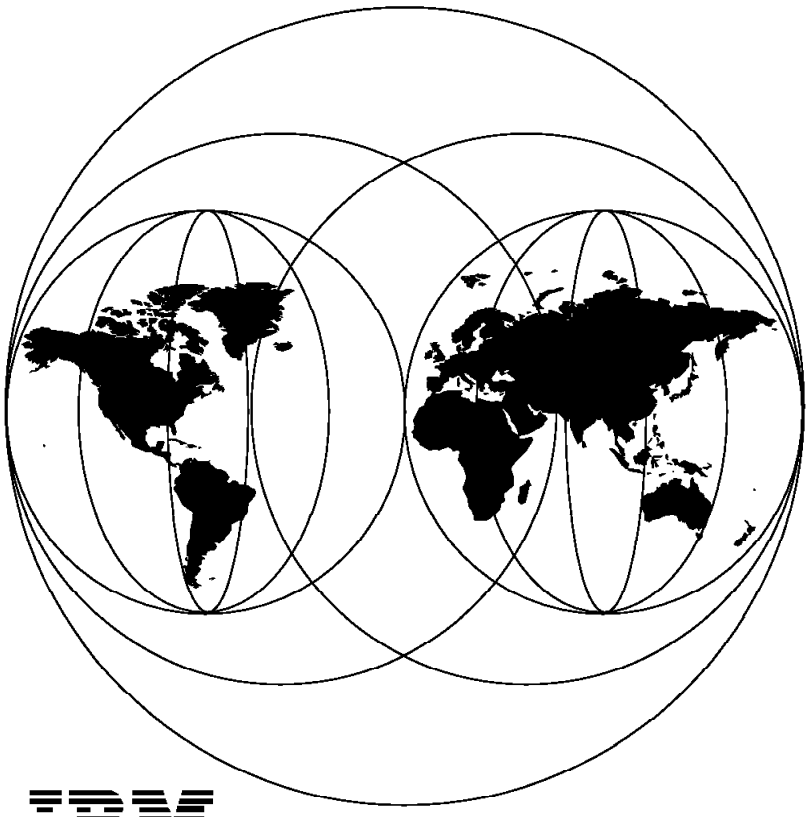
International Technical Support Organization

SG24-4593-00

Planning for CICS Continuous Availability in an MVS/ESA Environment

November 1995



IBM

International Technical Support Organization
San Jose Center

International Technical Support Organization

SG24-4593-00

Planning for CICS Continuous Availability in an MVS/ESA Environment

November 1995

Take Note!

Before using this information and the product it supports, be sure to read the general information under “Special Notices” on page xv.

First Edition (November 1995)

This edition applies to Version 3, Release 3 of the IBM licenced program Customer Information Control System/Enterprise Systems Architecture (CICS/ESA), program number 5685-083, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader′s feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 471 Building 80-E2
650 Harry Road
San Jose, California 95120-6099

Abstract

Do you want to build an online system that is available to users 24 hours a day, every day of the year? This book is designed to help you achieve a continuously available online system by using IBM′s Customer Information Control System (CICS).

We explain how you can build an environment to maximize the availability of your system, the management and operational considerations for supporting that environment, the application programming requirements, and how to exploit the features of IBM′s hardware and software to achieve continuous availability.

This book is for anyone who is involved in planning for or implementing a continuously available system based on CICS.

(176 pages)

Contents

Abstract . . . iii

Special Notices . . . xv

Preface . . . xvii
How This Document Is Organized . . . xvii
Related Publications . . . xviii
Books for CICS/ESA Version 3 . . . xviii
Books for CICS/ESA Version 4 . . . xviii
Books for Both CICS/ESA Version 3 and CICS/ESA Version 4 . . . xix
Other Books . . . xix
International Technical Support Organization Publications . . . xix
ITSO Redbooks on the World Wide Web (WWW) . . . xx
Acknowledgments . . . xxi

Chapter 1. Introduction . . . 1
1.1 The Changing Environment . . . 1
1.2 Why Continuous Availability? . . . 2
1.3 Continuous Availability, Continuous Operation, and High Availability . . . 2
1.4 Do You Need CICS Continuous Availability? . . . 3

Chapter 2. The Business Perspective . . . 5
2.1 Assessing the Cost of Unavailability . . . 5
2.2 Agreeing on Your Target Level of Continuous Availability . . . 7
2.3 Understanding Where You Are . . . 7
2.3.1 System Availability . . . 8
2.3.2 Network Availability . . . 8
2.3.3 Terminal Availability . . . 8
2.4 Estimating the Cost of Getting Where You Want to Be . . . 9
2.4.1 Causes of Unavailability . . . 9
2.4.2 Managing Changes to Your System . . . 10
2.4.3 Avoiding Errors . . . 10
2.4.4 Remove Stress Points . . . 12
2.4.5 Removing Single Points of Unavailability . . . 14
2.4.6 Improving Restart Times . . . 17
2.4.7 Choosing the Most Cost Effective Action . . . 17
2.5 Taking an Action, and Reviewing Its Effect . . . 18
2.6 An Outline Plan for Improving System Availability . . . 18
2.6.1 Identify the Major Causes of Unavailability . . . 19
2.6.2 Reduce Repeated Outages . . . 20
2.6.3 Analyze Restart Times . . . 20
2.6.4 Reduce Restart Times . . . 20
2.6.5 Track Results . . . 21
2.6.6 Repeat at Regular Intervals . . . 21

Chapter 3. Our Recommendations . . . 23
3.1 Recommended System Topology . . . 23
3.1.1 The Terminal . . . 25
3.1.2 The Network . . . 25
3.1.3 MVS Sysplex . . . 26
3.1.4 A CICSplex . . . 27

3.1.5 CICSPlex SM . . . 28
3.1.6 The Data Subsystem . . . 29
3.1.7 The I/O Subsystem . . . 30
3.2 Management Infrastructure . . . 32
3.3 Application Considerations . . . 32
3.4 Recommended Software Components . . . 32
3.4.1 ACF/VTAM . . . 33
3.4.2 MVS/ESA . . . 33
3.4.3 CICS/ESA . . . 33
3.4.4 CICSPlex SM . . . 34
3.4.5 DB2 . . . 35
3.4.6 IMS/ESA . . . 35
3.4.7 VSAM . . . 35
3.4.8 CICSVR . . . 35
3.5 Recommended Hardware Components . . . 36
3.5.1 Processors . . . 36
3.5.2 DASD and Controllers . . . 36

Chapter 4. More on System Topology . . . 37
4.1 The CICSplex . . . 37
4.2 Dynamic Transaction Routing . . . 38
4.2.1 High Availability . . . 39
4.2.2 Continuous Operations . . . 39
4.2.3 CICSPlex SM . . . 40
4.2.4 Roll Your Own . . . 40
4.3 Data Sharing . . . 40
4.3.1 DB2 . . . 41
4.3.2 IMS/ESA . . . 41
4.3.3 VSAM . . . 43

Chapter 5. Operations, Procedures, and Systems Management . . . 45
5.1 Managing Changes . . . 45
5.1.1 Running Different CICS Releases on the Same MVS System Image . . . 45
5.1.2 Adjusting for Daylight Savings Time . . . 46
5.1.3 Introducing New Versions of Application Programs . . . 46
5.2 Avoiding Errors . . . 48
5.2.1 Maintenance Strategy . . . 48
5.2.2 Testing . . . 51
5.2.3 Procedures . . . 55
5.2.4 Automation . . . 57
5.3 Removing Stress Points . . . 60
5.3.1 CICSPlex SM Real Time Analysis . . . 60
5.3.2 Performance Monitoring . . . 60
5.3.3 History Reporting . . . 61
5.4 Removing Single Points of Unavailability . . . 61

Chapter 6. Application Design . . . 63
6.1 Dealing with Errors . . . 63
6.1.1 Exceptional Conditions . . . 63
6.1.2 Soft Errors . . . 64
6.1.3 Abnormal Termination Recovery . . . 65
6.1.4 IMS/ESA DB Processing Options . . . 66
6.1.5 Program Error Program . . . 66
6.1.6 Node Error Program for VTAM Terminals . . . 67
6.1.7 Terminal Error Program for TCAM Terminals . . . 67

6.2 Avoiding Stress Points . . . 67
6.3 Avoiding Single Points of Unavailability . . . 68
6.3.1 Avoiding Transaction Affinities . . . 68
6.3.2 Removing the Requirement for Immediate Access to Another System Component . . . 70
6.3.3 Avoiding Any Requirement for Exclusive Use of Resources . . . 71
6.3.4 Using the External CICS Interface . . . 71
6.3.5 Choosing the Best Data Subsystem . . . 72
6.4 Third Party Software . . . 73

Chapter 7. CICS Functions That Support Continuous Availability . . . 75
7.1 Changing Your System . . . 75
7.1.1 Defining Resources to CICS . . . 75
7.1.2 Changing the CICS System Configuration . . . 78
7.1.3 Changes for IMS/ESA . . . 78
7.2 Avoiding Errors . . . 78
7.2.1 Use the Latest Version of CICS . . . 78
7.2.2 Minimize Storage Addressing Errors . . . 79
7.2.3 Errors in Definition . . . 79
7.3 Avoiding Overstressed Systems . . . 79
7.3.1 Storage Management . . . 79
7.3.2 Intersystem Session Queue Management . . . 80
7.3.3 System Managed Storage . . . 82
7.3.4 Limiting System Usage . . . 82
7.4 Avoiding Single Points of Unavailability . . . 82
7.4.1 VTAM Generic Resource Registration . . . 83
7.4.2 Failure of a FOR in a CICSplex . . . 83
7.4.3 CICS/ESA Data Sets . . . 83
7.5 Improving Restart Times . . . 86
7.5.1 Support for the MVS/ESA Automatic Restart Manager . . . 86
7.5.2 Persistent Session Support . . . 86
7.5.3 Using Autoinstall to Improve Restart Times . . . 87
7.5.4 Extended Recovery Facility . . . 87

Chapter 8. DB2 Considerations . . . 89
8.1 Changing Your System . . . 89
8.2 Errors . . . 90
8.3 Avoiding Overstressed Systems . . . 90
8.3.1 Monitoring DB2 Databases . . . 90
8.4 Single Point of Unavailability . . . 91
8.4.1 Data Sharing . . . 92
8.4.2 Copying Databases . . . 92
8.4.3 Reorganizing Databases . . . 93
8.4.4 Database Design to Avoid Reorganization . . . 93
8.5 Improving Restart Times . . . 98

Chapter 9. IMS/ESA DB Considerations . . . 99
9.1 Changing Your System . . . 99
9.1.1 Online Change Utility . . . 99
9.2 Errors . . . 99
9.3 Avoiding Overstressed Systems . . . 99
9.3.1 Use the DBCTL Interface . . . 100
9.3.2 Monitoring Databases . . . 100
9.4 Avoiding Single Points of Unavailability . . . 100
9.4.1 Data Sharing . . . 100

9.4.2 Copying Databases . . . 103
9.4.3 Reorganizing Databases . . . 103
9.4.4 Database Design to Avoid Reorganization . . . 103
9.5 Improving Restart Times . . . 105
9.5.1 Types of Recovery . . . 106
9.5.2 Recovery Strategy . . . 107

Chapter 10. VSAM Considerations . . . 109
10.1 Errors . . . 109
10.1.1 Backout Failure Control and Batch Backout . . . 109
10.1.2 Automation of the Recovery Process . . . 109
10.1.3 Monitoring . . . 110
10.2 Single Point of Unavailability . . . 111
10.2.1 Data Sharing . . . 111
10.2.2 Copying Data Sets . . . 112
10.2.3 Reorganizing Data Sets . . . 112
10.2.4 Data Set Design to Avoid Reorganization . . . 112
10.3 Improving Restart Times . . . 113
10.3.1 Forward Recovery of Data Sets . . . 114

Chapter 11. Batch Processing Considerations . . . 115
11.1 BatchPipes . . . 115
11.1.1 How BatchPipes Changes Traditional Batch Processing . . . 116
11.2 Alternatives to BatchPipes . . . 117

Chapter 12. Hardware That Supports Continuous Availability . . . 119
12.1 Parallel Sysplex . . . 119
12.1.1 Processors . . . 120
12.1.2 Sysplex Timer . . . 120
12.1.3 ESCON Director and Control Units . . . 121
12.1.4 Coupling Facility . . . 121
12.1.5 Coupling Facility Channels . . . 122
12.2 3990 Storage Control Functions . . . 122
12.2.1 Concurrent Copy . . . 122
12.2.2 Dual Copy . . . 123
12.2.3 Remote Copy . . . 124
12.3 RAMAC DASD . . . 124
12.4 Environmentals . . . 125
12.4.1 Electricity Supply Failure . . . 125
12.4.2 Cooling Failure . . . 125

Chapter 13. Disaster Recovery . . . 127
13.1 Why a Disaster Recovery Plan? . . . 127
13.2 Disaster Recovery Testing . . . 129
13.3 Six Tiers of Solutions . . . 130
13.3.1 Tier 0 - No Offsite Data . . . 130
13.3.2 Tier 1 - Physical Removal . . . 131
13.3.3 Tier 2 - Physical Removal with Hot Site . . . 133
13.3.4 Tier 3 - Electronic Vaulting . . . 134
13.3.5 Tier 0 - 3 Solutions . . . 135
13.3.6 Tier 4 - Active Secondary Site . . . 136
13.3.7 Tier 5 - Two Site, Two Phase Commit . . . 138
13.3.8 Tier 6 - Minimal to Zero Data Loss . . . 139
13.3.9 Tier 4 - 6 Solutions . . . 140
13.4 Disaster Recovery and High Availability . . . 142

13.4.1 Peer-to-Peer Remote Copy and Extended Remote Copy 142
13.4.2 Remote Site Recovery 144
13.4.3 Remote Recovery Data Facility 146
13.4.4 Choosing among RSR, RRDF, and 3990-6 Solutions 147
13.5 More Thoughts on Disaster Recovery 147
13.5.1 Disaster Recovery Personnel Considerations 147
13.5.2 Returning to Your Primary Site 148
13.5.3 Disaster Recovery Further Information 148

Chapter 14. Unavailability Cause and Solution Checklist 149

Appendix A. Service Level Management 151
A.1 An Overall Service Model 152
A.2 Service Level Agreements 153

Appendix B. Hardware Features 155
B.1 Parallel Transaction Servers 155
B.2 RAMAC DASD 157

Appendix C. Products and Tools 161
C.1 IBM CICSPlex System Manager for MVS/ESA 161
C.2 IBM CICS VSAM Recovery MVS/ESA 162
C.3 NetView for MVS/ESA 163
C.4 Enterprise Performance Data Manager/MVS 163
C.5 IBM Service Level Reporter for MVS 164
C.6 IBM Automated Operations Control/MVS 165
C.7 Information/Management for MVS/ESA 166
C.8 Teleprocessing Network Simulator 166
C.9 IBM CICS Transaction Affinities Utility MVS/ESA 167

List of Abbreviations 171

Index 173

Figures

1. A Simple Service Level Agreement 7
2. Splitting a CICS Region to Reduce Stress 13
3. Single Component: System Availability 14
4. Independent Duplicated Component: System Availability 15
5. Linked Duplicated Component: System Availability 16
6. Recommended System Configuration 24
7. I/O Hardware Configuration for Continuous Availability 31
8. Example of a CICSplex 38
9. Preparing and Replacing an Application Program 47
10. The EXEC CICS HANDLE CONDITION Command 63
11. The EXEC CICS HANDLE CONDITION Command 64
12. Database Sharing (Block level database sharing at the interhost level.) 102
13. Timeline Showing a Traditional Two-Job Stream 116
14. Timeline Showing Two Jobs Using BatchPipes 116
15. Disaster Recovery: Cost, Completeness, Speed, and Risk 128
16. Disaster Recovery Tier 0: No Offsite Backup 130
17. Disaster Recovery Tier 1: Physical Removal 131
18. Disaster Recovery Tier 2: Physical Removal to a Hot Site 133
19. Disaster Recovery Tier 3: Electronic Vaulting 134
20. Disaster Recovery Tier 0-3: Summary of Solutions 135
21. Disaster Recovery Tier 4: Active Secondary Site 136
22. Disaster Recovery Tier 5: Two Site, Two Phase Commit 138
23. Disaster Recovery Tier 6: Minimal to Zero Data Loss 139
24. Disaster Recovery Tier 4-6: Summary of Solutions 140
25. Information Systems Management Discipline Inter-relationships 151
26. Overall Service Model 152

Tables

1. Choosing the Most Effective Action 18
2. CICS/ESA N-Way Data Sharing Prerequisites 41
3. Selecting a Tier 6 Implementation 147
4. Problems and Possible Solutions 149

Special Notices

This publication is intended to help you plan for a continuous availability environment for CICS. It discusses the causes of unavailability, and suggests solutions for removing them, or for minimizing their effect. The information in this publication is not intended as the specification of any programming interfaces that are provided by MVS, ACF/VTAM, CICS, IMS, DB2, or any of the other programs mentioned in this book. See the PUBLICATIONS section of the IBM Programming Announcement for these products for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM
Advanced Peer-to-Peer Networking
BatchPipes
CBIPO
CBPDO
CICS
CICS/ESA
CICS/MVS
CICSPlex
CICSPlex SM
DATABASE 2
DB2
DFSMS
DFSMS/MVS
Enterprise Systems Architecture/390
ES/9000
ESA/370
ESA/390
ESCON
IBM

IMS	IMS/ESA
MQSeries	MVS/ESA
MVS/SP	MVS/XA
NetView	PR/SM
RACF	RAMAC
RMF	S/390
Sysplex Timer	System/390
SystemView	VisualGen
VTAM	

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Windows is a trademark of Microsoft Corporation.

E-NET	E-NET Corporation

Other trademarks are trademarks of their respective companies.

Preface

This book is intended to help you build and maintain a continuously available system based on IBM′s Customer Information Control System (CICS) under the control of the Multiple Virtual Storage/Enterprise Systems Architecture (MVS/ESA) operating system.

How This Document Is Organized

The document is organized as follows:

- Chapter 1, “Introduction”

  In this chapter we discuss the current and future environments that CICS is expected to support, and how the environmental changes are placing more and more demands on information systems departments to provide continuously available systems.

- Chapter 2, “The Business Perspective”

  In this chapter we describe some of the choices facing your business when you implement a continuously available system. We discuss some general solutions to building a continuously available CICS system, the costs of implementing the solutions, and the costs of not implementing them.

- Chapter 3, “Our Recommendations”

  In this chapter we describe the best environment you can build for continuous availability of CICS in an MVS/ESA environment using technology available today.

- Chapter 4, “More on System Topology”

  In this chapter we describe CICS and database system design features that can eliminate or minimize the occurrence of events leading to scheduled outages of a CICS/ESA system.

- Chapter 5, “Operations, Procedures, and Systems Management”

  In this chapter we describe how you can manage your systems to achieve continuous availability.

- Chapter 6, “Application Design”

  In this chapter we discuss application development issues which affect the availability of your system.

- Chapter 7, “CICS Functions That Support Continuous Availability”

  In this chapter we discuss the features of CICS that enable continuous availability, and the remaining causes of unavailability in a single CICS region.

- Chapter 8, “DB2 Considerations”

  In this chapter we discuss DB2 considerations relating to achieving continuous availability.

- Chapter 9, “IMS/ESA DB Considerations”

  In this chapter we discuss IMS/ESA considerations relating to achieving continuous availability.

- Chapter 10, “VSAM Considerations”

  In this chapter we discuss VSAM considerations relating to achieving continuous availability.

- Chapter 11, “Batch Processing Considerations”

  In this chapter we discuss techniques to minimize the impact of scheduled outages by reducing the batch window. We explain BatchPipes and some alternative techniques.

- Chapter 12, “Hardware That Supports Continuous Availability”

  This chapter describes the hardware features that support continuous availability.

- Chapter 13, “Disaster Recovery”

  In this chapter we describe what you need to consider when planning for disaster recovery in a CICS environment. If the normal availability of your CICS system is being extended into the 99 percent range, you should look at your disaster recovery plan. The same pressure that drives continuous availability drives timely and current disaster recovery.

- Chapter 14, “Unavailability Cause and Solution Checklist”

  This chapter provides a checklist of possible causes of unavailability, and where in this book and in other publications you can find information about solutions to these problems.

Related Publications

The publications listed in this section are particularly suitable for a more detailed discussion of the topics covered in this document.

Books for CICS/ESA Version 3

- *CICS/ESA Performance Guide*, SC33-0659-02
- *CICS/ESA CICS-IMS Database Control Guide*, SC33-0660
- *CICS/ESA XRF Guide*, SC33-0661
- *CICS/ESA Resource Definition (Online)*, SC33-0666-02
- *CICS/ESA Resource Definition (Macro)*, SC33-0667-02
- *CICS/ESA Application Programming Guide*, SC33-0675-02
- *CICS/ESA Application Programming Reference*, SC33-0676-02
- *CICS/ESA Recovery and Restart Guide*, SC33-0658
- *CICS/ESA Release Guide*, GC33-0792

Books for CICS/ESA Version 4

- *CICS/ESA Resource Definition Guide*, SC33-1166
- *CICS/ESA Application Programming Guide*, SC33-1169
- *CICS/ESA Application Programming Reference*, SC33-1170
- *CICS/ESA Performance Guide*, SC33-1183
- *CICS/ESA CICS-IMS Database Control Guide*, SC33-1184
- *CICS/ESA External CICS Interface*, SC33-1390

- *CICS/ESA Recovery and Restart Guide*, SC33-1182
- *CICS/ESA Release Guide*, GC33-1161

Books for Both CICS/ESA Version 3 and CICS/ESA Version 4

- *CICS/ESA Dynamic Transaction Routing in a CICSplex*, SC33-1012
- *CICS Transaction Affinities Utility MVS/ESA Users Guide*, SC33-1159

Other Books

- *IBM BatchPipes/MVS Introduction*, GC28-1214
- *CICSPlex SM Concepts and Planning*, GC33-0786
- *TPNS General Information*, GH20-2487
- *IBM DATABASE 2 Version 3 DB2 Administration Guide*, SC26-4888
- *IBM DATABASE 2 Version 3 DB2 Command and Utility Reference*, SC26-4891
- *IBM DATABASE 2 for MVS/ESA Version 4 Administration Guide*, SC26-3265
- *IBM DATABASE 2 for MVS/ESA Version 4 Command Reference*, SC26-3267
- *IMS/ESA Utilities Reference System*, SC26-8035
- *SystemView: AOC/MVS CICS Automation General Information*, GC23-3813
- *IBM 3990 Storage Control Introduction*, GA32-0098
- *Systems Analysis for High Availability, An Availability Management Technique*, GG22-9391
- *Workstation Interactive Test Tool Users Guide*, SC26-3054
- *SYSTEM/390 MVS Sysplex Overview: An Introduction to Data Sharing and Parallelism*, GC28-1208
- *SYSTEM/390 MVS Sysplex Systems Management*, GC28-1209
- *SYSTEM/390 MVS Sysplex Hardware and Software Migration*, GC28-1210
- *SYSTEM/390 MVS Sysplex Application Migration*, GC28-1211

International Technical Support Organization Publications

- *Accessing CICS Business Applications from the World Wide Web*, SG24-4547
- *CICS Workload Management Using CICSPlex SM and the MVS/ESA Workload Manager*, GG24-4286
- *Automating CICS/ESA Operations With CICSPlex SM and NetView*, GG24-4424
- *Disaster Recovery Library: Planning Guide*, GG24-4210
- *CICS Transaction Design: Pseudo-Conversational or Conversational*, GG24-3805
- *IBM RAMAC Array Family*, GG24-2509
- *Planning for IBM Remote Copy*, SG24-2595 (Available first quarter 1996)
- *A Comparison of System/390 Configurations - Parallel and Traditional*, SG24-4514

A complete list of International Technical Support Organization publications, known as redbooks, with a brief description of each, may be found in:

International Technical Support Organization Bibliography of Redbooks, GG24-3070.

To get a catalog of ITSO redbooks, VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

A listing of all redbooks, sorted by category, may also be found on MKTTOOLS as ITSOPUB LISTALLX. This package is updated monthly.

How to Order ITSO Redbooks

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-445-9269. Almost all major credit cards are accepted. Outside the USA, customers should contact their local IBM office. Guidance may be obtained by sending a PROFS note to BOOKSHOP at DKIBMVM1 or E-mail to bookshop@dk.ibm.com.

Customers may order hardcopy ITSO books individually or in customized sets, called GBOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain redbooks on a variety of products.

ITSO Redbooks on the World Wide Web (WWW)

Internet users may find information about redbooks on the ITSO World Wide Web home page. To access the ITSO Web pages, point your Web browser to the following:

```
http://www.redbooks.ibm.com/redbooks
```

IBM employees may access LIST3820s of redbooks as well. Point your web browser to the IBM Redbooks home page:

```
http://w3.itsc.pok.ibm.com/redbooks/redbooks.html
```

Acknowledgments

This project was designed and managed by:

Hugh Smith
International Technical Support Organization, San Jose Center

The authors of this document are:

Tom Hall
IBM Canada

Peter Henningsen
IBM Australia

John Joro
IBM Finland

Neil Leedham
IBM United Kingdom Laboratories

Hugh Smith
International Technical Support Organization, San Jose Center

This book replaces *Planning for Continuous Availability in a CICS and MVS/XA Environment*, GG24-1591. The authors of the original document were:

Peder Aberg, IBM Sweden
Michael Bredenkamp, IBM United Kingdom Laboratories

The project leader for the original document was:

Osvaldo Santoro, IBM Argentina

This publication is the result of a residency conducted at the International Technical Support Organization, San Jose Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Dave McAuley
International Technical Support Organization, San Jose Center

John Wade
IBM United Kingdom

Jim Coolbaugh
IBM USA

Chapter 1. Introduction

Since 1968, when CICS was first made available as a product, the world has undergone revolutionary changes. The public now takes as normal, even commonplace, things that seemed unreachable in 1968. The public's expectations have changed, and meeting these expectations becomes more challenging every year.

This book addresses one area where expectations have changed dramatically; computer based systems are available upon demand, no matter what time of the day, or day of the week. This book helps meet that expectation if you are running business applications based upon IBM's Customer Information Control System (CICS) under the control of the Multiple Virtual Storage/Enterprise Systems Architecture (MVS/ESA) operating system.

1.1 The Changing Environment

Let us briefly look at the current and future environments that CICS is expected to support, and how environmental changes are placing more and more demands on information system departments.

In the late 1960s, a large terminal network consisted of 50 terminals. These terminals were typically used for cost-saving, back-office functions such as general ledger and payrolls. The terminal operators were paid employees of the provider of the service (typically data entry clerks), who were trained in the processes that they were executing. Interacting with the computer formed a significant part of their daily activities, and the process of interaction was highly formalized. They also tended to work a "normal," 9-to-5 workday, providing plenty of time for batch and other work to run without interfering with the online operation. Typically, they represented about 10 percent of the total workload of the data processing center.

By the late 1970s, the picture had changed. Although many users were still installing cost-saving systems, they were also installing front-office applications, such as order entry and customer service applications. Support for automatic teller machines is an example of the type of new function that was appearing. The terminal networks were significantly bigger, 500 and more terminals with correspondingly higher transaction volumes, sometimes even as high as 15 transactions in a second. The terminal users did not receive as much formal training in using the computer systems as their predecessors, often because using the terminal was only a small part of their job and other functions. For example, customer interaction formed the major part of their workload. The hours these people worked also extended typical service level requirements from 6 a.m. until 8 p.m.

By the late 1980s expectations had changed again. Terminal networks of over 10,000 terminals were not uncommon. Enterprises interconnected their networks, for example, to allow customers to place orders directly with their suppliers. The services offered on these networks moved from being cost-saving to being revenue-earning, direct-customer services, for example, Videotext and home banking. If a service like this is not available, the organization loses revenue, and the users of the service become dissatisfied with the company providing the service. In today's highly competitive business environment this

may be an unacceptable business risk, since dissatisfied customers can easily take their business elsewhere.

The networks that grew in the late 1980′s and early 1990′s have different characteristics from the older ones. They have high transaction rates (often in excess of 500 transactions a second), with many installations today already having workloads that exceed 1000 transactions a second. Even higher transaction rates are possible. Recent tests made in a controlled laboratory environment have shown that a CICS/ESA system can support as many as 10,000 transactions per second. Networks can be huge, with thousands of connected users. Some terminal networks are already growing towards 500,000 terminals, and the recent explosion in use of the World Wide Web (WWW) has created possibilities for even larger networks of terminals. For information on how CICS participates as a server in the WWW, see *Accessing CICS Business Applications from the World Wide Web*.

Today′s users are computer sophisticates, in that they know what they expect from the computer system, even if they have had very little or no training in computer principles. In addition, they may be highly trained specialists in their own fields. These users have little or no understanding of the traditional requirements of data processing services, and little patience if such requirements result in the unavailability of the services they need at the time they need it. Not only do these people require (or demand) access to the systems at all times, but the networks now span the earth.

1.2 Why Continuous Availability?

Some examples of systems demanding very high levels of availability and continuous operation are:

- Medical services systems assisting in patient monitoring and diagnosis
- On-line financial services networks
- Continuous-operation manufacturing installations
- Information services
- International business services spanning multiple time zones

The CICS family of products has been developed since 1968 to evolve with these environmental changes. It can support users in the increasing sophistication that they expect from their on-line data communications system, providing them with protection of their investment in application code. At the same time it can exploit new technology and products.

This book contains the information that you need to move towards continuous availability with CICS/ESA.

1.3 Continuous Availability, Continuous Operation, and High Availability

In this manual, we use the terms continuous operation, high availability and continuous availability. To ensure a common base of understanding, there is a brief definition of these terms below.

Continuous operation Theoretically, operation of a system for 24 hours per day, 365 days per year.

In a continuous operation system we make **no provision for scheduled unavailability**. All changes and

maintenance can be applied in a non-disruptive manner.

High availability — A system that delivers an agreed upon level of service during scheduled periods of availability. To achieve a high availability system, we have to **eliminate any unscheduled unavailability** that may occur.

Practically, high availability systems strive for at least 99.9 percent availability during scheduled time.

Continuous availability — Combines the characteristics of both continuous operation and high availability. **No unavailability of any nature** is acceptable.

In practice it is extremely difficult for most installations to achieve continuous availability, since planned unavailability is scheduled in nearly all installations and unscheduled unavailability is, by their nature, often unexpected and difficult to eliminate. What we usually have to do is plan to minimize the frequency and length of both planned and unplanned unavailability.

The frequency of scheduled unavailability varies between once a day, to once every few weeks. The length of the scheduled unavailability is seldom more than two hours.

In all cases, **availability** reflects the perception of the end user of the service. This means that end users will perceive that their system is unavailable if:

- The response time exceeds the maximum normal response time they are accustomed to and interrupts their workflow.
- The response from the system is a request for an activity outside the normal work activities, such as a request to perform a sign-on operation again.

Unavailability, therefore, is defined as any interruption that presents end users with an interruption to their normal pattern of workflow.

1.4 Do You Need CICS Continuous Availability?

This may sound like a strange question, but it is one that you should answer before you go further. Typically the costs of increasing the availability of your system increase rapidly as you approach 100 percent availability. You should easily be able to achieve availability that exceeds 95 percent, without going through the full formality of procedures and techniques which can improve availability even further. Remember that 95 percent availability is the same as five percent unavailability, or approximately eight and a half hours of unavailability in a week.

You can improve availability substantially by careful planning and effective system management. Together these can take you to and beyond the 99 percent level (still one and a half hours unavailability in a week).

To achieve even higher availability (continuous availability), you will have to use special technical implementations, such as duplication of system components or other redundancy techniques, but these must be based on a good initial implementation. Any single point of failure opens the door to unscheduled unavailability. To achieve continuous availability, you need to eliminate these single points of failure. This level of availability also requires a higher level of

management support to provide the necessary personnel and financial resources. The decision to attempt to attain these high figures is a business decision to be considered at the strategic level. We discuss this further in Chapter 2, “The Business Perspective” on page 5.

Chapter 2. The Business Perspective

In this chapter we describe some of the choices facing your business when you implement a continuous availability system.

Before you start planning for continuous availability systems, you must understand that there is a cost involved in implementing them. The cost can be in time, to implement and test new procedures, or to review new application development standards and test them. The cost can be more visible, for example, in new releases of software, new hardware, or in duplication of existing system components, even duplication of the entire system.

The main steps in planning for continuous availability are:

1. Assessing the cost of unavailability
2. Agreeing on your target level of continuous availability
3. Understanding where you are
4. Estimating the cost of getting where you want to be
5. Taking an action, and reviewing its effect.

2.1 Assessing the Cost of Unavailability

The first question you must answer is: "What does a period of unavailability cost my business?" Be realistic when you answer this question, for you need to make a sound business judgement based on this assessment. Examples of the typical costs that you need to include are:

- Direct additional costs

  This may include overtime payments. If the system was unavailable long enough so that you need to pay users to stay late and complete their normal day's work in overtime. If you are using your system to control a manufacturing process, perhaps a batch produced during the period of unavailability has to be rejected. This is a direct cost.

  One area that many installations overlook is the direct cost of unavailability in their development and testing systems. Such unavailability can delay projects and leave staff idle and unproductive.

- Lost business

  If your business receives orders and enters them directly into the system, you cannot carry out this part of your business if the system is not available. You need to assess whether these orders are lost forever, or whether your customers will wait and place an order later when the system is available. Whether or not the order is lost may depend on the length of the period of unavailability. If the orders are lost, you need to assess how many orders you receive a minute and their average value.

