABC, LMN, XYZ), and ACCESS3 (TOP_SECRET). The SYSHIGH SECLABEL would contain (SECRET, ABC, LMN, XYZ) and the SYSNONE and SYSLOW SECLABELS would contain (UNCLASSIFIED). Note that the SECLABEL ACCESS2 has the same level and categories as the SECLABEL SYSHIGH and that these two SECLABELS are equal.

The representation of a SECLABEL is a profile. The following sections address the access control policy, tranquility, and labeling of subjects and objects.

5.3.1 Access Control Policy

The MVS/ESA implements the Bell and La Padula access control model except for a few features which are not implemented in MVS/ESA (execute access for objects and directory trees). The following rules are enforced:

1. A subject may read an object only if the following two conditions are met:
 - the security level of the subject is greater than or equal to the security level of the object,
 - the security category set of the subject includes the security category set of the object as a subset.
2. A subject may read and write an object only if the following two conditions are met:
 - the security level of the subject is equal to the security level of the object,
 - the security category set of the subject is the same as the security category set of the object.

5.3.2 Tranquility

Tranquility is defined as the inability of users to access any resource while either the user's or the resource's security label is being changed (through direct security label change or through change in the security level or category representations). In a B1 system only the security administrator (a user with the SPECIAL attribute) may change security labels or the representations of security levels or categories. There are two RACF commands (SETROPTS MLSTABLE and SETROPTS MLQUIET) which must be issued before any security label changes can be made.

5.3.3 Subjects

A user may have the ability to log in at one or more sensitivity levels. A sensitivity level is represented by a SECLABEL. Users are given access to resources protected by a given SECLABEL by giving them access to that SECLABEL profile. The RACF segment of the user's profile contains a default SECLABEL used during logon. If no SECLABEL is specified when the user logs in, the default is used. For batch jobs, the SECLABEL is specified on the job card. If not specified on the job card and the job is entered through an internal reader, the submitter's SECLABEL is propagated to the job. Users can issue a RACF command to find out their default SECLABEL.

In creating a subject on the system (user logs on, batch job starts, operator logs on, started task starts), RACF is called to create the UTOKEN for the subject. A UTOKEN contains the SECLABEL, the userid, the groupid, and the type of token it is (address space, command, operator, STC, mount, login, batch job or execution batch monitor). A UTOKEN stays with the subject and is used by RACF to obtain the security label when RACF performs the MAC check.

5.3.4 Objects

Objects have labels assigned to them. In general objects have profiles associated with them which contain the security label. When a subject creates an object not covered by a profile the UTOKEN of the subject is copied and becomes the RTOKEN for the object. The RTOKEN contains the same information as the UTOKEN, and it is given to RACF for the MAC check so RACF can find the security label of the object.

DASD Data Sets

All DASD data sets (both VSAM and non-VSAM) are protected via RACF. All MAC checking is done via the normal RACF MAC checking procedures. With the PROTECTALL option (mandatory for B1), all data sets must be created with either a generic or a discrete profile covering them. The SECLABEL for the data set is kept in the RACF profile.

SYSOUT Data Sets

A SYSOUT data set is given the SECLABEL of the job that creates it. The JES maintains the security label with the data itself as an RTOKEN in the JES spool, not in a RACF profile. Only the creating job can write the SYSOUT data set. Only the creating job or an authorized TSO user can read the data set. A TSO user can access the data set only if the user's SECLABEL dominates the SECLABEL of the SYSOUT data set.

When data is offloaded from the spool to another device, JES copies the security information for the data to the offload job and the data set headers. No verification is made of the security information

written to the offload data set. The JES calls RACF to ensure the operator starting the offload operation has sufficient authority to issue the command to start the offload. When the data is reloaded to the spool from an offload data set, the security data is reverified to insure it is still valid. The data will not be reloaded if any of the security data is no longer valid.

Tape Volume Sets

The first object (e.g., data set) written to a tape defines the security label for the whole tape volume. All other data sets placed on the tape must have the same security label. The tape volume set's SECLABEL is kept in its RACF profile in the RACF database.

TSO/E Messages

The four message commands which require MAC controls are SEND, LISTBC, TRANSMIT, and RECEIVE. The security label of a message is kept in an RTOKEN.

For the SEND command the sender's security label becomes the label of the message. If the "Now" parameter is specified and the receiver is not logged in or is logged in at a lower security label than the message, the message is discarded. If the "Save" or "Logon" parameters are specified and the receiver will never be able to read a saved message because of the message's security label, the message is discarded.

For the TRANSMIT command the sender's security label becomes the security label of the message or data set being sent. The receiver must issue the RECEIVE command to get the message. If the receiver's current security label is insufficient to read the message, the receiver will get no indication that any message has arrived. If the receiver will never be able to read the message, it is discarded.

5.4 Object Reuse

Reuse of the protected objects and of storage is handled by various hardware and software controls, including the RACF ERASE option, and by administrative practices.

5.4.1 RACF ERASE Option

The RACF provides an ERASE option which augments the operation of deletion. This option is set using the set RACF options (SETROPTS) command to specify the condition as to when data sets are to be erased or physically overwritten with zeros. The four possible arguments are outlined below:

Final Evaluation Report IBM MVS/ESA
CHAPTER 5. PROTECTION MECHANISMS

- **ALL:** All data sets are overwritten.
- **SECLEVEL:** Data sets above a specified security level are overwritten.
- **NOSECLEVEL:** Data sets which have their profile erase indicator on are overwritten.
- **NOERASE:** No data sets under RACF control are overwritten.

For the evaluated system, the ERASE parameter must be set to ALL, ERASE(ALL), so that the physical extents of DASD data sets are marked to be erased (overwritten with zeros at the time of deletion).

The RACF does not erase any of the extents, but instead maintains the argument specified. The deletion is actually executed by the DFP DELETE utility which references the ERASE argument. This utility is called by the DELETE command or by JCL instructions.

DASD Data Sets

The ERASE option affects the erasure of VSAM data sets and all non-VSAM data sets, including single and multiple volume data sets.

5.4.2 Controlling Reuse of Other Objects

The following objects are not controlled by the ERASE option: SYSOUT data sets, tape volumes, TSO TPUT messages, and RACF data set and resource profiles. Data reuse for these objects is handled by administrative practices or by the other software control, as detailed below.

SYSOUT Data Sets

The reuse of storage occupied by SYSOUT data sets is under strict JES control. Whenever space is released, it cannot be read until new information is first written there. If the entire pool data set is deleted, rules pertaining to DASD data sets are in effect.

Tape Volumes

The reuse of tape volumes is controlled by administrative practices. Specifically, a security administrator is responsible for maintaining a scratch pool of tapes. A scratch pool consists of either new or degaussed tapes.

The administrator may execute the SEARCH command using the EXPIRE operand to determine which volumes are beyond their security retention period. The tape's RACF volume definition is then removed before the tape is manually degaussed and placed back in the scratch pool.

Additionally, a user may relinquish a tape before the retention period is over by notifying the administrator. At this point, the data on the tape has not been deleted, and it is the responsibility of the administrator to degauss the tape.

TSO TPUT Messages

The TPUT macro supported by TSO is used to pass TPUT messages to VTAM. Specifically, they are placed in the CSA which is protected by its own storage key and therefore not addressable by users.

The TSO/E SEND command uses TPUT to transmit messages between users. The TPUT operation allows messages to be displayed when the receiving user is logged on or saved until the receiving user logs in. Once the message is displayed, it eventually is overwritten by new text sent to the terminal. At the end of a logon session, the terminal screen is cleared.

Messages can be saved to mail logs or into the SYS1.BROADCAST data set. The SYS1.BROADCAST data set is writable only by the system operator and is used for public announcements, whereas the user mail log data sets hold messages for individual users. In either case, the data sets are protected by RACF control and belong to specified users. Therefore when deleted by the user, the erase on scratch option will be invoked.

5.4.3 Controlling Reuse of System Structures

The following are some of the more prominent data structures under direct control of the TCB.

Address Spaces, Data Spaces, and Hiperspaces

Data reuse for address, data, and hiperspaces is controlled by the translation from a virtual address space to real pages in memory and whether or not that page had been previously referenced. This control is true for all spaces.

Specifically, a page-fault will occur only when an operation attempts to address a page not currently in real memory. The acquired real page is then filled with data from the virtual address space (page-in). If a real page is acquired to be used as a virtual page that has never been referenced, the page is filled with zeros (an operation of RSM).

Virtual pages are acquired either by the GETMAIN or the STORAGE (for AR mode programs) macros, and released through the FREEMAIN or the STORAGE macros. These macros ensure that

Final Evaluation Report IBM MVS/ESA
CHAPTER 5. PROTECTION MECHANISMS

the page tables for newly acquired virtual pages indicate that the pages have never been referenced. Any attempt to access memory not obtained via the above macros will cause a page-fault to occur and the task to ABEND.

An initiator performs further operations in its address space to control data reuse. After the last job step completes, the terminating portion of the initiator performs the following actions: all control blocks associated with the job in LSQA and SQA are released, and the ASM is notified to free all of its control blocks for VIO data sets created by the job.

Catalogs

Catalogs are DASD data sets under strict control of the TCB. Users may be allowed to manipulate specific entries. Users are allowed to add and remove their own entries but cannot manipulate the entire catalog. When an entry is deleted and the storage it occupied is reused, it is first overwritten with new information. When the entire catalog is deleted, it is treated just like a data set.

Volume Table of Contents

Entries in the Volume Table of Contents (VTOC) are overwritten when a data set residing on a volume is deleted. This action is executed during deletion and occurs regardless of the argument specified by ERASE.

The VTOC data set (the physical extent on the volume) can only be overwritten when the entire volume is re-formatted.

RACF Data Set and Resource Profiles

The data set and resource profiles are data structures found in the RACF database. When they are deleted, e.g., when the resource they are protecting has been deleted, the storage they occupied becomes eligible for reuse. However, that storage cannot be read until new information has been placed there. The new profile owner can then legitimately retrieve that data.

Temporary Data Sets

The VIO temporary data sets are controlled as a logical group of pages which reside in an address space. These data sets will remain in the address space for the duration of the job, although the pages may physically reside in auxiliary or real memory. When the job terminates, the pages occupied will be released. This release is controlled by the same paging operations, as explained earlier.

Data reuse for VSAM and non-VSAM temporary data sets is controlled in the same manner as other VSAM or non-VSAM data sets. The only difference is that the temporary data sets are overwritten (controlled by the ERASE option) at the termination of the job or the job step releasing these data sets.

Non-SYSOUT Spool Data Sets

Spool data sets are controlled by JES and are located on a spool volume specifically allocated for JES. Access to the data in the spool data set is allowed only to the owners and authorized programs. If the spool data sets are deleted, then the ERASE option causes the data sets to be overwritten.

5.4.4 Storage Reuse

Main Memory

There are three abstractions, all previously described, which pertain to this resource: address spaces, data spaces, and hiperspaces. The reuse of this type of storage falls under these three abstractions.

Disk Storage

There are several abstractions, again all previously described, which organize the storage in this resource: DASD data sets, SYSOUT data sets, RACF profiles, catalogs, VTOCs, VIO data sets, temporary data sets, and non-SYSOUT spool data sets. The reuse of this type of storage falls under these abstractions.

Terminal Storage

Terminals have internal memory limited to one screenful. The information displayed on the terminal's screen is scrolled up when subsequent screens are displayed. The final screen of a session is overwritten with a logon screen by TCAS.

Printer Storage

System printers do not store information. However, information is stored in their controllers, but this information is unavailable even to the privileged channel commands. Each time a new data set is sent to the printer, the page segment and overlay information is overwritten. Similarly, a new page overwrites the old one without any possibility of printing it first.

5.5 Auditing

The MVS/ESA audits job initiation, job start, and job termination; PSF audits printing events; and RACF has the ability to audit identification and authorization events, operator actions, and user accesses to resources. The products mentioned above generate audit records for all security relevant events and pass these records to SMF, which stores them in data sets owned by SMF. The RACF Report Writer generates tailored audit reports from data in the SMF data sets.

5.5.1 SMF Records and Data set Management

There are over 60 types of SMF records that record different categories of data such as machine data, auxiliary storage data, VSAM data set activity, JES data, and RACF data. The RACF produces three SMF record types: 80, 81, and 83. Two security relevant SMF record types produced by MVS are types 20 and 30. PSF produces security relevant SMF record type 6.

When the current recording data set cannot accommodate any more records, the SMF writer automatically switches recording from the active SMF data set to a empty SMF data set, and passes control to the SMF dump exit. The operator is then notified that the data set needs to be dumped. When the last available SMF data set is opened for use, a message is sent to the operator. If all SMF data sets are full, SMF will be unable to record data until an SMF data set is dumped. When this condition occurs, the system will be halted if the appropriate parameter has been set (PARMLIB member LASTDS(HALT)); otherwise the system continues to operate, and a message is sent to the operator indicating all the data sets are full.