- Lost opportunity, and other intangible costs

  You may want to place an intangible value on having a continuous availability system. For example, your customers may not even notice if your system is unavailable for five minutes a month, but they will almost certainly notice if your system is unavailable for 20 minutes every Monday afternoon. You have to assess how much this is worth to you. Is it possible that new customers will choose to do business with a competitor because

they have heard people say “Don’t do business with Company X. They can never help you. They always claim that their system is down.”

So far we have briefly discussed some of the costs you need to assess. You may need to refine your assessment, or repeat it several times for different areas of your business. Some of the additional questions that you need to ask yourself are:

- Do all the applications I run have the same value?

  What happens if a specific application is unavailable? Different applications have different characteristics. For example, if your order entry system is unavailable for an hour, the unavailability can have a high cost for your business. If your personnel system is unavailable for an hour, the unavailability may be inconvenient, but is unlikely to have a significant cost to your business. Remember to include your testing and development systems in your assessment. Your programmers are a valuable resource and should not be left idle.

  Do all your applications have equal value? Which are most important? Try ranking your applications in order of importance to your business.

- How is my business structured?

  Do you have branch offices? Is everything done centrally? Do you have a network of departmental machines?

  The answer to these questions can significantly alter your view of the impact of system unavailability. For example, if all your employees in a business area use the same system, the impact of unavailability of that system is widespread and could be extremely costly. If your business is highly departmentalized, the unavailability of one system may affect one department, but leave others unaffected. If one department is able to cover for another, the impact of unavailability may be minimal. For example, if you have Eastern, Central, and Pacific sales departments, customer inquiries could potentially be directed to operators in a different department if the system supporting the local operators is unavailable.

- Which systems are absolutely critical to my business?

  What if your systems are unavailable for a day or more? Can you run your business? How long can you afford to be without facilities?

  This leads us into another aspect of continuous availability. How do I provide essential services with the minimum of lost data and the minimum interruption of service if there is a disaster (earthquake, fire, or similar event)? Disasters like these are uncommon, but did you know that an estimated 80% of companies that have a fire in their machine room go out of business as a result?

We have tried to show in this section that the first step in planning for continuous availability is to understand your business and how your CICS-based applications support it. Once you have done this, you can confidently plan for continuous availability knowing which systems need it, how important it is that these systems achieve it, the costs of unavailability, and how much you should spend on achieving it.

2.2 Agreeing on Your Target Level of Continuous Availability

You may already have a formal agreement with the users of your system about what level of availability they need. If not, now is the time to negotiate one. When you have assessed the cost to your business of unavailability, you should translate this information into a Service Level Agreement (SLA), which defines the level of availability required to support your entire business.

An SLA typically describes:

The hours during which a service is required
The level of availability required during these hours
The responsiveness of the system during these hours.

Figure 1 shows an example of an SLA, showing these elements.

System:	**Branch Office Customer Inquiry System**
Hours:	8am EST to 6pm PST
Availability:	No more than one period of unavailability per month. Service must be restored within 20 minutes. Measurement applies to availability as measured for system XYZ at the central site.
Responsiveness:	90% of customer data transactions (AAAA, BBBB) complete in less than two seconds. Maximum response time is 15 seconds.

Figure 1. A Simple Service Level Agreement

The SLA is a contract between the service users and the service providers. It is a method of determining what the users of the system really expect in terms of availability, and what the providers can provide for that cost. There is no point in spending money to achieve 99.99 percent availability, when the users of the system can easily tolerate a one hour outage once a week.

Creating your SLA has to be an iterative process. The service users′ initial expectations may be too high, or unrealistically expensive to achieve. The service providers may then have to offer a lower level of service. If the lower level is not acceptable to the users, the service providers will need to estimate the cost of providing the level of availability required, and the users and providers between them will have to find the money to improve the availability of the system.

See Appendix A, “Service Level Management” on page 151.

2.3 Understanding Where You Are

You need to measure what your level of service is, both to know if the actions you take are effective and to assess the most cost effective place for further action. For a CICS-based system the main things we need to be concerned with are the unavailability of the system as a whole (for example, the unavailability of all the regions that can support a particular application), the network, or an individual terminal. Measuring the availability of the system requires a balancing act. End users are concerned with their availability and response time, but it is difficult to track and manage this centrally (imagine trying to assess the availability of the system as seen by 10,000 different users!).

Conversely, it is comparatively easy to measure the availability of a CICS region and get an overall view of the availability you are providing to the majority of end users. But the end users who have not been able to access their application for a week due to a LAN problem will not be impressed by a report that shows the system was 100% available.

2.3.1 System Availability

You must measure the availability of your CICS regions. The availability or unavailability of your CICS systems has a wide ranging effect, potentially covering all users. One way you can do this is to use the SAM or RTA functions of CICSPlex SM. You can set your own triggers, which will cause CICSPlex SM to generate alerts. You can use other software, such as NetView, to pick up these alerts and record them. CICSPlex SM is particularly suited to this task, since you can define the hours during which the system should be available.

Another way you could measure availability is to use normal SMF job step termination records. You should consider whether this is accurate enough for your needs, since the SMF records:

- Are not reliable if MVS stalls, or the processor fails
- Do not indicate when the system is stalled
- Include system initialization and termination time

You could set up a system for sampling records which reflects actual transaction activity. Possible sources include CICS automatic statistics and CMF records (which can be written to SMF as type 110 records). If you use a sampling technique you have to decide what constitutes unavailability. For example, if you collect automatic statistics every 10 minutes, a period exceeding 10 minutes without a statistics report suggests that the CICS system is no longer available. If you collect CMF records, sort them by end time, and count frequency by time period, say a minute or two. If the rate is significantly lower than expected (or zero) for a limiting number of periods, then this may provide a useable indication that the system has gone down, or at least is not processing transactions. If you further refine your analysis and analyze the periodic transaction rates by groups of transaction names, you can use this approach to measure availability of applications as well as CICS.

2.3.2 Network Availability

Session statistics from NetView′s Session Monitor (NLDM) record when PUs and LUs are active and when they are not. You could extract that information from SMF records, but you need a tool to analyze them. You could write your own program, or use a package like Service Level Reporter (SLR) or Enterprise Performance Data Manager/MVS (EPDM) to read and analyze the data.

2.3.3 Terminal Availability

We do not recommend that you try to measure availability on a terminal basis. The overheads and administration involved are typically prohibitive. Deal with local unavailability of terminals on an exception basis.

2.4 Estimating the Cost of Getting Where You Want to Be

Now that you understand what you want to achieve, how important it is to achieve it, and what your current problems are, you can assess the risk and the cost of preventing the problem.

In this section we describe some of the basic causes of unavailability, and discuss general approaches to eliminating them or reducing their effect. This should give you a starting point in comparing the cost and effectiveness of different solutions.

2.4.1 Causes of Unavailability

In this section we discuss changing your system, errors, overstressed systems, and single points of availability. Most causes of unavailability can be placed in one of these categories. Remember that we are interested in both unplanned and planned unavailability.

2.4.1.1 Changing Your System

Changes occur as new applications are written, existing applications are rewritten, new software is installed to introduce new features to the system, and new hardware is installed to provide increased capacity, or to reduce costs. Change is inevitable, but change can cause your system to be unavailable. Change can require a period of planned unavailability, for example, so that you can change the definition of part of the system. It can also cause a period of unplanned unavailability if the change has an unexpected effect, or introduces an error to the system.

2.4.1.2 Errors

Errors are usually associated with unplanned unavailability, but a non-catastrophic error may require a planned period of unavailability. Nearly all errors are caused by humans, and are typically:

- Programming errors
- Operational errors (making an incorrect decision)
- Procedural errors (failing to follow a procedure correctly)

2.4.1.3 Overstressed Systems

Overstressed systems are more likely to fail than unstressed ones. Consider one example of stress, a CICS region which is short-on-storage (SOS). One of the first actions that CICS takes is to stop accepting new tasks in the SOS region (which means that incoming tasks have to either queue, or be routed to other regions). Normally one or more tasks will complete processing and release storage, the SOS condition is resolved, and activities return to normal. It is, however, quite possible that all of the tasks in the system require ***more*** storage to complete their processing. Because the region is SOS, we cannot give them this storage, and the region stalls. Tasks that would have run without a problem cannot complete because the region is overstressed, and eventually this can cause the region, or application, to become unavailable.

2.4.1.4 Single Point of Unavailability

A single point of unavailability is often called a single point of failure, but this focuses solely on unplanned unavailability. We need to consider the planned unavailability case too. Unavailability can be caused by the failure of a critical component, or by the requirement for two processes to have exclusive access to a resource at the same time (for example database reorganization). Typically we can avoid this cause of unavailability by duplicating system components, or by removing the requirement for exclusive access, or at least shortening the time for which exclusive control is required.

2.4.1.5 Extended Restart

The unavailability of your system is worsened if it takes a long time to restart it. While this is not a cause of unavailability, it is something that you need to take into account when planning how to improve the availability of your system and reach continuous availability.

2.4.2 Managing Changes to Your System

Change is inevitable. There are two aspects to changes; testing in advance to ensure that the changes you make have the effect you expect, and managing the process of implementing the change. We discuss testing in section 2.4.3.2, “Testing” on page 11.

Managing the change requires good change management techniques. We discuss this in more detail in Chapter 5, “Operations, Procedures, and Systems Management” on page 45. The most important considerations for continuous availability are:

- Using facilities like CICS resource definition online (RDO) to make dynamic changes to the system. We discuss this in more detail in section 7.1, “Changing Your System” on page 75.
- Designing your system so that you can remove components to make changes without stopping the service completely. We discuss this briefly in section 2.4.5, “Removing Single Points of Unavailability” on page 14 and in more detail in Chapter 4, “More on System Topology” on page 37.
- Timing the change to have minimum impact on users of the system.
- Making several changes at once to reduce the number of periods of planned unavailability. Note that this conflicts with one of the basic ideas of change management, which is that you should only make one change at a time.
- Always have a backout or fallback plan.

2.4.3 Avoiding Errors

We have briefly reviewed the major causes of error. These are programming errors (either in your own applications, or in applications or system code that you have bought from others), operational errors, and procedural errors. You can reduce these causes of unavailability by developing a maintenance strategy, thorough testing, having correct procedures in place, and increased automation of procedures and operations.

2.4.3.1 Developing a Maintenance Strategy

Maintenance is the process of removing known errors before they affect your system. You have to ask:

Is this error likely to affect my system?
Will maintaining the system cause a period of unavailability?
What is the risk of delaying the maintenance?
How much testing is required?

We discuss the question of how and when to apply maintenance to your CICS system in section 5.2.1, “Maintenance Strategy” on page 48. You can use the ideas here to develop your own maintenance strategy for other system components, such as your own applications.

2.4.3.2 Testing

We cannot over-emphasize the importance of extensive testing. Your system will change and evolve over time, but with good testing procedures in place the changes need not be disruptive. The amount of testing you do depends, among other things, on the importance you place on availability. There will always be a trade-off between the time and expense required for extensive testing, and the benefits in increased reliability and availability of your systems. For example:

- You can use application generators, such as IBM′s VisualGen, and other tools to develop your applications. The resulting application systems are typically more reliable and need less testing than traditionally coded programs because the blocks of function used have been pretested. However, you can find that application execution costs will be higher.
- You can use testing and debugging tools to thoroughly test how your application works. The main difficulty you face with this is imagining every combination of circumstances that your application will encounter, and testing that it works correctly in all of them.
- You can use tools like Teleprocessing Network Simulator (TPNS) to stress test complex configurations under realistic loadings. This is more important in very large networks that offer services to the public, such as ATMs, but requires significant investment in the test configuration and specialist staff.

You must test both your applications and the system as a whole. It could be that the application is well designed and error free, but that it is so heavily used that your system fails under the stress of running the new application. Testing of your applications removes errors during development, reducing the chance that an error will cause a failure, or require a planned period of unavailability to correct an error. Testing of the system, particularly stress testing, will show whether your system has enough capacity to run your applications in the future, and if not where the resource constraints, or stress points, will occur.

We discuss testing in more detail in section 5.2.2, “Testing” on page 51.

2.4.3.3 Procedures

Your procedures (for example, how to stop a CICS region) must be well documented. They should be the definitive source of operational knowledge in your organization. They should detail the *correct* way to perform a process, and be based on the collective knowledge and experience in your organization. A good set of procedures can improve availability by reducing human errors and giving all your staff access to the knowledge they need to do their jobs well.

If you do not have well documented procedures for a task, then you are likely to have errors as your staff try to perform tasks in a ″seat of the pants″ fashion.

2.4.3.4 Automation

Automating operations and procedures repetitively uses tested techniques to ensure that the same (and hopefully correct) action is taken when required, reducing the possibility of unavailability caused by operator error.

Your operators issue commands and receive messages to communicate with and control operating systems, subsystems, and the network. Each of these systems, subsystems, and networks has its own command language syntax and message formats. Your operators have to keep in mind many different commands and messages, and sometimes they have to enter a complex series of commands and replies to messages to resolve a problem or carry out a task. The more complex the sequence of commands and replies, the greater the possibility of making a mistake.

Automation helps in two ways if your operators have to recover an online application system in a production environment. The first way is to either eliminate the operator completely, or to reduce the complexity of the action they have to take. To recover an online application system, they often have to make an immediate decision on a complex problem, which increases the possibility of making a mistake. The second way in which automation helps is in reducing delays in the detection of problems. Delays cause a reduction in the quality of service that your end users experience. Automation can help you by speeding the detection of the problem, as well as by reducing the complexity of the decisions your operators must make. See *Automating CICS/ESA Operations With CICSPlex SM and NetView* for more information.

2.4.4 Remove Stress Points

One of the most important parts of achieving continuous availability is to plan for the future, to ensure that your system can cope with expected usage, identify reasons why it cannot (for example, there may not be enough capacity in your existing I/O subsystem to deal with expected usage in six months time, or you may experience virtual storage constraint), and plan how you will change the system to remove these constraints, or stress points. For example, you have to ask yourself:

Can the hardware handle the expected increase in activity?
Will there be enough disk space for paging, data base growth, and so on?
What is the impact on virtual storage requirements?
What is the impact on real storage requirements?

You need to address these and many other questions, and make plans for all foreseeable situations. How do you know what questions to ask, or what to look for? The answer is different for every installation, and ideally you need to couple this with stress testing of the system, as discussed in section 2.4.3.2, “Testing” on page 11.

Whether you discover stress points in your system through testing, or usage, there are several ways to deal with the problem. Broadly speaking you can choose to:

- Limit usage of the system. You could, for example, choose to limit the overall number of tasks in the system (using CICS maximum tasks - MXT) or the number of tasks of a particular type (class maximum tasks - CMXT - in

CICS/ESA 3.3 or earlier, transaction classes - TRANCLASS - in CICS/ESA 4.1 or later). This is a good short-term solution, easy to implement, and often has an immediate and dramatic effect on both performance and availability in severely stressed systems.

- Give different priorities to different users of the system. You could, for example, use the MVS workload manager to give a higher priority to certain workloads, ensuring that they can be processed without being constrained.
- Remove the stress point, for example, by buying additional hardware.

One very effective method of reducing stress in CICS systems is to implement multiple regions, as shown in Figure 2.

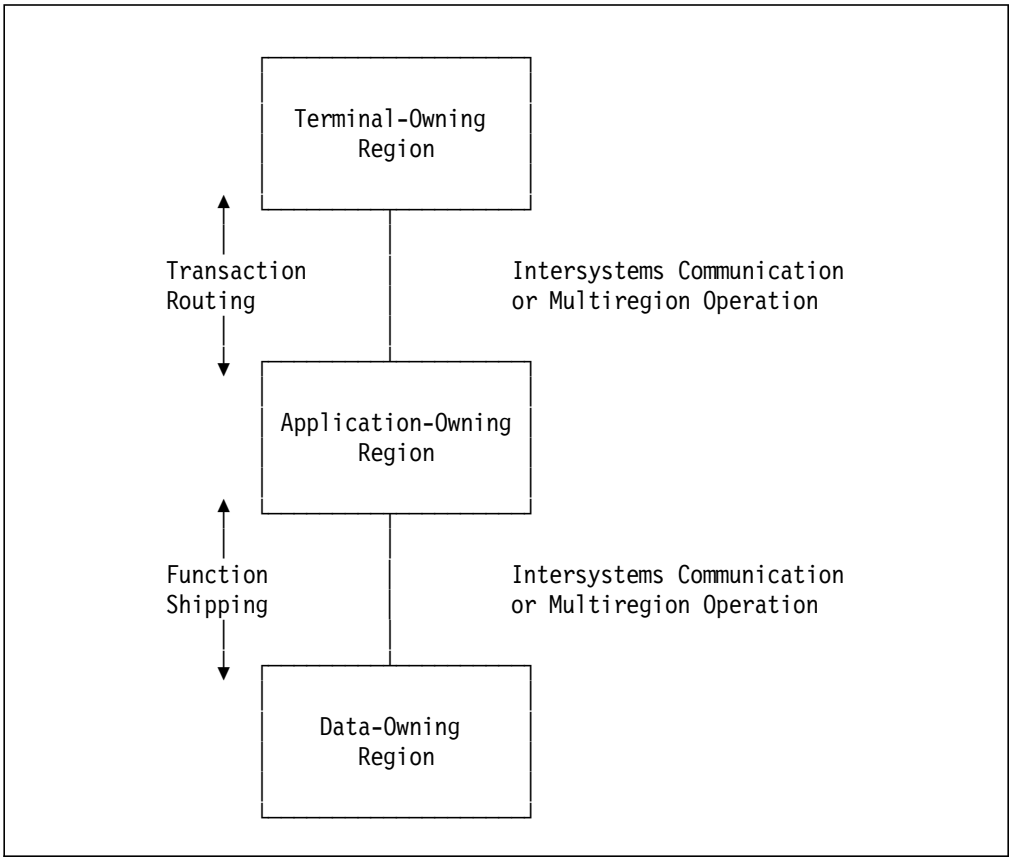


Figure 2. Splitting a CICS Region to Reduce Stress

At first sight this is not an obvious solution, since it increases the complexity of the system and introduces two extra components that can fail. Arithmetic says that if a single region is 99% available, then a three tier structure like this is only going to be 97% available. In fact this is one case where simple arithmetic breaks down.

Let us look at an example: A single region is having storage problems. Frequent SOS conditions cause stalls and the entire system locks and has to be re-started. Moving to a TOR-AOR-DOR structure relieves stress on AOR, and then allows further cloning or replicating of AORs to distribute storage requirements across multiple AORs. Typically this results in an improvement in availability if the cause of unavailability was a stressed region. Again, you must analyze your system and understand what is causing your unavailability.

Adopting a TOR-AOR-DOR structure has two advantages: It removes stress on the individual systems and makes them more reliable. It also reduces restart times, restoring service to users of the system earlier. Typically the TOR and DOR are the components of the system that take longest to recover if there is a failure (the TOR has to reestablish connections with the terminals in the network, the DOR has to do all the processing for file backout and recovery during an emergency restart). Fortunately, the TOR and the DOR are usually the most reliable parts of the system, containing very little of your own code. They run IBM supplied code, which is used in thousands of sites every day.

No matter how extensive your testing, your applications are usually specific to your site. The AORs running this code are likely to be less reliable that the TOR or the DOR. However, the AOR does not contain any recoverable resources, and only has a limited number of connections to other regions. Restarts are very quick.

Adopting a TOR-AOR-DOR structure thus separates the components that are most likely to fail from those that are most reliable.

2.4.5 Removing Single Points of Unavailability

Unavailability can be caused by the failure of a critical component, or by the requirement for two processes to have exclusive access to a resource at the same time (for example database reorganization). This section contains some simple examples to show you how to estimate the effects of duplicating components of your system to remove a dependency on a critical component. We then discuss how to remove the need for exclusive access to a resource.

The principles discussed here can be applied to MVS systems, a group of MVS systems (sysplex), CICS regions, a group of CICS regions (CICSplex), or to any other system component. Figure 3 shows our starting point, a one component system. Component A is 90% available, and therefore our whole system is 90% available.

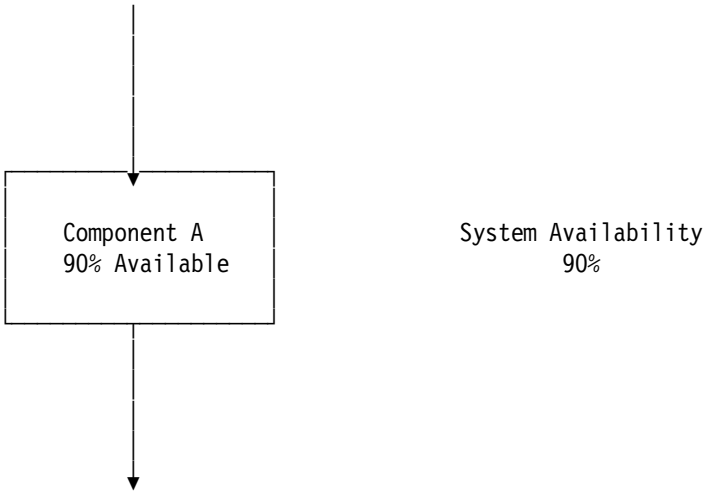


Figure 3. Single Component: System Availability

2.4.5.1 Duplication of Independent Components

Looking at a simple case, arithmetic tells us that if the component in Figure 3 on page 14 is 90% available, then we can increase the overall availability of the system to 99% by duplicating this component in a configuration like that shown in Figure 4.

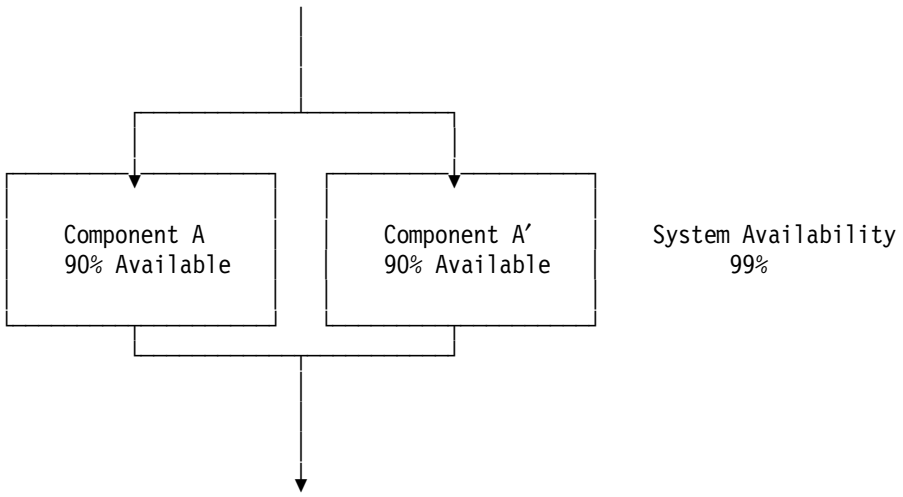


Figure 4. Independent Duplicated Component: System Availability

If Component A is 90% available, then it is 10% unavailable. Component A′ is an exact duplicate of Component A, with the same availability characteristics. The chances that both Component A and Component A′ are unavailable at the same time is therefore 10%×10%, or 1% (or 0.1×0.1=0.01). The system is 99% available, a dramatic improvement in availability.

The effect of duplicating components is more marked in cases of low availability. The effect of duplicating a component that is 99% available is to improve overall availability to 99.99%, less than a 1% improvement.

2.4.5.2 Duplication of Linked Components

We said that the example in section 2.4.5.1, “Duplication of Independent Components” is a simple case. It assumes that any failure of Component A is completely independent of Component A′. This is not always so. Imagine that Component A is a light bulb, and that our objective is to ensure that there is always some lighting in a room when we want it. Both Component A (Light Bulb A) and Component A′ (Light Bulb A′) run off the same electrical supply. The room can be dark because:

- We switched the lights off (planned unavailability, or operator error)
- The electrical supply failed (affecting both lights)
- Both light bulbs failed at the same time.

Figure 5 on page 16 shows what this system looks like. You can see that even though we have duplicated the light bulb, the whole system can fail because we have two further single points of availability. We have made some further assumptions; the electrical supply is 99.99% available, but we are having some problems with the light switch and it is only 97% available, reducing the overall system availability to approximately 96%.

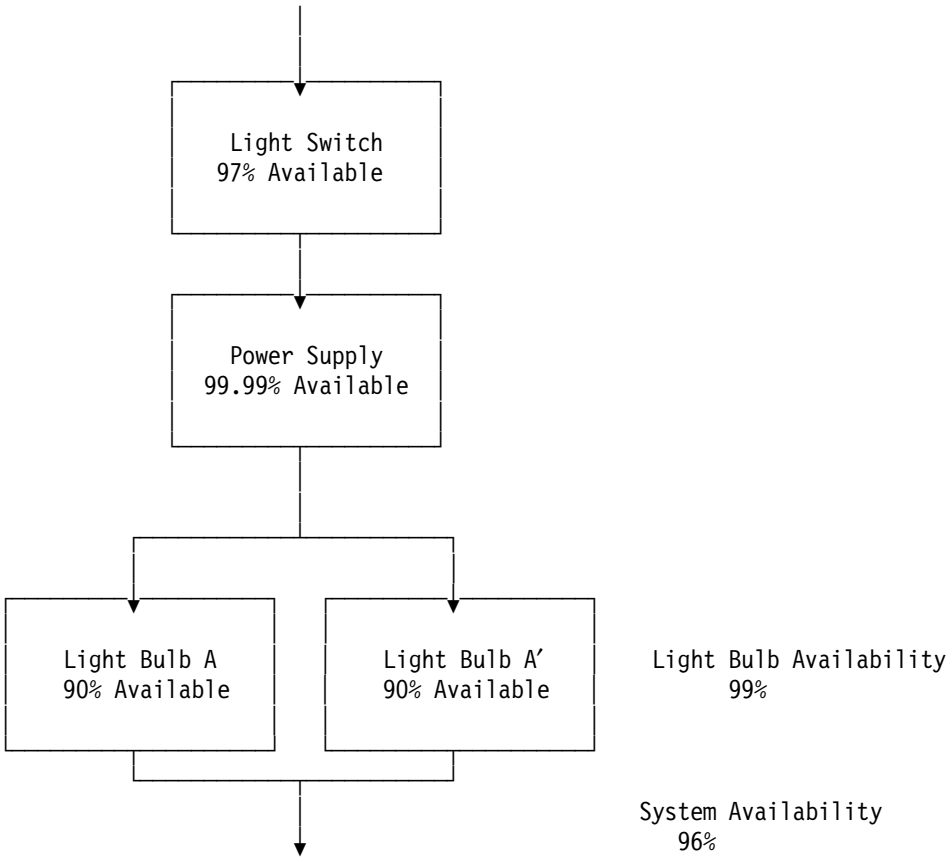


Figure 5. Linked Duplicated Component: System Availability

The major problem in this system is the light switch. Duplicating the other components will have a limited effect on improving availability until the problem with the light switch is resolved.

2.4.5.3 Duplication and Change Management

What if we wanted to upgrade the light bulbs in the system shown in Figure 5? Maybe replacing 50 watt light bulbs with 100 watt light bulbs? If we only had a single light bulb we could not change the bulb without making the system unavailable. Now that we have duplicated the light bulb, we can replace one even when the lights are on, and the other light bulb remains on (the system is still available).

This is an important principle, and is key to building a system that is continuously available.

2.4.5.4 Removing the Need for Exclusive Access

This is one of the key problems in providing a continuously available system. If any one part of your system requires exclusive access to a resource, then the whole system becomes unavailable if the resource is being used by another system.

Today the need for exclusive access usually applies to data, and is caused by the need to:

- Update a VSAM file from a batch program
- Reorganize a VSAM file
- Reorganize a DB2 database
- Reorganize a full function IMS database.

Using the latest levels of software reduces the impact of this problem. For example, you should use DFSMS/MVS 1.2 or later so that you can take backup copies of VSAM files while CICS is updating them, or even eliminate them (for example, you should use the DBCTL interface between CICS/ESA and IMS/ESA DB so that you can use data-entry databases, which can be reorganized while they are being updated).

2.4.6 Improving Restart Times

To restore an unavailable CICS region you must detect that it is unavailable, take any corrective action that is required, and then restart the system. One of the most effective things you can do is to automate this process. There are a number of tools and software components that can detect the unavailability of a CICS region and automatically restart the system. For example, CICSPlex SM can detect that a CICS region is not available when it should be and raise an alert. An automated operations tool, such as NetView, detects the alert and acts upon it. Alternatively, you could use the MVS automatic restart manager (ARM) to both detect the unexpected termination of a CICS region and restart it. The extended recovery facility (XRF) performs a similar function by automatically detecting that the active CICS region has stopped writing information to the CICS availability manager (CAVM) data set, and automatically switching work to an alternate CICS region.

Having detected the problem, and issued the command to restart CICS, you should take action to ensure that the restart completes as quickly as possible. The two things that are most important here are to reduce the amount of work that needs to be done during a restart, and to tune the system to make restarts as fast as possible. You can reduce the amount of work that needs to be done during a restart by implementing a TOR-AOR-DOR workload split (reducing the number of resources controlled by each region), by deferring work (using autoinstall, for example), and by specifying a low activity keypoint frequency in the SIT.

Tuning the system means, for example, making sure that the datasets used by CICS during a restart are not on highly active disk devices. Remember that if several CICS regions are restarted at the same time, and their logs and catalogs are on the same volumes, device activity may be unusually high while restart is taking place.

2.4.7 Choosing the Most Cost Effective Action

Let us return to the system shown in Figure 5 on page 16. Further analysis shows that the main problem with our light availability is that the light switch is in the wrong position, usually because of human error. We can fix this easily by installing a simple timer. To determine whether this is the most effective action, we compare the cost and effect of this action with other actions. Table 1 on page 18 summarizes the actions we could take, with their estimated effect on availability and their estimated cost.

Table 1. Choosing the Most Effective Action			
Action	**Estimated Cost**	**Estimated Improvement**	**Relative Effectiveness**
Install Timer	$45	3%	Best
Add Third Light Bulb	$45	1%	Middle
Duplicate Power Supply	$150,000	0.1%	Worst

We can see quickly that buying a timer for the light switch is a very cost effective solution, while duplicating the power supply is the least cost effective.

2.5 Taking an Action, and Reviewing Its Effect

The final stage is to decide which action you will take. This may not be the most cost effective action. The most cost effective action may be beyond your current means, or may take a year to implement. You may choose to take the middle ranking action that has an immediate beneficial effect and can be implemented swiftly and cheaply as a tactical solution. This is fine, as long as you understand that it may not cure your problem.

One important thing that you must do is measure the impact of what you did. If your action was more effective than you expected, you should review your list of recommended actions to see if they still have the same ranking and expected effectiveness. If your action was less effective than you expected, you should spend some time understanding why. In some cases you can achieve the effect you expected with a small additional change. In the worst case you can learn from your experience, and become a better planner for the future.

2.6 An Outline Plan for Improving System Availability

You are probably reading this redbook because you want to improve the availability of your system. In this section we take the ideas presented in the rest of the chapter and consolidate them into an outline plan that you can use to improve the availability of your existing systems.

You need to make a thorough analysis of your existing CICS system and any causes of unavailability in order to build your action plan to move towards continuous availability. The following list is a guideline on how to improve the availability in an existing CICS system:

1. Analyze service interruptions to identify the major causes of unavailability. Look at:

   - Hardware

     The central mainframe, the network, the environment, and so on.

   - Software

     The operating system, NCP, VTAM, CICS, data sets and databases, your own application programs, bought in application packages, and so on.

   - Operations

     Incorrect procedures, documentation, human errors, and so on.

2. Create an action plan to reduce the frequency of interruptions.

   When you have completed your analysis, you should understand what is causing unavailability in your system. You should be able to identify the most frequent causes of unavailability. From this you can propose an action to reduce the frequency of interruptions. Section 2.4, “Estimating the Cost of Getting Where You Want to Be” on page 9 discusses causes of unavailability and possible actions. You could, for example, decide to use duplication to increase redundancy in your system, to implement or improve change control procedures, to use automation to simplify operations, or to improve the documentation for operators.

3. Analyze restart time to identify major components. These could include:

   - Operational delays in noticing the failure, deciding what to do, doing it, and so on.
   - Application related delays, such as a requirement to run housekeeping procedures, delays due to PLT programs, and so on.
   - CICS related delays, such as dumping the failed CICS system, archiving logs, reinitializing the network, opening files, backing out transactions, and so on.

4. Create an action plan to reduce restart time.

   When you have completed your analysis, you should understand the components of restart time in your system, and be able to identify those components that can be reduced or eliminated. From this you can propose an action to reduce your restart time. You could, for example, isolate applications with slow housekeeping routines in a separate AOR (thus improving restart times for other applications), reduce the amount of PLT processing, or reduce the activity keypoint frequency in your systems (that is, reduce the time between activity keypoints by reducing the value of AKPFREQ in your SIT).

5. Predict the resulting improvements in availability.
6. Monitor the actual improvements.

You can perform these actions on your existing system no matter what hardware or software you are using, or your system design.

2.6.1 Identify the Major Causes of Unavailability

You must record each period of scheduled and unscheduled unavailability. This will give you a base to work from. Your record should include at least:

- A description of the period of unavailability
- The action taken to resolve it
- The root cause analysis
- Start and end time
- Length of outage
- Impact on users, or to your business.

The description should be brief and contain any ABEND messages or other symptoms. It should be enough to allow a person to review the documentation and be able to group incidents together for further review.

You should record all actions taken to end the period of unavailability, even if they did not succeed. You should review these periodically. From this you can identify whether the most appropriate action was taken, whether automation would help, or whether education or improved procedures are required. This information can be used as a base for updating procedures.

You should go back and perform root cause analysis after you have ended the period of unavailability. Looking back at the situation without pressure often lets you see what was the root cause of unavailability. If you can remove this root cause you may avoid future problems whose descriptions are very dissimilar, but share the same root cause. Root cause analysis may point out the need for changes to hardware, software, or system design. Therefore the root cause analysis should be shared, in order to gather input from a wide scope of sources.

You should record the start and end time of any period of unavailability. This can illustrate patterns of usage, and may also help with root cause analysis. For unscheduled unavailability this can help in assessing the impact of the outage. For scheduled unavailability you can check that the time was that planned.

If you record the length of a period of unavailability, reviewers can see at a glance the comparative severity or impact of the incident. The impact of any outage can be from total unavailability, to reduced functions for a few users. Some sites use a 1-10 scale, some a complex formula based on function usage versus total usage. Whatever system you use, be consistent. Changing scales makes future comparisons very difficult.

2.6.2 Reduce Repeated Outages

Repeated outages should be the easiest to reduce. You should be able to identify these outages from the description, and root cause analysis. From this group you can create a list of priorities, based on impact and length of outage. The solution to the problem can be as straight forward as operator education, or as complex as system redesign. Whatever action you choose to take, remember to predict the expected improvement in availability and the cost of implementing it. As with most business decisions, the benefit must outweigh the cost of implementing the solution.

2.6.3 Analyze Restart Times

You should use the times on the messages produced by CICS during startup to identify which parts of the process take the most time. You should pay particular attention to any part of the startup process that seems to take significantly longer in one region than in others, or to any part that takes a long time in all of your regions, or to any part that sometimes seems to take longer than usual.

2.6.4 Reduce Restart Times

You can do several things to influence the startup time of your CICS regions. Ensure that your global catalog data set (GCD), local global catalog data set (LCD), and restart data set (RSD) have the appropriate VSAM parameters specified.

You should ensure that you are using all the features of CICS that improve restart times, for example, CICS/ESA 4.1 support for the MVS ARM, persistent sessions, autoinstall of LU6.2 devices, and so on. See section 7.5, “Improving

Restart Times" on page 86. Remember to predict the expected improvement in availability and the cost of implementing it.

2.6.5 Track Results

Section 2.3.1, "System Availability" on page 8 states the importance of measuring the availability of your systems. The records you keep not only allow you to identify causes of unavailability, they also allow you measure the improvements in availability you achieve. You should regularly compare the information discussed in section 2.6.1, "Identify the Major Causes of Unavailability" on page 19 with your availability reports. This will allow you to determine whether the actions you are taking are increasing availability. If you cannot see an improvement in availability, then you may be dealing with symptoms of a problem, not the underlying problem itself.

2.6.6 Repeat at Regular Intervals

Maintaining continuous availability is an ongoing process that requires regular review and changing focus. Changing requirements most often mean higher availability, faster throughput and less scheduled down time. Knowing where you need to improve, and what will enable you to reach those goals will keep you one step ahead.

Chapter 3. Our Recommendations

In this chapter we describe the best environment you can build for continuous availability of CICS in an MVS/ESA environment using technology available today. You should use this as a guide to what you can achieve. After reading Chapter 2, “The Business Perspective” on page 5 you may decide that this environment is not appropriate for your company. In this case you may choose to implement a subset of the configuration and recommendations that we make here. For example, you may decide that you cannot justify running two MVS systems on separate processors; you can still implement multiple CICS regions on one MVS image to gain many of the benefits of our ideal configuration.

Where a specific level of CICS, the operating system, or a database is required we make a note of the level you need.

Our ideal system has many duplicate parts, and the unavailability of one or more will have little or no impact on users of the system. This allows you to make changes to parts of the system, while still providing full service to the users. Our recommended configuration uses the latest technology and design principles. We realize however, that it is not always possible to implement all of our recommendations because of various constraints. We suggest that you use this chapter as a guide to review your system and as a road map to bring your availability to the level you require.

Our recommended configuration covers the following:

- Recommended system topology
- Supporting management infrastructure
- Application considerations
- Recommended software components
- Recommended hardware components

3.1 Recommended System Topology

Our topology eliminates many of the causes of unavailability discussed in 2.4.1, “Causes of Unavailability” on page 9. Duplication is a basic principle in our design, eliminating any one piece of hardware or software as a single point of unavailability.

Figure 6 on page 24 shows the components of our ideal system. The flow of the arrows down from the terminal shows the various paths that a transaction can take.

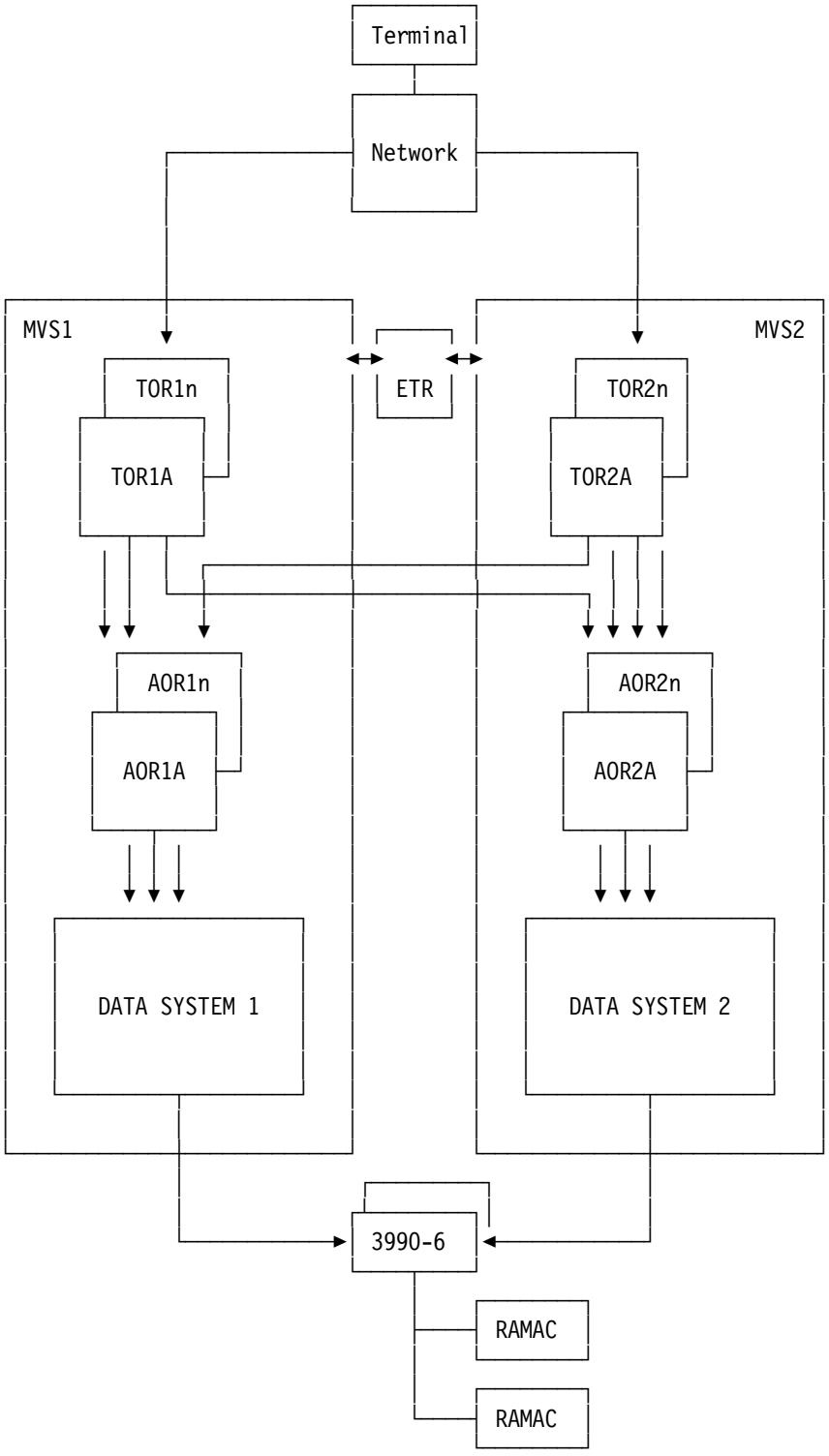


Figure 6. Recommended System Configuration

The components of our recommended solution are:

- The terminal
- The network
- An MVS sysplex
- A CICSplex, consisting of multiple CICS regions
- CICSPlex SM to manage dynamic transaction routing in the CICSplex (not shown in Figure 6 on page 24)
- A data subsystem (for example, IMS Version 5, DB2 Version 4, or VSAM (in DFSMS/MVS Version 1 Release 3)) which allows data sharing
- The I/O subsystem.

3.1.1 The Terminal

If a terminal is unavailable (because of failure, or the need to apply maintenance), the user of the terminal is unable to access applications. The impact of the unavailability of the terminal depends on how it is used. If the terminal is shared by many users, for example, an automated teller machine (ATM), the preferred solution is to duplicate the terminal. If one of the ATMs fails, your customers can still process their transactions through the other ATM, even if they have to wait longer to do so. Duplication in this case could mean physically having a second machine at the same site, or it could mean giving clear instructions on where a nearby ATM is located.

If the terminal serves only one user (for example, it is the terminal usually used by a systems programmer or application programmer), you may choose not to duplicate the terminal but to ask the terminal user to use an alternative until the terminal is available again. For example, you can ask the terminal user to use the terminal normally assigned to a colleague who is currently on vacation.

Whichever solution you adopt, unavailability of a terminal typically means some inconvenience to a small number of users, and has a low overall impact.

3.1.2 The Network

Designing your network for maximum availability is a topic in its own right, and we do not cover it in detail in this redbook. There are many possible configurations. You should remember that any unavailability in the network can affect a small department (LAN outage) to large number of users (large scale network failure or VTAM failure).

The elements to remember and look for when building your network are those that help you avoid the causes of unavailability discussed in 2.4.1, “Causes of Unavailability” on page 9.

- Changing your system

  ACF/VTAM Version 3 Release 4 and later releases support dynamic definition of independent and dependent logical units (LUs), dynamic modification of ACF/VTAM exits, and dynamic modification of the default logon mode (DLOGMOD) for an LU. Advanced peer-to-peer networking (APPN) enables dynamic reconfiguration of your network. Together these significantly enhance the availability characteristics of your network by making it possible to change the network without interrupting service.

- Avoiding errors

  Tools, such as NetView, help you to automate operation of your network, reducing the chance of human error.

- Remove stress points

  You can stress test your network using TPNS to remove potential stress points before they affect availability.

- Remove single point of unavailability

  ACF/VTAM and APPN allow automatic use of alternate routes in the event of a failure in the network.

3.1.3 MVS Sysplex

A sysplex consists of a connected set of MVS system images and system hardware components, treated as building blocks which may be replicated. While an MVS sysplex could in theory consist of a single MVS system image, a continuously available sysplex requires the use of two or more MVS images on two or more central processing complexes (CPCs), as shown in our configuration. The main software component is the cross system coupling facility (XCF), an integral part of the MVS base control program (BCP) for MVS/ESA Version 4 and later versions of MVS/ESA. XCF controls members and groups in the sysplex, and provides intermember communications and monitoring services for members. The hardware components are:

- A sysplex timer (if the MVS images are operating on two or more processors) to synchronize the clocks in the sysplex. The sysplex timer is also called an external time reference (ETR).
- Channel-to-channel links, ESCON channels or high-speed coupling facility links, for XCF signaling.
- An XCF couple data set. XCF requires a DASD data set that is shared by all systems in the sysplex.
- DASD controllers with enough paths to dedicate one path to each MVS image in the sysplex, for data sharing.
- A coupling facility. A coupling facility is not required for a basic sysplex, but many of our components for achieving continuous availability require a coupling facility.

We also require an MVS sysplex to eliminate an MVS system image as a single point of unavailability.

An MVS sysplex helps you to achieve continuous availability by addressing the following causes of unavailability:

- Changing your system

  Dynamic Reconfiguration Management allows you to change I/O configurations without interrupting system service. Enterprise Systems Connection Architecture (ESCON) provides dynamic connectivity between processors and I/O devices through switched point-to-point topology and the ESCON Director.

- Removing stress points

  The workload manager components of MVS/ESA 5.1 and MVS/ESA 5.2 allocate resources on an ″as required″ basis. This dynamic reallocation of resources means that a workload is less likely to be constrained by

processor capacity or by real storage requirements. The workload manager can only provide efficient allocation of the resources available; you still need enough overall capacity to run your work. The workload manager avoids the situation where one workload is overstressed (may be real storage constrained) while another less important workload is allowed to overuse the resources of the system.

- Removing single point of unavailability

  Changes in MVS/ESA 5.1 made it easier for you to create multiple cloned MVS system images, duplicating MVS and effectively removing it, or any one physical processor, as a single point of unavailability. If any one MVS system image or CPC is unavailable, the work normally run there can be redistributed to the other MVS system images in the sysplex. In effect, the MVS system images in the sysplex are seen from outside as a single system.

- Improving restart times

  MVS/ESA 5.2 introduced the automatic restart manager (ARM). This automatically restarts a failed component. You can control whether the failed component is restarted on the MVS system image it was originally running on, or on another MVS system image in the sysplex.

For more information, see *SYSTEM/390 MVS Sysplex Overview: An Introduction to Data Sharing and Parallelism*, and *A Comparison of System/390 Configurations - Parallel and Traditional*.

3.1.4 A CICSplex

A CICSplex is a connected set of CICS regions (or a complex of CICS regions). The essential components are:

- One or more terminal-owning regions (TORs)
- One or more application-owning regions (AORs)
- Optionally, a data-owning region[1] (DOR).
- A mechanism to control routing of incoming transactions from the TOR to an appropriate AOR for execution. Ideally this should be done using dynamic transaction routing.

We describe a CICSplex, TORs, AORs, FORs, QORs, DORs, and transaction routing in more detail in Chapter 4, “More on System Topology” on page 37.

Our recommended configuration consists of at least two TORs and multiple AORs on each MVS system image in the sysplex. The two TORs provide duplication and avoid a single point of unavailability. VTAM Generic Resource Registration allows the two TORs to have the same external name (and indeed allows the TORs on any other MVS system image in the sysplex to share that name), so that if any one TOR is unavailable end users are not aware of it as they log on. The multiple AORs protect the system from a single point of unavailability. They also allow the system to use sophisticated workload balancing, health checking, and ABEND avoidance logic in our dynamic routing

[1] We have used the slightly more general term data-owning region here. In fact this could be a region that processes requests only for CICS transient data and temporary storage queues, a queue-owning region (QOR), or it could be a region that processes requests only for CICS file control, a file-owning region (FOR).

program. This chooses the region which is least stressed and least likely to have an error as the target for transaction routing. Ideally each AOR should be capable of running any transaction.

- Changing your system

  Our CICSplex configuration allows you to remove any TOR, AOR, or DOR from the CICSplex, apply changes, and then reintegrate the region with the CICSplex. Duplication of the components means that work is redistributed across the remaining elements while you make your changes.

- Avoiding errors

  Our CICSplex configuration allows your dynamic routing program to route transactions away from AORs which are experiencing errors when running the transaction.

- Remove stress points

  Our CICSplex configuration allows dynamic balancing of work across the AORs in the CICSplex, thus reducing the chance that any one region will become a stress point.

- Remove single point of unavailability

  Our CICSplex configuration removes any TOR, AOR, or DOR as a single point of unavailability, as long as at least two AORs in the CICSplex are capable of running any specific transaction and a second DOR, capable of data sharing, is available to provide an access path to your data. Your dynamic routing program routes incoming transactions to whichever available AOR is capable of running them.

- Improving restart times

  The TOR-AOR-DOR split allows work to be spread over different regions during a restart.

For more information, see section 4.1, “The CICSplex” on page 37 and section 4.2, “Dynamic Transaction Routing” on page 38.

3.1.5 CICSPlex SM

You must provide a dynamic routing program to control CICS dynamic transaction routing. You can either write your own routing logic, or buy an application to control routing. The workload management functions of CICSPlex SM provide a dynamic routing program which tells the TORs in a CICSplex to which AOR they should route a transaction. The dynamic routing program first checks to see if there are any constraints on routing; it then applies a balancing algorithm and chooses the most suitable AOR.

CICSPlex SM also provides automatic resource monitoring. It can perform system availability and resource health checks automatically, generating alerts for further, possibly automated, operator processing when it detects an abnormal situation.

- Changing your system

  The workload balancing functions of CICSPlex SM allow dynamic balancing of work across the AORs in the CICSplex, thus reducing the chance that any one region will become a stress point.

- Avoiding errors

  The workload management functions of CICSPlex SM automatically avoid routing transactions to AORs where they are likely to abend, thus minimizing the effect of programming errors. CICSPlex SM also minimizes operator error by providing a single point of control. The real-time analysis (RTA) functions can detect error situations (such as a component not being available when it should be). RTA creates an alert. You can either use an automation tool, such as NetView, to take an action, have your operators respond to the alert, or leave the workload management function to deal with the error by routing transactions elsewhere until it is resolved.

- Remove stress points

  The workload balancing functions of CICSPlex SM allow for dynamic balancing of work across the AORs in the CICSplex. This reduces the chance that any one region will become a stress point. You can use the RTA functions to warn you of an impending stress situation, and then take actions to prevent it.

- Remove single point of unavailability

  The workload balancing functions of CICSPlex SM are aware of the status of the AORs. If you remove an AOR, or if it abends, the workload balancing functions automatically reroute transactions to other AORs in the CICSplex. If a DOR (or DB2 or DBCTL in a particular MVS system image) is unavailable then the workload management functions either detect that transactions are abending in the AORs using that DOR and route transactions to AORs connected to an active DOR, or you can configure RTA to raise an alert if the DOR (or DB2 or DBCTL) are unavailability, causing the workload management functions to avoid routing to AORs that depend on the unavailable DOR.

- Improving restart times

  You can use the RTA functions to detect that a component of the system is not available, and trigger an action to restart it. This could be particularly useful if you have not yet implemented MVS/ESA 5.2, and so do not have access to the ARM functions.

3.1.6 The Data Subsystem

A data subsystem which allows multiple CICS regions on different MVS system images concurrent access to data is an important component of a continuously available system. If you can only access the data through one MVS system image, or through one DOR, unavailability of that MVS system image or DOR makes the data unavailability to all users. If you only have one MVS system image on one CPC, your data subsystem is almost certainly a single point of unavailability.

The data subsystems that enable data sharing are:

- IBM DATABASE2 (DB2) Version 4 Release 1
- IBM IMS/ESA Database Manager (IMS/ESA DB) Version 5 Release 1[2]
- DFSMS/MVS 1.3 (providing VSAM function).

[2] IMS/ESA DB 5.1 provides full data sharing. IMS/ESA DB 4.1 and earlier versions allow two-way data sharing.

These components are discussed in more detail in 4.3, “Data Sharing” on page 40.

- Changing your system

  You can remove any one data subsystem from your configuration to make changes to it. Ideally you should do this by first quiescing the AORs attached to this data subsystem. The CICS dynamic routing program provided by CICSPlex SM is aware that the AOR is quiesced, and automatically routes work to the AORs connected to the corresponding data subsystems on other MVS system images.

- Remove stress points

  Workload balancing, as provided by CICSPlex SM, tends to route work to those AORs with the best response times. If the data subsystem on one MVS system image is overstressed, the dynamic routing program routes more work to CICS regions on the other MVS system image and so avoids aggravating the problem.

- Remove single point of unavailability

  If any one data subsystem is unavailable, work is routed to the AORs connected to the data subsystem on another MVS system image.

3.1.7 The I/O Subsystem

The effect of other failures, such as those to controllers or channels, can be avoided by correct configuration planning. Alternate paths must exist to all critical devices, so that a unit failure in a path does not isolate other units further down the path.

An example of a correct configuration for a disk device is shown in Figure 7 on page 31. Refer to *IBM 3990 Storage Control Introduction* for 3990s configurations.

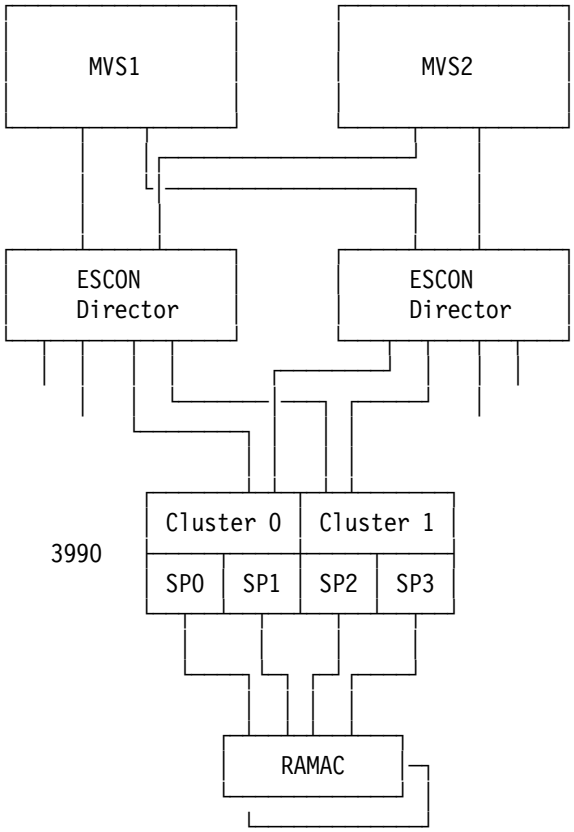


Figure 7. I/O Hardware Configuration for Continuous Availability

This configuration permits dual access from either MVS system image to the RAMAC string. The paths are configured so that failure of a single component will not cause unavailability of the data.

- Changing your system

  The I/O subsystem is configured so that any one path, ESCON director, 3990 cluster, or 3990 storage path can be removed for maintenance. The RAMAC disk drives have built-in features that allow non-disruptive change, including hot-plugging and the ability to remove a drawer and replace it without interrupting service.

- Remove single point of unavailability

  The I/O subsystem is configured so that any one path, ESCON director, 3990 cluster, or 3990 storage path can fail or be removed. The 3990 can write out duplicate copies of data, removing any disk drive as a single point of unavailability. In addition to this, the RAMAC devices have built-in redundancy and duplication, removing any component as a single point of unavailability.

3.2 Management Infrastructure

You must have an effective management infrastructure in place to maintain continuous availability for your systems. The management infrastructure consists of all the procedures for operating your system, changing it, handling problems, testing new applications, testing new levels of system software, and so on. If you do not have these procedures in place, you will quickly find that errors in carrying out these tasks, or errors introduced as a result of inadequate testing, quickly become your greatest source of unavailability, and negate all the effort that you have put into creating a flexible and robust system topology.

The management infrastructure you must create is discussed in Chapter 5, “Operations, Procedures, and Systems Management” on page 45.

3.3 Application Considerations

Your applications must be designed to be robust and to be capable of exploiting our recommended system topology. For example, if your application must run in a particular CICS region, that CICS region becomes a single point of unavailability for this application, negating any benefit you would have gained by adopting our recommended system topology.

When you build your applications, you must consider:

- Dealing with unexpected conditions, so that an unusual condition does not cause your application to become unavailable
- Building efficient applications, so that your application does not cause the system to become overstressed
- Avoiding affinities, so that your application can run in any AOR in your CICSplex, and a single AOR does not become a single point of unavailability for your application
- Avoiding requirements for exclusive use of resources, thus avoiding another cause of single point of unavailability.

We discuss application design in more detail in Chapter 6, “Application Design” on page 63.

3.4 Recommended Software Components

The following section covers:

- VTAM
- MVS/ESA
- CICS/ESA
- CICSPlex SM
- DB2
- IMS/ESA
- VSAM
- CICSVR

3.4.1 ACF/VTAM

We recommend that you use a version and release no earlier than ACF/VTAM Version 4 Release 2. VTAM 4.2 supports VTAM generic resources as well as persistent sessions. VTAM persistent session support is available with VTAM Version 3.4.1 and current maintenance, but only VTAM 4.2 supplies both generic resources and persistent sessions.

3.4.2 MVS/ESA

We recommend that you use a version and release no earlier than MVS/ESA Version 5 Release 2 as your base. It provides you with these benefits over MVS/ESA Version 5 Release 1:

- Increased availability of data using the new automatic restart manager (ARM)
- Better workload management by supporting workload balancing across multiple servers in a sysplex
- System management improvements using a new system logger facility and dynamic reconfiguration of coupling facility structures
- A number of hardware configuration definition enhancements.

MVS/ESA 5.1 provides software support for data sharing and parallelism in a sysplex, including support for the coupling facility and 9672 Parallel Transaction Server. MVS/ESA 5.1 also provides increased availability of data, and simplified workload management. The system management enhancements improve multisystem management, and may improve productivity in a sysplex.

3.4.3 CICS/ESA

Each release of CICS includes enhancements designed to increase the overall availability of the CICS systems. We recommend that you use a version and release no earlier than CICS/ESA Version 4 Release 1 in your configuration. CICS/ESA 4.1 includes several availability enhancements: transaction isolation, VTAM generic resource registration, VTAM single-node persistent session support, a new global user exit (GLUE) called XZIQUE, and dynamic transaction routing enhancements.

Transaction isolation provides storage protection between application transactions, preventing one application program from accidentally overwriting storage belonging to another and thus causing an error. For more information on transaction isolation see section 7.2.2, “Minimize Storage Addressing Errors” on page 79.

VTAM generic resource registration allows multiple TORs to share the same generic VTAM APPLID. This reduces the dependency on a single TOR. If a TOR is unavailability, users can log on to any other TOR that shares the same generic APPLID. They will not see any unavailability.

VTAM single-node persistent session support improves restart time in the event of a failure. It can prevent LUs from being unbound after a CICS failure, removing the need for the sessions to be rebound thus reducing the amount of processing required to restart CICS. See the *CICS/ESA Resource Definition Guide* for details.

XZIQUE, an new GLUE, prevents overstressing of systems by limiting the amount of queueing on sessions between CICS regions. XZIQUE is invoked for any request that causes a session to be allocated, including function shipping requests, transaction routing, and distributed program link (DPL). Managing queues for intersystem sessions in this way helps prevent a region from becoming overstressed because of many tasks which are enqueued on a link to another region. The other region may be stalled, or "sick." This phenomenon of one region becoming overstressed because of the number of tasks enqueued on a link or waiting for activity in another sick region is sometimes called "sympathy sickness." For more information on XZIQUE see section 7.3.2, "Intersystem Session Queue Management" on page 80.

Dynamic transaction routing enhancements provide integration with the MVS/ESA WLM and also provide more information to your dynamic routing program. For example, CICS now reinvokes the dynamic routing program when a task ends, so that the dynamic routing program knows the number of tasks currently assigned to each AOR.

The enhancements in CICS/ESA 4.1 are built on CICS/ESA 3.3. CICS/ESA 3.3 provides availability enhancements with storage protection, and a new GLUE, XISCONA, which limits intersystem queueing for function shipping requests.

Storage protection prevents application programs from overwriting CICS code. This reduces the chance of an application programming error causing an ABEND in CICS code which brings down the region. Storage protection is implemented by using storage keys. An application running in USER key is prevented from accessing storage held in a CICS key data area. As an extension to this support, reentrant programs are placed into a separate storage area, the extended read-only dynamic storage area (ERDSA). The ERDSA is obtained in either CICS key (protected from being overwritten by an application program, but accessible to CICS), or in read-only key (protected from being overwritten by both application programs and CICS). ERDSA support is available in all CICS/ESA 3.3 or later environments. You need to implement MVS/ESA 4.2.2 or later on a processor that supports the MVS subsystem storage protection function to use CICS storage protection. For more information on storage protection, see section 7.2.2, "Minimize Storage Addressing Errors" on page 79.

CICS/ESA 3.3 also introduced a new GLUE, XISCONA, which can limit the queuing of function shipped requests only. As with XZIQUE, XISCONA helps control intersystem queueing and so prevents "sympathy sickness" when one region is not responding. For more information on XISCONA see section 7.3.2, "Intersystem Session Queue Management" on page 80.

3.4.4 CICSPlex SM

We recommend that you include CICSPlex SM in your configuration, using a version and release no earlier than CICSPlex SM Version 1 Release 2. CICSPlex SM provides:

- A workload manager component. This allows workload balancing across a CICSplex, on one or more MVS sysplexes. CICSPlex SM workload balancing allows you greater flexibility in system design and improves availability, since CICSPlex SM will route your transactions to the most appropriate AOR at that time. If an AOR is unavailable, or in a condition that would impact the timely processing of the transaction, that AOR will not be selected if there is a more suitable AOR. CICSPlex SM uses one of two algorithms for workload

balancing, queue mode or goal mode. For more information, see *CICS Workload Management Using CICSPlex SM and the MVS/ESA Workload Manager*.

- A single system image. Your operators can control all the CICS regions in a CICSplex as if they are one. One command operates on all selected regions, reducing the possibility of an operator error.
- A single point of control. All the regions in your CICSplex can be controlled from one session, and your operators can gain a consolidated view of the CICSplex rather than individual CICS regions. This simpler view and improved information reduces the possibility of an operator error.
- Monitoring functions that automate the detection of abnormal conditions in your CICSplex, allowing speedy response.
- An application programming interface (API). CICSPlex SM 1.2 adds a new API to CICSPlex SM, making it easier for you to automate operation of your CICSplex.
- Support for the NetView Resource Object Data Manager (RODM). CICSPlex SM 1.2 reports the status of all non-transitory CICS resources to RODM.

For more information, see 4.2.3, “CICSPlex SM” on page 40.

3.4.5 DB2

We recommend that you use a version and release no earlier than IBM DATABASE 2 Version 4 Release 1. DB2 4.1 provides full data sharing for DB2 databases in a sysplex. Along with the IMS resource lock manager (IRLM) Version 2 Release 1 and cross system coupling facility (XCF), you can access DB2 databases from a CICSplex which spans multiple MVS system images.

3.4.6 IMS/ESA

We recommend that you use a version and release no earlier than IMS/ESA Version 5 Release 1. IMS/ESA 5.1 supports block-level data sharing between up to 32 IMS subsystems across multiple MVS system images. The remote site recovery (RSR) option of IMS/ESA 5.1 aids with disaster recovery by allowing you to resume online operations at a remote tracking site with minimal delay and minimum, if any, loss of data.

3.4.7 VSAM

We recommend that you use a version and release no earlier than DFSMS/MVS Version 1 Release 3. DFSMS/MVS 1.3 provides the VSAM support required for VSAM record level sharing (RLS). DFSMS/MVS 1.3 and a future release of CICS will enable the sharing of VSAM data across a sysplex. This will eliminate a file-owing region (FOR) as a single point of failure for VSAM data.

3.4.8 CICSVR

We recommend that you use a version and release no earlier than IBM CICS VSAM Recovery MVS/ESA Version 2 Release 2. CICSVR 2.2 automates VSAM recovery job creation, speeding the process and reducing the chance of an error. CICSVR 2.2 can track VSAM backups created while the VSAM data set is closed to CICS, or VSAM backups created using VSAM backup while open (BWO) while the VSAM data set is open and being used by CICS, as well as CICS logs.

For more information, see C.2, “IBM CICS VSAM Recovery MVS/ESA” on page 162.

3.5 Recommended Hardware Components

In the following section we cover the hardware used by our recommended system.

3.5.1 Processors

Our recommended system runs CICS/ESA 4.1 and exploits subsystem storage protection and transaction isolation. This requires an Enterprise Systems Architecture/390 (ESA/390) processor with the subspace-group facility.

We recommend that you use a parallel CMOS processor, as it offers these availability advantages over Bipolar processors:

- It does not require chilled water
- Additional processors can be added without interrupting service
- It allows higher total processing capability reducing overstressed systems
- It allows more parallel processing, limiting the effect of unavailability of any one part of the system.

3.5.2 DASD and Controllers

Our recommended design has 3990-6 controllers and RAMAC DASD.

3.5.2.1 Controllers

The 3990-6 controller provides higher availability features than previous models. The 3990-6 and the 3990-3 both provide two functions that help improve the availability of CICS systems: concurrent copy reduces the time required to take copies of data sets, and dual copy removes certain CICS data sets as single points of unavailability.

The 3990-6 introduced two new availability benefits: peer-to-peer remote copy (PPRC) and extended remote copy (XRC). For more information, see 13.4.1, “Peer-to-Peer Remote Copy and Extended Remote Copy” on page 142.

3.5.2.2 DASD

To maximize availability, we recommend that you use RAMAC DASD. RAMAC DASD uses the RAID-5 process to record your information on the DASD. Each block of data to be written to DASD is divided into three parts, and parity information is added to each. The three parts are then written to three different disks. If any one disk becomes unavailable, the data and parity information from the other two are used to recreate the data of the third.

Another feature of RAMAC is the ability to recreate the lost disk without interrupting the use of the data. Once the lost disk is recreated, the data can be used directly, not derived from the parity information.

The benefit of RAMAC DASD is its high availability and fault tolerance.