The SMF address space contains buffers to temporarily store records before writing them to the SMF data set. Because I/O to the SMF data set may be slower than the rate at which records are arriving in the SMF address space, the buffers may run out of space. When the SMF buffers are 80% full, a message is sent to the operator. If the buffers fill up audit data could be lost. When this occurs the system will send a message to the operator or halt the system to prevent loss of audit data depending on the PARMLIB member NOBUFFS setting. If MVS fails, the internal buffers can be recovered from a system dump.

5.5.2 Auditor

Only the auditor is able to perform RACF audit functions. An auditor is a user with the AUDITOR or group-AUDITOR attribute. The group-AUDITOR attribute allows the same authorities as the AUDITOR attribute except that the authorities only apply to a specific group and its subgroup(s).

The auditor can specify audit controls by using the RACF SETROPTS command or the "Set Audit Options" ISPF panels. These controls can direct RACF to log the following: accesses to specific data sets; accesses to specific general resources; changes to profiles; the activities of users with the SPECIAL, group-SPECIAL, OPERATIONS, and group-OPERATIONS attributes; command

violations; access attempts to resources with a specified SECLABEL; MVS operator commands; and changes to SECLABELs. The auditor may also read the contents of profiles to determine the current auditing being performed on the system.

5.5.3 Logging

The RACF produces SMF record types 80, 81, and 83. Record type 81, the RACF Initialization Record, is written at the completion of the initialization of RACF. The information contained in this record includes the name and volume identification of each RACF data base, the data set name and volume identification of the UADS data set, RACF options, and the maximum password interval. Record type 80, the RACF Processing Record, is written for most other RACF events. RACF writes one record for each event. Included in the type 80 record are: time and date, event code and qualifier, processor identification, userid, group name, reason for logging, terminal id, SECLABEL, and relocate sections. Relocate sections record the name of the object that caused the security event to take place. Record type 83, the RACF Processing Record for Auditing Data Sets, is written to record data sets that are affected when a SECLABEL is changed. The type 83 record includes the same information as a type 80 record.

RACF always logs the following:

- Use of the RVARY of the SETROPTS command
- Use of the RACINIT SVC during user logon and job starts
- Action performed by the console operator to grant access to a resource as part of the failsoft processing performed when RACF is inactive

Additionally, the RACF auditor can direct RACF to log the following events:

- Use of RACDEF SVC
- Changes to RACF profiles
- RACF commands issued by a SPECIAL or group-SPECIAL user
- RACF command violations
- RACF-related activities of specific users
- Access to resources allowed because the user has OPERATIONS or group-OPERATIONS attribute
- All or some accesses to specific data sets
- All or some accesses to specific general resources

Final Evaluation Report IBM MVS/ESA
CHAPTER 5. PROTECTION MECHANISMS

- All or some accesses to specific classes
- Use of operator commands (MVS and JES)
- Use of PSF printers.

The owner of a resource can specify in the resource profile what type and level of accesses to log, or that no logging is to occur. Owner-controlled logging is not directly under the control of the auditor; however, the auditor can expand a resource owner's logging specification by issuing RACF auditor commands. The security auditor cannot change or delete the resource owner's logging specification. There is no auditing of access granted by Global Access Checking or by FRACHECK.

The MVS produces SMF record types 20 and 30. Record type 20, the Job Initiation record, is written at job initiation which includes TSO/E logon. This record includes: time and date, processor identification, userid, group name and terminal id. Record Type 30, Common Address Space Work record, is used for job start and termination information. Included in the type 30 record are: job start and end times, processor identification, job name, userid, group name, terminal id, and job step completion codes.

Record type 6, the Print Services Facility record, is written whenever data set processing is complete for a job. The information contained in this record includes: the data set name, device name SECLABEL, date and time, job name, userid, status of the printed job, and whether labeling was overridden.

5.5.4 JES Complex Auditing

Each machine within a JES complex maintains its own SMF data sets. In order to determine the audit trail for the entire JES complex, the SMF data set from each system may be dumped to a common data set and the Report Writer then run using the common data set as input, or each SMF data set may be specified in JCL as input to the Report Writer. The system id of each system is recorded in the SMF log records.

5.5.5 RACF Report Writer

The RACF Report Writer is used to generate reports from the SMF audit records. A user with the AUDITOR or SPECIAL attribute can run the Report Writer. The RACF Report Writer reformats SMF records and uses these reformatted records as inputs to the modules that produce the report. The RACF Report Writer may be run as a batch job or it may be invoked by a RACF command during a TSO/E session. The input file to the RACF Report Writer consists of SMF record types 6, 20, 30, 80, 81, and 83.

The RACF Report Writer operates in three phases. The first phase, command and subcommand processing, invokes the Report Writer and allows subcommands SELECT, EVENT, LIST, SUM-

MARY, and END to be entered. The SELECT and EVENT subcommands specify which of the input records the Report Writer uses to generate the reports. Records may be selected by date, time, user, group, job name, event type, SECLABEL, or various other criteria. The events that can be used to select records include TSO/E logon, use of the ADDUSER command, use of the ADDSD, and use of SETROPTS commands. The LIST subcommand formats and prints a listing of each SMF record selected, while the SUMMARY subcommand formats and prints a summary listing of the SMF records. END terminates subcommand processing and the first phase of the Report Writer. During the second phase, record selection, the Report Writer compares the SMF records against the criteria specified in the SELECT and EVENT subcommands, and reformats the selected records if necessary. During the third phase, report generation, the reports requested with the LIST and SUMMARY subcommands are generated. The Report Writer produces reports for the LIST subcommand by listing all SMF records from the work data set in the sequence that has been specified. For each SUMMARY subcommand, the report writer produces a separate summary report of the SMF record by group, resource, command, RACF event, or owner activity (depending on what was specified).

5.5.6 Data Security Monitor

In addition to the RACF Report Writer, the RACF Data Security Monitor (DSMON) produces 11 standard reports regarding the status of RACF controls. To run DSMON, a user must have the AUDITOR attribute or at least READ authority in the DSMON profile access list. The 11 standard reports produced are:

- System report - contains the identification of the system control program and specifies which version of RACF is installed and whether it is active.
- Group tree report - lists all subgroups for the SYS1 or the user supplied group.
- Program properties table report - lists all the programs in the program properties table, indicates whether each is authorized to bypass password protection, and indicates whether each runs with a system key.
- RACF authorized caller table report - lists the names of all programs in the RACF authorized-caller table and indicates whether each program is authorized to issue the RACINIT or RACLIST SVC. RACF authorized callers are products authorized to modify ACEEs or profiles by calling RACF.
- RACF class descriptor table report - lists class name and status for all general resource classes in the class descriptor table.
- RACF exits report - lists the names of all the installation-defined RACF exit routines.
- RACF global access checking table report - lists all entries in the global access checking table.

Final Evaluation Report IBM MVS/ESA
CHAPTER 5. PROTECTION MECHANISMS

- RACF started procedures report - lists each entry in the started procedures table.
- Selected user attribute report - lists all RACF users with the SPECIAL, OPERATIONS, AUDITOR, or REVOKE attributes and indicates if they are at the system or group level.
- Selected user attribute report summary - shows total number of installation-defined users and users with the SPECIAL, OPERATIONS, AUDITOR, and REVOKE attributes at the system and group level.
- Selected data sets reports - lists all data sets that meet one or more of the selection criteria (e.g., APF authorized, RACF backup, RACF protected).

Chapter 6

Other Assurances

This chapter discusses the assurances provided by IBM for the MVS/ESA system.

6.1 Functional Testing

Software and hardware development on IBM systems is controlled by engineering cycles. These cycles include design, development, and test phases. This section provides an overview of the testing involved in these phases.

6.1.1 Software Testing

IBM employs a development guide which has a list of objectives to be met before a software product can be shipped to customers. Two major checkpoints are *Design Verification Test and Pre-Shipment Test*. The test department bases its goals on these two checkpoints. The test department is independent of the design and development departments.

Software engineering is performed during every phase of product design and development, from walkthroughs of the high level design to a complete system test with finished products. Regression testing is employed to ensure compatibility between releases. Special tests have been devised to provide an upgrade path for those sites which have no current RACF environment, or for those with a previous version of RACF.

To control the testing at each phase of development of a product, a comprehensive test plan is produced for every new software product or new version of a software product. This plan includes definition of test objectives, entrance and exit criteria, responsibilities, test schedules, tools to be used, and dependencies of the product on other products. There are several phases at which a product is tested:

- High level design
- Low level design
- Unit test
- Driver build

Final Evaluation Report IBM MVS/ESA
CHAPTER 6. OTHER ASSURANCES

- Function test
- System test
- Installation walkthrough
- Performance test
- Field test
- Install test

After all the above phases, IBM, through its early support program, provides the product to a limited number of customers. This gives the final assurance that a product is ready for general release. Descriptions of each phase follow.

The high level design phase consists of developing an English description of the new or changed product. Reviews of the specifications and walkthroughs are utilized to accomplish this phase. In the low level design phase, algorithms are described using formal tools or methods, recovery considerations are defined, and data layouts are created. This phase has the goal of producing the code.

During unit test the execution of the smallest program unit is verified. For example, all entry points are invoked, each conditional branch is executed both ways, and defined inputs are checked to see if they give the expected outputs. During the driver build phase, unit tested code is collected into functioning subsets of the product called drivers. The product's comprehensive test plan is also produced in this phase.

The function test phase exercises the algorithms and the interfaces at a low level. Sources for tests include specifications, participation in design and code walkthroughs, code listing, pre-existing test cases and documentation drafts. The system test phase has the goal of exposing the whole product to a real, production-like environment. There are four parts to this test including the initial functional evaluation, the basic test of the product, the load and stress test of the product, and the characteristic evaluation of the product.

During the installation walkthrough phase, the instability of a product is examined. Representatives from development organizations participate in typical installations and migrations of the product. Performance testing is the next step, where the goal is to obtain an estimate on the capacity and the throughput of the product.

The field test phase involves running a pre-install test version of the operating system with all the necessary products and hardware at an internal IBM site. The purpose of this phase is to observe the product in a production environment. The IBM Software Distribution (ISD) sends products to customers. The install test phase makes sure that the materials the IBM Software Distribution receives are a complete set.

6.1.2 Hardware Testing

Hardware functional testing is an important part in the total test scheme of a modern computer product. Rather than assume the hardware works by default (i.e., the software runs), a set of tests designed to ensure that the architecture implemented by the machine conforms to the specification given in the *Principles of Operation* is run on every machine that leaves the manufacturing facility. These tests have the additional advantage of being able to be run after the system is in place to ensure correct functioning of the hardware and also to aid in hardware failure detection.

The systems that IBM has devised to perform this function for the ESA/370 architecture are the Systems Assurance Kernel (SAK) for the 3090 processors and the System Test Program (STP) for the 4381 processors. Each system consists of approximately one million lines of code, and it can be used to test other architectures such as S/370 and 370-XA. An appropriate system is used at all points in the manufacturing life-cycle of a hardware product, including sub-system simulation, element simulation, system simulation, assurance verification testing and manufacture verification testing, final verification testing and field verification and maintenance testing.

Both SAK and STP can support a number of environments, such as multiprogramming, multiprocessing, extended addressing, and V=V and V=R modes. The test programs are based on a series of random instructions or command sequences with random data patterns where ever possible. Testing covers all aspects of the final system, including the central processor, cache, main and extended memory, and the storage controller, and portions of the channel subsystem such as the channels themselves, the DASD and tape device operations, CTC functioning, and other device operation (e.g., printers, card readers).

Failing instructions and operations are logged, and can be isolated so that correction of the error is facilitated. A number of new test generation techniques, as well as testing methodologies, were devised and implemented in the SAK and the STP. A typical testing scenario involves hosting the majority of the testing software on a separate machine, compiling the test case, and then downloading it to the target machine. After the run, all data collected from the run are analyzed on the host.

The SAK and STP are run on all machines before leaving the factory to validate the machine conformance to the ESA/370 architecture. Field engineers also have a scaled-down version of the test program which they can use to supplement the diagnostics that are resident on the machines which they service.

6.2 Security Model

The *Security Model of the MVS/ESA B1 System*, dated 9 May 1990, informally models the security policy supported by the TCB. The document describes the relationship of the security policy to the Bell and La Padula security model, and shows that it satisfies the axioms of that model. It provides a

description of subjects, objects, and access modes. It also summarizes the security-relevant features of the hardware architecture and discusses the use of these features by the operating system.

6.3 System Modification Program

IBM provides a Program Update Tape (PUT) usually distributed every 6 weeks which contains updates and revisions to MVS/ESA. Other products are updated on a demand basis, but somewhat less frequently than MVS/ESA. The revisions are referred to as a Program Temporary Fix (PTF).