Chapter 4. More on System Topology

In this chapter we describe CICS and database system design features that allow you to duplicate components of your system, removing them as single points of unavailability.

4.1 The CICSplex

Up until the late 1970s most enterprises could run all of their data-processing operations on a single computer, and a single CICS system could be expected to handle an enterprise′s entire online transaction processing (OLTP) workload. Later, as a result of fast developments in hardware and operating system technology, and even faster developments in requirements to provide high availability, high capacity transaction processing systems, the concept of a CICSplex[3] evolved (though the name CICSplex was invented much later). A CICSplex is a connected set of CICS regions (or a complex of CICS regions). Figure 8 on page 38 shows a very simple CICSplex. Factors that led to the evolution of the CICSplex included:

- Lack of resources, particularly virtual storage
- Security
- Application isolation
- Performance
- Operational requirements (separate test and production, differing availability times)
- Integrity (isolate rogue applications).

CICS regions in a CICSplex are classified according to their main function, even though each is a full-function CICS system.

- Terminal-owning regions (TORs): These are used to manage terminal sessions. A TOR routes transactions initiated from terminals to another region for execution. This function is known as transaction routing. When transaction routing was first implemented in CICS, you had to define to which region the TOR should route a particular transaction by including a SYSID option in the transaction definition in the TOR. Once the transaction was defined to the TOR, all instances of that transaction would be routed to a particular region for processing, which makes the system inflexible (for example, if the region you want to route to is not available, the TOR cannot dynamically select an alternative region).[4] CICS/ESA Version 3 introduced a new form of transaction routing—dynamic transaction routing (see section 4.2, “Dynamic Transaction Routing” on page 38). Dynamic transaction routing uses a program (the dynamic transaction routing program) to decide to which region a particular transaction should be routed. The previous method of fixed definitions (now called static transaction routing) can still be used.
- Application-owning regions (AORs): These are used to actually process transactions and route the results back to the originating TOR. If the

[3] We use CICSplex, with a lowercase p, when referring to a complex of CICS regions. We use CICSPlex, with an uppercase P, when referring to the product CICSPlex SM.

[4] It is possible to use a CICS global user exit, such as DFHZATDX, to control routing by changing the transaction identifier (transid) of the incoming transaction before it is attached in the TOR. This allows some flexibility.

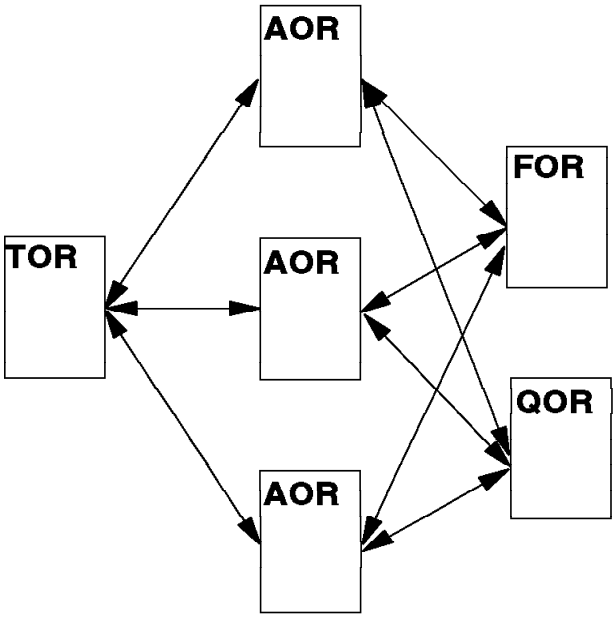


Figure 8. Example of a CICSplex

transaction running in the AOR needs to access remote data (that is, data located in yet another region), CICS sends a request from the AOR processing the transaction to the region that owns the resource (file, or transient data queue, or temporary storage queue) in which the data is located. This function is known as function shipping.

- File-owning regions (FORs): These are used to manage VSAM files, BDAM files, and CICS data tables. The FOR processes read, update, and delete requests for data records that have been function shipped to the FOR from an AOR, and returns the results to the originating AOR. FORs can also manage access to local DL/I databases, in which case the more general term data-owning region (DOR) is often used, instead of FOR.
- Printer-owning regions (PORs), queue-owning regions (QORs), and resource-owning regions (RORs): As with files, many other CICS resource types have been moved from AORs to dedicated owner regions. This allows multiple AORs to share access to external resources, usually through function shipping.

4.2 Dynamic Transaction Routing

CICS transaction routing is an intercommunication facility that allows terminals or logical units connected to one CICS region to initiate, and communicate with, transactions in another CICS region. This means that you can distribute terminals and transactions around your CICS systems and still have the ability to run any transaction with any terminal.

When you define transactions to CICS, you define them as local or remote. Local transactions always execute in the terminal-owning region (TOR), that is, in the CICS region to which the terminal initiating the transaction is directly logged on. Remote transactions are routed to other CICS regions connected to the terminal-owning region by MRO links, or to other systems that are connected

by intersystem communication (ISC) links. If the remote system is another CICS region, it is referred to as an application-owning region (AOR).

CICS supports two forms of transaction routing for remote transactions:

- *Static transaction routing* where the transaction is predefined with the name of the remote CICS region to which it is to be routed for execution.
- *Dynamic transaction routing* where one of the attributes on the transaction resource definition specifies DYNAMIC(YES), indicating that the transaction is to be routed dynamically. Alternatively, if the transaction is not defined CICS/ESA assumes that it is to be dynamically routed. The name of the remote destination is determined by a user-written dynamic transaction routing program at the time the transaction is invoked from a user′s terminal.

CICS/ESA Dynamic Transaction Routing in a CICSplex contains detailed information on this subject.

Using dynamic transaction routing across a CICSplex rather than static routing is fundamental to building a continuously available CICS system.

4.2.1 High Availability

Dynamic transaction routing can help you to provide high availability by enabling you to clone AORs. With multiple AORs available to choose from, the dynamic transaction routing program can bypass any failed or “sick” AOR. Faster restart is possible for smaller AORs, and fewer end users are affected by any single failure. All these factors contribute to:

- Reducing the number of failures seen by end users
- Reducing the length of an outage as seen by end users
- Reducing the incidence of “sympathy sickness” between CICS regions.

4.2.2 Continuous Operations

Dynamic transaction routing helps you to achieve continuous availability by allowing you to:

- Quiesce an AOR and direct transactions to a alternate AORs
- Remove an AOR once it has been quiesced
- Stop and restart an AOR to apply service changes to the AOR when no transactions are active
- Add an AOR clone and notify the dynamic transaction routing program that a new target AOR is available.

The capability provided by dynamic transaction routing enables you to add and remove AORs, and apply changes to AORs, in a way that is quite transparent to end users. The kind of changes you might want to make to an AOR, while still maintaining service to end users, could be to introduce modified CICS/ESA tables, to apply service to CICS/ESA, or apply service to application code.

4.2.3 CICSPlex SM

Implementing a CICSplex with dynamic transaction routing allows the CICS system to grow, but is not without its problems. For example, in a CICSplex you have to consider:

- Effective use of resources
- Multiple operation points
- Increased difficulty in keeping track of what's happening.

CICSPlex SM, introduced by IBM at the time CICS/ESA 4.1 was announced, resolves these problems and provides a dynamic transaction routing program.

CICSPlex SM is a systems management tool for CICSplex environments, providing services as described in C.1, "IBM CICSPlex System Manager for MVS/ESA" on page 161. In this book we reference the CICSPlex SM functions that contribute to increased availability:

- Workload management
- Availability monitoring
- Support for the automatic restart manager.

For more information on CICSPlex SM, you should read *CICSPlex SM Concepts and Planning*.

4.2.4 Roll Your Own

You can code your own dynamic routing program, trading off customization against cost. If you code your own, you should get exactly what you want, at a higher cost. The logic involved is very complex. You may also need to update the code when you move to a new release of CICS/ESA or MVS/ESA.

4.3 Data Sharing

Data sharing is the concept that applications, running in different CICS regions, IMS processing regions, or even as batch jobs, can share access to data no matter which MVS image in a sysplex they happen to be running on. In order to fully implement data sharing in a CICS/ESA workload management environment you have to implement the prerequisites, shown in Table 2 on page 41.

Table 2. CICS/ESA N-Way Data Sharing Prerequisites			
Prerequisite	**Data Environment**		
	IMS DBCTL	DB2	VSAM
MVS/ESA 5.1	X	X	
MVS/ESA 5.2	O	O	X
Coupling facility	X	X	X
CICS/ESA V4R1	X	X	
Future CICS Release	O	O	X
IMS DB 5.1	X		
IRLM 2.1	X	X	
DB2 V4R1		X	
DFSMS/MVS 1.3			X
Note: X=required, O=optional			

Ideally, all CICS/ESA AORs should be able to access all the data needed to run any transaction. If you have only one MVS system image, you should find it easy to achieve this. If you have two or more MVS system images in your sysplex, then you may have to implement specific levels of database or access method software to allow all AORs to access the same data at the same time, no matter which MVS system the AOR is running in.

By implementing data sharing between online and batch applications, you can eliminate the need for many of the scheduled outages that would otherwise be required to accommodate batch processing requirements. With IMS/ESA and DBCTL, you can use batch message-driven programs (BMPs) to achieve sharing between batch and online without the need for full data sharing.

Data base systems such as DB2 and IMS/ESA provide data sharing with integrity; this is partially true for VSAM files through DFSMS.

4.3.1 DB2

IBM DATABASE 2 Version 4 Release 1 provides full data sharing support for DB2 databases in a sysplex. IBM DATABASE 2 Version 4 Release 1 and the IMS Resource Lock Manager (IRLM) Version 2 Release 1 use the Cross System Coupling Facility (XCF) lock structures to hold information about locks on the database, and use the XCF signalling facility to invalidate database pages held in the storage of any other DB2 subsystems sharing access to a database when a row is updated. If you do not have DB2 Version 4 installed, and your CICSplex spans more than one MVS image, you need to use the workload separation facilities of CICSPlex SM to ensure that transactions are routed to AORs that can access the appropriate DB2 databases.

4.3.2 IMS/ESA

CICS/ESA supports two forms of access to IMS databases, DBCTL and local DL/I access.

DBCTL

DBCTL in IMS/ESA Version 5 Release 1 Database Manager (IMS/ESA DB) provides n-way data sharing support at block level within a sysplex. The DBCTL

subsystems in multiple MVS images can share access to the databases at the same time. IMS/ESA DB 5.1 and IRLM 2.1 use XCF lock structures to hold information about locks on the database, and use the XCF signalling facility to invalidate database blocks held in the storage of any other IMS/ESA DB 5.1 subsystems sharing access to a database when a segment is updated. The other IMS/ESA DB systems would then refresh their copy of the segment from disk when they next need to access it.

DBCTL in IMS/ESA DB 4.1 provides two-way data sharing support at block level within a sysplex. The DBCTL subsystems in two MVS images can share access to the databases at the same time. The IRLMs of the two MVS systems communicate using VTAM to invalidate database blocks held in the storage of the other IMS/ESA DB 4.1 subsystem sharing access to a database when a segment is updated. The other IMS/ESA DB system would then refresh its copy of the segment from disk when it next needs to access the segment.

Local DL/I

You can implement sharing of IMS databases between multiple AORs in a single MVS image by using a single data-owning region (DOR). The DOR controls access to the databases, and transactions in the AORs access the databases by function shipping their DL/I requests to the DOR. You can extend this support to allow sharing of IMS databases between multiple AORs over two MVS images using the block level sharing facilities of IMS Database Recovery Control (DBRC). You must implement the recovery control level of DBRC before you can use block level data sharing. Using block level sharing has significant implications for the CICS user. For example, all batch jobs that update the databases *must* produce a log; dealing with all these logs increases the complexity of operations.

Use DBCTL, not local DL/I

We strongly recommend that you use the DBCTL interface rather than try to implement block level sharing with local DL/I.

DBCTL provides a release-independent interface between CICS and IMS/ESA DB. If you implement DBCTL you avoid the requirement to regenerate CICS or IMS every time a new release of the other is installed.

Note: CICS/ESA 4.1 and IMS/ESA 4.1 are the last releases of these products that support local DL/I.

If a database is specified not to be shared, the IMS data sharing facility ensures that only one system at a time has exclusive use of a database. Non-sharing is a special case of database-level sharing, and is a significant database integrity enhancement, in that it can prevent a database from being accessed in case of errors, such as:

- After a batch DL/I ABEND but before database backout has been performed.
- After a CICS abnormal termination but before incomplete transactions have been backed-out by emergency restart processing.
- After a disk I/O error, but before the database has been recovered.
- After a database reorganization, but before an image copy has been made to establish a known recovery point.

Batch processing

With IMS/ESA, you can have concurrent access of databases from Batch Message Processing programs (BMPs) and from CICS/ESA applications. Separate sub-system address spaces for CICS and the IMS database manager, and multiple DL/I thread TCBs provide isolation and enable multi-processor utilization. This is independent of data sharing and is a good solution if all your workload executes on a single MVS/ESA image.

4.3.3 VSAM

IBM has stated that it intends to provide record level sharing (RLS) for CICS VSAM files. The full text of the announcement is:

> *To facilitate customer planning, IBM intends to enhance CICS/ESA to provide VSAM Data Sharing between CICS regions. This will include VSAM record level sharing (RLS) between multiple CICS systems and centralized sysplex-wide log and journal management. Support for the underlying facility will be provided by a future MVS/ESA release. Further parallel sysplex support will provide for temporary storage data sharing between CICS systems in the same or different MVS images.*
>
> *IBM may change its product plans in the future for business or technical reasons.*

Multiple CICS/ESA applications in one MVS/ESA image can share VSAM data sets with integrity by using function shipping to a single FOR (see 4.1, “The CICSplex” on page 37). This approach has limitations. It does not solve the problem of sharing data sets between CICS/ESA and batch, and the performance overhead of function shipping can be high.

VSAM record level sharing (RLS) is a function provided by DFSMS/MVS 1.3. IBM has announced its intent to support RLS in a future release of CICS/ESA. RLS is designed to allow VSAM data to be shared, with full update integrity, between many applications running in many CICS regions in one or more MVS system image in a sysplex.

If your CICSplex includes AORs on multiple MVS images, and you want CICSPlex SM to manage the workload across these AORs, you need to implement a temporary solution until the RLS support is available. You could use either of the following as a temporary solution:

Function ship to a single FOR. The FOR would access your VSAM data sets on behalf of all the AORs. If you implement this solution, we strongly recommend that you use CICS/ESA 4.1 with MVS/ESA 5.1 or MVS/ESA 5.2, so that you can take advantage of the multiregion operation (MRO) XCF support available with these releases of software. You may find that you cannot implement this solution because it would cause your FOR to become processor constrained. Most of the work in the FOR takes place under the CICS quasi-reentrant TCB, so it is possible for a single FOR to become processor constrained, even when there is spare processor capacity on your CPC. This may force you to use multiple FORs.

Split the files between multiple FORs. You must ensure that your transactions update files in only one FOR in any single logical unit of work (LUW). Your transactions can send read requests to any number of FORs in one LUW. If you split the files across multiple FORs you either have to allow all AORs to access all FORs, or otherwise group your files by functional areas (for example, payroll or accounting), and then use CICSPlex SM

workload separation to route transactions through to AORs that are connected to the appropriate FORs.

You might be able to reduce the function shipping overhead of using FORs considerably by using the shared data tables (SDT) feature of CICS/ESA. You benefit most from using SDTs if all (or at least, most) of your AORs accessing the SDT are on the same MVS image. The path length for a request to an SDT is significantly lower than that needed to read a record using function shipping. Locking in SDTs is at the record level, whereas locking in conventional VSAM is done at the control interval level. This means less contention on the access to data during I/O.

With SDT, read requests from the local AORs are executed through MVS cross-memory services instead of function shipping them to a mirror transaction through CICS MRO connections. For remote AORs, the requests would be function shipped through MRO/XCF connections (for CICS/ESA 4.1 or later with MVS/ESA 5.1 or later) or using ISC (with earlier levels of CICS or MVS). In most cases SDTs give a better response time to transactions that are routed to the AORs running in the same MVS as the FOR. The CICSPlex SM dynamic routing program is able to detect this difference, and routes transactions to these AORs.

Chapter 5. Operations, Procedures, and Systems Management

In this chapter we describe how you can manage your systems for continuous availability. We discuss:

- Managing changes
- Avoiding errors
- Removing stress points
- Removing single points of unavailability.

5.1 Managing Changes

Change management is the process of recording and controlling changes to the system. Recording is performed by the person or persons wishing to make the change; controlling is performed by the person or persons responsible for system availability. The key goals are to keep a record of changes and to maintain central control so that changes are not made indiscriminately.

Without well planned and managed change control, a system can become a house of cards, resulting in extended outages in order to track down problems and apply all necessary fixes. Change control is the process of recording and controlling changes to the system.

Effective change control can prevent your system from being corrupted and reduce the time taken to identify the problem and recover should this occur.

You may want to consider running different CICS releases on the same MVS system image, adjusting for daylight savings time, and introducing new versions of application programs.

5.1.1 Running Different CICS Releases on the Same MVS System Image

The MVS link pack area (LPA) comprises several areas, both above and below the 16MB line. We use the term LPA to refer to the pageable link pack areas (PLPAs) above and below the line, into which modules are normally installed to be used from the MVS link pack area.

Note: The MVS link pack area has both pageable and fixed parts. Although you can install CICS modules into the fixed parts, we recommend that you use the pageable areas for performance reasons.

The LPAs below the 16MB line are specifically referred to by the term LPA, and those areas above the 16MB line are referred to by the term ELPA.

If you install a module into the LPA or ELPA, that module will not be used from the MVS link pack area until you re-IPL your MVS system. However, you can use the MVS modified link pack area (MLPA) to provide a temporary extension to the PLPA, existing only for the life of the current IPL. You can use this area to add or replace LPA eligible modules without having to recreate the MVS link pack area.

You could make use of the MLPA when adding a new higher release of CICS to your MVS image. By adding the LPA eligible modules for the new higher release to the MLPA you avoid the need for a scheduled outage for this purpose.

If you plan to run different releases of CICS on the same MVS system image you have to make a decision on which levels of CICS SVCs and eligible CICS modules to place in the LPA. The wrong decision may lead to an unscheduled period of unavailability.

We recommend:

- *CICS SVCs*. Use the latest level of the CICS SVC. This is downward compatible with all earlier releases. The High Performance Option (HPO) SVC is unchanged between releases.
- *LPA only modules*. Some modules, such as DFHIRP, must reside in the LPA. These modules are downward compatible, so use the latest level.
- *LPA eligible modules*. Some modules are eligible for the LPA but may also reside in the CICS private area. The level of these modules in the LPA should be for the most widely used version of CICS. Other systems with a different level of CICS can use the LPA=NO system initialization parameter. This prevents CICS from using the version of these modules in the LPA.

For more information see the *CICS/ESA Installation Guide*.

5.1.2 Adjusting for Daylight Savings Time

Some countries change their local time twice a year to gain advantage of the longer daylight times during summer.

The system logs for CICS, DB2, and IMS contain a time stamp for recovery purposes. This introduces a potential problem. Moving the clock forward at the start of summer is not a major problem, your logs may have a gap of an hour when nothing appears to happen but their integrity is retained. During the time adjustment at the end of summer, the local time is adjusted to one hour earlier. If the system starts logging immediately the logs will contain overlapping timestamps.

Two possible solutions are:

Do not change the system to reflect summer time.

This is a good solution for availability but probably unacceptable to your users.

Shut the system down for an hour.

This is not acceptable if you require continuous availability but may be your only realistic option. You could use this as an opportunity for planned maintenance.

5.1.3 Introducing New Versions of Application Programs

For minor changes, such as a new copy of a single program, use the master terminal NEWCOPY function. CICSPlex SM single point of control allows you to issue NEWCOPY in all CICS systems in a defined group with one command.

You can use the external CICS interface (EXCI) to issue the NEWCOPY command under program control immediately after a successful translate, compile, and

link-edit, as shown in Figure 9 on page 47. This makes the successfully updated program available for use immediately.

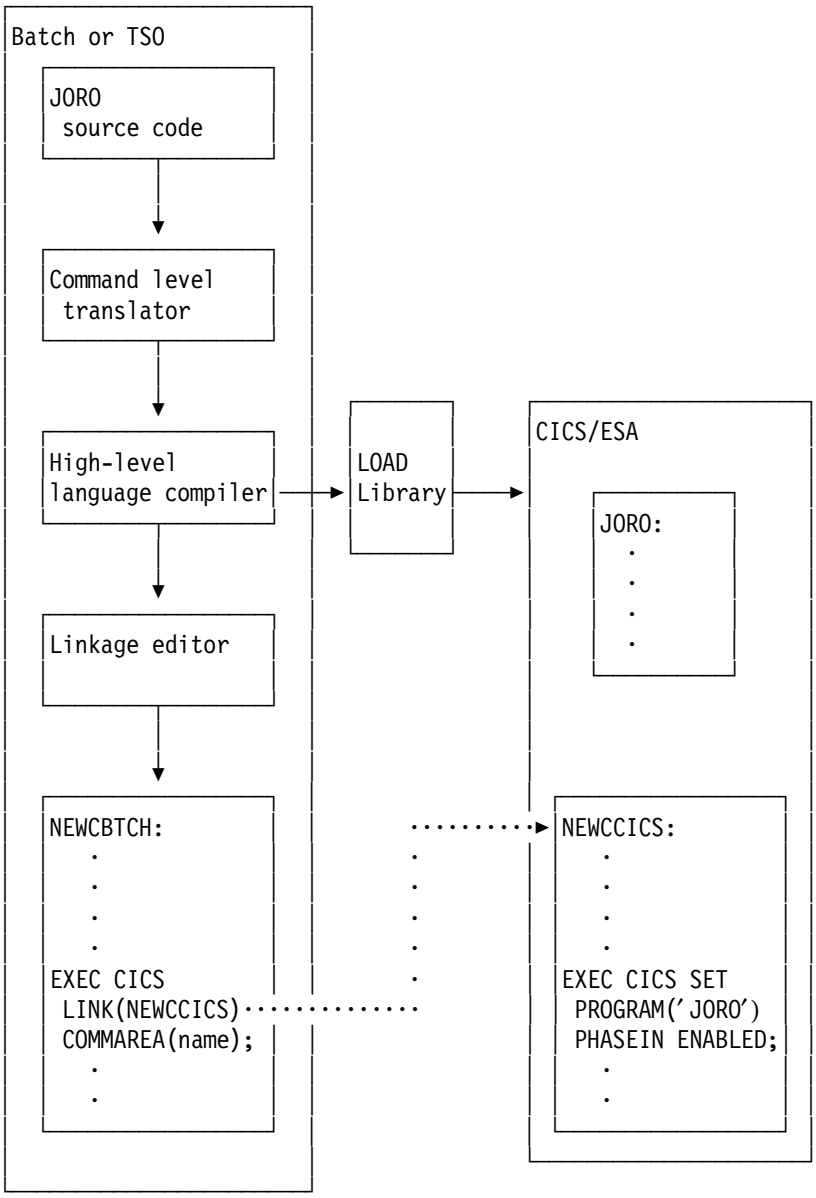


Figure 9. Preparing and Replacing an Application Program

In the example shown in Figure 9, we have added a step following the link-edit. This step executes program NEWCBTCH which picks up the name of the link-edited program, places it in a COMMAREA, and then uses the EXCI to link to program NEWCCICS in the CICS/ESA region where JORO is being executed. NEWCCICS uses the CICS/ESA system programming interface to refresh (PHASEIN) the existing copy of JORO.

5.2 Avoiding Errors

You must have a maintenance strategy, sufficient testing, operating procedures (especially procedures for change and problem management), and use automation as much as possible if your are to avoid errors as a source of unavailability.

5.2.1 Maintenance Strategy

Applying maintenance to any resource makes it unavailable for a certain period of time. If you have enough parallel or duplicated resources, unavailability of one component due to maintenance need not result in an unavailability of the system as a whole. If you cannot avoid scheduling an outage, your objective must be to minimize the duration and impact of the outage.

Maintenance is either corrective or preventive. Corrective maintenance is applied to resolve problems as they arise. However with preventive maintenance you have to decide how much to apply and when to apply it. Your options for applying preventive maintenance are discussed in section 5.2.1.1, “Preventive Maintenance Strategies” on page 50.

A planned outage is usually necessary to apply CICS maintenance. You should research your change in advance to determine what other associated actions you will have to take such as applying prerequisite PTFs and reassembling CICS table definitions. If you do this research, the duration of the planned outage can be kept to a minimum.

As part of your research, you should check the effect of these changes on software you have bought from non-IBM vendors.

In a CICSplex planned maintenance can usually be undertaken without interrupting service to end users, unless the maintenance needs to be applied to an FOR. For more information see Chapter 3, “Our Recommendations” on page 23.

The two main types of software maintenance are:

Authorized Program Analysis Report
: An Authorized Program Analysis Report (APAR) temporary fix is maintenance to *correct* or *circumvent* a current problem which is being investigated by the change team. APAR temporary fixes have not been evaluated and tested as extensively as other types of maintenance and may be changed significantly before the final version is incorporated into a Program Temporary Fix (PTF).

Program Temporary Fix
: A PTF is the final version of the APAR temporary fix which has successfully completed all evaluation and testing. One or many APARs can be incorporated into a single PTF. In addition, small enhancements to the system can be shipped as PTFs. Such an enhancement is known as a small programming enhancement (SPE).

APAR temporary fixes are normally obtained from a Support Center on an individual basis. PTFs can also be obtained individually from a support center. Other common sources are a Data Link Library (DLL) and ServiceLink.

Individual maintenance (APARs or single PTFs) should only be used to circumvent a current problem, or to maintain prerequisite levels required by other software products.

PTFs are normally obtained in packages. Some of the options are:

Replacement package options

- Custom Built Installation Productivity Option (CBIPO)

  The CBIPO provides a complete system replacement which incorporates all the product maintenance.

- SystemPac/MVS

  SystemPac is a software package for creating or replacing an MVS system or subsystem (such as CICS).

Upgrade package options

- Custom Built Product Delivery Option (CBPDO)

  CBPDO has the facility to collect all service since a specified level in a single source.

- Program Update Tape (PUT)

  This distributes all newly available service to customers on a regular basis (approximately once a month).

  Note: PUTs may have been withdrawn in some countries.

- ProductPac/MVS

  ProductPac is a software package for upgrading the product and service content of an existing system.

- ServicePac/MVS

  ServicePac is a software package for upgrading the service content of an existing system.

CBIPO, CBPDO, and PUT tapes are traditional methods for implementing change. SystemPac, ProductPac and ServicePac, however, are new offerings that are collectively known as CustomPacs. CBIPOs and CBPDOs differ from CustomPac offerings mainly in the extent of customization.

CBIPOs and CBPDOs are tailored to an order checklist, a specified service level, and a customer profile, but not to the customer's actual system.

CustomPac offerings are tailored to an order checklist, as well as to customization data provided about the customer's actual system.

With the CustomPac offerings service is automatically provided in follow-on packages.

For more information on these maintenance packages see *MVS Software Manufacturing Offerings*.

5.2.1.1 Preventive Maintenance Strategies

With so many types of maintenance and delivery options, it is possible to adopt any one of a number of maintenance strategies. Each has benefits, but also drawbacks. Some of those options are outlined below:

- **Keep software up to the latest available maintenance level**
  - This strategy results in the highest frequency of planned outages.
  - This strategy has a high potential to reduce unplanned outages since the largest possible amount of preventive service is applied.
  - There is a small risk of a new PTF itself causing errors. To reduce this risk, higher levels of testing may be required than for the PTF package options.
  - The effort required to resolve PTF prerequisite chains is low.
- **Use an upgrade package**
  - The frequency of planned outages is lower than for the previous option.
  - Unplanned outages are reduced from a no-maintenance strategy.
  - The probability of PTFs causing unscheduled outages is reduced compared to the first option because the PTFs have been tried by other customers.
  - More research is required to resolve PTF prerequisite chains than for the first option.
- **Use a replacement package**
  - The disruption of system testing and the amount of time for scheduled outages is more than for the upgrade package options.
  - The system is current up to the level of the replacement package.
- **Defer all preventive maintenance**
  - There are no planned outages.
  - There may be unplanned outages. These outages may increase in frequency as the transaction volumes increase, or as new applications exploit specific (new) functions.
  - If corrective maintenance has to be applied, the work required to resolve PTF prerequisite chains will be considerable.
  - Outages may be forced by the requirements of other products (for example, VTAM) to have prerequisite service applied.
  - The time taken to resolve an outage may be extended due to the additional work in rediscovering old errors.

Recommendation

Apply preventive maintenance regularly using one of the CustomPac offerings.

Planned outages to apply the preventive service reduce unplanned outages occurring as a result of software errors. A planned outage can be controlled, both in timing and duration, an unplanned outage cannot.

It is possible to have fall-back strategies (like removing the PTFs) for an error resulting from the application of preventive maintenance. There is no place to fall back to if an unplanned outage occurs, because the reason for the outage is not as clear.

We recommend the CustomPac offerings, since these offer the best, most efficient vehicle for installing systems, products, or service.

5.2.2 Testing

With the complexity of commercial information technology systems it is unrealistic to expect new or changed applications to always work without error. You need to test your applications as they are developed and introduced into your business to identify and remove problems before they can threaten the availability of your systems.

There are three general categories of testing you should consider:

- Functional testing
- Regression testing
- Performance testing.

TeleProcessing Network Simulator (TPNS) can help you with all three types of testing. It is discussed in section 5.2.2.4, “Testing with the TeleProcessing Network Simulator” on page 53.

5.2.2.1 Functional Testing

Functional testing refers to testing a new program or subsystem (or to testing a new component of an existing program or subsystem) to ensure that it performs according to specification. You should prepare test cases for each new function to make sure that the new code does the job it was designed to do. These test cases should test all possibilities, including user errors.

If the application developer is doing the functional testing there can be a tendency to assume that certain areas of code are error free and not test them. Testing is there to catch the unforeseen and unexpected and you should avoid such assumptions.

Three CICS supplied transactions, which assist application development and functional testing, are CEDF, CEBR, and CECI; for more information on these transactions see *CICS/ESA CICS-Supplied Transactions*.

CECI interacts with your test system to allow you to create or delete test data, manipulate temporary storage queues, or deliberately introduce wrong data to test out error logic. You can also use CECI to repair corrupted database records on your production system.

If you can not determine why your application is failing CICS trace and dump facilities may help. For more information see the *CICS/ESA Problem Determination Guide*.

Your functional testing should include any changes to procedures, documentation, and system settings. This will help you implement any necessary changes as you install the new program or subsystem into your system.

5.2.2.2 Regression Testing

Regression testing refers to testing a modified program or subsystem to ensure that the change has no effect on the unmodified portion. For example, do the parts of the program that were not changed still work the same? In regression testing, a standard set of test cases is run against the program or subsystem before and after it has been modified. Each test case must produce the same results in each case.

You should aim to incorporate new test cases from functional testing into your regression test set as they become available.

A prime consideration in testing is repeatability. The ability to repeat a test case ensures that situations that caused errors can be duplicated, a condition often impossible when people carry out testing at terminals. One approach to testing is the use of a simulator or driver, such as TPNS or Workstation Interactive Test Tool (WITT). See section 5.2.2.4, “Testing with the TeleProcessing Network Simulator” on page 53 and the *Workstation Interactive Test Tool Users Guide* for further information. A driver is a program, or programs, that simulates the actions of a number of applications, terminals, or terminal operators. These applications, terminals, or terminal operators access programs and data within another computer or within another region of the same computer. The driver collects all information returned from the driven system and records that information for further analysis.

The driver you use and the test cases you write for it must be very reliable so that any errors are only attributable to the program under test.

If you use a driver for regression testing one of your main problems is likely to be analyzing the vast amount of data generated by the test to determine if there were any errors. As far as possible you should design your test cases to be self-checking so that unexpected results are reported to an exception log. If you have to compare large output logs, there are utilities that may help you; the TPNS log compare utility is one example (see *TPNS General Utilities*).

5.2.2.3 Stress Testing

Stress testing (or performance testing) refers to testing the system to ensure that it can handle the projected throughput with acceptable response times. Performance testing ensures that the appropriate number of terminals can be supported or that the application programs can handle the anticipated transaction loads. You can change parameters, such as the size and number of buffer pools, during performance testing to see what effect these changes have on the system. This is known as tuning the system. For a realistic performance test a driver which can simulate large numbers of terminals is required, such as TPNS.

Performance testing can be used to develop your systems so that they will be able to cope in times of high stress, preventing unscheduled outages which might otherwise result.

As with regression testing you need a way to interpret your results. The TPNS response time utility provides an example of what can be done; it takes the TPNS log from the test run as input and outputs a response time report, which may be in a graphical format. For more information see *TPNS General Utilities*.

For more information on testing see the *CICS/ESA Application Programming Guide*.

5.2.2.4 Testing with the TeleProcessing Network Simulator

The TeleProcessing Network Simulator (TPNS) is a terminal and network simulation tool. You can use TPNS to determine system performance and response time, to evaluate network design, to perform functional testing, and to automate regression testing.

TPNS controls the generation of message traffic to a teleprocessing subsystem or application, which eliminates the need for large amounts of terminal hardware and terminal operator time. You can use TPNS to perform tests that evaluate the reliability and approximate performance characteristics under expected operating conditions. TPNS can simulate a user-specified network of terminals and the associated messages and allows you to alter network conditions and message loads during a test run.