For an installation to handle and manage PTFs effectively, IBM furnishes the System Modification Program/Extended (SMP/E). This program provides the system administrator with a means to add or change the system or its parameters. In addition, SMP/E provides a means to update library functions, revise system modules, define macros not available during the initial product load; it is used during system generation.

The SMP/E executes authorized, but since it is used only by system programmers during maintenance and system generation (before the system is operating in a trusted mode), it is not part of the evaluated software.

Hardware changes are less frequent than software changes, but a similar distribution system exists for hardware microcode fixes. The microcode patches are shipped to the user site and are propagated in the system by doing a microcode load and re-IPLing the system.

6.4 System Generation

The purpose of system generation is to build the required MVS libraries that suit the needs of an installation. Through this sysgen process, modules are selectively chosen from the distributed library and then placed into the actual system library. In addition, the sysgen will create the necessary I/O blocks required by the selected system.

An upgrade path is also provided for users that have a C2-rated system. IBM provides step-by-step instructions for performing such an upgrade, and these steps have been found to provide accurate guidance in bringing the system to the B1 level.

Presently, IBM offers its customers four methods to generate an MVS system from scratch. The customer is responsible for choosing one of these methods: direct product order, Custom Build Product Delivery Option (CBPDO), Custom Build Installation Productivity Option (CBIPO), and MVS express.

The direct product order method delivers separate tapes and manuals for each product. A product program directory provides the necessary information to cover installation. The SMP/E plays a

vital role for this method of installation because each product must be loaded into the SMP/E data base before it can be installed on the system.

The CBPDO helps to automate the direct method by delivering one logical tape and to standardize the licensing agreement for all of the products ordered. The customer is responsible for selecting all PTFs to be included. SMP/E is used to install the system from the tape.

The CBIPO option delivers a full system which has had the software configuration checked. It also automates the ordering and licensing process, similar to CBPDO. One package is delivered containing all software and documentation required to install the system. To help assist the customer, the documentation is centered around a model installation. The customer is responsible for loading the tape to a disk and selecting the system configuration.

The MVS express is a service which is intended for first time customers who have been recommended by their IBM representative. This service minimizes the customer's effort by selecting system components and actually IPLing the system.

6.5 System Integrity

IBM provides several utility programs that can be used to verify the correct operation of the hardware and firmware.

6.5.1 On-Line Test Executive Program (OLTEP)

The OLTEP is an IBM utility for executing the online programs that test all evaluated control units and devices. The OLTEP is a standard component of the operating system, and resides in the system libraries. It runs as a system job, causing minimum interference with the normal system operations (i.e., other jobs can be run while OLTEP is running). This program performs diagnostics, prints the diagnostic information, and verifies repairs for peripheral devices.

6.5.2 SYS1.LOGREC Error Recording Data Set

The system error recording programs (ERP) write information about all hardware failures, selected software errors, and system conditions into the SYS1.LOGREC data set. These programs consist of the Recovery Termination Manager and the Error Recovery Programs (one for each device type). The records in this data set can contain either error statistics or environmental data. Examples of error statistics are channel or I/O device count failures, times of system failure and hardware status at time of failure. Examples of environmental data include time and circumstances for each failure, and device/control unit and software system recovery attempt results. These records are recorded in chronological order.

Final Evaluation Report IBM MVS/ESA
CHAPTER 6. OTHER ASSURANCES

The types of events recorded in SYS1.LOGREC include operator initiated system termination, serious errors which result in abnormal termination (operator intervention required), system initialization (IPL), buffer overflow, paging I/O errors (for permanent channels), and I/O device failures (both temporary and permanent).

6.5.3 Processor Complex Exerciser

The Processor Complex Exerciser (PCX) is a system exerciser program for the IBM 3090 Processor Complex. It tests the following functions: the dynamic address translation mechanism, the address space instructions (i.e., program calls, program transfers, set primary ASN, load address space parameters), the general processor instruction set, the multi-processing instructions, contention between processors, storage protection, main storage and the processor buffers, the page in/page out instructions, and the I/O channel subsystem.

6.5.4 Built-In Diagnostics

Both the 3090 and 4381 processors have extensive diagnostics built in which can be run on demand to ensure the correct operation of the hardware, and help isolate problems in the machines. These diagnostics are available at the MVS operator's console on the 4381, and on the system and service (level 2) consoles at the PCE for the 3090. Diagnostics which are activated at the time of fault can actually detail to the field engineer the part which is in error on the machine, and has proven to be extremely reliable thus far. A facility exists that will call an IBM service center with a problem report at the time of failure. This option is site-configurable and requires operator confirmation before it proceeds.

Chapter 7

Evaluation as a B1 System

7.1 Discretionary Access Control

Requirement

The TCB shall define and control access between named users and named objects (e.g., files and programs) in the ADP system. The enforcement mechanism (e.g., self/group/public controls, access control lists) shall allow users to specify and control sharing of those objects by named individuals, or defined groups of individuals, or by both, and shall provide controls to limit propagation of access rights. The discretionary access control mechanism shall, either by explicit user action or by default, provide that objects are protected from unauthorized access. These access controls shall be capable of including or excluding access to the granularity of a single user. Access permission to an object by users not already possessing access permission shall only be assigned by authorized users.

Applicable Features

MVS/ESA mediates interaction between users and the system's protected resources. The TCB uses an access control list mechanism on protected resources to perform access mediation. All protected resources have a corresponding RACF profile (access control list) that is compared to the user attributes contained in the user profile to mediate access.

The RACF has the capability to categorize users into groups. Access can be specified on both a user and group basis.

A resource owner (anyone specified with the ALTER or CONTROL attribute) or an appropriately privileged user can control the sharing of the system's protected resources. Only those personnel can control who is granted access to a protected resource.

All newly created objects are protected by default. This is accomplished by activating the required RACF SETROPTS options listed in the Trusted Facility Manual, *MVS/ESA Planning: B1 Security*.

The RACF has the capability of specifying access to the granularity of a single user.

Final Evaluation Report IBM MVS/ESA
CHAPTER 7. EVALUATION AS A B1 SYSTEM

Conclusion

MVS/ESA satisfies the B1 Discretionary Access Control requirement.

Additional Requirement

The following changes are made in this requirement at the B3 level:

CHANGE: The enforcement mechanism (e.g., access control lists) shall allow users to specify and control sharing of those objects. These access controls shall be capable of specifying, for each named object, a list of named individuals and a list of groups of named individuals with their respective modes of access to that object.

ADD: Furthermore, for each named object, it shall be possible to specify a list of named individuals and a list of groups of named individuals for which no access to the object is to be given.

Applicable Features

The RACF resource profiles can include an arbitrary number of users and groups, each of which can either be granted or denied access. Access modes can be specified that determine what type of access is permitted.

Conclusion

MVS/ESA satisfies the additional provisions of the B3 Discretionary Access Control requirement.

7.2 Object Reuse

Requirement

All authorizations to the information contained within a storage object shall be revoked prior to initial assignment, allocation or reallocation to a subject from the TCB's pool of unused storage objects. No information, including encrypted representations of information, produced by a prior subject's actions is to be available to any subject that obtains access to an object that has been released back to the system.

Applicable Features

MVS/ESA uses the RACF ERASE option, administrative practice, and various software and hardware features to address the object reuse requirement.

The RACF ERASE option, when set to ALL, ERASE(ALL) causes all DASD data sets to be overwritten with zeros at the time of deletion (release). The deletion is executed by the DFP DELETE function which is initiated by JCL commands or by the DELETE command.

The SYSOUT data sets are controlled by JES with the access allowed only to owners and authorized programs. JES performs a logical delete function when a spool data set is no longer needed. The space previously occupied by a data set is then overwritten prior to its assignment to another job.

Tapes volumes are handled in a procedural manner. Operators are instructed to degauss a tape before placing it in the tape pool. Once properly degaussed, residual information cannot be obtained from the tape volume.

The TSO TPUT messages place information in the VTAM address space which is key protected and inaccessible by users. If the address space is terminated, storage is released and is zeroed as it is being allocated to another user. If a message is saved in a data set, that object's protection scheme applies.

Main storage, obtained via system services, initially contains zeros. The system does not present any means of accessing other types of system storage. Disk, terminal, and printer storage are not directly addressable by unauthorized users. TCB software and hardware components manipulate that storage under a well-defined set of rules.

Conclusion

MVS/ESA satisfies the B1 Object Reuse requirement.

7.3 Labels

Requirement

Sensitivity labels associated with each subject and storage object under its control (e.g., process, file, segment, device) shall be maintained by the TCB. These labels shall be used as the basis for mandatory access control decisions. In order to import non-labeled data, the TCB shall request and receive from an authorized user the security level of the data, and all such actions shall be auditable by the TCB.

Applicable Features

MVS/ESA provides and maintains sensitivity labels over the subjects and objects under its control. For batch jobs, the sensitivity label is generated based on the label of the user submitting the batch job unless another label is specified in the job statement. For started tasks, the sensitivity label

Final Evaluation Report IBM MVS/ESA
CHAPTER 7. EVALUATION AS A B1 SYSTEM

is based on predefined values specified by the system administrator for that particular subject. Operators' and TSO users' labels are based on the intersection of either the default or the specified label values with the terminal sensitivity label. The label information, kept in the RACF data base and in system data sets, is protected from tampering using mandatory and discretionary controls, and by exclusive lock on the RACF data base. The label abstraction is carried in the UTOKEN while the subject is active.

Similarly, sensitivity labels are provided for the objects. The DASD and SYSOUT data sets, tape volumes, and TSO TPUT messages are defined in the RACF data base. The label abstraction for all these objects is carried in a RACF profile or an RTOKEN. The values of UTOKENs and profile fields, and of UTOKENs and RTOKENs, are compared when making mandatory access decisions.

Information can be imported to MVS/ESA only through tape volumes. The system administrator is responsible for associating a sensitivity label with the volume and registering it with the system. Failure to do so will not disclose any information, as volumes without sensitivity labels will fail all MAC checks with unprivileged users. Volume registration is an auditable event.

Conclusion

MVS/ESA satisfies the B1 Labels requirement.

7.4 Label Integrity

Requirement

Sensitivity labels shall accurately represent security levels of the specific subjects or objects with which they are associated. When exported by the TCB, sensitivity labels shall accurately and unambiguously represent the internal labels and shall be associated with the information being exported.

Applicable Features

The label information originates from the RACF data base and from JES data sets. Both are protected by the mandatory and discretionary controls, and through exclusive allocation. Only authorized users are allowed to manipulate the values in these data bases. Subsequently the label information is placed in key-protected system memory in the form of UTOKENs, RTOKENs, and profiles.

Whenever the system exports an object, it does so by including the object's label and maintaining a link between the two.

Conclusion

MVS/ESA satisfies the B1 Label Integrity requirement.

7.5 Exportation of Labeled Information

Requirement

The TCB shall designate each communication channel and I/O device as either single-level or multilevel. Any change in this designation shall be done manually and shall be auditable by the TCB. The TCB shall maintain and be able to audit any change in the current security level or levels associated with a communication channel or I/O device.

Applicable Features

MVS/ESA supports five types of devices: disk units, tape units, terminals, printers, and console terminals. Disk units (containing disk volumes with data sets), terminals, console terminals, and printers are the multilevel devices. Tape units are the single-level devices. The designations of these devices remain fixed although their operating ranges may change under the TCB's control. The tape units are considered single-level since that is their mode of operation. Once a unit is released, it may be assigned to another user at another level. All such changes are auditable.

Conclusion

MVS/ESA satisfies the B1 Exportation of Labeled Information requirement.

7.6 Exportation to Multilevel Devices

Requirement

When the TCB exports an object to a multilevel I/O device, the sensitivity label associated with that object shall also be exported and shall reside on the same physical medium as the exported information and shall be in the same form (i.e., machine-readable or human-readable form). When the TCB exports or imports an object over a multilevel communication channel, the protocol used on that channel shall provide for the unambiguous pairing between the sensitivity labels and the associated information that is sent or received.

Final Evaluation Report IBM MVS/ESA
CHAPTER 7. EVALUATION AS A B1 SYSTEM

Applicable Features

When exporting DASD data sets (to a disk unit), MVS/ESA maintains the explicit sensitivity label of that object in the RACF data base. The pairing between the data set on the disk and its corresponding entry (and the associated profile) in the data base is provided by the means of unique data set and disk names. Multiple instances of cataloged data set names are not allowed.

The TCB protects the information about the data set (found in the catalog and an appropriate VTOC) from tampering. The RACF data base is subject to mandatory and discretionary controls, and the system has it opened for its exclusive use.

When a user attempts to logon at a terminal, the TCB performs an intersection of the default or the specified security label and the security range associated with the terminal. If the intersection is not an empty set, the logon is authorized. The user can display the security label at any time during the session. Concurrent logons all are subject to this kind of access control.

Whenever a print request is made, the TCB verifies that the security label associated with the information (data set) to be printed (found in its profile) is within the range associated with the printer. If this is so, the information is exported and printed.

The information kept on a multilevel I/O device (DASD volume) and that representing its sensitivity label are both machine-readable. Terminals, console terminals, and printers do not retain information.

Conclusion

MVS/ESA satisfies the B1 Exportation to Multilevel Devices requirement.

7.7 Exportation to Single-Level Devices

Requirement

Single-level I/O devices and single-level communication channels are not required to maintain the sensitivity labels of the information they process. However, the TCB shall include a mechanism by which the TCB and an authorized user can reliably communicate to designate the single security level of information imported or exported via single-level communication channels or I/O devices.

Applicable Features

Tape units are the only type of single-level device in MVS/ESA. The TCB relies on procedural and automated means (i.e., human lookup of sticky labels and the system recognizing tape's internal la-

bel) by which it and the user implicitly designate the single security label of information transferred to or from this device.

A security label is associated with a tape volume when that resource is allocated to the user. When the tape volume is written for the first time, its security label, based on the label of the information being transferred to tape, is saved in a profile. The tape volume is also associated with the tape unit in which it is placed. This association remains in effect as long as the volume is allocated to a user.

Conclusion

MVS/ESA satisfies the B1 Exportation to Single-Level Devices requirement.

7.8 Labeling Human-Readable Output

Requirement

The ADP system administrator shall be able to specify the printable label names associated with exported sensitivity labels. The TCB shall mark the beginning and end of all human-readable, paged, hardcopy output (e.g., line printer output) with human-readable sensitivity labels that properly ¹ represent the sensitivity of the output. The TCB shall, by default, mark the top and bottom of each page of human-readable, paged, hardcopy output (e.g., line printer output) with human-readable sensitivity labels that properly represent the overall sensitivity of the output or that properly represent the sensitivity of the information on the page. The TCB shall, by default and in an appropriate manner, mark other forms of human-readable output (e.g., maps, graphics) with human-readable sensitivity labels that properly represent the sensitivity of the output. Any override of these marking defaults shall be auditable by the TCB.

Applicable Features

MVS/ESA supports 254 sensitivity levels and 11,761 security categories. Initially only three names are assigned: SYSNONE, SYSLOW, and SYSHIGH. Subsequently the system administrator may further define and activate other sensitivity labels.

The system attaches a printer job separation page to the beginning and end of every printed output. The printing of these pages can be disabled by the system administrator. Such action can be audited

¹The hierarchical classification component in human-readable sensitivity labels shall be equal to the greatest hierarchical classification of any of the information in the output that the labels refer to; the non-hierarchical category component shall include all of the non-hierarchical categories of the information in the output the labels refer to, but no other non-hierarchical categories.

Final Evaluation Report IBM MVS/ESA
CHAPTER 7. EVALUATION AS A B1 SYSTEM

by the TCB. Each set of header and trailer pages also contains a random sequence number. This number can be used by the person handling the output to detect attempts at spoofing the banner pages.

MVS/ESA also labels the top and bottom of each page of printer output with the security label of the data being printed. Printing of these labels may be disabled only by previously-authorized users. The system administrator may specify that such overrides of labeling be audited by the system.

Output produced at a user terminal is implicitly labeled at the start of the terminal session. During the session the user has an option to query the system for the session label.

Other devices capable of producing human-readable output are not included in the evaluated configuration.

Conclusion

MVS/ESA satisfies the B1 Labeling Human-Readable Output requirement.

7.9 Mandatory Access Control

Requirement

The TCB shall enforce a mandatory access control policy over all subjects and storage objects under its control (e.g., processes, files, segments, devices). These subjects and objects shall be assigned sensitivity labels that are a combination of hierarchical classification levels and non-hierarchical categories, and the labels shall be used as the basis for mandatory access control decisions. The TCB shall be able to support two or more such security levels. The following requirements shall hold for all accesses between subjects and objects controlled by the TCB: A subject can read an object only if the hierarchical classification in the subject's security level is greater than or equal to the hierarchical classification in the object's security level and the non-hierarchical categories in the subject's security level include all the non-hierarchical categories in the object's security level. A subject can write an object only if the hierarchical classification in the subject's security level is less than or equal to the hierarchical classification in the object's security level and all the non-hierarchical categories in the subject's security level are included in the non-hierarchical categories in the object's security level. Identification and authentication data shall be used by the TCB to authenticate the user's identity and to ensure that the security level and authorization of subjects external to the TCB that may be created to act on the behalf of the individual user are dominated by the clearance and authorization of that user.

Applicable Features

MVS/ESA enforces a form of the Bell and La Padula mandatory access control policy over its resources. This policy allows a subject to read an object only if the sensitivity level of the subject dominates the sensitivity level of the object and the security category set of the subject includes the security category set of the object. It further allows a subject to read and write an object only if the sensitivity level of the subject is equal to the sensitivity level of the object and the security category set of the subject is identical to the security category set of the object.

Labels are assigned to each subject and object under the TCB control and are maintained by the TCB. MVS/ESA provides 254 sensitivity levels and 11,761 security categories.

During the identification and authentication process the system computes user's sensitivity labels. This information is then used in assuring that subsequent tasks and batch jobs created by that user are dominated by that user.

Conclusion

MVS/ESA satisfies the B1 Mandatory Access Control requirement.

7.10 Identification and Authentication

Requirement

The TCB shall require users to identify themselves to it before beginning to perform any other actions that the TCB is expected to mediate. Furthermore, the TCB shall maintain authentication data that includes information for verifying the identity of individual users (e.g., passwords) as well as information for determining the clearance and authorizations of individual users. This data shall be used by the TCB to authenticate the user's identity and to ensure that the security level and authorizations of subjects external to the TCB that may be created to act on behalf of the individual user are dominated by the clearance and authorization of that user. The TCB shall protect authentication data so that it cannot be accessed by any unauthorized user. The TCB shall be able to enforce individual accountability by providing the capability to uniquely identify each individual ADP system user. The TCB shall also provide the capability of associating this identity with all auditable actions taken by that individual.

Applicable Features

MVS/ESA requires all users to identify themselves before they can request any other action to be performed. This identification is accomplished via userids. Each user has a unique userid.

Final Evaluation Report IBM MVS/ESA
CHAPTER 7. EVALUATION AS A B1 SYSTEM

This applies to console operators, TSO/E users, and batch jobs. Started tasks are identified and authenticated via the replacement table.

MVS/ESA uses passwords to authenticate users. When an interactive user logs on the system, a UTOKEN is created by RACF. The UTOKEN contains the userid, groupid, and current SECLABEL for the user. At logon each interactive user is required to supply a userid and password, and the user may specify a SECLABEL other than the default. For batch jobs, either the userid, password, and optionally a SECLABEL dominated by the user's current SECLABEL are supplied by the submitter, or the current validated userid, password, and SECLABEL are propagated along with the batch job. Started tasks have a userid and password defined in the userid/groupid replacement table, which is checked before the task can execute.

Password data are stored in a hashed, masked, or encrypted form in the RACF database. Only authorized users can access the RACF database.

The MVS/ESA TCB maintains RACF profiles in the RACF database, which is protected from access and modification by unauthorized users. Since each subject has a unique userid, individual accountability is enforced. Each userid is used to associate all auditable actions with the subject represented by that userid.

Conclusion

MVS/ESA satisfies the B1 Identification and Authentication requirement.

7.11 Audit

Requirement

The TCB shall be able to create, maintain, and protect from modification or unauthorized access or destruction an audit trail of accesses to the objects it protects. The audit data shall be protected by the TCB so that read access to it is limited to those who are authorized for audit data. The TCB shall be able to record the following types of events: use of identification and authentication mechanisms, introduction of objects into a user's address space (e.g., file open, program initiation), deletion of objects, actions taken by computer operators and system administrators and/or system security officers, and other security relevant events. The TCB shall also be able to audit any override of human-readable output markings. For each recorded event, the audit record shall identify: date and time of the event, user, type of event, and success or failure of the event. For identification/authentication events the origin of request (e.g., terminal ID) shall be included in the audit record. For events that introduce an object into a user's address space and for object

deletion events the audit record shall include the name of the object and the object's security level. The ADP system administrator shall be able to selectively audit the actions of any one or more users based on individual identity and/or object security level.

Applicable Features

RACF has the capability to audit all security relevant events and log them in an SMF data set. To prevent the loss of audit data when the SMF buffers or data sets are full, an installation has the option to halt the system when these conditions occur. The SMF data set is protected by RACF. Only users with the AUDITOR or SPECIAL attribute may generate reports from the SMF audit records.

The owner of each resource and the system auditor specify which events are to be recorded in the SMF data set. Types of events that may be audited include use of the RACINIT SVC, accesses to data sets and general resources, RACF command violations, use of operator commands, changes of security labels, overriding of human readable output labels, and RACF-related activities of specific users. The auditor may specify a userid or SECLABEL when indicating events to be audited. The information recorded in the SMF record includes userid, groupid, event code, date and time of event, success or failure, terminal id, security label, object name (if applicable), and command text (if applicable).

Conclusion

MVS/ESA satisfies the B1 Audit requirement.

7.12 System Architecture

Requirement

The TCB shall maintain a domain for its own execution that protects it from external interference or tampering (e.g., by modification of its code or data structures). Resources controlled by the TCB may be a defined subset of the subjects and objects in the ADP system. The TCB shall maintain process isolation through the provision of distinct address spaces under its control. The TCB shall isolate the resources to be protected so that they are subject to the access control and auditing requirements.

Final Evaluation Report IBM MVS/ESA
CHAPTER 7. EVALUATION AS A B1 SYSTEM

Applicable Features

The System 370-XA architecture provides many mechanisms which are used by MVS/ESA to preserve the integrity of the TCB. Address spaces, data spaces, and hiperspaces are isolated from one another by the hardware, using the address space translation mechanism. All TCB components reside in address spaces, data spaces and hiperspaces separate from user address spaces, data spaces and hiperspaces, and are protected from modification by means of the Authorized Program Facility, the two state architecture, and key-controlled memory protection. User spaces are separate from one another, and can be shared only in well defined, TCB controlled ways which have been previously described.

All subjects and objects defined in the system are controlled by the TCB. These subjects and objects have been clearly identified and are governed by the access control policy implemented by the system. In addition, extensive auditing can be performed on these subjects and objects.

Conclusion

MVS/ESA satisfies the B1 System Architecture requirement.

7.13 System Integrity

Requirement

Hardware and/or software features shall be provided that can be used to periodically validate the correct operation of the on-site hardware and firmware elements of the TCB.

Applicable Features

MVS/ESA provides many different mechanisms for allowing a site to gain and maintain correct hardware operation assurance. The On-Line Test Executive Program can be used to extensively test all processors, channels, and control units in the hardware configuration. It can also test devices such as DASD and tape drives for correct operation. The error recording functions included with MVS/ESA itself allow for recording in the SYS1.LOGREC error recording data set, and can log various statistics regarding the performance and current availability of hardware components of the machine. Processor complex exerciser and channel subsystem exerciser are also provided for use with the 3090 machines. These exercisers can be used to test each CP individually for critical functions, including address space manipulations and I/O subsystem operation. The 4381 series machines have an equivalent function through the system test program.

Low level diagnostic tests can be executed on all of the system hardware components. These are frequently run at the component power-up time but can also be invoked while the system is operational.

Conclusion

MVS/ESA satisfies the B1 System Integrity requirement.

7.14 Security Testing

Requirement

The security mechanisms of the ADP system shall be tested and found to work as claimed in the system documentation. A team of individuals who thoroughly understand the specific implementation of the TCB shall subject its design documentation, source code, and object code to thorough analysis and testing. Their objectives shall be: to uncover all design and implementation flaws that would permit a subject external to the TCB to read, change, or delete data normally denied under the mandatory or discretionary security policy enforced by the TCB; as well as to assure that no subject (without authorization to do so) is able to cause the TCB to enter a state such that it is unable to respond to communications initiated by other users. All discovered flaws shall be removed or neutralized and the TCB retested to demonstrate that they have been eliminated and that new flaws have not been introduced.

Applicable Features

The evaluation team performed testing of the security features of MVS/ESA in March and June, 1990 at an IBM site. The testing involved different hardware configurations and included the following hardware:

3090-600J Processor	3090-600J Processor
3090-600J Processor	3090-600J Processor
4381 Model 92 Processor	3880-3 DASD Controller
3990 DASD Controller	3390 DASD
3380 (AE4,BE4) DASD	3380 (AD4,BK4) DASD
3480-A22 Tape Controller	3480-B22 Tape Device
3827 Page Printer	3835 Page Printer
3174-1L Terminal Controller	3274 (31D) Terminal Controller
3279 Terminals	3088 Model 2 CTCA

**Final Evaluation Report IBM MVS/ESA
CHAPTER 7. EVALUATION AS A B1 SYSTEM**

The evaluation team first generated the operating system using IBM's CBIPO system generation process. The CBPDO process was used because some of the TCB products had not yet been prepared for the CBIPO process.