TPNS is a powerful tool for functional testing which can simulate an extensive range of device types and complicated message flow scenarios. TPNS can also be used to reproduce error situations resulting from faulty hardware or unusual events.

TPNS can perform load tests of CICS, IMS, VTAM, VSAM and MVS. This includes:

- Determining the ability of the software to operate without failure.
- Testing system performance.

  Examples of tests in this area include the ability of a certain processor to process anticipated transaction loads at the required response time, and within acceptable limits of CPC utilization.
- Measuring internal resource utilization.

  Examples are the load on IRC links for function shipping, or, the usage of CICS DSA at specific transaction volumes.
- Providing a vehicle for testing much of the online operating and recovery procedures.

A large range of utilities are available to assist with automatic script generation, test run analysis, and other processes. For more information about TPNS refer to *TPNS General Information*.

Test Case Design Considerations

- All applications or a subset?

  If some of the applications are not suitable for TPNS testing, then the installation must consider whether running with a subset of the applications and adjusting the volumes correspondingly is adequate.
- All terminals or a subset?

  It may be possible to simulate a suitable load through a smaller number of terminals than are used in the network by specifying a high transaction rate per terminal.

- All databases and data sets, or a subset?

  One problem in using TPNS is that the user must duplicate his operational data bases in the TPNS system. It may be possible to use a subset of the data, with some slight application changes.

  If large databases are compressed onto fewer actual disks, the device utilization may become a bottleneck during testing.

TPNS Configuration: TPNS can be installed in several different configurations. The configuration of the network that contains the resources you want to simulate (known as the logical configuration) determines the configuration of the system you use to run TPNS (known as the physical configuration).

VTAMAPPL is the simplest and most popular physical configuration. You use this configuration to simulate primary and secondary logical units in the same subarea as VTAM. These logical units can have a session with any other logical unit that VTAM will allow a session to be started with. In VTAMAPPL configuration TPNS will simulate all available LU types from LU0 to LU6.2 parallel sessions. This range includes 3270 terminals, pool pipeline terminals, printers, client-server applications and even another CICS region connected by LU6.2 parallel sessions to the test region.

VTAMAPPL is the easiest configuration to use since you do not need a communication controller. Other configurations give further features but require physical hardware. For example, the link attach configuration requires its own dedicated communication controller running a TPNS version of the network control program. TPNS in the VTAMAPPL configuration has been used successfully in many installations for functional, regression and performance testing.

Implementation Requirements: You need the CPC capacity in the test environment to run TPNS at the same time as the system being tested. You must have enough processor capacity to run TPNS itself as well as your CICS test region(s) and associated programs. In a performance test where TPNS and your test region(s) are running on the same CPC, TPNS itself may take up to half the CPC capacity. You should avoid sharing CPC capacity between your production and test systems. If you do, the performance of your production system may become unacceptable.

You require sufficient disk space to reproduce the application databases and data sets.

Conclusion: Each release of TPNS has provided new useability enhancements that make it easier for you to quickly and effectively establish your TPNS test environment. The automatic script generation utilities allow you to create TPNS scripts by capturing keystrokes entered at a real terminal, or by analyzing traces of system activity. Using these new features makes it possible to establish your test environment with little or no knowledge of the TPNS scripting language. TPNS provides a comprehensive range of features and functions which can help you in many areas of testing. TPNS is uniquely suited to performance testing because of it′s ability to simulate thousands of terminals and to run external to the system under test.

IBM offers installation, training, implementation and consultancy services for TPNS to help you implement your TPNS test solution.

5.2.3 Procedures

Your written procedures should be the definitive source of knowledge in your organization. They should detail the *correct* way things are done and make the collective knowledge and experience in your organization available to all. A good set of procedures can improve availability by reducing human errors and giving all your staff access to the knowledge they need to do their jobs well.

For your procedures to be of real benefit, they must be accurate, up-to-date and comprehensive, otherwise they will not become a trusted source of knowledge. The only way to achieve this is to reference and update your procedures on a day-to-day basis. Your staff must learn to rely on the information they contain as the definitive source of knowledge on how to do their jobs. Errors and omissions should be reported as they are found.

To make this possible you need one definitive set of procedures readily accessible to all. These are best maintained in softcopy format. However you should have hardcopy versions of your recovery procedures because the online procedures will be unavailable if you lose your systems.

The operational procedures recommended here are what we consider the minimum necessary to be used in the daily operation of your CICS system. You should customize them to your environment, and supplement them with other material necessary to run your CICS system. They include:

- Startup and shutdown procedures
- CICS online sessions log and incident report
- Transaction tables
- Change and problem management.

The *CICS/ESA Operations and Utilities Guide* for CICS/ESA 4.1 has very detailed information about procedures for starting and stopping CICS. The *CICS/ESA Operations Guide* for CICS/ESA 3.3 has information on the other procedures, and suggested forms layout.

5.2.3.1 Startup, Shutdown and Error Procedures

Your operators should have access to online, and hard copies of each of these procedures. They should be reviewed regularly, and include updates from your system testing, when appropriate. Clear and easy-to-follow procedures can increase your availability. Don′t forget they are often referred to at times of high stress.

5.2.3.2 Online Sessions Log and Incident Report

These forms, usually completed by your operations staff, provide valuable data to different people. Your system programmers, or application programmers can be notified of problems from the incident report, and use the session log to gather background data. Change control can check the status of installs from the session log, and problem control can be alerted from the incident report.

Since these forms, electronic or hard copy, are used by different groups for different purposes, you should ensure that the data collected is complete. You should also ensure that all data collected is required, and if not, update your forms.

5.2.3.3 Transaction Tables

Your data set and database-to-transaction table should have all your data set and databases mapped against all your transactions.

Your program-to-transaction table should have all your programs mapped to your transactions.

This enables your operations staff to quickly determine what action should be taken if problems occur. For example, if a database is unavailable, you may need to disable the transactions that access it. Or if you have a copy of a database you don't want updated, you can disable the transactions that update that database.

Any application changes that require changes to these tables should be included as part of the deliverables to production.

5.2.3.4 Change and Problem Management

Change and problem management in this quickly changing and complex environment is a key to your overall system management, including CICS availability. As you increase the number of components in a CICSplex, you reduce the impact of any single failure. However, since changes may need to be cascaded through the system, knowing the state of a change is also more complex. Determining why you have a problem in one region and not in another can tie problem and change management together.

Change management is essential for continuous availability. You must be able to plan and track all changes that you make to the system. This can also help you reduce the number of planned outages by consolidating changes to various systems. Tracking changes is also an important step. You must be able to determine which changes apply to all systems or regions, and which are made only to specific systems or regions. We also discuss change management in section 5.1, "Managing Changes" on page 45. Problem management is the process of recording, tracking and resolving any problem which has the potential to impact the end user. The problem may be the hardware, software, operational or environmental in origin. Understanding the cause, impact, and frequency of outages is critical to ensuring that the proper resources are expended to correct the problem. If current problem tracking procedures are inadequate, new and better procedures must be implemented. Problems must be defined and listed according to their impact. Problem management must be well organized so that all the appropriate personnel are involved early in the cycle to diagnose, debug, document and deliver solutions.

Problem management must be well organized so that all the appropriate personnel are involved early in the cycle to diagnose, debug, document and deliver solutions. Frequent (daily) meetings for general information exchange and less frequent (weekly) meetings for planning future changes are necessary between representatives of operations, systems programming, applications, and end users. Management attendance and support of these planning sessions will help to emphasize the importance of systems management in providing extended availability to all users of the system.

Problem management is in place to help you track, report and resolve all problems. This gives you the ability to build a database of problems and solutions to your problems. You can often reduce an outage by knowing exactly what changes have been made, and what the backout plan is, and if they happen

again, may reduce resolution time. This will also help you highlight repeated problems.

You can also refer to *Systems Analysis for High Availability, An Availability Management Technique* for further discussion of change and problem management.

5.2.4 Automation

Automation offers improved availability, increased operator productivity, and simplified growth. CICS/ESA automation is an important component of your enterprise automation strategy. Automatic operations result in an immediate and consistent response to situations that arise, reducing unscheduled outages caused by operator error. Before you can automate your operations, however, you must have documented procedures that describe what the required action is in a particular circumstance.

IBM provides several products that you can use to automate operations of your CICS regions and of your CICSplex. We give a brief overview of them here. For more information, and some examples of how to use these products to automate your systems, refer to *Automating CICS/ESA Operations With CICSPlex SM and NetView*.

5.2.4.1 NetView

NetView is the foundation of IBM′s strategy for providing automation and network management and is a component of the SystemView structure on the MVS/ESA platform. Product information can be found in Section C.3, “NetView for MVS/ESA” on page 163. NetView automation may be based on messages as well as management service units (alerts or any other data, for example SNA architected data). NetView supports the following features for automation:

- Connection to the Sub-System Interface (SSI) for MVS/ESA for the retrieval of all messages
- Message Automation table, where messages, alerts or Management Service Units (MSUs) are examined against set criteria and if a match occurs can then:
  - Issue any command (for example, MVS, JES2, or VTAM)
  - Invoke a command list
  - Invoke a command processor
  - Suppress the message
  - Display the message
  - Highlight the message
  - Sound the beeper on the screen
  - Turn on an important-message-waiting indicator
  - Log the message (System log and/or NetView log)
  - Copy the message to any number of operators
  - All or any combination of the above
- Messages can also come from Terminal Access Facility (TAF) operator control sessions, allowing the automation of subsystems such as CICS and IMS.
- Automation Tasks, which allow the execution of automation without the necessity for real operators to log on. The workload can be split between any number of automation tasks, and allow the automation functions to use the multi-tasking capabilities of NetView.

NetView′s automation facilities provide base functions to a wide variety of applications that provide sophisticated “drop in,” “ready to run” system and network automation, such as Automated Operations Control/MVS (AOC/MVS) for MVS system and subsystem automation.

5.2.4.2 Automated Operations Control/MVS

The Automated Operations Control/MVS (AOC/MVS) program, a NetView application, automates system console operator tasks normally handled by a console operator. These tasks include the start-up, monitoring, recovery and shutdown of MVS subsystems, components, and applications including VTAM, Resource Measurement Facility (RMF), JES2, JES3 and Time Sharing Option (TSO). In addition, AOC/MVS provides facilities to automate operator console messages, initiate timer-based actions, and prevent critical MVS resource shortages.

Through integration with NetView, AOC/MVS delivers a graphical operator interface built on an OS/2 workstation. The operator sees the entire enterprise represented as icons.

Operators and systems programmers can off-load many critical control activities to AOC/MVS applications and services. AOC/MVS automation allows console messages to be identified and commands to be issued automatically at computer speed. As a result, AOC/MVS can react to a potential problem faster than a human operator can, minimizing the impact of a situation on end users. Since AOC/MVS reacts immediately and consistently to situations, it can reduce the number of unscheduled outages caused by operators overlooking critical messages or taking inappropriate actions.

AOC/MVS provides three additional (optional) applications to perform specific subsystem automation and further console consolidation. These are CICS Automation Option, IMS Automation Option and the OPC Automation Option.

To optimize CICS/ESA system availability, AOC CICS Automation (CICSAO) provides:

- Automated startup and shutdown activities
- Interfaces to other software components
- Recovery from CICS ABEND situations
- Active and passive monitoring functions
- Timers and external triggers that help you to adhere to service level standards and agreements.

CICSAO tracks startup processing to verify timely completion of the CICS startup process. In addition, CICSAO notifies operators when the CICS startup process exceeds predefined startup processing standards. Should an application require several CICS regions to be active, they may be grouped under one CICSAO group name. Every region defined in a group is started or shut down when the group name is specified for the startup or shutdown request.

Operators are notified when intervention is required for critical recovery situations or when program application failures exceed predefined error thresholds. CICSAO is also designed to provide help with problem determination, and recovery for severe CICS processing failures.

You can find additional information in *SystemView: AOC/MVS CICS Automation General Information*.

5.2.4.3 CICSPlex SM

CICSPlex SM is part of the growing trend towards automation, not only of routine system-management tasks, but also of exceptional situations. For example, CICSPlex SM′s real-time analysis functions can issue SNA generic alerts to NetView, thereby allowing NetView to take corrective action before problems become apparent to end users. External messages, which are directed to the console by default, can be intercepted for automatic handling by other products. For example, they may be intercepted and processed by AOC/MVS CICS Automation.

CICSPlex SM and CICSAO are complementary products, each having functions that the other does not. You can obtain additional benefits by using them together, taking advantage of:

- Alerts and messages from CICSPlex SM to drive CICSAO automation actions
- CICSAO′s ability to dynamically start a new CICS/ESA address space
- CICSAO as a bridge, allowing CICS/ESA resource status to be coordinated and reported with the status of resources from other managed subsystems
- CICSAO′s ability to use the CICSPlex SM 1.2 API to issue commands to multiple CICS/ESA systems by means of a single request to CICSPlex SM.

For more information on CICSPlex SM see section 3.4.4, “CICSPlex SM” on page 34.

5.2.4.4 Automatic Restart Manager

The Automatic restart manager (ARM), introduced in MVS/ESA 5.2, is a recovery function that provides improved availability for batch jobs and started tasks by automatically restarting them after they unexpectedly terminate. ARM also provides a benefit in system failure scenarios by restarting registered clients on another system in the sysplex.

CICS/ESA 4.1 contains code to:

- Register with ARM when CICS starts
- Tell ARM when CICS is ready to accept work
- Tell ARM when CICS is deliberately shutting down
- Enable you to specify on “shutdown immediate” that you want to shut down without automatic restart
- Change the link to DB2, DBCTL, and VTAM to avoid the need for human intervention. This is done through timer driven retries.

CICSPlex SM 1.1.1, through APAR PN65642, can initiate an ARM restart of a CICS/ESA 4.1 (or higher) region being managed by a CMAS in the same MVS/ESA 5.2 system image by either using the ARMrestart action of the CICSRGN family of views or by using an RTA EVENT associated with an action definition (ACTNDEF) in which the ARM restart option was specified.

You should never make your CMAS eligible for automatic restart because, regardless of the option specified, a restarted CICS/ESA always performs an emergency start, and a CMAS must be started COLD. For more information see section 3.4.4, “CICSPlex SM” on page 34.

5.3 Removing Stress Points

With effective setting and monitoring of thresholds, you can usually get early warning that a system is becoming stressed. This allows you to take corrective action, either dynamically or by planning a scheduled period of unavailability, thus preventing an unscheduled period of unavailability.

There are several tools that you can use to get early warning of stress in a CICS region.

5.3.1 CICSPlex SM Real Time Analysis

CICSPlex SM's Real Time Analysis (RTA) functions provide automatic, external notification of unusual or otherwise interesting conditions.

System Availability Monitoring (SAM) can cause notification to be issued when a CICS system becomes unavailable during its normal hours of operation. It can also be used to notify the CICSPlex SM operator when any of the following conditions occurs in an active CICS system:

- CICS DSA short-on-storage
- CICS system dump
- CICS transaction dump
- Maximum tasks
- System stalled, or likely to stall.

SAM causes both CICS/ESA event notifications and external messages to be generated by default. All you have to do to implement SAM is specify the usual hours of operation of the set of monitored CICS systems. Thus, SAM can be of benefit as soon as CICSPlex SM is installed and configured.

CICSPlex SM RTA resource monitoring enables CICS resources, other than the CICS/ESA system itself, to be evaluated against a predefined status requirement. RTA resource monitoring can be applied to both a group of CICS systems and to an individual CICS system. Analysis point monitoring (APM) is RTA resource monitoring applied to a group of CICS systems, which is the CICSplex by default. For example, the status of particular files used by a group of CICS systems can be reported in a single external notification. MAS (managed address space) resource monitoring (MRM) is RTA resource monitoring applied to a single CICS system.

The real-time analysis functions of CICSPlex SM are described in greater detail in *CICSPlex SM Concepts and Planning*.

5.3.2 Performance Monitoring

If you have a CICS/ESA performance monitoring tool installed, you can use it to report on out-of-line conditions, as defined by you, when they occur. Manual or automated intervention can then be used to correct the condition before it results in an outage.

5.3.3 History Reporting

Reporting on availability requires a complex analysis of data from many areas because users depend on the availability of multiple types of resources:

- Central site hardware and the operating system environment in which the CICS region runs
- Network hardware, such as communication controllers, teleprocessing lines, and terminals through which users access the CICS region
- CICS region
- Application programs and data. Application programs can be distributed among several CICS regions.

Products that can provide this reporting include Service Level Reporter (see appendix C.5, “IBM Service Level Reporter for MVS” on page 164) and Enterprise Performance Data Manager/MVS (see appendix C.4, “Enterprise Performance Data Manager/MVS” on page 163).

For CICS/ESA availability reporting, you should produce and review daily an exception report that highlights any instances where a CICS/ESA application was unavailable in terms of your service level agreements.

5.4 Removing Single Points of Unavailability

Batch processing is one of the biggest and most serious causes of single points of unavailability. Batch processing has traditionally been carried out over night. The CICS regions are taken down or the data is made unavailable to the CICS regions. The overnight time period was often a period with no online access, so this was not a problem. As businesses change and spread across time zones, there is pressure to reduce the period where the data is unavailable. Overnight processing, be it updates, database copies or reorganization, has to be done, but we must find new and more efficient ways of doing it.

The impact of not being able to access a data set or database because it is still being used by a batch process can vary between unavailability of part of the system (one database used by one application is unavailable), through unavailability of all applications (for example, you cannot start your FOR because a vital VSAM data set is still being processed).

We discuss DB2 processing considerations in Chapter 8, “DB2 Considerations” on page 89, IMS processing considerations in Chapter 9, “IMS/ESA DB Considerations” on page 99, and VSAM processing considerations in Chapter 10, “VSAM Considerations” on page 109.

Chapter 6. Application Design

In this chapter we discuss application development issues which affect the availability of your system, specifically:

- Dealing with errors
- Avoiding stress points
- Avoiding single points of unavailability by:
  - Avoiding transaction affinities
  - Removing the requirement for immediate access to another system component
  - Avoiding any requirement for exclusive use of resources
  - Using the external CICS interface (EXCI)

6.1 Dealing with Errors

In this section we describe ways you can minimize the impact of errors in your system to retain maximum availability.

6.1.1 Exceptional Conditions

A CICS exceptional condition is, in brief, a reason why a CICS command cannot be completed normally. Exceptional conditions may occur at any time during the execution of a command and, unless you specify otherwise in your application program, a standard system (default) action for each condition will be taken. Usually, this default action is to terminate the task abnormally. You can handle these exceptional conditions by either using EXEC CICS HANDLE CONDITION commands, or by using the RESP and RESP2 options.

6.1.1.1 HANDLE CONDITION

You use this command to specify the label within the program to which control is to be passed if an exceptional condition occurs.

For example, a read from a VSAM data set can fail for some reason. What happens to the application if the data set is disabled? The program would be terminated abnormally by CICS. CICS allows the programmer to cope with this situation by inserting a CICS HANDLE CONDITION command before the data set READ command, as shown in Figure 10.

```
EXEC CICS HANDLE CONDITION DISABLED(DISABL)
.
.
.
EXEC CICS READ DATASET('name')
```

Figure 10. The EXEC CICS HANDLE CONDITION Command

If the file is disabled when the program issues the EXEC CICS READ FILE command, the program goes to the paragraph labeled DISABL. When you write your program you should include logic at DISABL to deal with this situation. This happens without the program being terminated abnormally by CICS. The program may now continue or return control to CICS.

You can handle general errors by using the EXEC CICS HANDLE CONDITION ERROR(label) command.

The HANDLE CONDITION command is similar to coding a GO TO statement in your application program. This is inconsistent with structured programming techniques, and for this reason many designers and programmers prefer to use the RESP and RESP2 method of dealing with exceptional conditions. One other reason for using RESP and RESP2 is that HANDLE CONDITION applies to all commands in your program until you disable or change the HANDLE.

6.1.1.2 RESP and RESP2 Options

Another useful method for treating exceptional conditions is to check the RESP and RESP2 values that you can request with each command-level call. For example, to check for a DISABLED condition you can use code similar to that shown in Figure 11.

```
EXEC CICS READ DATASET('name') RESP(xxx) RESP2(yyy)
.
.
.
IF xxx=DFHRESP(DISABLED) THEN .....
```

Figure 11. The EXEC CICS HANDLE CONDITION Command

You can test the responses within the application program and provide code to handle exceptional conditions without using HANDLE CONDITION before the call. As with HANDLE CONDITION the standard system default action for the condition, usually to ABEND the task, is disabled when you use the RESP option.

RESP2 is optional, but may only be specified when RESP is used. Since CICS does not provide a function like DFHRESP for testing RESP2 values, you have to code the testing explicitly in your application program. RESP2 contains information such as VSAM return codes.

If you are rewriting macro-level applications, the RESP and RESP2 method is more akin to macro-level ways of handling exceptional conditions.

We recommend that you use RESP and RESP2 instead of HANDLE CONDITION, because they allow you to write structured programs.

For more information see the *CICS/ESA Application Programming Guide*.

6.1.2 Soft Errors

Application code is unlikely to cause a CICS/ESA region to fail, other than by a storage violation (see section 7.2.2, “Minimize Storage Addressing Errors” on page 79). It can however cause the region to become unavailable to varying numbers of users because of deadlocks or code that is looping. The more thorough your testing, the less likely it is that these types of errors will occur in production systems.

6.1.2.1 Deadlocks

You should use the deadlock time-out (DTIMOUT) parameter on the resource definition for a transaction to minimize the impact of deadlocks. In conjunction with the RESTART parameter, it limits the unavailability, resulting from a deadlock, of a CICS/ESA system to an end-user.

When a get-for-update request is issued for a record in a VSAM data set, locking occurs at a control interval (CI) level. Implementation of VSAM RLS in a sysplex changes the locking to record level. When CICS/ESA fulfills its stated intention of supporting VSAM RLS, the potential for deadlocks in a CICS/VSAM environment will be reduced.

6.1.2.2 Looping

An error in program logic could result in the program looping without giving up control to CICS, monopolizing the CPU and thus preventing other transactions from being dispatched.

You should use the RUNAWAY parameter in the transaction resource definition to control looping. It specifies the amount of time, in milliseconds, for which any task running under this transaction definition can have control of the processor before it is assumed to be in a runaway condition (logical loop). When this interval expires, CICS can abnormally terminate the task. This means that the CICS/ESA region is again available to other transactions.

6.1.3 Abnormal Termination Recovery

CICS provides a program-level ABEND exit facility, invoked during abnormal termination of a task, so that you can include an exit routine of your own that can be executed when required. The EXEC CICS HANDLE ABEND command allows your application programmers to specify the name of a program or a routine which will receive control when a task ends abnormally.

An example of a function performed by such a routine follows:

- You have a multiple CICS environment where two CICS systems are communicating via distributed transaction processing (DTP). This means that there is an application in each CICS.
- The CICS that started the conversation is waiting for a reply from the application in the second CICS.
- For some reason, the session connecting the two CICS has been broken. Rather than waiting any longer, the application in the first CICS system issues an `EXEC CICS ABEND ABCODE('AC11')` and the application abends with a specific ABEND code of AC11.
- The ABEND command terminates the task abnormally, and causes an active exit routine to be executed. The HANDLE ABEND exit routine intercepts the ABEND code and sends a message that the session is down to the CICS console or to the end user. This will not prevent further abends when running this transaction, but the CICS operator (or an automated operator) can take action and the end users are informed why they cannot get a reply.

For more information see the *CICS/ESA Application Programming Guide*.

6.1.4 IMS/ESA DB Processing Options

A CICS application program needs a database program communication block (PCB) to access the database. The PCB defines the application program′s view of the database. The PCB also defines how the application program is allowed to process the segments in the database, whether the program can only read the segments, or whether it can update them as well. You do this by specifying a processing option for the PCB.

An application program with a processing option of GO can only read segments. The program can retrieve a segment even if IMS has locked that segment for another updating application program. If the GO application program retrieves a segment that contains an invalid pointer, IMS terminates the program abnormally.

To prevent a GO application program from being terminated abnormally in this situation, there are two additional processing options you can use with GO: N and T.

- When you use the N with GO (GON), IMS returns a GG status code to your program if the segment you are retrieving contains an invalid pointer. You can then decide what you want to do; for example you might continue processing by reading a different segment.
- When you use T with GO (GOT), IMS retries the request the program just issued. You will then be able to retrieve the updated segment if the program that was updating the segment has reached a commit point or had updates backed out since you last tried to retrieve the segment. If, when IMS retries the request for you, the pointer is still invalid, IMS returns a GG status code to the program.

The GG status code is only used in the ways described above, to indicate that the segment being retrieved contains an invalid pointer and the application program has a processing option of GON or GOT.

These options are limited to read-only applications. This increases availability because the database is still accessible for read while utilities such as Image Copy are running. The only disadvantage of these options is that integrity is not guaranteed. If this limitation is taken into account during design, then the GON and GOT options can be useful for increasing application database availability.

For more information see the *IMS/ESA Utilities Reference System*.

6.1.5 Program Error Program

You can specify your own processing to take affect after program ABENDs by using a program error program (PEP). The PEP is driven for all program ABENDs, whereas a HANDLE ABEND command is specific to the program in which it is coded. You can use the sample PEP supplied with CICS/ESA, change the sample, or write your own. Alternatively you can run your system without a PEP.

You could use a PEP to disable transaction IDs (to prevent their use) for failing programs pending diagnostic and corrective actions. This would avoid the need for master terminal operator intervention and the risk of several more ABENDs in quick succession. You could also use the PEP to disable other transactions that depend on the satisfactory operation of the failing program.

For more information see the *CICS/ESA Recovery and Restart Guide*.

6.1.6 Node Error Program for VTAM Terminals

You can specify your own processing for VTAM errors in a node error program (NEP). You can use the sample NEP supplied with CICS/ESA, change the sample, or write your own. Not all communication errors represent communication system failures. Some errors (such as trying to write zero-length data) may reflect special situations in applications, needing special action.

You may need to perform application-related activity when a node error occurs. For example, if a message cannot be delivered to a terminal, you may want it to be redirected to another. If you send a message with exception-response only, CICS may not have the data available to resend it, but the application might be able to recreate it. For example, if an error occurred while sending a document to a printer, the requesting application may be able to restart from the beginning, or from a specific page.

For more information see the *CICS/ESA Recovery and Restart Guide*.

6.1.7 Terminal Error Program for TCAM Terminals

You can specify your own processing for non-VTAM communication errors in a terminal error program (TEP). You can use the sample TEP supplied with CICS/ESA, change the sample, or write your own.

There are some situations in which CICS may try to send a message to an input-only terminal; for example, a message to say that the transaction ID used is invalid, or your application program could just wrongly send a message. You could provide a TEP to reroute these messages to a system transient data destination such as CSTL, which is used for terminal I/O messages.

Another example of where you might use a TEP is if you need to carry out application-related activity when a terminal error occurs. For example, if a message is not delivered to a terminal because of an error condition, you may need to notify the application that the message needs to be redirected.

For more information see the *CICS/ESA Recovery and Restart Guide*.

6.2 Avoiding Stress Points

One of the most important things you can do to avoid overstressing your CICS regions is to design and code your applications efficiently. The *CICS/ESA Application Programming Guide* contains a chapter on “Designing Efficient Applications.” Rather than repeating the information presented there, we recommend that you refer to that guide. For more detailed information on conversational and pseudo-conversational programming, see *CICS Transaction Design: Pseudo-Conversational or Conversational*.

One additional recommendation is that you use a 31-bit compiler. Many CICS sites today are still running code created and compiled using 24-bit compilers, and are suffering storage problems as a result. Write all new applications using a 31-bit compiler, and investigate the possibility of reworking and recompiling some of your most heavily used 24-bit applications with a 31-bit compiler.

6.3 Avoiding Single Points of Unavailability

One absolutely key aspect of application design for continuous availability is that you must design your applications so that they can exploit a multi-region CICSplex environment. If you design applications that are forced to run in a specific CICS region, you immediately introduce a single point of unavailability into your system.

6.3.1 Avoiding Transaction Affinities

You must avoid introducing transaction affinities into your application programs. Transaction affinities prevent you from exploiting our recommended system topology, and make one AOR a single point of unavailability for your application. Applications that include transaction affinities are far more vulnerable to unavailability than those which do not.

CICS transactions use many different techniques to pass data from one to another. Some of the techniques require that the transactions exchanging data must execute in the same CICS region and therefore impose restrictions on the dynamic routing of transactions. If transactions exchange data in ways that impose such restrictions, there is an “affinity” between them.

The significance of transaction affinities to CICS availability is their role in limiting the benefits of dynamic transaction routing. Dynamic transaction routing maintains continuous availability by routing transactions away from an unavailable AOR to an another AOR which is capable of running the transaction. For more information see section 4.2, “Dynamic Transaction Routing” on page 38.

We recommend that you use the CICS Transaction Affinities Utility to identify potential affinities in your existing applications. For more information see appendix C.9, “IBM CICS Transaction Affinities Utility MVS/ESA” on page 167 and the *CICS Transaction Affinities Utility MVS/ESA Users Guide*.

Two types of affinity affect dynamic transaction routing; intertransaction affinity and transaction-to-system affinity.

6.3.1.1 Intertransaction Affinity

Intertransaction affinities occur when CICS transactions use techniques to pass information between themselves, or to synchronize their activity, that require them to execute in the same CICS region. Intertransaction affinities can occur in the following circumstances:

- One transaction terminates, leaving *state data* in a place that a second transaction can access only by running in the same CICS region as the first transaction.
- One transaction creates data that a second transaction accesses while the first transaction is still running. For this technique to work safely, the first transaction usually waits on some event, which the second transaction posts when it has read the data created by the first transaction. This technique requires that both transactions be routed to the same CICS region.
- Two transactions synchronize, using either an event control block (ECB) or an enqueue (ENQ) mechanism. Because CICS has no function shipping support for these techniques, this type of affinity means the two transactions must be routed to the same CICS region.

If you have intertransaction affinities, there is always the potential for a loss of availability. Once the first transaction in a particular affinity transaction group has run, all other transactions in that group must use the same AOR for the lifetime of the affinity. If that AOR is unavailable, the subsequent transactions will fail. The only action you can take to prevent loss of availability due to intertransaction affinities is to eliminate them.

If you have existing applications with intertransaction affinities, then you can use your dynamic transaction routing program to manage those affinities, allowing your applications to run successfully. However, managing your intertransaction affinities with a dynamic transaction routing program does not ensure continuous availability if an AOR is unavailable.

You must eliminate intertransaction affinities from your applications. For a full discussion of intertransaction affinities, what causes them, and techniques for eliminating them, refer to *CICS/ESA Dynamic Transaction Routing in a CICSplex* or to the chapter entitled “Affinity Between Transactions” in the *CICS/ESA Application Programming Guide* for CICS/ESA 4.1. For more information on how to manage your intertransaction affinities with a dynamic transaction routing program see 4.2, “Dynamic Transaction Routing” on page 38 and 4.2.3, “CICSPlex SM” on page 40.

6.3.1.2 Transaction-to-System Affinity

A transaction-to-system affinity is an affinity between a transaction and a particular CICS region, where the transaction interrogates or changes the properties of that CICS region.

Transactions with affinity to a particular system, rather than another transaction, are not eligible for dynamic transaction routing. In general, they use INQUIRE and SET commands or have some dependency on global user exit programs, which also have an affinity with a particular CICS region.

There is no remote (that is, function shipping) support for INQUIRE and SET commands, nor is there a SYSID option on them. Hence transactions using those commands must be routed to the CICS region that owns the resources to which they refer. In general, such transactions cannot be dynamically routed to any AOR; they should be statically routed.

Global user exits running in different CICS regions cannot exchange data. It is unlikely that user transactions pass data or parameters by means of user exits, but if such transactions do exist, they must run in the same AOR as the global user exits.

Transaction-to-system affinities prevent you from dynamically routing transactions. If the region on which your transactions depend is lost the transactions become unavailable. The only action you can take to improve availability is to remove the transaction-to-system affinities from your applications. For a full discussion of transaction-to-system affinities, refer to *CICS/ESA Dynamic Transaction Routing in a CICSplex* or to the chapter entitled “Affinity Between Transactions” in the *CICS/ESA Application Programming Guide* for CICS/ESA 4.1.

6.3.1.3 Working with Intertransaction Affinities

You can use dynamic transaction routing and CICSPlex SM to provide a partial solution for intertransaction affinities that you cannot eliminate from your applications. Any affinities that cause a dependence on a particular AOR will cause some unavailability if that AOR becomes unavailable. For example, if you have an affinity transaction group with a pseudo-conversational lifetime that is active (the first transaction in the group has run and created an affinity with a particular AOR, and that affinity has not yet been broken), then unavailability of that AOR causes other transactions in this affinity transaction group to fail. However, if a new user uses the first transaction in the affinity transaction group, dynamic transaction routing and CICSPlex SM route the transaction to an available AOR, and the user is not aware of any unavailability.

6.3.1.4 Recommendations

Transaction-to-system affinities should be avoided. If you have any, they inevitably lead to a loss of availability for transactions which depend on a particular AOR if the AOR becomes unavailable.

Although intertransaction affinities do not initially appear to make transactions dependent on a particular CICS system, there are circumstances in which this is the case. The only way to remove the potential impact of such intertransaction affinities on availability is to eliminate them.

We recommend you use the CICS Transaction Affinities Utility to identify the cause of intertransaction affinities in existing applications. For more information see appendix C.9, “IBM CICS Transaction Affinities Utility MVS/ESA” on page 167 and the *CICS Transaction Affinities Utility MVS/ESA Users Guide*.

6.3.2 Removing the Requirement for Immediate Access to Another System Component

If your business and application requirements allow it, you should consider using asynchronous techniques to pass data and information between components of your system. This is a very powerful technique for systems that do not require an immediate response.

Consider a simple business transaction: withdrawing money from an ATM. Logically there are several steps in the transaction:

1. You enter your card, and identify yourself to the system using your personal identification number (PIN)
2. You ask for money
3. The system checks various factors, including whether you have enough money in your account to satisfy the request
4. The system issues the money and updates your balance

This is a very simple view of the processing that takes place, but we want to illustrate only one point; some of the processing must be synchronous, but some may be asynchronous. Checking the PIN and checking the balance are synchronous (our bank insists that they must be completed before we can issue the money), but the final part, updating the account balance, can be performed asynchronously. We do not have to update the account balance immediately, as long as we can guarantee that the balance is updated in a reasonably short time (maybe a few minutes).

If your application can use asynchronous techniques, then you can considerably increase the availability of your application; your application is available and can process normally, even if the AOR you want to use is unavailable.

You can use the IBM MQSeries for MVS/ESA to implement this sort of asynchronous processing.

6.3.3 Avoiding Any Requirement for Exclusive Use of Resources

The *CICS/ESA Application Programming Guide* contains a section "Exclusive Control of Resources," within the chapter "Designing Efficient Applications." Rather than repeating the information presented there, we recommend that you refer to that guide.

6.3.4 Using the External CICS Interface

In this section we describe the external CICS interface (EXCI), introduced in CICS/ESA 4.1, and how it can be used to reduce the need for scheduled outages. The EXCI is an application programming interface that enables a non-CICS program (a client program) running in MVS to call a program (a server program) running in a CICS/ESA 4.1 region, and to pass and receive data by means of a communications area. The CICS program is invoked as if linked-to by another CICS program. For more information about the EXCI see the *CICS/ESA External CICS Interface*.

We use two examples to show how you can use the EXCI to increase the availability of your systems. The first example shows how you could perform batch updates to a crucial CICS file while it remains online and available. The second example shows how the elapsed time of a batch job can be reduced by using the EXCI to control CICS resources.

6.3.4.1 EXCI Example 1: File Updates

You have a banking application that uses a file to hold account balances. The file is accessed constantly by transactions initiated at ATMs. Once a month you need to run a batch job against the file to credit monthly interest to certain accounts.

If the batch job is an EXCI client program, you can credit the monthly interest while all the ATMs remain fully functional. The EXCI client program passes requests to update the account balances file to the EXCI server program. The server issues an `EXEC CICS WRITE FILE(filename)` command on behalf of the client without the need for the file to be taken offline.

Due to the performance overhead of using the EXCI, the batch job would take much longer to run than if the file had been taken offline and accessed directly. You should use the EXCI to directly access data in a CICS region only for very limited access to crucial data that cannot be made unavailable to the CICS region.

6.3.4.2 EXCI Example 2: Controlling CICS Resources

Your EXCI batch job (client) requests that the server program issues system programming interface (SPI) commands on it's behalf. You could, for example, close a VSAM dataset using an `EXEC CICS SET FILE(filename) CLOSED` command, update the file from your batch job, make a backup copy of the file, and then open the file again using an `EXEC CICS SET FILE(filename) OPEN` command. This eliminates the need for operator intervention and removes the possibility of

operator error. The elapsed time of the batch job is reduced, resulting in a smaller batch window.

6.3.5 Choosing the Best Data Subsystem

You have two choices to make when designing your system: Which data subsystem do I use, and which data organization within that subsystem do I use?. For example, you can choose to use VSAM, and organize your data in a key-sequenced data set (KSDS), or you can choose IMS/ESA DB and organize your data in a HIDAM database.

Your choice will be influenced by many factors, such as whether the data you need to access is already available in one form on your system, which data subsystem is the most appropriate for the processing you want to do, and which offers the best general application development and operational environment.

Each data subsystem has different characteristics for continuous availability.

6.3.5.1 DB2

DB2 provides comprehensive facilities for continuous availability of data. DB2 Version 4 Release 1 also supports data sharing, enabling duplication of DB2 subsystems across MVS system images in a sysplex.

The only single point of unavailability in DB2 is reorganization. When you reorganize your data you can reorganize table spaces and partitions separately, and read the data during the unload phase. However, the entire table space is unavailable for other purposes during the later phases of reorganization.

See Chapter 8, “DB2 Considerations” on page 89 for further information.

6.3.5.2 IMS/ESA DB

IMS/ESA provides comprehensive facilities for continuous availability of data. IMS/ESA 5.1 also supports n-way data sharing, enabling duplication of DB2 subsystems across MVS system images in a sysplex. Earlier releases support two-way data sharing.

IMS/ESA allows you to take backup copies of your data without making it unavailability to your CICS applications. The online image copy utility creates an as-is copy of your database while it is being updated. This utility can be used with HISAM, HDAM, and HIDAM databases only. With IMS/ESA 4.1 and later releases you can use concurrent image copy for all full function DL/I databases, while your database is being updated. The backups produced by online and concurrent image copy can be used for recovery purposes.

The IMS online change utility enables you to update ACBLIBs, which contain PSBs and data management blocks (DMBs), and security information belonging to full function databases, without bringing down the system. This enables continuous access from CICS when ACBs need to be changed.

Data entry databases (DEDBs) can be reorganized without taking them offline. The DEDB direct reorganization utility enables you to reorganize DEDBs while CICS is updating the data. Reorganization is a single point of unavailability for full function databases.

See Chapter 9, “IMS/ESA DB Considerations” on page 99 for further information.

6.3.5.3 VSAM

VSAM facilities are not as comprehensive as those of DB2 and IMS/ESA DB for continuous availability of data. We discuss the IBM statement of direction for RLS in section 4.3.3, “VSAM” on page 43. DFSMS/MVS 1.3 supports n-way data sharing, enabling duplication of VSAM subsystems across MVS system images in a sysplex, however CICS support for this function is not available yet (as of November 1995).

Backup while open (BWO) allows you to take a backup of the file while CICS has the file open for update. If a control-area or control-interval split occurs during the backup, the backup cannot be used and you must repeat the process.

Reorganization of a VSAM file and batch updating both make the file unavailable to CICS. We recommend that you take a copy of the VSAM file before and after any batch changes and after a reorganization. This gives you the ability to recover, without rerunning the batch job.

See Chapter 10, “VSAM Considerations” on page 109 for further information.

6.4 Third Party Software

You should include “suitability for continuous availability” in your selection criteria when evaluating packaged solutions from independent software vendors, or when evaluating application development software. Some of the things you need to consider are:

- Is it supported with your current release of CICS/ESA?
- What transaction affinities, if any, exist? Do they prevent me from using dynamic transaction routing?
- Does the software support use of multiple transaction IDs?
- Can I apply maintenance without causing unavailability?
- Can I change any definitions without causing unavailability?
- Does the database design support continuous availability?
- Does the database used support data sharing?

Chapter 7. CICS Functions That Support Continuous Availability

In this chapter we discuss the features of CICS that enable continuous availability, and the remaining causes of unavailability in a single CICS region. We discuss:

- Changing your system
- Avoiding errors
- Avoiding overstressed systems
- Single points of unavailability
- Improving restart times

7.1 Changing Your System

In this section we describe the features that allow you to make non-disruptive changes to an individual CICS system, and the changes that are disruptive.

7.1.1 Defining Resources to CICS

Your CICS system must know which resources to use, what their properties are, and how they are to interact with each other. You supply this information to CICS by using one or more of the four methods of resource definition:

- Resource definition online (RDO) uses the CICS-supplied online transactions, CEDA, CEDB, and CEDC. Definitions are stored on the CICS system definition (CSD) file, and are installed into an active CICS from the CSD.
- Autoinstall minimizes the need for a large number of definitions by dynamically creating new definitions based on a model definition.
- The DFHCSDUP offline utility stores definitions in the CSD. DFHCSDUP allows you to make changes to definitions in the CSD by means of a batch job submitted offline.
- Macro tables can be used to define some resources. Definitions are stored in assembled tables in a program library, and are installed from there during CICS initialization.

7.1.1.1 Resource Definition Online

The ability to dynamically add or change resource definitions and have them immediately usable by your systems can eliminate the need for a scheduled outage. The CICS master terminal facility (CEMT) provides this function for many system initialization table (SIT) parameter changes, and RDO provides it for individual resource definitions. As CICS/ESA has evolved, we have seen numerous enhancements to RDO. CICS/ESA 3.1 introduced RDO for VSAM files, CICS/ESA 3.1.1 extended that support to cover data tables, and CICS/ESA 4.1 introduced support for transaction classes and for dynamic addition of MRO connection definitions.

With CICS/ESA 4.1 RDO is available for transactions, programs, profiles, map sets and partition sets, terminals and typeterms, connections and sessions, transaction classes, and files and LSR pools.

Using RDO you can change the definition of a resource in the CSD at any time or you can add a completely new definition. These new or changed definitions do not affect the running system until you either COLD start it (so that it uses the definitions in the CSD), or you use the CEDA INSTALL command to install the definition.

Generally you can use the INSTALL command to install a new resource definition at any time. In CICS/ESA 3.3 or earlier you must close inter-region communication (IRC) before installing a new CONNECTION definition (this restriction is removed in CICS/ESA 4.1). The installation should succeed unless you are installing a group of definitions, and one or more definitions in the group are not new, but alter existing definitions.

If you are changing an existing definition, the install fails if the resource is being used (for example, a transaction could be executing and using the transaction definition as you try to change it). If this happens when you are installing a group of definitions, all your changed definitions are backed out and no changes are made to the running system. For this reason you may want to install changed definitions individually rather than in groups. If a resource is very heavily used, you may have to disable it before you can install the new definition, causing a very short period of unavailability.

You cannot reinstall existing CONNECTION definitions while IRC is open, you must close it first. Connections must be installed by group, they cannot be installed as an individual resource. Because of these restrictions you must put any new connections in a group of their own if you want to install them while IRC is open.

See the *Resource Definition Guide* for your CICS release for more information.

7.1.1.2 Autoinstall

Before the introduction of autoinstall, you had to define all resources individually to CICS before they could be used. With the introduction of autoinstall for terminals in CICS/OS/VS 1.7, VTAM terminals can be defined dynamically on their first use, thereby saving both storage for terminal entries and time spent creating the definitions.

CICS/ESA 4.1 extends the autoinstall function to include user programs, map sets and partition sets, and LU6.2 parallel sessions.

If you enable the autoinstall program function, and an implicit or explicit load request is issued for a previously undefined program, map set, or partition set, CICS dynamically creates a definition, and installs and catalogs it, as appropriate.

Autoinstall for programs also removes one source of unavailability caused by an error. Before this support was available, an error in defining a program to the CICS region meant that the application using the program was unavailable. This could happen if the program was accidentally missed from the list of programs that need to be defined, was removed by mistake, or was misspelled. With autoinstall these errors are removed as a source of unavailability.

CICS/ESA 4.1 also introduces autoinstall for dynamically routed transactions in TORs. You need to define only one transaction definition for all dynamically routed transactions in a TOR. This makes it easier to create and manage the CICSplex you need to achieve continuous availability.

CICS/ESA 4.1 extends the autoinstall function to include support for APPC parallel session and APPC single sessions. In each of these cases the additional support is for BIND requests from primary nodes. CICS/ESA 4.1 continues to support autoinstall for APPC single sessions initiated by a CINIT request, as in earlier releases. This works in the same way as autoinstall for terminals, in that you supply a model TERMINAL definition and its associated TYPETERM definition. Autoinstall for APPC sessions has the follow advantages for continuous availability:

- Restart of any kind should be noticeably faster, especially when large numbers of terminals are involved.
- You can reduce storage requirements, as autoinstalled resources do not occupy space until they are required.
- The end users of APPC parallel sessions benefit by getting faster access to their CICS systems. They do not have to wait for the system programmer to define the connections to them.

7.1.1.3 The DFHCSDUP Utility

You can change resource definitions in your CSD using DFHCSDUP, or you can add new resource definitions. You cannot install resources in an active CICS system with DFHCSDUP. If you are sharing a CSD among regions, it must not be in use by another region while you are running DFHCSDUP.

7.1.1.4 Macro Tables

The evolution in all CICS releases has been from macro definitions to RDO. Macro definitions are a problem when your goal is continuous availability. You can change the definitions contained in the tables while CICS is running, but this has no effect on the running system. To change the definitions in CICS you must stop and restart CICS.

RDO does not support the following resource definitions:

DCT for both intrapartition and extrapartition data sets
FCT for non-VSAM files
JCT for journals
MCT for monitoring data
DDIR directory of data management blocks (local DL/I only)
PDIR directory of program specification blocks (local DL/I only)
PLT for initialization and shutdown processing
SIT for initialization and customization parameters
Many SIT parameters can be changed dynamically, by either program or operator commands.
SRT for system recovery
TCT for non-VTAM terminals and networks
TST for remote and recoverable temporary storage queues

Any changes to these resource definitions cannot take effect until the CICS region has been stopped and restarted to load the updated table. Orderly recycling of cloned AORs with CICSPlex SM allows you to change even these resources in a non-disruptive manner. However, if a region is a single point of unavailability, then making changes to any of these resources results in a period of unavailability for any applications dependent upon this region.

7.1.2 Changing the CICS System Configuration

Generally speaking you cannot make major changes to the configuration of your CICS system without stopping it and restarting it. For example, you cannot change the number of journals in the system, in CICS/ESA 3.3 you cannot change the sizes of the dynamic storage areas (DSAs), but you can work around this by changing the storage cushion sizes (in CICS/ESA 4.1 you can change the DSA sizes), and you cannot change the number of buffers for temporary storage or intrapartition transient data. Again, duplicating CICS regions in a CICSplex is key to avoiding unavailability due to the need to make changes like these.

7.1.3 Changes for IMS/ESA

If you use DBCTL you avoid the need for changes to CICS/ESA when a new release of IMS/ESA is installed. With DBCTL, the levels of CICS/ESA and IMS/ESA products installed are independent of each other; this is not true with local DL/I. In both cases you need to stop the IMS subsystem (either DBCTL or your DOR), and then restart it using the new level of IMS/ESA. This will cause a period of unavailability unless you are operating in a data sharing environment.

DBCTL also allows you to use the IMS online change facility to update ACBLIBs without bringing down the system. See section 9.1.1, “Online Change Utility” on page 99.

7.2 Avoiding Errors

CICS has several functions that help you deal with errors. Usually these functions diminish the effect of programming errors.

7.2.1 Use the Latest Version of CICS

IBM is continually enhancing the reliability and availability of CICS/ESA by the internal restructure of CICS/ESA code and control blocks, using modern software engineering techniques. This commenced with the first release of CICS/ESA and has been extended with each subsequent release. Selected parts of the CICS/ESA code are restructured to improve reliability and serviceability, and to take advantage of the capabilities of MVS/ESA Domains are functionally isolated parts of CICS/ESA that communicate with the rest of the CICS/ESA system, and with external applications, by means of strictly-defined, standard, internal interfaces. This improves the reliability of CICS/ESA, makes problem determination easier, and limits the effects of program failures. Many CICS users have recorded noticeable increases in availability after they have migrated to later releases of CICS/ESA.

We recommend that you migrate to the latest release of CICS/ESA as soon as possible.

Despite the significant improvements in code quality, there is still a possibility that CICS/ESA may fail. Using a CICSplex minimizes the impact of CICS/ESA software failure.

7.2.2 Minimize Storage Addressing Errors

Before CICS/ESA Version 3 Release 3, CICS and application code executed mainly in a single storage key (key-8). This meant that user application code had the same access to CICS code and control blocks as CICS itself, and hence there was no protection against erroneous applications inadvertently overwriting parts of storage, except to place some modules into the Link Pack Area (LPA). CICS/ESA 3.2.1 brought some relief to the problem by putting the CICS areas in lower addresses than user areas. This dealt with many of the problems (since they are often caused by programs that try to write overlength data into a storage area).

These errors tend to either cause an ABEND of CICS (causing a period of unplanned unavailability), or an ABEND of another task. Even worse, it is possible that the error will overwrite storage in such a way that there is no ABEND in the system, but data is corrupted.

CICS/ESA 3.3 introduced storage protection, which uses an extension to MVS storage keys to physically separate CICS code and control blocks from user storage. Storage protection is described in detail in the *CICS/ESA Release Guide* for CICS/ESA 3.3. Storage protection is only a partial solution. It helps to prevent failures of the CICS region due to addressing problems. CICS/ESA 4.1 introduced transaction isolation. This can be used to prevent user applications from overwriting the storage of other user applications. CICS/ESA 4.1 also provides validation of addresses passed to CICS which the caller requires to be updated by CICS (command protection). Transaction isolation and command protection are described in detail in the *CICS/ESA Release Guide* for CICS/ESA 4.1.

7.2.3 Errors in Definition

Autoinstall for programs can help with availability. If a program has not been defined to CICS, autoinstall will build a definition when it is required, removing one small cause of unavailability.

7.3 Avoiding Overstressed Systems

CICS/ESA Version 3, in its various releases, has dramatically improved the way storage is handled, thus reducing the chance that storage will become a stress point. CICS/ESA 3.3 and CICS/ESA 4.1 also introduced global user exits that prevent inter-system queueing from becoming a source of stress.

7.3.1 Storage Management

Dynamic storage area evolution goes hand in hand with subsystem storage protection and transaction isolation.

7.3.1.1 Storage Management in CICS/ESA 3.3

CICS/ESA 3.3 has five DSAs, each with its own storage cushion. You must specify each DSA and cushion size. You can change the cushion sizes dynamically, but the DSA sizes are fixed for the lifetime of CICS. We recommend that you over allocate the DSAs and then control them using cushion sizes. This avoids a period of planned unavailability to adjust the DSA sizes.

7.3.1.2 Storage Management in CICS/ESA 4.1

CICS/ESA 4.1 has eight DSAs. To facilitate continuous operations and to simplify system management, CICS determines individual DSA sizes and varies them dynamically. You have to specify only the upper limits of the total DSA requirement above and below the 16MB line; you do not have to specify DSA and cushion sizes. You specify the upper limits of the total DSA requirement using two new SIT options; DSALIM and EDSALIM. They specify upper limits for DSA storage below and above the 16MB line respectively. You can change DSALIMIT and EDSALIMIT while CICS is running using either the CEMT master terminal command, or an EXEC CICS SET command from within a program.

7.3.1.3 Program Compression

As storage becomes constrained, dynamic program compression progressively releases programs which are not being used. However, short-on-storage conditions can still occur and are reported as “Times went short on storage.” If this value is not zero, you might consider increasing the size of the dynamic storage area. Otherwise you should consider the use of MXT and transaction classes to constrain your system′s virtual storage requirements.

Compared with previous versions of CICS, CICS/ESA Version 3 and CICS/ESA Version 4 handle storage stress in a different way. The intention is to avoid the major disturbance to transaction throughput and response time that could result from non-dynamic program storage compression (the removal of all nonresident, not-in-use programs which previous versions implemented when a GETMAIN request could not be satisfied.)

Nonresident, not-in-use programs may be deleted progressively with decreasing free storage availability as CICS determines appropriate, on a least-recently-used basis. The dispatching of new tasks is also progressively slowed as free storage approaches a critically small amount. This self-tuning activity tends to spread the cost of managing storage. There may be less or more program loading overall, but the heavy overhead of a full program compression is not incurred at the critical time.

The loading or reloading of programs is handled by CICS with an MVS subtask. This allows other user tasks to proceed if a processor of the MVS image is available and even if a page-in is required as part of the program load.

Storage compression is part of storage management that helps to prevent unscheduled periods of unavailability. You still have to monitor storage statistics to be aware of the storage stress situations and make the appropriate actions to solve the situation.

7.3.2 Intersystem Session Queue Management

In a perfect intercommunication environment, queues would never occur because work flow would be evenly distributed over time, and there would be enough intersystem sessions available to handle the maximum number of requests arriving at any one time. However, in the real world this is not the case, and, with peaks and troughs in the workload, queues do occur. We need to avoid the situation where an unacceptably high level of queuing causes a bottleneck in the work flow between interconnected CICS regions, which leads to availability problems for the terminal end-user as throughput slows down or stops. In extreme cases large parts of a CICSplex can stall, requiring the stalled systems to be restarted to recover. This abnormal and unexpected queuing

should be prevented, or dealt with when it occurs: a “normal” or optimized level of queuing can be tolerated.

For example, function shipping requests between CICS AORs and connected FORs can be queued in the issuing region while waiting for free sessions. Provided a FOR deals with requests in a responsive manner, and outstanding requests are removed from the queue at an acceptable rate, all is well. But if an FOR is unresponsive, the queue can become so long and occupy so much storage that the performance of connected AORs is severely impaired. Further, the impaired performance of the AOR can spread to other regions. This condition is sometimes referred to as “sympathy sickness,” although it should more properly be described simply as intersystem queuing, which, if not controlled, can lead to performance degradation across more than one region.

CICS/ESA 4.1 provides an internal solution based on values you specify for the QUEUELIMIT and MAXQTIME parameters of the connection resource definition. If you specify values for these parameters, CICS restricts any queues on the connection using the limits specified.

CICS/ESA 4.1 provides a new GLUE, XZIQUE. This allows you to write an exit program to detect queuing problems early and take action to deal with them. CICS/ESA 3.3 provides the XISCONA GLUE to limit the queuing of function shipping requests. The XZIQUE exit extends the function provided by XISCONA, which is driven only for function shipping requests. XZIQUE allows you to deal with allocate requests originating from function shipping, transaction routing, and other forms of intercommunication request. This provides increased flexibility to help you handle bottlenecks caused by these requests. An XZIQUE global user exit program can respond to situations in ways most appropriate for the special circumstances of each customer installation.

The exit enables allocate requests to be queued or rejected, depending on the length of the queue. The exit also allows a connection on which there is a bottleneck to be cleared of its backlog.

There is no interaction between the XZIQUE and XISCONA global user exits. If you enable both exits, XISCONA and XZIQUE could both be driven for function shipping requests, which is not recommended. You should ensure that only one of these exits is enabled.

For those situations where simple control requirements are adequate, you can specify the “queue limit” parameter for connections. If you also need to detect and react quickly to bottlenecks, you can use the ′maximum queue time′ parameter.

You can write an XZIQUE global user exit program to detect bottlenecks early and take actions that might resolve a problem before the situation becomes really serious and begins to affect other interconnected regions. Detecting undesirable queuing early can help to improve reliability and availability in busy communicating systems.

7.3.3 System Managed Storage

System Managed Storage (SMS) is an approach to DASD storage management in which the storage management subsystem determines data placement and an automatic data manager handles data backup, movement, space, and security.

You must use DFHSMS to manage your VSAM data sets if you use BWO. If you use DFHSMS to manage your VSAM data sets, you should consider carefully the period after which your production VSAM data sets are migrated to primary or secondary storage, or if they should be migrated at all. If a migrated data set has to be recalled for CICS, it can take several minutes from primary storage, or longer from secondary storage. While the recall is taking place, the application is unavailable to the user, and no other open or close operations on that data set can be performed until the data set has been recalled. The system could eventually become stressed due to a cascading effect as other transactions also queue for the data set.

If a migrated data set has to be recalled, CICS issues message DFHFC0989 to the system console. This notifies the system operator that a recall is taking place, and indicates whether it is from primary or secondary storage.

Problems can occur at startup time as the CICS/ESA initialization task (III) builds the LSR pool. The information to build the pool is taken from the file definition, or from the VSAM data set. If a VSAM file definition belonging to an LSR pool does not have all the necessary information supplied to build the LSRPOOL, a VSAM SHOWCAT command is issued to obtain the missing information. If the VSAM file has been migrated, the CICS region recalls the data set to obtain the data. If the VSAM file has been migrated to level two, the wait can be significant. This causes CICS to wait at startup, often giving the appearance of being hung.

You must ensure that the CICS log, transient data and intrapartition data sets are not migrated. Because these data sets rely on format and/or placement on DASD device, the migration compression and recall decompression can adversely affect their format and recalling them to DASD with different geometry causes access problems.

DASD management systems offer you many benefits, but we recommend careful consideration be made before placing any CICS/ESA data sets under their control.

7.3.4 Limiting System Usage

You can use CICS parameters, such as MXT and transaction classes, to limit the number of tasks in a CICS region, thus limiting the use of system resources such as virtual storage. MXT limits the overall number of tasks, while transaction classes allow you to be more selective and limit only a subset of transactions.

7.4 Avoiding Single Points of Unavailability

Possible single points of failure due to system design are the TOR and the FOR.

End users in a CICSplex experience unscheduled unavailability when a TOR fails. If there is only one TOR, all users are affected; with two or more TORs, only those users connected to the failed TOR are affected. To minimize the impact of TOR unavailability we recommend that you use VTAM generic

resource registration, in conjunction with VTAM persistent session support, with multiple TORs in a sysplex with multiple network nodes. If you only have one TOR, use VTAM persistent session support (refer to section 7.5.2, “Persistent Session Support” on page 86) to improve restart time and minimize the period of unavailability.

7.4.1 VTAM Generic Resource Registration

A VTAM application program such as CICS can be known by its generic resource name, in addition to its own application program network name as defined on the APPL definition statement. Multiple CICS regions can use the same generic resource name. For the generic resource function to operate, each VTAM application must register with VTAM under its generic resource name. CICS/ESA 4.1 automatically registers with VTAM when it is ready to receive VTAM logon requests (CICS/ESA 3.3 and earlier releases of CICS do not support generic resource registration).

A terminal user, wishing to start a session with a CICSplex that has several TORs, uses the generic resource name in the logon request. VTAM establishes the session with one of the TORs that is registered as a member of the generic resource name.

VTAM generic resource registration minimizes the impact of scheduled or unscheduled TOR outages, because the end-user can logon to any available TOR with the same generic APPLID.

To use VTAM generic resources in CICS/ESA 4.1, you need ACF/VTAM Version 4 Release 2 or a later, upward-compatible, release. In addition, VTAM 4.2 must be running under an MVS system image that is part of a sysplex and it must be connected to a coupling facility.

VTAM generic resource registration and XRF are mutually exclusive.

7.4.2 Failure of a FOR in a CICSplex

You can contain the impact of an outage due to failure of a FOR to a subset of users by having separate FORs for each group of related data sets. You should place all data sets that can be updated in a single unit of work in the same FOR. By setting up multiple FORs, you also reduce the restart time for any one of them and reduce the possibility that processor resource usage by a FOR could become a bottleneck.

7.4.3 CICS/ESA Data Sets

Loss of a CICS/ESA system data set does not always result in unavailability of CICS/ESA; it may only lead to a specific function being unavailable. For example, an I/O error on a dump data set means that dumping is terminated but processing continues. In this section we look at the CICS/ESA data sets as potential single points of unavailability.

You can use the dual copy function of 3990-3 or 3990-6 in an appropriately configured DASD subsystems to eliminate the possibility of an outage due to an I/O error on a CICS/ESA system data set. Any decision to use dual copy must take performance implications into account. Data sets with high write activity, such as the system log, are not good candidates for dual copy because of the potential for performance degradation. Conversely, data sets with low write activity such as the global and local catalogs and restart data sets are excellent

candidates for dual copy. For information on dual copy refer to section 12.2.2, “Dual Copy” on page 123.

In the following individual discussion of each of the CICS/ESA data sets, the comments relating to I/O errors assume no dual copy function is available.

- DFHRPL library:

  If an I/O error occurs, you can continue to use those modules that have already been loaded into the region. Some applications are unavailability because they cannot load the modules they require. You should restart CICS/ESA as soon as you can using backup copies of the data sets.
  Impact: Widespread if many regions use the affected data set in their DFHRPL concatenation.
  Recommendation: Ensure that you have recent backup copies of the data set in your DFHRPL concatenations. You will need these for disaster recovery anyway.

- DFHCSD:

  The CICS/ESA system definition data set (DFHCSD) is not used by a running system unless you are using resource definition online (RDO) to install new definitions. If an I/O error or data set full condition occurs, it does not cause CICS/ESA to fail. However, you can only recover by restarting CICS/ESA using a backup copy of the data set. When you do this depends on how critical it is to introduce the new definition. The CSD is managed through CICS file control and can be a candidate for forward recovery.
  Impact: Widespread if many regions share the same CSD.
  Recommendation: Ensure that you have a recent backup copy of the CSD, or make the CSD eligible for forward recovery. You will need a copy of the CSD for disaster recovery anyway.

- DFHGCD:

  CICS/ESA fails if there is an I/O error while writing to the global catalog (DFHGCD) or if the global catalog fills. If an I/O error occurs, you should first attempt an emergency restart. If this fails, a cold start is required and any dynamically installed resource definitions must be reinstalled. If a data set full condition occurs, emergency restart is probably required. It may be possible to copy the data to a larger data set.
  Impact: Limited: Each global catalog is used by one region (except in the case of XRF, where it is shared between the active and alternate regions).
  Recommendation: Use dual copy facilities for crucial regions (TORs and FORs, for example). Ensure that you define a secondary extent so that the catalog does not fill.

- DFHLCD:

  For both an I/O error and a data set full condition, CICS/ESA can continue. However, if you need to reinitialize the local catalog, you are forced to reinitialize the global catalog, and you must perform a cold start when you next start the region.
  Impact: Limited: Each local catalog is used by only one region.
  Recommendation: Use dual copy facilities for crucial regions (TORs and FORs, for example). Ensure that you allow for secondary extents.

- DFHRSD:

  The CICS/ESA restart data set is used only during emergency restart. If an I/O error occurs, redefine DFHRSD and retry the restart. If a data set full

condition occurs, the job again needs to be restarted but with a larger data set size specified.
Impact: Limited: Each restart data set is used by only one region, and only when the region is started.
Recommendation: Delete and redefine the data set if an error occurs during startup.

- DFHINTRA

  Loss of the intrapartition transient data set causes CICS/ESA to fail. If an I/O error occurs, you need to restart the CICS/ESA region, specifying a cold start for transient data.

  If a data set full condition occurs, further attempts to write fail with a NOSPACE condition. The CICS/ESA logic tries to trigger automatic transaction initiation (ATI); if this is not successful, a restart is required. It is impossible to copy the data to a larger data set.
  Impact: Limited: Each intrapartition transient data set is used by only one region.
  Recommendation: Use dual copy facilities for crucial regions (QORs where there are recoverable intrapartition transient data queues, for example). Ensure that you allow for secondary extents.

- DFHTEMP

  DFHTEMP is the auxiliary temporary storage data set. If an I/O error occurs, information may be returned to the application. Ultimately you need to perform a restart specifying a cold start for temporary storage. If a data set full condition occurs, temporary storage tries to format additional control intervals (CIs). If none are available TS waits and retries. Eventually, a restart may be required, possibly after copying to a larger data set while retaining the CI size.
  Impact: Limited: Each temporary storage data set is used by only one region.
  Recommendation: Use dual copy facilities for crucial regions (QORs where there are recoverable temporary storage queues, for example). Ensure that you allow for secondary extents.

- Dump data sets

  The CICS/ESA dump data sets, DFHDMPA and DFHDMPB, are not critical to CICS/ESA availability. If an I/O error occurs, dumping is terminated but CICS/ESA continues to run. If a data set full condition occurs, dumping is stopped or switched to a new data set and again CICS/ESA continues.

- Trace data sets

  If an I/O error occurs, tracing is terminated, but CICS/ESA keeps on running. If a data set full condition occurs, tracing is stopped or switched to a new data set.

- Extrapartiton TD

  If an I/O error occurs, it is handled by the access method (QSAM) and information is returned to the application. If a data set full condition occurs, the information is returned to the application and no further records are written to the data set. The application must close the data set, process the data and then reopen the data set.

- Journals

  If a journal suffers an I/O error, CICS issues a message recommending shutdown. The journal is probably unreadable. If the journal is the system

log, then a subsequent COLD start (or a WARM start if a NORMAL shutdown was successful) is required.

- If a data set full condition occurs, journaling is stopped or switched to a new data set.
  Impact: Limited: Each journal data set is used by only one region.
  Recommendation: Use dual copy facilities for crucial regions (any region that owns recoverable resources).

7.5 Improving Restart Times

You should ensure that you are using all the features of CICS that improve restart times. These include CICS/ESA 4.1 support for the MVS ARM, persistent session support, and autoinstall of LU6.2 devices. You should also review the *CICS/ESA Performance Guide* for information on improving CICS startup time.

7.5.1 Support for the MVS/ESA Automatic Restart Manager

CICS/ESA 4.1 makes use of the MVS ARM to automatically detect the unscheduled unavailability of a CICS region and restart it. This significantly reduces the time for a restart, since detection of the failure is at machine speeds rather than human speeds, and the restart action is predefined.

For more information on the MVS ARM see 5.2.4.4, “Automatic Restart Manager” on page 59 and the *CICS/ESA Release Guide* for CICS/ESA 4.1.

7.5.2 Persistent Session Support

Persistent session support improves the availability of CICS. It uses VTAM 3.4.1 persistent LU-LU session improvements to provide restart-in-place of a failed CICS without rebinding.

CICS support of persistent sessions includes the support of all LU-LU sessions except LU0 pipeline and LU6.1 sessions. CICS determines for how long the sessions should be retained from the PSDINT system initialization parameter. This is a user-defined time interval. If a failed CICS is restarted within this time, it can use the retained sessions immediately. There is no need for network flows to rebind them.