The team went on to IPL the generated system on several hardware configurations in order to execute all of the vendor's test cases. The configurations tested were a 3090-600J (no partitioning), a 3090-300J (partitioned 3090-600J), a JES2 complex with a 4381 and a 3090-300J, and a JES3 complex with a 4381 and a 3090-300J.

The team executed all of the vendor test cases at least once. Some tests were repeated to ensure they ran successfully on both a JES2 and a JES3 system. The test cases focused on the correct functionality of the security mechanisms. Each test case contained a description of the test, one or more variations of the test, instructions, and set-up requirements. The test cases exercised operator and administrator command authorization, TSO/E logon and batch job submission (Identification and Authentication), data set and resource protection (DAC and MAC), auditing of RACF commands and security events, labeling of printed output, printing to labeled printers, and use of the RACF ERASE(ALL) option (Object Reuse).

Conclusion

MVS/ESA satisfies the B1 Security Testing requirement.

7.15 Design Specification and Verification

Requirement

A formal or informal model of the security policy supported by the TCB shall be maintained over the life cycle of the ADP system and demonstrated to be consistent with its axioms.

Applicable Features

The *Security Model of the MVS/ESA B1 System*, dated 9 May 1990, informally models the security policy supported by the TCB. The document describes the relationship of the security policy to the Bell and La Padula security model, and shows that it satisfies the axioms of that model. It provides a description of subjects, objects, and access modes. It also summarizes the security-relevant features of the hardware architecture and discusses the use of these features by the operating system.

Conclusion

MVS/ESA satisfies the B1 Design Specification and Verification requirement.

7.16 Security Features User's Guide

Requirement

A single summary, chapter, or manual in user documentation shall describe the protection mechanisms provided by the TCB, guidelines on their use, and how they interact with one another.

Applicable Features

There are several documents which collectively provide users with guidance on the security features offered by the evaluated system.

TSO Extensions User's Guide and *TSO Extensions Command Language Reference* present information about the interactive interface to MVS/ESA and the allowable user commands. These manuals describe the logon and logoff procedures and general data manipulation.

The *RACF General Information Manual* provides summary information about RACF. The *RACF User's Guide* describe the protection mechanisms, their interactions, and functions. The manual, *RACF Command Language Reference*, defines the syntax and functions of RACF commands supplying information on password manipulation and access controls. The manuals are supplemented with examples of most features previously described. References to other applicable manuals are also included.

Conclusion

MVS/ESA satisfies the B1 Security Features User's Guide requirement.

7.17 Trusted Facility Manual

Requirement

A manual addressed to the ADP system administrator shall present cautions about functions and privileges that should be controlled when running a secure facility. The procedures for examining and maintaining the audit files as well as the detailed audit record structure for each type of audit

Final Evaluation Report IBM MVS/ESA
CHAPTER 7. EVALUATION AS A B1 SYSTEM

event shall be given. The manual shall describe the operator and administrator functions related to security, to include changing the security characteristics of a user. It shall provide guidelines on the consistent and effective use of the protection features of the system, how they interact, how to securely generate a new TCB, and facility procedures, warnings, and privileges that need to be controlled in order to operate the facility in a secure manner.

Applicable Features

One manual, *MVS/ESA Planning: B1 Security*, provides guidelines on the consistent and effective use of the protection features of the system. This manual describes the main steps required to install and operate the products in a secure manner, and contains pointers to several other documents which collectively provide system administrators with the details of the security features of each product. This collection of manuals is referred to as the Trusted Facility Library.

In order to obtain complete guidelines on the consistent and effective use of the protection features of the system, how they interact, and full instructions on how to operate the facility in a secure manner, the system administrator must have ready access to the complete library. This library consists of the following 36 separate manuals and describes the security features and uses of each product within the TCB.

1. MVS/ESA Conversion Notebook for System Product Version 3, Volume 2
2. MVS/ESA System Programming Library: Initialization and Tuning
3. MVS/ESA JCL Reference
4. MVS/ESA System Programming Library: Installation Exits
5. MVS/ESA System Programming Library: System Management Facilities (SMF)
6. MVS/ESA Interactive Program Control System (IPCS) Command Reference
7. MVS/ESA Operations: System Commands
8. MVS/ESA System Programming Library: Initialization and Tuning
9. MVS/ESA Message Library: System Messages Volume 1
10. MVS/ESA Message Library: System Messages Volume 2
11. MVS/ESA System Programming Library: JES2 Initialization and Tuning
12. MVS/ESA System Programming Library: JES2 Customization
13. MVS/ESA System Programming Library: JES2 Commands

14. MVS/ESA JES3 Conversion Notebook
15. MVS/ESA System Programming Library: JES3 Initialization and Tuning
16. MVS/ESA System Programming Library: JES3 Customization
17. MVS/ESA System Programming Library: JES3 Commands
18. MVS/ESA Data Administration Guide
19. MVS/ESA Data Facility Product Version 3: Customization
20. MVS/ESA Storage Administration Reference
21. MVS/ESA Catalog Administration Guide
22. MVS/ESA Checkpoint/Restart User's Guide
23. MVS/ESA Integrated Catalog Administration: Access Method Services Reference
24. MVS/ESA Interactive Storage Management Facility User's guide
25. MVS/ESA Magnetic Tape Labels and File Structure Administration
26. MVS/ESA System-Data Administration
27. MVS/ESA VSAM Administration Guide
28. ACF/VTAM Network and Implementation
29. TSO Extensions Version 2 Command Reference
30. TSO Extensions Version 2 Customization
31. Resource Access Control Facility (RACF) Security Administrator's Guide
32. Resource Access Control Facility (RACF) Auditor's Guide
33. System Program Library: Resource Access Control Facility (RACF)
34. Resource Access Control Facility (RACF) Command Language Reference
35. Print Services Facility Security Guide
36. Print Services Facility/MVS: System Programming Guide

The functions and privileges to be controlled for secure operation in MVS/ESA guidelines on use of protection features, interactions of protections features, and warnings are called out in *MVS/ESA Planning: B1 Security*. Generation of a new TCB also requires the documents numbered 1, 2, 6, 4, 8, 10, 11, 15, and 31. Examining the audit files, maintaining the audit file, and finding descriptions of audit records also requires the documents numbered 5, 22, 31, 32, 33, and 34. Security operator and administrator activities are also described in the documents numbered 7, 9, 10, 13, 17, 18, 25, 26, 27, 29, 31, 34, 35.

Conclusion

MVS/ESA satisfies the B1 Trusted Facility Manual requirement.

7.18 Test Documentation

Requirement

The system developer shall provide to the evaluators a document that describes the test plan, test procedures that show how the security mechanisms were tested, and results of the security mechanisms' functional testing.

Applicable Features

IBM's objective for testing is to insure that a product behaves as defined in the initial specifications. This is achieved through various tests at each stage of development of a product. At every stage of development there are four goals of testing: elimination of defects, exercise of usable functions, reduction of maintenance costs, and achievement of RAS (Reliability, Availability, Serviceability).

A comprehensive test plan is produced for every product on the system. This test plan addresses which products will be tested, unique hardware and software requirements, migration paths, and load and stress levels. Suites of test cases for each product in the system are created based on Final Programming Functional Specifications and drafts of customer documentation. Each test case contains a number of variations, a description of that test case, and setup/execution instructions.

Conclusion

MVS/ESA satisfies the B1 Test Documentation requirement.

7.19 Design Documentation

Requirement

Documentation shall be available that provides a description of the manufacturer's philosophy of protection and an explanation of how this philosophy is translated into the TCB. If the TCB is composed of distinct modules, the interfaces between these modules shall be described. An informal or formal description of the security policy model enforced by the TCB shall be available and an explanation provided to show that it is sufficient to enforce the security policy. The specific TCB

protection mechanisms shall be identified and an explanation given to show that they satisfy the model.

Applicable Features

IBM has developed a very extensive set of publications describing the philosophy, architecture and design logic of MVS/ESA and other evaluated products. At the highest level, *System 370 Enterprise Systems Architecture Principles of Operation* spells out the underlying architecture. The MVS/ESA described in *MVS/ESA General Information Manual*, adheres to this architecture. Likewise, general information manuals and system programmer manuals for DFP, JES2, JES3, ACF/VTAM, TSO/E, and RACF provide a corresponding level of detail. Program Logic Manuals, product workbooks and Final Programming Functional Specifications are available for product developers as well.

MVS Security describes the interfaces and interactions between RACF and other TCB components.

Conclusion

MVS/ESA satisfies the B1 Design Documentation requirement.

7.20 Additional Requirements

7.20.1 Discretionary Access Control

The following changes are made in this requirement at the B3 level:

CHANGE: The enforcement mechanism (e.g., access control lists) shall allow users to specify and control sharing of those objects. These access controls shall be capable of specifying, for each named object, a list of named individuals and a list of groups of named individuals with their respective modes of access to that object.

ADD: Furthermore, for each named object, it shall be possible to specify a list of named individuals and a list of groups of named individuals for which no access to the object is to be given.

Applicable Features

The RACF resource profiles can include an arbitrary number of users and groups, each of which can either be granted or denied access. Access modes can be specified that determine what type of access is permitted.

**Final Evaluation Report IBM MVS/ESA
CHAPTER 7. EVALUATION AS A B1 SYSTEM**

Conclusion

MVS/ESA satisfies the additional provisions of the B3 Discretionary Access Control requirement.

7.20.2 Trusted Facility Management

Requirement

The TCB shall support separate operator and administrator functions.

Applicable Features

MVS/ESA provides operator functions at the MVS consoles. Users with the SPECIAL attribute are system administrators, and users with the AUDITOR attribute perform auditing functions. It is also possible to assign users the administrator and auditor roles at the group level.

Conclusion

MVS/ESA satisfies the Trusted Facility Management requirement at the B2 level.

Chapter 8

Evaluator's Comments

SYSTEM CHANGES

Program Temporary Fixes (PTF) are shipped by IBM approximately every 6 weeks. They should be analyzed by each installation before they are applied to the system. The application of PTFs, other than those evaluated by the team, causes an installation to run a system that has not been tested and is not an evaluated system.

Since IBM elected not to participate in the RAMP program as defined today, the National Computer Security Center is unable to evaluate future changes to the system. IBM has expressed interest in the program but not to the level of requirements as described in the RAMP Program Document. However, IBM indicated its intent to continue providing the evaluated system to its customers even after new PTFs were generated.

SYSTEM COMPLEXITY

The evaluation team found this system particularly complex and, in some ways, obscure. This stems from the fact that the system originated as a batch processing system. Introduction of products such as TSO, VTAM, and RACF further increased the interfaces presented to an end user, making the interfaces less consistent. The team experienced this extensively during SYSGEN and testing, and advises that only experienced personnel attempt to do the SYSGEN.

The effects of the complexity of the system, the interplay of all of its products and parts, were felt in many areas: resolving technical questions was a slow process; coordinating the documentation details, especially test plans, from all the "product owners" was cumbersome; and the lack of a centralized, system-wide technical focus caused delays. These effects were equally well noticed by the team and by IBM. In testing, for example, there is no attempt to standardize the philosophy, design, and implementation of security related tests, or for that matter any system tests.

DOCUMENTATION

At the end-user level the documentation is often repetitious, inconsistent, and lacking in detail. Although the necessary information is contained within the documentation, it is poorly organized and difficult to use. The team was able to locate the necessary information because of the extensive training which was received from IBM.

Final Evaluation Report IBM MVS/ESA
CHAPTER 8. EVALUATOR'S COMMENTS

USER INTERFACE

User interactions with the system are significantly simplified with Interactive System Productivity Facility (ISPF), a tool which provides interactive user with a menu-driven interface to TSO, RACF, and other utilities. The ISPF does a good job of partially masking the archaic batch and file systems. At the programmer level, a robust interface to MVS is evident.

The implementation of some TSO/E command messages is not user friendly. For some commands, when a user who has no access to a resource due to MAC controls issues the command, there is no message. This is the same response an authorized user would receive (no message), which generally indicates successful completion of the command. For other TSO/E commands, when a user who has no access to a resource due to MAC controls issues the command, a message indicating there is no such resource is returned by the system.

SYSTEM ADMINISTRATION

The auditing performed by MVS/ESA can be very thorough and can result in a large number of audit records being produced. The RACF Report Writer supplied with the system to generate audit reports provides many options to produce reports containing desired events. However, in some cases the selection criteria do not discriminate enough, creating a large number of audit records for the auditor to examine. The online ISPF menus and data set searching functions offer some assistance.

Auditing events occurring within a complex is simple. The Auditor needs to include the audit logs from the separate machines in the complex in the job stream and run the RACF Report Writer sorting records by time.

The RACF gives the system administrator flexibility in password management. This allows the administrator to control the minimum and maximum length of passwords, password lifetimes, control over password syntax, password history, how many consecutive password verification attempts RACF is to permit before it revokes a userid, and the authority to cause RACF to revoke the user's right to use the system if the userid has remained unused beyond a specified number of days.

TESTING

The team has mixed feelings about the overall testing approach. A test suite was assembled for the evaluation containing tests for new features as well as existing ones. The tests for the existing features utilized tailored environments which made them incompatible with a B1 environment without substantial updates. Audit tests were manual which makes them time consuming and which lessens the likelihood that they will be repeated. Other tests which were developed in an automated manner had to be performed manually by the team. On the more positive side, the team is impressed with IBM's responsiveness in providing a good testing environment and in addressing all the problems which surfaced during the testing phase of the evaluation.

Appendix A

Evaluated Hardware Components

A.1 Processors

3090 Processors

Models 120E 150E 180E 200E 280E 300E 400E 500E 600E

Models 100S 120S 150S 170S 180S 200S 250S 280S 300S 380S 400S 500S 600S

Models 110J 120J 150J 170J 180J 200J 250J 280J 300J 380J 400J 500J 600J

Models 150JH 170JH 250JH

4381 Processors

Models 90E 91E 92E

A.2 DASD Controllers

3880-3, 3880-21 (paging only), 3880-23

3990 002, G03, J03, L03, Q03

A.3 DASD Devices

3380 (AD4, BD4, AE4, BE4, AJ4, AK4, BJ4, BK4, CJ2)

3350 (A2, B2)

3390 (A14, A18, A24, A28, B14, B18, B24, B28, B1C, B2C)

A.4 Tape Controllers

3480-A11, 3480-A22

Final Evaluation Report IBM MVS/ESA
APPENDIX A. EVALUATED HARDWARE COMPONENTS

A.5 Tape Devices

3480-B11, 3480-B22
3480 (A11, A22)
3490-D31, 3490-D32 (combination controller & device in 1 unit)
3490-A01, 3490-A02, 3490-B04
3422-A01, B01

A.6 Printers

3825 Page Printer
3827 Page Printer
3835 Page Printer

A.7 Terminals

3x7x Controllers
3174 (1L), 3274 (21A, 21B, 21D, 31A, 31D)
3xxx Terminals
3178-1, 3179 (1, 2, G1, G2)
3180 (1, 2), 3278 (2A, 3, 4, 5)
3279 (2A, 2B, 3A, 3B)

A.8 Other Hardware

Multisystem Channel Communication Units
3088 (A1, 1, 2)

Appendix B

Evaluated Software Components

B.1 TCB Software

When ordering the evaluated system from IBM a customer will indicate either number 5695-052 for the JES2 package or number 5695-053 for the JES3 package. The package received will consist of the following products and PTFs:

PRODUCTS:

MVS Version 3 Release 1.3

JES2 Version 3 Release 1.3 OR JES3 Version 3 Release 1.3

TSO/E Version 2 Release 1.1

ACF/VTAM ESA Version 3 Release 3

PSF Version 1 Release 3

DFP Version 3 Release 1.1

RACF Version 1 Release 9

PTFs:

UR03386	UR03387	UR03815	UR04540	UR04551	UR04732	UR05251	UR05372
UR05565	UR05840	UR08050	UR08356	UR09316	UR10931	UR12812	UR13349
UR15994	UR17644	UR19590	UY00293	UY00729	UY00891	UY00999	UY01975
UY02165	UY03283	UY05101	UY05119	UY05260	UY06637	UY09934	UY11554
UY11736	UY12664	UY15207	UY15787	UY16087	UY17682	UY20289	UY25685
UY26815	UY26943	UY27102	UY27643	UY27729	UY27989	UY27990	UY28204
UY28389	UY28452	UY28550	UY28673	UY28683	UY28684	UY28723	UY28793
UY28820	UY28878	UY28879	UY28889	UY29055	UY29117	UY29203	UY29206
UY29256	UY29343	UY29352	UY29557	UY29569	UY29634	UY29635	UY29704
UY29758	UY29957	UY30157	UY30231	UY30232	UY30284	UY30417	UY30441
UY30460	UY30465	UY30466	UY30467	UY30468	UY30469	UY30535	UY30550
UY30553	UY30600	UY30601	UY30604	UY30668	UY30718	UY30786	UY30847
UY30848	UY30880	UY30881	UY30942	UY31077	UY31196	UY31234	UY31362
UY31365	UY31443	UY31445	UY31447	UY31455	UY31575	UY31619	UY31730

Final Evaluation Report IBM MVS/ESA
 APPENDIX B. EVALUATED SOFTWARE COMPONENTS

UY31753	UY31757	UY31821	UY31844	UY31846	UY31847	UY31912	UY31941
UY32132	UY32133	UY32142	UY32236	UY32270	UY32324	UY32325	UY32346
UY32424	UY32427	UY32449	UY32608	UY32621	UY32628	UY32646	UY32648
UY32730	UY32763	UY32775	UY32887	UY32990	UY33028	UY33070	UY33071
UY33126	UY33127	UY33175	UY33330	UY33332	UY33333	UY33341	UY33342
UY33370	UY33405	UY33417	UY33418	UY33436	UY33601	UY33603	UY33640
UY33687	UY33806	UY33812	UY33876	UY33887	UY34145	UY34279	UY34368
UY34396	UY34397	UY34410	UY34411	UY34453	UY34479	UY34480	UY34659
UY34667	UY34689	UY34740	UY34744	UY34805	UY34807	UY34808	UY34809
UY34841	UY34900	UY34935	UY34940	UY34994	UY34996	UY35011	UY35084
UY35085	UY35221	UY35317	UY35318	UY35349	UY35368	UY35419	UY35474
UY35540	UY35541	UY35552	UY35619	UY35635	UY35734	UY35735	UY35772
UY35811	UY35820	UY35828	UY35858	UY35861	UY35894	UY35933	UY36048
UY36122	UY36127	UY36128	UY36130	UY36133	UY36143	UY36144	UY36193
UY36195	UY36196	UY36233	UY36269	UY36380	UY36431	UY36448	UY36557
UY36558	UY36559	UY36560	UY36561	UY36567	UY36609	UY36610	UY36613
UY36617	UY36698	UY36699	UY36880	UY36947	UY36949	UY37060	UY37069
UY37116	UY37118	UY37148	UY37150	UY37250	UY37342	UY37348	UY37349
UY37478	UY37519	UY37520	UY37544	UY37567	UY37590	UY37591	UY37731
UY37789	UY37833	UY37886	UY37964	UY37989	UY38165	UY38277	UY38278
UY38331	UY38332	UY38352	UY38443	UY38495	UY38497	UY38498	UY38503
UY38521	UY38670	UY38673	UY38700	UY38774	UY38777	UY38827	UY38829
UY38843	UY38888	UY38920	UY38960	UY38963	UY38967	UY39007	UY39023
UY39024	UY39100	UY39101	UY39114	UY39128	UY39163	UY39164	UY39176
UY39222	UY39228	UY39229	UY39230	UY39243	UY39257	UY39264	UY39269
UY39290	UY39293	UY39294	UY39296	UY39297	UY39317	UY39324	UY39343
UY39344	UY39374	UY39396	UY39406	UY39412	UY39445	UY39463	UY39494
UY39498	UY39507	UY39512	UY39558	UY39564	UY39593	UY39595	UY39621
UY39642	UY39644	UY39663	UY39686	UY39689	UY39697	UY39706	UY39708
UY39734	UY39737	UY39738	UY39750	UY39770	UY39795	UY39815	UY39816
UY39827	UY39843	UY39844	UY39857	UY39893	UY39895	UY39901	UY39902
UY39909	UY39916	UY39917	UY39923	UY39924	UY39925	UY39934	UY39943
UY39966	UY39979	UY39996	UY40013	UY40014	UY40023	UY40024	UY40027
UY40030	UY40051	UY40057	UY40096	UY40100	UY40101	UY40106	UY40111
UY40123	UY40153	UY40158	UY40161	UY40162	UY40168	UY40169	UY40175
UY40176	UY40195	UY40198	UY40218	UY40222	UY40224	UY40251	UY40255
UY40258	UY40287	UY40301	UY40302	UY40327	UY40338	UY40339	UY40352
UY40359	UY40384	UY40402	UY40448	UY40461	UY40467	UY40494	UY40499
UY40500	UY40503	UY40504	UY40506	UY40508	UY40540	UY40544	UY40549
UY40569	UY40571	UY40578	UY40606	UY40626	UY40627	UY40638	UY40639
UY40640	UY40645	UY40646	UY40666	UY40667	UY40723	UY40724	UY40725
UY40726	UY40727	UY40730	UY40733	UY40741	UY40791	UY40792	UY40819

Final Evaluation Report IBM MVS/ESA
B.I. TCB SOFTWARE

UY40820	UY40835	UY40840	UY40842	UY40866	UY40867	UY40874	UY40879
UY40884	UY40885	UY40888	UY40889	UY40904	UY40919	UY40953	UY40981
UY40988	UY40989	UY41049	UY41065	UY41066	UY41083	UY41086	UY41087
UY41088	UY41109	UY41110	UY41112	UY41139	UY41143	UY41147	UY41169
UY41170	UY41171	UY41182	UY41183	UY41192	UY41199	UY41200	UY41208
UY41209	UY41225	UY41231	UY41232	UY41275	UY41291	UY41292	UY41354
UY41363	UY41372	UY41373	UY41403	UY41453	UY41462	UY41475	UY41476
UY41493	UY41494	UY41501	UY41508	UY41509	UY41518	UY41547	UY41548
UY41549	UY41556	UY41558	UY41565	UY41566	UY41589	UY41614	UY41615
UY41622	UY41652	UY41653	UY41670	UY41671	UY41688	UY41689	UY41718
UY41737	UY41738	UY41743	UY41744	UY41755	UY41756	UY41760	UY41761
UY41776	UY41777	UY41783	UY41786	UY41800	UY41801	UY41820	UY41821
UY41847	UY41848	UY41875	UY41894	UY41895	UY41915	UY41916	UY41924
UY41925	UY41940	UY41955	UY41956	UY41959	UY41962	UY41963	UY41972
UY41974	UY42011	UY42015	UY42016	UY42032	UY42033	UY42039	UY42052
UY42053	UY42054	UY42062	UY42063	UY42065	UY42066	UY42067	UY42083
UY42084	UY42085	UY42086	UY42098	UY42101	UY42102	UY42107	UY42110
UY42115	UY42128	UY42129	UY42140	UY42141	UY42228	UY42251	UY42268
UY42282	UY42328	UY42335	UY42336	UY42375	UY42475	UY42478	UY42480
UY42481	UY42500	UY42504	UY42505	UY42506	UY42559	UY42567	UY42568
UY42569	UY42594	UY42595	UY42637	UY42638	UY42640	UY42647	UY42648
UY42660	UY42662	UY42675	UY42676	UY42693	UY42694	UY42706	UY42708
UY42713	UY42717	UY42719	UY42720	UY42722	UY42733	UY42742	UY42758
UY42759	UY42792	UY42820	UY42821	UY42829	UY42830	UY42835	UY42836
UY42854	UY42855	UY42865	UY42889	UY42890	UY42892	UY42893	UY42895
UY42913	UY42926	UY42927	UY42932	UY42951	UY42952	UY42965	UY42966
UY42970	UY42971	UY42975	UY42976	UY42981	UY42986	UY42987	UY42992
UY42993	UY42999	UY43002	UY43003	UY43006	UY43007	UY43017	UY43018
UY43034	UY43035	UY43036	UY43043	UY43045	UY43060	UY43061	UY43066
UY43067	UY43070	UY43071	UY43072	UY43073	UY43074	UY43077	UY43078
UY43079	UY43080	UY43082	UY43111	UY43112	UY43115	UY43134	UY43143
UY43144	UY43158	UY43170	UY43172	UY43173	UY43180	UY43181	UY43182
UY43183	UY43200	UY43205	UY43206	UY43207	UY43223	UY43226	UY43259
UY43260	UY43311	UY43313	UY43320	UY43321	UY43331	UY43332	UY43336
UY43337	UY43341	UY43342	UY43344	UY43345	UY43355	UY43356	UY43361
UY43362	UY43368	UY43369	UY43379	UY43381	UY43382	UY43385	UY43386
UY43403	UY43418	UY43419	UY43421	UY43433	UY43434	UY43455	UY43456
UY43458	UY43465	UY43473	UY43474	UY43475	UY43476	UY43484	UY43486
UY43487	UY43488	UY43489	UY43490	UY43491	UY43492	UY43493	UY43501
UY43502	UY43520	UY43526	UY43527	UY43528	UY43529	UY43576	UY43577
UY43584	UY43585	UY43620	UY43621	UY43624	UY43625	UY43646	UY43647
UY43649	UY43650	UY43685	UY43687	UY43688	UY43697	UY43698	UY43707