You can change the interval using the CEMT SET VTAM command, or the EXEC CICS SET VTAM command, but the changed interval is not stored in the CICS global catalog, and therefore is not restored in an emergency restart.

If you terminate CICS through CEMT PERFORM SHUTDOWN IMMEDIATE, or if CICS fails, its sessions are placed in “recovery pending” state.

During emergency restart, CICS restores those sessions pending recovery from the CICS global catalog and the CICS system log to an “in session” state. This happens when CICS opens its VTAM ACB. Without persistent session support, all sessions existing on a CICS system are lost when that CICS system fails. In any subsequent restart of CICS, the rebinding of sessions that existed before the failure depends on the terminal′s AUTOCONNECT option. If AUTOCONNECT is specified for a terminal, the user of that terminal waits until the GMTRAN transaction has run before being able to continue working. If AUTOCONNECT is not specified for a terminal, the user of that terminal has no way of knowing (unless told by support staff) when CICS is operational again unless the user

tries to log on. In either case, users are disconnected from CICS and need to reestablish a session, or sessions, to regain their working environment.

With persistent session support, sessions are put into recovery pending state on a CICS failure. If CICS starts within the specified interval, and RECOVOPTION is set to CLEARCONV or SYSDEFAULT, terminal users do not need to reestablish their session, or sessions, to regain their working environment. The sessions for autoinstalled terminals persist in a bound state, subject to the operation of the AIRDELAY system initialization parameter. This means that if an autoinstalled terminal is not used during the period specified on the AIRDELAY system initialization parameter, its session is unbound and the terminal entry is scheduled for deletion.

The terminal user is notified of the successful recovery if MESSAGE is specified on RECOVNOTIFY of the TYPETERM resource definition.

Persistent session support improves availability of CICSplexes, particularly those with one or more terminal-owning regions that attach a large number of sessions. It provides a faster restart of failed TORs and so improves the availability of an entire CICSplex. Without persistent session support, such a TOR may take a considerable time to restart after failure. If a CICSplex has only one TOR and it has failed, the entire CICSplex may become unavailable to end users. If a CICSplex has more than one TOR and one (or more) fails, large parts of the CICSplex may become unavailable to end users.

Persistent session support should be used with care for LU6.2 sessions between CICS/ESA regions since it can result in excessive queueing.

For CICS persistent session support, you need the VTAM persistent LU-LU session enhancements in VTAM 3.4.1 or later. CICS/ESA 4.1 functions with releases of VTAM earlier than 3.4.1, but in the earlier releases, sessions are not retained in a bound state in the event of a CICS failure.

Persistent session support and XRF are incompatible; if XRF=YES is specified, PSDINT is ignored.

7.5.3 Using Autoinstall to Improve Restart Times

If you are using autoinstall with cataloging, restart times are similar to those of restarting a CICS region that is not using program autoinstall. If you are using autoinstall without cataloging, CICS restart times are improved because CICS does not install definitions from the CICS global catalog. Instead, definitions are autoinstalled as required whenever programs, map sets and partition sets are first referenced following the restart. Cold starts are faster with autoinstall for programs because you can omit all predefined program definitions from startup group lists.

7.5.4 Extended Recovery Facility

The extended recovery facility (XRF) improves availability to end users. Active and alternate CICS/ESA systems work together, monitoring each other′s well-being. Only the active CICS carries out normal CICS processing. The alternate CICS waits for a failure of the active CICS, or a command, and then takes over as a new active system. It does this by taking over the resources of the active CICS, including databases, terminals, and the system log, and effecting an emergency restart. The speed of the restart is enhanced by the

automatic or semiautomatic restart initiation, and backup sessions between XRF-capable terminals and the alternate system.

CICS with XRF can cope with CICS failures, and also with CPC, operating system, and VTAM failures when the alternate is in a second CPC.

It is important to note that the XRF concept is based on an emergency restart, and the recovery and restart procedures in a single CICS system are still relevant when you use XRF. If your current recovery procedure delays the restart to allow, for example, post-processing or preprocessing, these procedures are no longer possible with XRF.

XRF can be used to eliminate the need for a scheduled outage for maintenance. You can force the alternate system to take over the processing, conduct the necessary maintenance activity, and then have the original system take back the processing.

The viability of XRF is compromised in a CICS/ESA environment using dynamic transaction routing in two ways. Firstly, the proliferation of CICS/ESA regions as a result of cloning adds complexity to the XRF setup procedures. Secondly, the functions available in sysplex, parallel processors, CICS/ESA and ACF/VTAM has obsoleted many of the benefits available through XRF. If you use CICS/ESA 4.1 with MVS/ESA 5.2, the MVS ARM provides automatic detection of failure and restart, including restart on an alternate MVS system image in the sysplex, if appropriate. Persistent session support allows fast reconnection with a terminal network. XRF still provides better support in some situations than the MVS ARM and persistent session support (restarting a TOR on an alternate MVS system image, for example), but in general we recommend that you plan to use the newer functions, rather than XRF.

Note: XRF and persistent session support are mutually exclusive.

If you need further information on XRF, you should refer to *CICS/ESA XRF Guide*. This manual was not updated for CICS/ESA 4.1.

Chapter 8. DB2 Considerations

In this chapter we discuss DB2 considerations relating to achieving continuous availability, especially:

- Changing your system
- Errors
- Avoiding overstressed systems
- Single points of unavailability
- Improving restart times

8.1 Changing Your System

DB2 is designed with the following capabilities for nearly continuous operation:

- Online definition and modification of database and authorization descriptors
- Online binding of application plans
- Online changing of buffer pool and hiperpool sizes
- Online execution of most utilities. For example:
  - You can recover online such objects as table spaces, partitions, data sets, a range of pages, a single page, and indexes.
  - You can recover several indexes or index partitions simultaneously to reduce recovery time.
  - You can read and update a table space while copying it.
- You can reorganize table spaces and partitions separately and read the data during the unload phase. However, the entire table space is unavailable for other purposes during the later phases of reorganization.
- Availability of a table space (provided it is not explicitly stopped) after an I/O error, except for any portions that span the error ranges
- Consistent reorganization time for a table, regardless of the cluster ratio, by specifying the SORTDATA parameter on the REORG utility.
- The use of packages significantly reduces the amount of time that your applications are unavailable because of the bind process. When an application changes, only the database request modules of the programs that have changed need to be rebound.
- Continuing operation of DB2 after an I/O error writing a log record. On the active log, it moves to the next data set; on the archive log, it dynamically allocates another data set.
- Remote site disaster recovery methods that allow you to prepare for disasters that could cause a complete shutdown of your local DB2 system.
- Typical continuation of DB2 during restoration of dual operation of the bootstrap data set, active logs, and archive logs if degradation to single copy mode was necessary.

8.2 Errors

It is important to avoid I/O errors on table spaces, indexes, logs, and bootstrap data sets (BSDSs). DB2 usually continues to operate after log write I/O errors. On the active log, DB2 moves to the next data set. On the archive log, another data set is dynamically allocated.

The DB2 subsystem usually does not have to be taken down to restore dual operation of the BSDS, active log, and archive log if degradation to single copy mode was necessary. When you have an I/O error, the outages have to be reduced to a minimum. Errors that have to be recovered are:

- Table space or index space I/O errors

  Any index spaces or table spaces affected must be recovered.

- DB2 catalog or directory I/O errors.

  Any catalog or directory table space must be recovered. DB2 remains active, but only the person with overall install SYSADM authority can perform the recovery (or do anything at all with the DB2 subsystem) until the recovery takes place.

- BSDS or log I/O errors

  - Write log error
    Logging goes to the next available log data set. The data set is not lost, it is reused on the next cycle. If errors persist on this data set, you have to stop DB2 after the next offload and use VSAM AMS and the offline utility Change Log Inventory to add a replacement log data set.
  - Read log error
    If the error occurs during offload, offload tries to use the dual copy of the log data set. If dual logging is not active, the log data set is stopped.
  - Read errors on archive log while using RECOVER
    If a second copy exists, it is allocated and used; otherwise, recovery fails.
  - I/O error on the BSDS

    Normally, DB2 keeps duplicate copies of the BSDS. If one copy fails, DB2 continues with a single BSDS. To restore the failed BSDS use the RECOVER BSDS command, which makes a copy of the good BSDS in a preallocated data set.

8.3 Avoiding Overstressed Systems

To avoid overstressing your DB2 subsystem, you should monitor your databases regularly.

8.3.1 Monitoring DB2 Databases

It is important for CICS continuous availability to monitor the databases regularly. Monitoring measures the efficiency of your database, in performance and space utilization. Most of your base tables and indexes are constantly being changed through updates, inserts, and deletions. Monitoring can expose changes in space that may lead to problems. The monitoring and tuning process of your databases can be done by using three main sources:

- **The RUNSTATS utility program**

  The RUNSTATS utility scans table spaces or indexes to gather statistics about data and indexes. To ensure that the information is available and current, you should run RUNSTATS at these times:

  – After loading a table space and the appropriate indexes have been created and before binding application plans that will access it.

  – When a table space or index has been reorganized. Then you should optionally re-bind applications plans.

  – After extensive inserts and updates. Again re-bind application plans for which performance is critical.

  Use the RUNSTATS information to tell when to reorganize your table spaces and indexes. RUNSTATS does not produce a report on the statistics gathered but normal SQL queries can be executed against the DB2 catalog tables to obtain the information needed to monitor space growth and the need for reorganization.

  RUNSTATS reads the entire table space. This may take a long time. It allows either read-only access or you can allow updates to the table space and indexes, which is likely to lead to an increased executing time for RUNSTATS.

- **The STOSPACE utility program**

  The STOSPACE utility program is an online program that will collect information on actual space allocated for storage groups and related table spaces and indexes. After executing STOSPACE, monitor space usage with SPUFI or QMF queries. Used periodically, the STOSPACE utility can help you determine if the defined disk space should be increased or decreased.

- **The DB2 catalog tables**

  Information in the DB2 catalog can help you determine when to reorganize table spaces and indexes. You cannot, however, reorganize catalog table spaces.

  Information from the SYSTABLEPART catalog table can tell you how well disk space is being used for table spaces. You can find out the percentage of ″dead″ space in a specific table space. DB2 reclaims space lost through DROP activity when you reorganize the table space.

8.4 Single Point of Unavailability

Databases are a critical resource for continuous availability. As long as you cannot update duplicate copies of the data, you have to plan for regular backup, reorganization and recovery of your data with the minimum of unavailability.

Planned outages can seriously affect the availability of databases so the scheduled frequency of backup and reorganization should be reduced. However, the backup and reorganization of a database cause database outage only during load or reload phases of the utilities. The planned outages are needed because they reduce the probabilities and duration of unplanned outages.

The following topics address data sharing, ways to minimize the time for backup and reorganization, and also methods for reducing the effect recovery has on

availability. This is not a complete description of the utilities; for more information, refer to *DB2 Utilities Guide,*.

8.4.1 Data Sharing

DB2 provides excellent support for shared access to data. Many CICS regions, batch jobs, and other users can concurrently access data held in DB2. We recommend that you use a version and release no earlier than DB2 Version 4 Release 1. It provides data sharing across multiple MVS system images, thus removing any one MVS system image as a single point of unavailability. Earlier versions of DB2 provide data sharing only within a single MVS system image. DB2 Version 4 Release 1 also provides row level locking. Earlier versions of DB2 use page level locking. Row level locking means that applications are less likely to find that the data they want to access is locked by another application, and so improves concurrency of access to data.

8.4.2 Copying Databases

To ensure that a table space can be recovered to a particular point, there must be a copy of it at some earlier state. This is called a backup. The principal tools for backups are:

COPY creates an image copy of a table space or of a data set belonging to a table space. It does not copy indexes, as that operation is unnecessary; indexes can be recovered from table spaces.

The data can be copied to a MVS sequential data set

- Disk
- MSS
- Tape

The COPY utility makes a **full** image copy or an **incremental** image copy, as you chose.

- Full image copy - copies all pages in a table space or a data set.
- Incremental image copy - copies only those pages which have been changed since last image copy.

It is advisable that more than one full image copy be available at all times, so that fallback recovery can occur if necessary. Concurrent updates can be made if they do not lock the entire table space.

MERGECOPY merges image copies for a table space or data set. This speeds up recovery because RECOVER will have to process fewer image copy data sets. It has two main options:

- Merge all outstanding incremental copies with the most recent full copy to produce a new full image copy.
- Merge all outstanding incremental copies to produce a new incremental image copy.

If frequent incremental image copies are created, MERGECOPY should be run regularly to limit the time required for recovery. Both COPY and MERGECOPY can create a full image copy. There are times when COPY is required, such as after LOAD or REORG. But in other cases, MERGECOPY is a valid alternative.

8.4.2.1 Alternative Backup

DSN1COPY is a service aid. It does not execute under the control of DB2 and can be executed even when the DB2 subsystem is not operational. This allows great flexibility. Because it bypasses the usual DB2 safeguards, DSN1COPY should be used with care. DSN1COPY copies DB2 VSAM data sets that contain table spaces or index spaces. Other data sets you can copy are:

- DB2 VSAM to sequential
- DB2 VSAM to other DB2 VSAM
- Sequential to DB2 VSAM
- Sequential to sequential

DSN1COPY can be useful in the following situations:

- Backing up data and index by using COPY and restoring them with DSN1COPY.
- Backing up data and index by using DSN1COPY and restoring them with DSN1COPY.

DFHSM manages your disk space efficiently by moving data sets that have not been used recently to less expensive storage. It also makes your data available for recovery by automatically copying new or changed data sets to tape, disk, or MSS backup volumes. All DFHSM operations can also be performed manually.

8.4.3 Reorganizing Databases

The REORG utility reorganizes a table space to improve access performance and reorganizes indexes so that they are more efficiently clustered. REORG does this by:

- Reorganizing table rows into clustering sequence (with some exception)
- Reclaiming space from dropped tables
- Resetting free space to their original settings
- Reorganizing index rows into physical key sequence

During the UNLOAD phase of a table space or partition, it will be available for read access only. During the next phases, the target table space is locked and contention will arise if any attempt is made to access the same table space or any of its data sets. Monitoring the table spaces gives you an idea on how to decide when REORG is needed (see 8.3.1, “Monitoring DB2 Databases” on page 90).

The reorganization elapsed time can be reduced by specifying LOG NO and larger BLKSIZE for SYSUT1, SORTOUT, and SORTWKnn data set. For very large table spaces, when index scan access is used, you should consider if it is possible to reorganize the indexes only.

8.4.4 Database Design to Avoid Reorganization

In this section we highlight those features and properties of database design that can directly affect CICS/ESA availability. For continuous availability, your database design goal should be no scheduled outage for either reorganization or backup.

A DB2 database can be designed in many different ways. It is important to understand how the design of a database can affect database availability. This section addresses physical and logical design considerations and their effect on

data availability. For detailed information on DB2 database design you should refer to *IBM DATABASE 2 Administration Guide*.

8.4.4.1 Logical Design

In this section we discuss the logical design of a DB2 database and the impact this can have on a database, and therefore CICS/ESA, availability. You should endeavor to make logical design correct before implementation as, generally, it is difficult to change after it has been implemented.

DB2 Objects: When you define a DB2 database, you name an eventual collection of tables and the table spaces in which they reside. When you create an index, DB2 automatically creates an index space for that index. There is a one-for-one correspondence between the index and index space.

In deciding whether to define a new database for a new set of objects, you should consider the fact that an entire database can be started and stopped as a unit and that the status of all a database's objects can be displayed by a single command that names only the database.

Therefore, it is often convenient to place a set of tables that are used together in the same database.

Some operations lock an entire database; for example, the LOAD utility prevents data definition statements from using the same database concurrently. Hence it may be inconvenient to place many unrelated tables in a single database. QMF users, especially, may do a great deal of data definition. For maximum availability, we recommend that each QMF user have a separate database.

In DB2, a database is a set of DB2 objects. DB2 objects that affect database availability are tables, table space and indexes.

- **Table space**

  You should create a table space before any tables are created, When you create a table space you can specify several options, some of which can affect data availability.

  A table space is an area of a database that contains one or more tables. It can be partitioned, segmented or simple. A table space divided into a number of parts is called a partitioned table space. Only one table can be stored in a partitioned table space; each partition contains one part of the table. Refer to the section "Partitioning" on page 97 for additional comments.

  A segmented table space holds more than one table, but each segment contains rows from only one table. Refer to the section "Segmenting" on page 97 for additional comments. If a table space is not partitioned or segmented, it is called a simple table space.

- **Table**

  Designing tables to be used by many applications is a critical task. Although you can add columns to tables and use views to make certain changes, generally you cannot change the design of the table after it has been implemented without adversely impacting availability.

  Both simple and segmented table spaces can contain multiple tables. Whether you put more than one table in a table space depends strongly on circumstances. When you lock a simple or partitioned table, you

actually lock an entire table space, preventing other users from accessing other tables in the same space. This can appear to a locked-out end user as loss of availability. If you are using a segmented table space when you lock a table, only that table is locked. We recommend that you put each table in its own table space and that you use the LOCK TABLE command with care.

- **Index**

  For each table you can create one or more indexes. There are three main reasons for creating indexes:

  – The most important is to improve performance, because an index may allow DB2 to make direct retrieval of table rows without scanning the entire table or table space.

  – An index ensures uniqueness of a row in a table.

  – An index can eliminate the need for sorting.

  The elapsed time needed for reorganization, recovery and load for large tables can be very critical. The number of indexes involved when reorganizing and loading tables has a significant effect on elapsed time and therefore database availability; recovery of data is not impacted by the number of indexes.

8.4.4.2 Altering Your Database Design

After using your database for a while, you may want to change some aspects of its design. You can use the SQL ALTER statement to change the following DB2 objects:

- Storage groups
- Databases
- Table spaces
- Tables
- Indexes

ALTER changes the way those objects are defined in the catalog, but it cannot accomplish every change; for example, you cannot drop a column from a table with ALTER. When you cannot make a change with ALTER, you must use the DROP statement to remove the object from the database and then use the CREATE statement to recreate the object. For more information about the ALTER command refer to *IBM DATABASE 2 Administration Guide* and *IBM DATABASE 2 Command and Utility Reference*.

8.4.4.3 Physical Design

After finishing your logical design, the next step is the physical design. Physical design is normally easy to change after implementation but still it is critical to application availability. The way space is allocated is important to continuous availability.

Disk Allocation

STOGROUP defined table spaces

A **storage group** is a set of disk volumes used to hold the data sets required for table spaces and indexes. The volumes of a storage group must be of the same device type. The description of a storage group includes its name, its volumes, and the Integrated Catalog Facility (ICF) catalog used to keep

track of the data sets. When you create table spaces and indexes, you specify the storage group from which you want space to be allocated.

Note that the first volume in a storage group must be full before a second one is used. Therefore, DB2 does not balance the load across a multivolume storage group.

You can create a storage group using the SQL statement CREATE STOGROUP.

User (non-STOGROUP) defined table space

Instead of using storage groups, you can define your own VSAM data sets with Access Method Services (AMS). You must define a data set for each of these items:

- A segmented table space
- A partition of a partitioned table space
- A simple table space
- A non-partitioned index
- A partition of a partitioned index.

Conclusions

STOGROUP is simpler to use when defining table spaces. DB2 defines, deletes, and extends the data sets in the storage group, using VSAM AMS.

STOSPACE records the space allocated to storage group table and index spaces. For user-defined spaces, STOSPACE does not record any statistics.

Managing your own DB2 data sets may be more flexible than using storage groups. You may choose to manage your own VSAM data sets for reasons like:

- You have a large table on several data sets. Under your own control, enlarge it if it becomes fully extended.
- You have a table whose size may increase or decrease significantly as you insert or delete data. If you control your own data sets, you can change the space allocation more easily.
- It is easier to move a data set from one device type to another if you do not use storage groups.

Space Allocation: The amount of space specified for a table space and how you choose to use buffer pools for a table space affect the availability of the data. This section addresses two ways to specify database space allocation, and how using them can help maintain continuous availability.

You specify the amount of table space to be allocated when a table space is loaded or reorganized using the PCTFREE and FREEPAGE clauses of the CREATE TABLESPACE statement.

PCTFREE specifies the percentage of each page that is to be reserved as free space, for use by insert and update operations. If there is no free space in a page when a row is inserted, or when an update increases the length of the row, then the row must be stored in some other page.

Every table has a clustering index. If you do not explicitly create one, the first index you create is treated as a clustering index for the purpose of insertion. The row may not be in the physical sequence of the clustering key. The performance of sequential operations on the clustering index suffers when there

are many records that are not in the physical sequence of the clustering key. For a table that can have many insert or update operations, you may want to assign a value of PCTFREE greater than the default of 5. Otherwise, you have to reorganize the table space frequently.

On the other hand, for a table that does not experience any inserts, and for which no updates lengthen a row, you can save space by assigning a PCTFREE of 0.

If you can foresee adding a column to a table after first creating it, you may want to provide additional free space in the table space. Otherwise, inserting values in the new column is likely to force rows that are not in the physical sequence of the clustering key.

FREEPAGE specifies how often DB2 is to leave a full page of free space. The space is available for inserts and updates in the same way that free space in a page would be used. It is particularly important for tables whose records are longer than half a page; in that case, only one record could fit in a page. But access to free space on the same page is faster than access to another page, so it is preferable, in general, to increase PCTFREE, and leave FREEPAGE at its default value of 0.

We recommend that you spend time during the design phase to develop the best physical design for your application requirements. This can reduce the requirement for database reorganization.

Segmenting: A segmented table space is intended to hold more than one table. The available space is divided into groups of pages called segments, each the same size. Each segment contains rows from only one table. To search all the rows for one table, it is not necessary to scan the entire table space, but only the segments that contain that table. If a table is dropped, its segments become immediately reusable. All the segments must reside in the same user-defined data set or in the same storage group.

Partitioning: Partitioning a table space provides several advantages for large tables. For example, it allows you to perform some operations, like loading data and reorganizing the space, using one part of the table at a time, and perhaps reducing the time required for the operation to a manageable time. Another reason for using partitioned table spaces is that you can put more frequently accessed data on faster devices.

We recommend partitioning of table spaces as the best option to enhance availability. Partition independence opens up the maintenance window, allowing utilities to be run on partitions of data while maintaining the availability of data in other partitions. A combination of storage management system (SMS) and DB2 utilities such as COPY and RECOVER allows an almost instantaneous availability of data.

8.5 Improving Restart Times

This section we address recovery and the types of failures that can cause unplanned outages. A detailed description is given in *DB2 Operation and Recovery Guide*, and in *DB2 Operation and Recovery Sample Procedures*.

8.5.1.1 DB2 Recovery

- **RECOVER utility**

  Use the RECOVER utility to restore a damaged DB2 object. RECOVER recovers data to its current state or to a previous state. It can act on:

  - An entire table space
  - A specific data set within a table space
  - An index
  - A single page
  - A page range within a table space that DB2 has found in error
  - The catalog and directory.

  To recover data, DB2 reads the catalog table SYSIBM.SYSCOPY to determine the most recent full image copy and any subsequent incremental image copies. These data sets are dynamically allocated. If DB2 is not able to allocate the full image copies that is needed, it attempts to use previous full image copies. This is called fallback recovery. If DB2 is not able to allocate the incremental image copy data set it needs, it will stop applying the incremental image copies, and proceed with the log data sets instead. RECOVER also recovers indexes by recreating them from the data.

  TOCOPY and TORBA are options of the RECOVER utility. Because they recover data to a prior time, and not to the present, they are referred to as **partial recoveries**. RECOVER without these options is referred to a **standard recovery**.

  The RECOVER utility will rebuild a table space. To ensure integrity, the table space should be stopped and restarted with ACCESS UT before you can run the RECOVER utility.

- **Recovering data to a prior point of consistency**

  Data can be **restored** to a prior point in time if it has been backed up appropriately. There are several ways to restore data:

  - Restoring data using the DSN1COPY service aid
  - Restoring data with a non-DB2 dump (see DFHSM)
  - Restoring data using RECOVER TOCOPY and RECOVER TORBA commands.

  After data is restored, take a full image copy in order to use the DB2 RECOVER utility for future recovery.

Chapter 9. IMS/ESA DB Considerations

We recommend that you use the DBCTL interface between CICS and IMS/ESA DB, and that you use IMS/ESA 5.1 rather than an earlier release. Using IMS/ESA 5.1 and DBCTL provides a number of advantages for continuous availability. In this chapter we discuss:

- Changing your system
- Errors
- Avoiding overstressed systems
- Single points of unavailability
- Improving restart times

9.1 Changing Your System

DBCTL gives access to more IMS database facilities, including Online Image Copy Utility and Online Change Facility.

9.1.1 Online Change Utility

The IMS online change utility enables you to update ACBLIBs, which contain PSBs and data management blocks (DMBs), and security information belonging to full function databases, without bringing down the system. This enables continued access from CICS when ACBs need to be changed. You can use the online change utility only if you are using the DBCTL interface to IMS.

For more information on this utility, see the *IMS/ESA System Administration Guide* and *IMS/ESA Utilities Reference: Database*.

9.2 Errors

You should use IMS Data Base Recovery Control (DBRC) for recovery and control purposes. DBRC reduces the chance of improper recovery by tracking the logs and image copies for you and generating the necessary JCL. See section 9.5, “Improving Restart Times” on page 105 for further information.

IMS data entry databases (DEDBs) provide a high level of availability for, and efficient access to, large volumes of data. The multiple area data sets (MADS) option allows you to replicate a DEDB up to seven times. Replication means that data continues to be available for both reading and writing even if multiple I/O errors occur.

9.3 Avoiding Overstressed Systems

To avoid overstressing your IMS subsystem, you should use the DBCTL interface (taking advantage of the performance benefits it offers), and monitor your databases regularly.

9.3.1 Use the DBCTL Interface

DBCTL uses a separate TCB for each application thread, while in the local DL/I case all requests run under the CICS main task TCB. This prevents your system from becoming processor constrained as easily, especially if you are using a multi-processor CPC.

DBCTL also provides direct access from the AORs. You do not need to function ship DL/I requests to a single DOR. The cost of making a request to DBCTL is lower than the cost of function shipping the request to a DOR.

9.3.2 Monitoring Databases

Regular database monitoring is important in a CICS continuous availability environment because it can expose factors that can lead to problems. Specifically, monitoring can:

- Provide timely warning of database space shortages
- Keep track of performance levels
- Establish
  - Initial characteristics of the database
  - Workload statistics
  - Capacity planning and performance prediction data
- Evaluate the results of tuning exercises
- Validate pointers (data content)

  Timely detection and correction of pointer discrepancies in a database can prevent unscheduled database outages. A planned outage for pointer correction will be shorter than an unplanned outage due to failure.

The database monitoring tools, such as the IBM DBTools product, can assist in determining when a database must be reorganized, or when database space allocation becomes insufficient.

9.4 Avoiding Single Points of Unavailability

There are several thrusts to the strategy for avoiding single points of unavailability. These revolve around the use of data sharing (to prevent a single application or IMS subsystem from being a single point of unavailability), and using the latest IMS utilities that do not need exclusive access to a database to function. We also need to make full use of IMS′s built-in capability for duplicating system components.

9.4.1 Data Sharing

Data sharing allows application programs running in different subsystems to make use of common databases, whether the applications are in the same MVS system image or on different MVS system images, removing any one MVS system image or IMS subsystem as a single point of unavailability. IMS/ESA 5.1 allows n-way data sharing between IMS subsystems in a parallel sysplex. IMS/ESA 4.1 and earlier releases allow two-way data sharing. DBRC provides the control functions that allow data sharing, and IRLM provides data locking.

When using the data sharing facilities of DBRC, no single system owns the database being shared. The database is, subject to specified constraints, available to all the participating subsystems. Such a database can thus be viewed as a global resource. The availability of a shared database to both the

online and batch systems is greatly enhanced, allowing more flexibility in scheduling these systems for execution.

IMS data sharing permits greater flexibility of configuration because shared databases can be available equally to all sharing subsystems. IMS data sharing allows multiple CICS regions, IMS systems, and batch jobs executing on one or more CPCs to access a database concurrently. DBRC allows two types of data sharing: Database level sharing and block level sharing.

9.4.1.1 Database Level Sharing

If you use database level sharing, DBRC controls concurrent access and update from the sharing subsystems at the database level; the resource enqueued or locked is the database. The database can be accessed for update by only one system at a time. The database is shared in one of three ways:

- One system has exclusive use of the database (PROCOPT=EX).
- One system is authorized to update the database while other systems can be authorized to read the database by specifying PROCOPT=GO and access RO. In this situation, database integrity cannot be guaranteed for the read-only systems, since the records they read may be in the process of being deleted or updated.

  Refer to 6.1.4, “IMS/ESA DB Processing Options” on page 66 for DL/I calls that may be used in these instances.
- Multiple subsystems can be authorized to read the database, all with full integrity.

Database-level sharing can be implemented across multiple CPCs in the same sysplex.

Database level sharing can impose severe restrictions on your system. Typically the database can either be available for read-only access, or it can be available to one subsystem for update and unavailable to all other subsystems.

9.4.1.2 Block Level Sharing

Block level data sharing allows multiple systems (batch or online) to concurrently access the same databases for both update and retrieval processing. As with database level sharing, block level sharing allows both intrahost and interhost data sharing. Intrahost sharing allows sharing when all subsystems are on the same host. Interhost sharing allows the sharing systems to be on different CPCs in a sysplex. Figure 12 on page 102 depicts an example of block level database sharing at the interhost level with IMS/ESA 5.1.

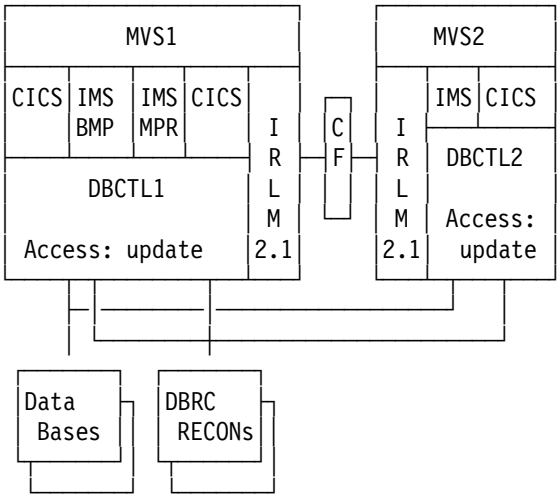


Figure 12. Database Sharing (Block level database sharing at the interhost level.)

Since block level sharing does not compromise database integrity, there must be some mechanism in the system to prevent two subsystems from updating the same database record or block (VSAM control interval) at the same time. This serialization and control is provided by a component of IMS called the IMS Resource Lock Manager (IRLM). Both DBRC and the IRLM are required for block level sharing. When block level sharing is used, IMS controls concurrent access and update at the database record level and VSAM control interval (CI) level; the resources enqueued upon and locked are the database record and the CI.

DBRC is required for both types of data sharing. In addition, all systems which share the databases must reference the same recovery control (RECON) data sets.

9.4.1.3 Selecting Data Sharing

If you want to use the data sharing facilities of DBRC, you must select the DBRC share control environment by specifying the SHARECTL parameter when you initialize the RECON data sets with the INIT.RECON command. In addition, to invoke the data sharing facilities for a particular database, that database and all of its component data sets must be defined to DBRC. You define a database to DBRC with the INIT.DB command; when doing this, use the SHARELVL parameter to specify the level of data sharing that you require for the database. This information is recorded in the RECON data set.

If a database is not to be shared, the IMS data sharing facility will ensure that only one system at a time has exclusive use of a database. Non-sharing is a special case of database level sharing, and is a significant database integrity enhancement, since it can prevent a database from being accessed in case of errors, such as:

- After a batch DL/I ABEND but before database backout has been performed.
- After a disk I/O error but before the database has been recovered.
- After a database reorganization but before an image copy has been made to establish a known recovery point.

9.4.2 Copying Databases

A database backup is the prime requisite for recoverability in case of failures. Without a backup, it may be impossible to recover the data lost.

The online image copy utility creates an as-is copy of your database while it is being updated. This utility is used for HISAM, HDAM, and HIDAM databases only. In IMS/ESA 4.1 and above you can use concurrent image copy for full function DL/I databases, while your database is being updated. The online and concurrent image copy can be used for recovery purposes. IMS concurrent copy creates a fuzzy copy with no impact on data availability. DFSMS concurrent copy creates a sharp copy, but you must stop the database before you can use it.

9.4.3 Reorganizing Databases

Database reorganization can be a major contributor to planned database unavailability. Reorganization is desirable for a number of reasons, such as improving performance, or improving space utilization of the database space.

Reorganization can be time consuming, so you should use DEBDs if you can (the DEDB direct reorganization utility enables you to reorganize DEDBs while CICS is updating the data), or design the database so that it does not require frequent reorganizations (refer to 9.4.4, “Database Design to Avoid Reorganization” for recommendations), or choose a fast method of reorganization. IBM offers a number of utilities and program offerings for fast reorganization, including:

- DBTools
- HISAM Reorganization Unload and Reload Utilities
- HD Reorganization Unload and Reload Utilities
- Partial Database Reorganization
- IMS/VS High Speed Sequential Retrieval (HSSR)
- Fast Reorganization Reload II

HSSR and Fast Reorganization Reload can reorganize databases faster than the IMS supplied utilities. They provide better data availability because databases are unavailable for shorter periods of time.