Final Evaluation Report IBM MVS/ESA
 APPENDIX B. EVALUATED SOFTWARE COMPONENTS

UY43708	UY43743	UY43745	UY43750	UY43751	UY43755	UY43756	UY43758
UY43759	UY43766	UY43767	UY43774	UY43780	UY43781	UY43797	UY43801
UY43803	UY43821	UY43828	UY43829	UY43835	UY43840	UY43846	UY43872
UY43873	UY43880	UY43886	UY43902	UY43905	UY43910	UY43911	UY43952
UY43989	UY43990	UY43997	UY44006	UY44010	UY44012	UY44016	UY44070
UY44072	UY44078	UY44080	UY44081	UY44082	UY44083	UY44085	UY44093
UY44099	UY44100	UY44115	UY44116	UY44129	UY44152	UY44153	UY44155
UY44156	UY44163	UY44164	UY44183	UY44200	UY44206	UY44207	UY44228
UY44229	UY44232	UY44233	UY44253	UY44285	UY44289	UY44292	UY44293
UY44303	UY44307	UY44310	UY44314	UY44315	UY44324	UY44325	UY44328
UY44331	UY44333	UY44347	UY44348	UY44364	UY44365	UY44390	UY44408
UY44409	UY44423	UY44427	UY44435	UY44438	UY44439	UY44451	UY44452
UY44465	UY44467	UY44473	UY44487	UY44489	UY44493	UY44504	UY44517
UY44519	UY44520	UY44532	UY44533	UY44551	UY44560	UY44568	UY44569
UY44572	UY44583	UY44584	UY44586	UY44587	UY44596	UY44597	UY44602
UY44603	UY44613	UY44615	UY44617	UY44619	UY44620	UY44631	UY44632
UY44634	UY44639	UY44640	UY44643	UY44648	UY44649	UY44650	UY44651
UY44657	UY44671	UY44682	UY44683	UY44689	UY44690	UY44711	UY44712
UY44725	UY44728	UY44729	UY44743	UY44744	UY44747	UY44751	UY44752
UY44754	UY44755	UY44758	UY44759	UY44777	UY44785	UY44786	UY44793
UY44794	UY44831	UY44832	UY44836	UY44838	UY44839	UY44840	UY44842
UY44843	UY44848	UY44849	UY44850	UY44854	UY44855	UY44867	UY44868
UY44879	UY44886	UY44890	UY44891	UY44893	UY44899	UY44900	UY44904
UY44911	UY44918	UY44919	UY44924	UY44940	UY44943	UY44944	UY44946
UY44948	UY45005	UY45006	UY45014	UY45015	UY45020	UY45021	UY45023
UY45026	UY45029	UY45030	UY45031	UY45032	UY45038	UY45039	UY45040
UY45041	UY45044	UY45047	UY45050	UY45054	UY45057	UY45060	UY45070
UY45079	UY45084	UY45108	UY45111	UY45130	UY45132	UY45150	UY45159
UY45166	UY45182	UY45187	UY45188	UY45194	UY45199	UY45208	UY45210
UY45212	UY45214	UY45254	UY45255	UY45264	UY45274	UY45275	UY45307
UY45309	UY45336	UY45338	UY45342	UY45343	UY45358	UY45365	UY45368
UY45369	UY45374	UY45379	UY45397	UY45401	UY45402	UY45408	UY45409
UY45410	UY45412	UY45413	UY45415	UY45419	UY45420	UY45422	UY45423
UY45445	UY45446	UY45456	UY45474	UY45475	UY45479	UY45484	UY45490
UY45492	UY45493	UY45502	UY45527	UY45528	UY45538	UY45551	UY45562
UY45603	UY45615	UY45624	UY45626	UY45627	UY45639	UY45640	UY45664
UY45672	UY45673	UY45689	UY45693	UY45694	UY45695	UY45700	UY45702
UY45704	UY45705	UY45708	UY45710	UY45711	UY45731	UY45732	UY45739
UY45745	UY45750	UY45752	UY45753	UY45755	UY45775	UY45779	UY45784
UY45808	UY45809	UY45813	UY45814	UY45824	UY45830	UY45831	UY45838
UY45851	UY45863	UY45864	UY45878	UY45889	UY45890	UY45901	UY45902
UY45906	UY45907	UY45965	UY45966	UY45975	UY45976	UY45982	UY45984

Final Evaluation Report IBM MVS/ESA
B.1. TCB SOFTWARE

UY45991	UY45992	UY45994	UY45995	UY46001	UY46056	UY46059	UY46062
UY46071	UY46078	UY46079	UY46106	UY46110	UY46124	UY46125	UY46131
UY46140	UY46145	UY46155	UY46171	UY46172	UY46174	UY46182	UY46183
UY46191	UY46192	UY46211	UY46216	UY46217	UY46219	UY46231	UY46256
UY46263	UY46265	UY46276	UY46277	UY46321	UY46322	UY46324	UY46329
UY46332	UY46334	UY46336	UY46339	UY46345	UY46346	UY46356	UY46360
UY46361	UY46364	UY46427	UY46428	UY46429	UY46431	UY46432	UY46437
UY46438	UY46440	UY46442	UY46461	UY46468	UY46487	UY46488	UY46490
UY46491	UY46518	UY46519	UY46522	UY46539	UY46540	UY46564	UY46565
UY46569	UY46570	UY46582	UY46583	UY46584	UY46588	UY46590	UY46591
UY46603	UY46604	UY46607	UY46608	UY46610	UY46611	UY46613	UY46614
UY46629	UY46630	UY46636	UY46640	UY46641	UY46643	UY46658	UY46659
UY46665	UY46666	UY46678	UY46679	UY46681	UY46686	UY46687	UY46696
UY46697	UY46722	UY46726	UY46727	UY46736	UY46737	UY46748	UY46749
UY46750	UY46752	UY46753	UY46765	UY46767	UY46775	UY46786	UY46792
UY46793	UY46794	UY46805	UY46806	UY46808	UY46814	UY46815	UY46826
UY46827	UY46831	UY46837	UY46845	UY46853	UY46861	UY46869	UY46873
UY46874	UY46885	UY46890	UY46892	UY46897	UY46905	UY46917	UY46918
UY46919	UY46920	UY46921	UY46933	UY46948	UY46949	UY46961	UY46962
UY46964	UY46965	UY46967	UY46968	UY46979	UY46983	UY46984	UY46987
UY47001	UY47003	UY47010	UY47017	UY47022	UY47028	UY47029	UY47034
UY47037	UY47046	UY47049	UY47052	UY47054	UY47055	UY47056	UY47057
UY47063	UY47065	UY47067	UY47078	UY47079	UY47089	UY47094	UY47100
UY47101	UY47103	UY47110	UY47111	UY47114	UY47116	UY47129	UY47130
UY47131	UY47139	UY47142	UY47143	UY47145	UY47160	UY47164	UY47168
UY47171	UY47176	UY47180	UY47183	UY47184	UY47187	UY47188	UY47190
UY47200	UY47201	UY47205	UY47210	UY47216	UY47231	UY47239	UY47247
UY47252	UY47256	UY47257	UY47259	UY47264	UY47269	UY47270	UY47276
UY47277	UY47301	UY47302	UY47305	UY47315	UY47324	UY47337	UY47345
UY47353	UY47361	UY47371	UY47387	UY47388	UY47393	UY47396	UY47406
UY47418	UY47422	UY47431	UY47442	UY47445	UY47451	UY47462	UY47468
UY47472	UY47473	UY47475	UY47477	UY47478	UY47493	UY47494	UY47499
UY47505	UY47506	UY47510	UY47512	UY47514	UY47518	UY47525	UY47526
UY47535	UY47537	UY47539	UY47557	UY47559	UY47568	UY47570	UY47573
UY47574	UY47576	UY47583	UY47587	UY47588	UY47592	UY47594	UY47595
UY47598	UY47606	UY47607	UY47608	UY47616	UY47622	UY47639	UY47641
UY47648	UY47649	UY47651	UY47652	UY47674	UY47675	UY47676	UY47677
UY47681	UY47682	UY47633	UY47686	UY47690	UY47691	UY47695	UY47696
UY47697	UY47712	UY47716	UY47717	UY47725	UY47734	UY47747	UY47748
UY47752	UY47753	UY47759	UY47773	UY47775	UY47776	UY47802	UY47814
UY47824	UY47827	UY47850	UY47851	UY47857	UY47880	UY47882	UY47891
UY47892	UY47896	UY47897	UY47920	UY47921	UY47924	UY47925	UY47926

Final Evaluation Report IBM MVS/ESA

APPENDIX B. EVALUATED SOFTWARE COMPONENTS

UY47927	UY47930	UY47935	UY47940	UY47965	UY47980	UY47997	UY48001
UY48002	UY48003	UY48009	UY48010	UY48021	UY48024	UY48030	UY48033
UY48034	UY48035	UY48048	UY48051	UY48052	UY48059	UY48060	UY48075
UY48076	UY48080	UY48086	UY48089	UY48090	UY48094	UY48095	UY48096
UY48099	UY48101	UY48102	UY48106	UY48115	UY48122	UY48124	UY48137
UY48149	UY48150	UY48152	UY48153	UY48155	UY48159	UY48160	UY48165
UY48181	UY48184	UY48192	UY48206	UY48214	UY48230	UY48234	UY48236
UY48244	UY48273	UY48277	UY48279	UY48282	UY48283	UY48289	UY48299
UY48310	UY48311	UY48312	UY48313	UY48314	UY48316	UY48317	UY48324
UY48344	UY48348	UY48364	UY48365	UY48373	UY48381	UY48385	UY48392
UY48393	UY48395	UY48408	UY48409	UY48411	UY48420	UY48421	UY48430
UY48435	UY48445	UY48447	UY48448	UY48454	UY48456	UY48461	UY48484
UY48493	UY48494	UY48497	UY48498	UY48502	UY48507	UY48508	UY48511
UY48512	UY48515	UY48516	UY48522	UY48525	UY48526	UY48528	UY48530
UY48534	UY48535	UY48548	UY48553	UY48554	UY48559	UY48560	UY48562
UY48563	UY48566	UY48578	UY48589	UY48598	UY48600	UY48615	UY48620
UY48626	UY48627	UY48630	UY48631	UY48633	UY48636	UY48637	UY48640
UY48641	UY48646	UY48647	UY48652	UY48653	UY48663	UY48679	UY48682
UY48687	UY48690	UY48694	UY48699	UY48700	UY48702	UY48703	UY48709
UY48719	UY48733	UY48741	UY48742	UY48744	UY48755	UY48762	UY48763
UY48771	UY48776	UY48778	UY48779	UY48780	UY48789	UY48790	UY48794
UY48803	UY48817	UY48823	UY48824	UY48836	UY48842	UY48853	UY48856
UY48865	UY48868	UY48871	UY48874	UY48875	UY48880	UY48882	UY48883
UY48886	UY48898	UY48899	UY48912	UY48920	UY48921	UY48922	UY48924
UY48925	UY48926	UY48932	UY48934	UY48938	UY48939	UY48949	UY48961
UY48963	UY48983	UY48986	UY48987	UY48989	UY48990	UY48992	UY48993
UY48994	UY48995	UY49003	UY49004	UY49006	UY49036	UY49041	UY49046
UY49060	UY49068	UY49071	UY49072	UY49074	UY49075	UY49076	UY49081
UY49083	UY49084	UY49098	UY49115	UY49119	UY49120	UY49127	UY49132
UY49133	UY49139	UY49140	UY49151	UY49152	UY49159	UY49161	UY49173
UY49174	UY49187	UY49192	UY49193	UY49204	UY49211	UY49212	UY49219
UY49220	UY49231	UY49232	UY49239	UY49242	UY49243	UY49244	UY49245
UY49246	UY49251	UY49253	UY49255	UY49256	UY49258	UY49261	UY49266
UY49274	UY49275	UY49278	UY49279	UY49282	UY49288	UY49289	UY49293
UY49299	UY49300	UY49303	UY49305	UY49306	UY49308	UY49317	UY49318
UY49320	UY49330	UY49331	UY49332	UY49333	UY49334	UY49345	UY49346
UY49353	UY49354	UY49355	UY49356	UY49363	UY49364	UY49365	UY49366
UY49372	UY49377	UY49380	UY49382	UY49383	UY49384	UY49385	UY49386
UY49389	UY49409	UY49416	UY49420	UY49422	UY49424	UY49433	UY49434
UY49435	UY49436	UY49443	UY49444	UY49452	UY49453	UY49467	UY49474
UY49475	UY49482	UY49483	UY49492	UY49500	UY49505	UY49507	UY49516
UY49523	UY49524	UY49540	UY49544	UY49561	UY49563	UY49566	UY49579

Final Evaluation Report IBM MVS/ESA
B.2. SOFTWARE OUTSIDE THE TCB

UY49596	UY49597	UY49626	UY49634	UY49652	UY49661	UY49667	UY49672
UY49675	UY49678	UY49687	UY49692	UY49702	UY49747	UY49748	UY49757
UY49759	UY49769	UY49774	UY49775	UY49777	UY49788	UY49803	UY49804
UY49812	UY49825	UY49826	UY49840	UY49841	UY49851	UY49852	UY49854
UY49857	UY49858	UY49880	UY49896	UY49898	UY49904	UY49914	UY49919
UY49933	UY49937	UY49938	UY49940	UY49969	UY49971	UY49991	UY49996
UY50023	UY50024	UY50025	UY50026	UY50041	UY50042	UY50081	UY50099
UY50166	UY50167	UY50224	UY50290	UY50291	UY50304	UY50309	UY50310
UY50327	UY50423	UY50464	UY50556	UY50558	UY50613	UY50667	UY50694
UY50699	UY50948	UY51070	UY51104	UY51175	UY51197	UY51198	UY51275
UY51284	UY51296	UY90335	UY90341	UY90343	UY90352	UY90360	UY90379
UY90407	UY90409	UY90414	UY90439	UY90441	UY90444	UY90449	UY90462
UY90463	UY90494	UY90503	UY90504	UY90515	UY90516	UY90526	UY90527
UY90531	UY90538	UY90543	UY90544	UY90545	UY90549	UY90550	UY90551
UY90568	UY90569	UY90570	UY90571	UY90572	UY90573	UY90574	UY90575
UY90576	UY90577	UY90578	UY90579	UZ38355	UZ38392	UZ38393	UZ38395
UZ38523	UZ38763	UZ38932	UZ39136	UZ39347	UZ39442	UZ39584	UZ39631
UZ39935	UZ40304	UZ40444	UZ40804	UZ41437	UZ41764	UZ42140	UZ42656
UZ44086	UZ46415	UZ47406	UZ49462	UZ49959	UZ49996	UZ51149	UZ51239
UZ52227	UZ57201	UZ57269	UZ57526	UZ58252	UZ58305	UZ58330	UZ58382
UZ59217	UZ59296	UZ59511	UZ59565	UZ59749	UZ59911	UZ59949	UZ59997
UZ60067	UZ60117	UZ60166	UZ61499	UZ61706	UZ61713	UZ61995	UZ62193
UZ62515	UZ63343	UZ63726	UZ64025	UZ64281	UZ65007	UZ66158	UZ67029
UZ67182	UZ67200	UZ67537	UZ67573	UZ67723	UZ68854	UZ68986	UZ69350
UZ69351	UZ69705	UZ70235	UZ71064	UZ71105	UZ71324	UZ71348	UZ71390
UZ71727	UZ72110	UZ72363	UZ72771	UZ73396	UZ73741	UZ73839	UZ75099
UZ75301	UZ75642	UZ76429	UZ77326	UZ77689	UZ78320	UZ78728	UZ78867
UZ80273	UZ80304	UZ81148	UZ81419	UZ82287	UZ90202		

B.2 Software Outside the TCB

Software outside the TCB that may be added to the system without affecting the rating must have the following characteristics and must not be run by privileged users:

- Cannot run in supervisor state
- Cannot run APF authorized
- Cannot run with key 0 through 7

Final Evaluation Report IBM MVS/ESA
APPENDIX B. EVALUATED SOFTWARE COMPONENTS

This page intentionally left blank

Appendix C

Acronyms

ACB	Access Method Control Block
ACEE	Accessor Environment Element
ACF/VTAM	Advanced Communications Function for the Virtual Telecommunications Access Method
ADSP	Automatic Data Set Protection
AFP	Advanced Function Printing
ALB	ART Lookaside Buffer
ALD	Access List Designator
ALE	Access List Entry
ALEAX	ALE Authorization Index
ALET	ALE Token
ANSI	American National Standards Institute
APAR	Authorized Program Analysis Report
API	Application Program Interface
APF	Authorized Program Facility
AR	Access Register
ART	Access-Register Translation
ASCB	Address Space Control Block
ASCII	American Standard Code for Information Interchange
ASM	Auxiliary Storage Manager
ASN	Address Space Number
ASR	Address Space Register
ASTE	Address Space Second Table Entry
ASXB	Address Space Control Block Extension
AT	Authority Table
AUK	Authorized User Key
BAM	Block Availability Mask
BCD	Binary Coded Decimal
BDAM	Basic Direct Access Method
BISAM	Basic Indexed Sequential Access Method
BLOB	Bunch of Loose Old Bits
BPAM	Basic Partitioned Access Method
BSAM	Basic Sequential Access Method
CBPDO	Custom Built Product Delivery Option
CBIPO	Custom Built Installation Productivity Option
CCE	Channel Control Elements

Final Evaluation Report IBM MVS/ESA
APPENDIX C. ACRONYMS

CCW	Channel Command Word
C/I	Converter/Interpreter
CP	Central Processor
CSA	Common Storage Area
CTC	Channel to Channel adaptor
DAC	Discretionary Access Control
DASD	Direct Access Storage Device
DAP	Design Analysis Phase
DAT	Dynamic Address Translation
DCB	Data Control Block
DEB	Data Extent Block
DFP	Data Facilities Product
DLF	Data Lookaside Facility
DoD	Department of Defense
DSCB	Data Set Control Block
DSMON	Data Security MONitor
DSP	Dynamic Support Program
DU-AL	Dispatchable Unit Access List
EAX	Extended Authorization Index
EBCDIC	Extended Binary-Coded-Decimal Interchange Code
ECB	Event Control Block
ECSA	Extended Common Storage Area
ELSQA	Extended Local System Queue Area
EPL	Evaluated Products List
ERP	Error Recording Program
ESA	Enterprise Systems Architecture
EXCB	Extended ECB
FLIH	First Level Interrupt Handler
FLPA	Fixed Link Pack Area
FSA	Functional Subsystem Application
FSI	Functional Subsystem Interface
FSS	Functional Subsystem
GMS	Global Main Scheduling
GRS	Global Resource Serialization
HASP	Houston Automatic spooling Priority
IBM	International Business Machines
ICB	Index Control Block
IML	Initial Microprogram Load
IOB	Input/Output Block
IOS	Input/Output Supervisor
IOSB	Input/Output Supervisor Block
IPL	Initial Program Load

IOS	Input/Output Supervisor
IOT	Input/Output Table
IRIM	IPL Resource Initialization Module
ISD	IBM Software Distribution
ISO	International Standards Organization
ISPF	Interactive System Productivity Facility
JCL	Job Control Language
JCT	Job Control Table
JES	Job Entry Subsystem
JES2	Job Entry Subsystem 2
JES3	Job Entry Subsystem 3
JOE	Job Output Elements
JQE	Job Queue Element
JST	Job Summary Table
JVT	Job Volume Table
LASI	Library Access System Interface
LLA	LNKLST Lookaside
LRU	Least Recently Used
LU	Logical Unit
LPA	Link Pack Area
LSA	Logic Support Adaptors
LSQA	Local System Queue Area
LSS	Logic Support Stations
MAC	Mandatory Access Control
MDS	Main Device Scheduler
MLPA	Modified Link Pack Area
MPL	Mandatory Print Labeling
MTGBM	Master Track Group Bit Map
MVS/ESA	Multiple Virtual Storage/Enterprise Systems Architecture
MVS/SP	Multiple Virtual Storage/System Product
MVS/XA	Multiple Virtual Storage/Extended Architecture
NAU	Network Addressable Units
NCSC	National Computer Security Center
NJE	Network Job Entry
NIB	Node Initialization Block
NIP	Nucleus Initialization Program
OLTEP	On-Line Test Executive Program
ORB	Operation Request Block
OS/VS1	Operating System/Virtual Storage 1
PASN-AL	Primary Address Space Number Access List
PC	Program Call
PCE	Processor Control Element

Final Evaluation Report IBM MVS/ESA
APPENDIX C. ACRONYMS

PCX	Processor Complex Extension
PDDDB	Peripheral Data Definition Block
PDS	Partitioned Data Set
PLPA	Pageable Link Pack Area
PMA	Processor Memory Array
PP	Physically Partitioned
PSA	Prefix Save Area
PSF	Print Services Facility
PSW	Processor Status Word
PTF	Program Temporary Fix
PTO	Page Table Origin
PTR	Preliminary Technical Review
PU	Physical Unit
PUT	Program Update Tape
QISAM	Queued Indexed Sequential Access Method
QSAM	Queued Sequential Access Method
RACF	Resource Access Control Facility
RAS	Reliability Availability Serviceability
RCT	Region Control Task
RIM	Resource Initialization Module
RPL	Request Parameter List
RSM	Real Storage Manager
RTOKEN	Resource Token
SAF	System Authorization Facility
SAK	System Assurance Kernel
SAT	System Authorization Table
SCE	System Control Element
SCII	Standard Code for Information Interchange
SI	Single Image
SLT	System Linkage Table
SMF	System Management Facility
SMP	System Modification Program
SMP/E	SMP/Extended
SMS	Storage Management Subsystem
SQA	System Queue Area
SRB	Service Request Block
SRM	System Resource Manager
SSCP	System Service Control Point
SSI	Subsystem Interface
SSIB	Subsystem Identification Block
SSOB	Subsystem Option Block
STC	Started Task Control

STD	Segment Table Descriptor
STO	Segment Table Origin
STP	System Test Parameter
SVC	Supervisor Call
SWA	Scheduler Work Area
TCAM	Terminal Communications Access Method
TCAS	Terminal Control Address Space
TCB	Trusted Computing Base
TCB	Task Control Block
TCSEC	Trusted Computer System Evaluation Criteria
TFM	Trusted Facility Manual
TGB	Track Group Blocks
TGAE	Track Group Allocation Entry
TLB	Translation Lookaside Buffer
TMP	Terminal Monitor Program
TOD	Time Of Day
TSO/E	Time Sharing Option Extensions
TVTOC	Tape Volume Table Of Contents
UACC	Universal Access Authority
UADS	User Attributes Data Set
UCB	Unit Control Block
UPA	User Printable Area
UTOKEN	User Token
VAP	Vendor Assistance Phase
VIO	Virtual Input/Output
VPA	Valid Printable Area
VSAM	Virtual Storage Access Method
VSM	Virtual Storage Manager
VTAM	Virtual Telecommunications Access Method
VTOC	Volume Table Of Contents
XBM	Execution Batch Monitor

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS			
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT UNLIMITED DISTRIBUTION			
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CSC-EPL-90/002		5. MONITORING ORGANIZATION REPORT NUMBER(S) S235,899			
6a. NAME OF PERFORMING ORGANIZATION National Computer Security Center	6b. OFFICE SYMBOL <i>(If applicable)</i> C71	7a. NAME OF MONITORING ORGANIZATION			
6c. ADDRESS <i>(City, State and ZIP Code)</i> 9800 Savage Road Ft. George G. Meade, MD 20755-6000		7b. ADDRESS <i>(City, State and ZIP Code)</i>			
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL <i>(If applicable)</i>	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER			
8c. ADDRESS <i>(City, State and ZIP Code)</i>		10. SOURCE OF FUNDING NOS.			
11. TITLE <i>(Include Security Classification)</i> Final Evaluation Report IBM MVS/ESA		PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
12. PERSONAL AUTHOR(S) Barbara A. Maguschak, Cynthia Reese, Robert L. Williamson					
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM ___ TO ___	14. DATE OF REPORT <i>(Yr, Mo., Day)</i> 90,09,17	15. PAGE COUNT 186		
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS <i>(Continue on reverse if necessary and identify by block number)</i> NSA, International Business Machines' MVS/ESA, TCSEC		
FIELD	GROUP	SUB GR			
19. ABSTRACT <i>(Continue on reverse side if necessary and identify by block number)</i> The National Security Agency (NSA) examined the security protection mechanisms provided by International Business Machines' MVS/ESA operating system. It was evaluated against the DoD Trusted Computer System Evaluation Criteria (TCSEC) and the evaluation team determined that the the system met all criteria for the B1 level of trust. This report documents the findings of the evaluation.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL PATRICIA L. MORENO		22b. TELEPHONE NUMBER <i>(Include Area Code)</i> (301)859-4458	8b. OFFICE SYMBOL C71		