9.4.4 Database Design to Avoid Reorganization

In this section we highlight those features and properties of database design that can directly affect CICS/ESA availability. For continuous availability, your database design goal should be no scheduled outage for reorganization.

Factors that can have significant impact on continuous availability that you should consider when designing a DL/I database are:

- Database organization
- Logical relationships
- Space allocation.

For detailed information on IMS database design, you should refer to *IMS/ESA Version 4 Database Administration Guide* or *IMS/ESA Version 5 Administration Guide: Database Manager*.

9.4.4.1 Data Base Organization

Evaluate the database organization according to the following considerations:

- **HISAM**

  Segments deleted from a HISAM database are not physically deleted, they are only flagged as deleted. Such space can only be reclaimed at reorganization time. This may result in unavailable space in the database if the database cannot be reorganized frequently, due to the need for continuous availability.

  Do not use HISAM if you expect a high number of root insertions or segment deletions.

  The result of large numbers of insertions may be many VSAM CI/CA splits, requiring a reorganization to improve performance.

- **HDAM/HIDAM**

  The HDAM/HIDAM organizations differ from the HISAM organization in that the space freed by deletions can be reused, making these organizations well suited for continuous availability.

  Even with HDAM/HIDAM organizations, space allocations must be carefully planned to minimize the reorganization requirements. Refer to 9.4.4.3, “Space Allocation” on page 105 for more details.

  If you need to process the data sequentially, as well as randomly, HIDAM is the preferred access method. Alternatively, you can use HDAM with a sequential randomizer.

- **DEDBs**

  DEDBs are unique in providing online database reorganization. You must use DBCTL to use DEDBs; they are not supported by local DL/I.

  The Concurrent Image Copy utility, applicable also to full function databases, allows databases to be copied without them having to be taken offline.

9.4.4.2 Logical Relationships

The type of pointers used in logical relationships can affect the length of a scheduled outage. You can use either direct or symbolic pointers.

If you need to reorganize a database and it is in a logical relationship using direct pointers, then all of the databases that are related to that database must be offline during reorganization.

If you use symbolic pointers, a logical parent database can be reorganized without affecting the logical child database. This results in fewer databases being unavailable, and for a shorter period.

We recommend that applications intended for continuous availability should not use logical relationships but, if they must, they should use symbolic pointers to reduce the requirement for scheduled outages.

9.4.4.3 Space Allocation

The amount of space allocated to a database and the way it is allocated can affect the frequency of scheduled outages. This section addresses two ways that correct database space allocation can maintain continuous availability.

- Initial free space for HDAM/HIDAM organizations

  After initial load of a HDAM/HIDAM database, a dependent segment insert is put as close as possible to the segments it is related to. This minimizes the I/O time needed to retrieve it. As the database grows, available space decreases and newly inserted dependent segments are put further away from their related segments. This slows performance and can only be eliminated by a database reorganization.

  To minimize the effect of such inserts, free space can be allocated within the databases originally. This reduces the performance effect of inserts, which in turn reduces the need for frequent reorganizations.

- Maximum size of databases

  The maximum size of a database is:

  – For DEDB: 960 gigabytes
  – For VSAM: 4 gigabytes
  – For OSAM: 4 gigabytes.

  If a database data set is approaching these limits, the options are:

  – For VSAM or OSAM, consider using DEDB if at all possible
  – Divide the database data set into multiple data sets
  – Use the IMS edit/compression exit to compress segments before writing them to disk
  – Use the IMS Compression - Extended Release 2 product (5655-085) to compress data before it is written to disk and to expand data on retrieval.

It is not possible to operate and update a DL/I database under CICS indefinitely. At some time you must schedule time for database backup and reorganization. If you do not do this, the recovery time after a period of unscheduled unavailability may be increased beyond the limits of acceptability. Reorganization of the database is also required to maintain the performance of the database at reasonable, consistent, and planned levels.

Unplanned outages will sometimes necessitate database recovery. The recovery needs to be as fast as possible to provide the smallest possible outage. In 9.5, “Improving Restart Times” we look at the recovery facilities of DBRC, and present some planning guidelines.

9.5 Improving Restart Times

The recovery control facilities of DBRC provide a high degree of control over the recovery of IMS databases. This control extends not only to the execution of the recovery operations themselves, but also to the previous executions of all of the other, recovery-related IMS utility programs necessary to enable recovery of databases.

Without DBRC, extensive manual recording has to be made of all of the logs created by systems which have updated databases. Manual recording is also

needed for the progress of the recovery related utilities such as Image Copy, Change Accumulation, and so on. All of this manually recorded information is necessary to enable the construction of a valid recovery operation. If some of this information is incorrect or missing, the wrong input can easily be supplied to a recovery operation. An invalid recovery can then result, which in turn will result in a loss in the integrity of the database.

The recovery control facilities of DBRC are designed to extensively reduce the manual effort required to manage this extremely important operational area. In so doing, DBRC significantly reduces the probability of an invalid recovery being made.

To assist in continuous availability of the CICS system, the recovery control facilities of DBRC can be used without necessarily also using DBRC′s data sharing facilities.

9.5.1 Types of Recovery

Three types of recovery are supported for database data sets controlled by DBRC. These are:

- Full recovery
- Time-stamp recovery
- Track recovery

DBRC GENJCL and utility JCL verification support is provided for all three recovery types.

The inputs to recovery are:

- The latest appropriate image copy data set.
- The latest appropriate change accumulation data set.
- All required logs concatenated in the correct sequence.

If using the optional DSLOG facility of DBRC, the DSLOG data sets would be used instead of any logs that had been processed by the DSLOG utility.

Full recovery

A full recovery of a database data set restores the data set to its current state as indicated by the RECON data set.

When the database recovery is executed, DBRC will validate the JCL to ensure that all the correct inputs have been supplied in the correct order. Additionally, DBRC will create a record of the recovery in the RECON data set when the recovery successfully completes.

Time stamp recovery

A time stamp recovery of a database allows recovery to some point other than its current state. Time stamp recovery is typically used in a batch environment to restore the database(s) to the state that they were in at the beginning of a batch run. If, for example, step N of a job abends, or had the wrong input, time stamp recovery can recover the database to the status prior to step N, without running batch backout for all steps preceding step N.

There are several caveats in the use of time stamp recovery, and the reader is referred to manuals referenced at the beginning of this section and at the beginning of this book for more details.

Track recovery

DBRC also provides support for the track recovery option of the IMS Data Base Recovery utility. Track recovery is used to recover one or more ″bad″ tracks on a VSAM data base data set to its current state. Since the entire database data set is not being recovered, a time stamp track recovery cannot be performed.

9.5.2 Recovery Strategy

Because of the way the Image Copy and Log Merge utilities operate, the recovery strategy for a continuous availability data sharing environment must be carefully considered. The basic consideration is which of the following two alternatives result in better data availability.

9.5.2.1 Regular Change Accumulation, Infrequent Image Copy

If you use this strategy you perform infrequent image copy of databases but perform regular change accumulation Change accumulation can be performed using either of the DBRC Data Set Log (DSLOG) utility, or the Data Base Change Accumulation utility (DFSUCUM0). In general, the DSLOG utility is only useful if database activity is relatively low. The IMS database change accumulation utility, DFSUCUM0, allows you to collect the changes from multiple log data sets onto a single log, thus helping to speed up recovery. You will have to balance the benefits of using it against the overhead it incurs, and the fact that you may not need to use its output.

This strategy can have a faster recovery time if you frequently update a small number of records.

9.5.2.2 Frequent Image Copy, Change Accumulation When Needed

This means taking frequent image copies of the databases and only performing change accumulation if needed for database recovery.

This strategy will have longer recovery time if the databases are shared. The criterion for deciding whether this strategy is viable, is the number of sharing systems, and therefore the number of logs that must be merged for recovery purposes.

The more logs that need merging, the longer recovery will take.

Chapter 10. VSAM Considerations

Many CICS applications hold their data in VSAM data sets. This is particularly true of packages and applications written by third party application vendors. In this chapter we discuss VSAM considerations relating to achieving continuous availability, especially:

- Errors
- Single Point of Unavailability
- Extended Restart

10.1 Errors

CICS/ESA introduced two functions that help deal with errors in processing VSAM data sets; backout failure control and batch backout. You should also use a product like CICSVR to automate and control recovery of your VSAM data sets if you need to recover them for any reason.

10.1.1 Backout Failure Control and Batch Backout

These functions were introduced in CICS/ESA. If CICS cannot write to a VSAM data set during dynamic transaction backout (DTB) or an emergency restart, the data set needs to be recovered. Before CICS/ESA, you had to ABEND CICS to ensure data integrity, since the only way to recover was to forward recover the data set, and then emergency restart CICS (to backout any uncommitted changes).

CICS/ESA improved the process. Now if a failure occurs during DTB or emergency restart, CICS continues but marks the data set as being in need of recovery. You can recover the data set offline by forward recovery, followed by a batch backout of uncommitted data (using a CICS-supplied utility). When you have recovered the data set, you can make it available to CICS again.

This support means that only those applications that need to access the failed data set are affected, rather than all applications that use this region.

10.1.2 Automation of the Recovery Process

CICS VSAM Recovery MVS/ESA (CICSVR), Program Number 5695-010, helps you to recover lost and damaged VSAM data sets.

CICSVR:

- Automatically determines the backups and journals required for recovery, if you have DFSMShsm
- Performs forward recovery to recover VSAM data sets that have been lost or damaged
- Provides backout to remove uncommitted updates present in the VSAM data sets
- Works with backups taken using the backup-while-open (BWO) facility
- Updates VSAM spheres directly to the base cluster. It applies all changes that were made to the base cluster, including those that were made to paths.

- Recovers multiple data sets in a single run
- Recovers VSAM KSDS, ESDS, and RRDS
- Can work independently of CICS
- Produces reports showing job statistics.

10.1.3 Monitoring

VSAM data sets should be monitored for potential problems that could cause an outage. You can obtain the information you need for monitoring from:

- The output from the AMS command LISTCAT
- The CICS statistics produced at CICS shutdown or using the CSTT command
- VTOC information provided with MVS utilities or using online dialogs with ISMF or ISPF

Typical problems that may occur in a VSAM data set are:

- No more usable space on the volume.

  If the available space on the volume(s) allocated to the data set is exhausted, or is fragmented into extents smaller than can be successfully allocated, an outage may occur.

  To avoid this situation we recommend that you implement the space management functions of DFHSM. This will provide an agreed upon amount of space for the application users through the deletion of old data and the migration of inactive data to DASD or tape.

  An alternate approach is to under allocate space on volumes (that is, to only allocate 80 percent of the available space on a volume), to avoid the situation.

- No more usable space in the data set.

  Monitor the allocated RBAs (Relative Byte Addresses) versus the high-use RBA for each extent. It may be that there is no more space in which to perform a Control Interval (CI) or Control Area (CA) split, or that all the available space is distributed across the data set in places where it does no good.

  You can circumvent this by revising the space allocation parameters for the data set, as well as the free space specification. You can also use AMS EXAMINE to monitor space usage within a data set.

- Excessive numbers of CI or CA splits.

  This may cause excessive delay for a transaction; also, the number of I/O operations to effect a CA split is very high, because VSAM has to effectively read in the whole CA, and write it out again as two CAs, as well as updating the index.

  Correct this problem by reorganizing the data set using a REPRO or IMPORT AMS command.

10.2 Single Point of Unavailability

You should consider several points when creating your strategy for avoiding single points of unavailability. You need to consider data sharing (to prevent a single application or subsystem from being a single point of unavailability), using backup while open to make copies of your data sets, and avoiding the need to reorganize your data sets.

10.2.1 Data Sharing

CICS provides very good support when sharing VSAM data sets between users from within a CICS region. VSAM sees each CICS region as a separate user, and CICS provides the functions needed to allow multiple tasks within the region to access the same data set at the same time with integrity.

Currently VSAM processing limits your ability to share a VSAM dataset between multiple users (multiple CICS regions or between CICS and batch). VSAM does not provide any logging mechanism, has very limited locking, and no understanding of the logical unit of work concept. This forces you to use a FOR if you require access to an updatable VSAM file from more than one CICS region with integrity. In this case the FOR is, again, a single user, and there are no integrity problems. If you update a VSAM file from CICS and batch simultaneously, neither CICS nor batch guarantee data integrity. You must develop your own mechanisms for enqueueing upon records, and then ensure that all applications wishing to use the data set go through the enqueueing mechanism. Even if you do this, the batch program is not producing any log records to record the changes it is making, so you cannot recover the file in the event of a failure.

In practice this means that it is not possible to do batch updates to a data set without de-allocating it from CICS, or without using CEMT to close the data set and re-open it as read-only, or at the best, without writing additional code in all the CICS applications as well as the batch applications requiring concurrent access to the file. You can minimize the time a VSAM data set is unavailable for online update because of batch update requirements by using the function provided by the external CICS interface (EXCI). This allows you, from a batch program, to have an online VSAM data set closed or changed to read-only by CICS/ESA. You can then perform the batch updates and, when completed, use the EXCI to return the data set to its original online status. Refer to section 6.3.4, “Using the External CICS Interface” on page 71 for more detail.

10.2.1.1 VSAM SHAREOPTIONS

VSAM SHAREOPTIONS (1) provides full integrity since it maintains both read and write integrity. VSAM SHAREOPTIONS (2) provides only write integrity. VSAM SHAREOPTIONS (3) or (4) do not guarantee data set integrity, they merely allows sharing. You have to provide for data set integrity in your applications requiring access to the data set with these SHAREOPTIONS. Refer to *DFSMS/MVS Access Method Services for VSAM Catalogs* for more detail about SHAREOPTIONS.

10.2.1.2 CICS and Record Level Sharing

IBM has a statement of direction regarding CICS/ESA and VSAM record level sharing, that address some of the restrictions when working with VSAM. The future support will provide:

- Improved performance as AORs can directly access the VSAM data sets
- Removal of the FOR as a single point of failure
- Improved availability:
  - Data sharing across CPCs allows transactions to be routed to AORs on another MVS system image and still have access to VSAM data sets
  - A backout failure only affects the records within the LUW not the whole data set
  - Reduces the scope of deadlocks, as records are locked, not CIs.
- Improves sharing between CICS and batch
- Less system management required
- Improves integrity.

10.2.2 Copying Data Sets

Backup while open (BWO) enables you to take a backup of the file while CICS has the file open for update. When running CICS for long periods of time, BWO reduces the amount of time taken to recover a VSAM file if a failure occurs. Only the updates after the BWO copy has started would need to be recovered, to bring the VSAM file to the state it was in just prior to the failure.

BWO produces a fuzzy copy of the data set. For KSDS files a CI or CA split while the backup is in progress invalidates the backup (you could possibly get either two copies of a record, or none, depending on where the record is moved to during the backup). In this case you need to repeat the backup.

10.2.3 Reorganizing Data Sets

You must make a VSAM file unavailable to CICS if you need to reorganize it or update it from a batch program. We recommend that you take a copy of the VSAM file before and after any batch changes. This gives you the ability to recover, without rerunning the batch job.

10.2.4 Data Set Design to Avoid Reorganization

In this section we highlight those features and properties of data set design that can directly affect CICS/ESA availability. For continuous availability, your database design goal should be no scheduled outage for reorganization.

You can do the following for applications that are written to use VSAM data sets:

- If many records will be added to the data set, allow enough space during initial allocation for expansion. DFP allows for more than 16 extents; allocate sufficient space in one extent and use the secondary extents only to prevent program failures.
- If record insertion is evenly distributed over the data set, initially allocate the data set with FREESPACE; this alleviates the impact of CI or CA splits, and allows longer intervals between data set reorganizations.
- If all the insertions are clustered closely together, then allocate the data set with FREESPACE(0 0). After the existing records have been loaded via REPRO, use the AMS command ALTER and add free space to the cluster. This optimizes the space usage and still serves to reduce the frequency of CI and CA splits.

10.3 Improving Restart Times

The method you use to recover a data set, and the time taken, depends on the type of backup you take. Each type of backup has its own advantages and we shall briefly discuss the available alternatives for backup as well as the recovery scenarios.

- Backup using DFHSM.

  DFHSM uses the EXPORT command internally to create a backup copy of the VSAM data set. The primary benefit of DFHSM is that the end user can initiate a backup or a recovery operation without intervention. If the backup exists online, the recovery operation does not even require a tape mount, much less the submission of JCL or the lookup of which data set contains the most current backup.

- Backup using DFDSS.

  You can dump DASD data to a sequential data set with the DUMP command of DFDSS. You can dump data sets, a full volume, or a range of tracks. Depending on the data organization and the DFDSS parameters, DFDSS will dump only the allocated and used space or the allocated and unused space For example, all allocated space for VSAM data sets is dumped; unallocated space is not dumped by DFDSS. The types of dump you can perform are:

  - Full volume dump - all data sets on the volume are dumped.
  - Track dump - a range of tracks is dumped. The application of this function is limited.
  - Data set dump - selected data sets using filtering are dumped. This is a powerful function. For example, DFDSS logical processing can be used to dump all VSAM data sets that belong to an application without being concerned about the device type and volumes on which the data set reside. The primary use of such a backup is when a volume is lost or for disaster recovery.

- Backup using AMS command.

  AMS provides the backup and restore of a data set in two different ways, EXPORT followed by IMPORT, or REPRO.

  EXPORT temporarily creates a portable copy of the data set that has its own definition imbedded in the copy. You do not need to delete or redefine the cluster if an exported copy is to be imported. In addition, most of the parameters originally specified in the DEFINE CLUSTER command can be altered in the IMPORT command. The backup copy is not usable in its exported state, and can only be processed by the IMPORT command.

  REPRO creates a sequential data set that can be used in its external form. If the data set is lost, you need to redefine the cluster before the unloaded copy can be reloaded using another REPRO command.

  The primary attraction of both of these facilities is that they are under the user's control, and they can therefore initiate a backup whenever required.

We recommend that you use DFHSM to automatically perform the backups of the VSAM data sets. A CICS time-initiated task using the command level systems programmer API can close and de-allocate the data sets before the specified time that the backup operation will commence. At a later specified time, the program can then re-open the data sets and processing can continue.

If reorganization is required, define a new cluster and change any parameters if necessary. Close the data set using CEMT (or using the EXCI) and REPRO the data set to the new cluster from the old one. After successful completion, change the name of the data set in the FCT via CEMT and set the status of the data set to ENABLED. On the next reference, CICS opens the data set and processing continues.

Do not forget to also change the name of the data set in the FILE definition, or else, on the next COLD start CICS will recover the data set name from the CSD.

10.3.1 Forward Recovery of Data Sets

If after-images of changed or new records are placed in a journal, either to the system log or to a separate user journal, user programs are required to support the forward recovery process, together with operational procedures for the use of those programs. These user programs, or, in some instances, VSAM Access Method Services (AMS) utilities, are required for the following functions:

- Taking backup copies of the files periodically.
- Journal changes made by batch jobs.

  Note: If you have facilities that allow concurrent access, the log(s) created must reflect the correct sequence of alterations to the data set.
- Extracting after-images from the CICS system logs or user journals and maintaining the correct logical sequence of the after-images.

  Note: If speed of forward recovery is important, then you should send the after-images to a user journal (DFHJ02x - DFHJ99x) and not to the system log. This reduces both the amount of processing needed to extract forward recovery data and the emergency restart time.
- Applying the after-images to the backup copy to reconstruct the file up to the point of failure.

If the system is used 24 hours per day, these procedures may necessitate taking files offline to make backup copies. Alternatively, housekeeping operations can be scheduled for a day when the system is not in use.

Chapter 11. Batch Processing Considerations

In this chapter we discuss techniques to minimize the impact of scheduled outages by reducing the batch window. We cover BatchPipes and some alternative techniques.

11.1 BatchPipes

Batchpipes minimizes the impact of scheduled outages by speeding up the processing of batch jobs and thus giving the potential to reduce the batch window.

BatchPipes offers a way to connect jobs so that data from one job can move through processor storage to another job without going to DASD or tape. It addresses a growing problem that faces installations with batch workloads: insufficient time to complete that work. Given a batch job stream with a data flow of certain characteristics, BatchPipes can dramatically shorten the elapsed time of the job stream. The jobs can run faster and process larger volumes of data in the available batch window; your processors are freed to run more work, perhaps extending the period of time that interactive applications are available or supporting your company's work in another time zone. In short, you might get more work from your processors. BatchPipes, running on MVS/ESA may:

- Allow two or more jobs that formerly ran serially to run concurrently.
- Reduce the number of physical I/O operations by transferring data through processor storage rather than transferring data to and from DASD or tape
- Reduce tape mounts and use of DASD.

What difference does parallel processing make to my batch workload?

Large tightly-coupled processing systems running with MVS can have many jobs in various stages of processing at one time. Batch applications have not traditionally taken advantage of this multiple processing capability. Rather, they often consist of multiple jobs that run one after another. With BatchPipes, the processor can run two or more jobs in a job stream at one time. Jobs that once ran sequentially can now run in parallel because data records or blocks are available to the next job immediately after they are written. That is, the whole sequential file does not have to be written and then closed before the next job can access the file.

What difference does keeping data in the processor make to my batch workload?

When data moves from one job to another through processor storage, the transfers take place in microseconds as compared to the milliseconds required to transfer data to and from tape or DASD. Keeping data in processor storage reduces the number of physical I/O operations, causes less I/O contention, and frees the device that holds the intermediate data set. An additional benefit of transferring data through processor storage is a reduction in the mounting and managing of tapes for applications using BatchPipes.

For a complete introduction to BatchPipes see *IBM BatchPipes/MVS Introduction*.

11.1.1 How BatchPipes Changes Traditional Batch Processing

Most installations have batch job streams that use intermediate data sets where output from one process (a job or step) is written to DASD or tape. This data is then read by a second process immediately after the data set closes or the first process ends.

Consider the following example of traditional batch processing. Job1 writes data to an I/O device. When all the records have been written and the data set is closed, Job2 starts. Job2 reads the data from the device. After Job2 finishes processing all the records, the data set is no longer required and can be deleted. In effect, the transfer of data from one job to the other is at a data set level.

Compare this with the same two jobs using BatchPipes. External storage devices are replaced by a processor storage buffer holding a small portion of the intermediate data set. In this case, Job1, a writer to the processor storage buffer, and Job2 , a reader from that buffer, run concurrently. Job2 can obtain data from the processor storage buffer as soon as Job1 writes the first block. Output from Job1 becomes immediately available as input to Job2. You can think of the data as ″flowing″ from Job1 to Job2. The processor storage buffer is called, in BatchPipes terms, a pipe, through which data flows, always in the same direction, from a writer job to a reader job. Writer-pipe-reader are known as a pipeline. To understand the elapsed time savings, compare timelines of traditional job-to-job processing and BatchPipes job-to-job processing. Figure 13 shows the timeline for the traditional processing of Job1 and Job2.

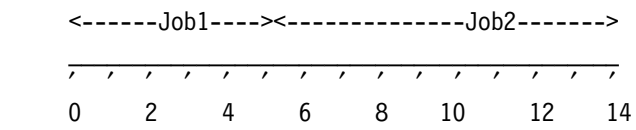


Figure 13. Timeline Showing a Traditional Two-Job Stream

With BatchPipes, the two jobs can run concurrently in less time than Job2 (the longer-running job) formerly needed. Elapsed time savings come from concurrent processing of the two jobs and from elimination of I/O wait times (time intervals during which a job waits for I/O transfers to and from DASD or tape), and reduction of I/O transfer times between jobs. Figure 14 shows the timeline for the two jobs using BatchPipes.

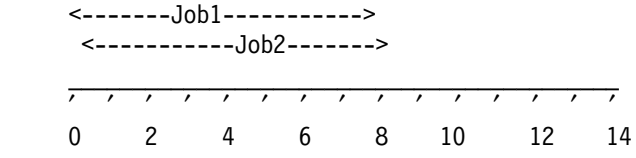


Figure 14. Timeline Showing Two Jobs Using BatchPipes

Running the two jobs concurrently demonstrates a form of parallelism, the simultaneous processing of many tasks (or jobs) within one application.

11.2 Alternatives to BatchPipes

BatchPipes is one of several software and hardware functions that IBM offers to improve the elapsed time of batch jobs that run on MVS. Some of your other options are:

- VIO to expanded storage
- Hiperbatch
- Batch Local Shared Resources (BLSR)
- Sequential Data Striping.

To answer questions you might have about how BatchPipes compares with these functions, this section looks at whether these functions benefit the same type of batch job stream that BatchPipes benefits, a job stream where:

- The jobs use sequential QSAM or BSAM to access the intermediate dataset
- Output from one job or job step is input to a second job or job step
- The data set is intermediate, that is, the jobs or job steps do not need a permanent copy of the data for backup or recovery.

Both **VIO to expanded storage** and BatchPipes target similar batch environments. They eliminate the I/O wait times that occur when a job waits for I/O transfers to and from storage devices; they handle data that is intermediate. VIO keeps all the data in expanded or auxiliary storage from the time the data is created until the entire data set is deleted; BatchPipes keeps only a small portion of the data set in virtual storage at one time. Therefore, there is no limit to the amount of data that BatchPipes can transfer.

VIO supports both random and sequential I/O; BatchPipes supports only sequential I/O.

VIO improves performance for jobs that use data sets that are relatively small. If the job requires large quantities of data (for example millions of records), VIO is not suitable because expanded or auxiliary storage will be exhausted, or because the storage will be paged (eventually to DASD) and performance could actually decrease.

Because BatchPipes allows writer and reader jobs to run concurrently, elapsed times generally improve more with BatchPipes than with VIO.

Hiperbatch targets a different type of batch application than BatchPipes. With Hiperbatch, the target application has many readers simultaneously accessing data that one writer job created. The shared data all resides in processor storage and on DASD; the data persists after being read. In contrast, for BatchPipes, the target application has one writer passing data to one reader. BatchPipes holds very little of the data in processor storage at one time, and the output from the writer job is temporary.

With Hiperbatch, readers run simultaneously with each other, but not with the writer; with BatchPipes, writer and reader jobs run simultaneously.

BLSR also targets a different type of batch application than BatchPipes. With BLSR, the target application uses direct or keyed VSAM in a non-sequential access pattern. In contrast, BatchPipes supports QSAM and BSAM in a completely sequential access pattern. Further, for data integrity reasons, BLSR

writers and readers must run serially. BatchPipes writers and readers must run concurrently.

BLSR is not generally suitable for sequential access patterns (because the performance could actually decrease) and BatchPipes is not suitable for random access patterns.

Sequential Data Striping benefits batch jobs that spend much of their elapsed time sequentially reading or writing large DASD files. When a data set is created, Sequential Data Striping allocates the data set on as many as 16 volumes. Jobs that later read the data have their read requests processed in parallel from those devices. Sequential Data Striping automatically ″reassembles″ the records as the data is transferred from DASD. Once all the records are reassembled, Sequential Data Striping returns control to the job.

With Sequential Data Striping, the data is kept on DASD; with BatchPipes, the data is temporary. Sequential Data Striping reduces I/O operations and I/O wait times; BatchPipes eliminates them.

Chapter 12. Hardware That Supports Continuous Availability

This chapter describes the hardware features that support continuous availability. It covers the following topics:

- Parallel sysplex
- 3990 storage control functions
- RAMAC DASD
- Environmentals.

12.1 Parallel Sysplex

A sysplex is a set of MVS systems communicating and cooperating with each other through multisystem hardware components and software services.

A sysplex implementation without a coupling facility is referred to as a base sysplex. When the implementation includes a coupling facility, it is called a parallel sysplex. For a description of a coupling facility see 12.1.4, “Coupling Facility” on page 121.

In a parallel sysplex, workload balancing can be dynamic, with available capacity being used when required, and an application can be spread across multiple systems for higher availability. For example in a CICS/ESA environment with cloned AORs, your workload would run regardless of CICS, MVS system or hardware failure. Parallel sysplex can also eliminate outages for planned system changes, by allowing the removal and reintroduction of a system nondisruptively.

High performance data sharing is fundamental for parallel processing. The coupling facility allows database managers and MVS components to implement high performance protocols to share data. This removes a FOR as a single point of failure in a CICS/ESA environment, increasing availability.

Hardware in a sysplex that implements coupling facility data sharing includes the following:

- Processors

  Processors which support a coupling facility include the 9021 711-based models, the 9121 511-based models, and the parallel CMOS processors. They include the Processor Resource Systems Management (PR/SM) logically partitioned mode (LPAR) feature. PR/SM LPAR allows you to define a special logical partition called a coupling facility logical partition (coupling facility LP) that provides coupling facility functions. More information can be found in B.1, “Parallel Transaction Servers” on page 155.

- Sysplex timer.

  The sysplex timer coordinates time-of-day clocks for systems in the sysplex.

- ESCON director and control units

  The ESCON director controls switching of point-to point connections between processors and control units.

Control units are needed for direct access storage devices (DASD), tape drives, printers, consoles, communications, and work stations.

- Coupling facility.

  The coupling facility provides data sharing functions for multiple systems in the sysplex.

- Coupling facility channels.

  Coupling facility channels connect the processors to the coupling facility.

These components are discussed in more detail in the following sections.

12.1.1 Processors

Parallel CMOS, 9021 711-based, and 9121 511-based processors allow you to use coupling facility channels to connect processors to the coupling facility.

At the system level, many functions are implemented that prevent or tolerate errors, thus increasing availability. Examples of these functions are:

- Subsystem storage protection (SSSP) - This is a generic protection capability. It implements storage protection for multiple components within a single address space. It is exploited by CICS/ESA 3.3 (or later) to protect CICS control blocks and data from user applications.
- CICS transaction isolation - This facility extends the protection provided under SSSP to protect one CICS user application from other user applications within the same CICS address space. It is exploited by CICS/ESA 4.1 (or later).
- Dynamic reconfiguration management (DRM) - DRM allows an installation to modify its hardware and software I/O configurations dynamically for devices, control units and channel paths without requiring a power-on reset (POR) of the hardware or IPL of the MVS/ESA operating system.
- LPAR logical CP vary on and off - This facility allows the reconfiguration of dedicated CP resources between partitions without partition deactivation.

We recommend parallel CMOS processors as they offer availability advantages over bipolar processors. Parallel CMOS processors:

- Do not require water cooling
- Allow processors and LPs to be added without interruption
- Extend parallel processing
- Allow non-disruptive upgrade of licensed internal code (LIC) (patches and engineering changes).

See B.1, “Parallel Transaction Servers” on page 155 for a list of the functions and features that support continuous availability.

12.1.2 Sysplex Timer

The IBM 9037 sysplex timer coordinates time-of-day (TOD) clocks for MVS systems that run on separate processors or CPCs in the sysplex.

The sysplex timer is a table-top unit that synchronizes the TOD clocks in up to 16 processors, processor sides, or System/390 (S/390) microprocessor clusters. The sysplex timer also provides automatic setting of the time and date in the processor.

Correct execution of MVS functions in a multisystem sysplex requires that all TOD clocks on all central processing complexes be synchronized. All processors in the sysplex must be connected to the same sysplex timer. The time in a sysplex timer can be periodically set from an external time source. This allows multiple sysplexes to keep their times synchronized by having all sysplex timers controlled by the same external time source.

We recommend that you duplex the sysplex timer and make sure that you have independent power supplies, otherwise it becomes a single point of failure.

12.1.3 ESCON Director and Control Units

The ESCON director performs dynamic switching for point-to-point connections between processors and control units. In a sysplex, ESCON directors can connect processors to other processors, and processors to control units used for storage or communication control.

ESCON capable storage control units control access to DASD or tape and provide functions that improve the availability of your system. See section 12.2, “3990 Storage Control Functions” on page 122 for details.

ESCON capable communication control units control access to terminals, allowing end users to enter OLTP transaction requests, and to networks for remote processing.

We recommend that you connect two or more ESCON directors to each processor and each control unit whenever possible, otherwise it becomes a single point of failure.

12.1.4 Coupling Facility

The coupling facility provides locking, caching, and list services between coupling-capable S/390 processors running MVS/ESA Version 5. Coupling facility channels are used to connect the coupling facility to the coupling-capable processors. The coupling facility functions are provided by:

- The IBM S/390 Coupling Facility 9674 which is a dedicated coupling facility; no operating systems run on it.
- The 9021 711-based processors and the S/390 parallel transaction server 9672 processors that can have a PR/SM logical partition (LP) running the coupling facility control code (CFCC - Licensed Internal Code) and thus make the LP a coupling facility.

We recommend a 9674 as the coupling facility to be used in production data sharing configurations.

The 9674 coupling facility is designed to run as a single, dedicated coupling facility to reduce the risk of long outages, achieve acceptable performance, provide large capacity shared storage, and maximize connectivity to the coupling facility.

For high availability, we recommend you to install a second, similarly configured, 9674 coupling facility to reduce the possibility of a single point of failure. A second 9674 improves application software availability by allowing fast recovery from one coupling facility to the other in the event of a coupling facility outage.

12.1.5 Coupling Facility Channels

Coupling facility channels connect the coupling facility to the coupling-capable processors:

- The 9021 711-based processors
- The 9121 511-based processors
- The 9672 parallel CMOS processors

Coupling facility channels are S/390 channels that use high bandwidth fiber optic cables to provide fast point-to-point connectivity between the coupling facility LP and the connected processors or LPARs. Standard ESCON I/O protocols are not used for the coupling facility data transfer.

Coupling-capable systems can be added or removed dynamically from the coupling facility. A correctly configured coupling facility allows coupling facility channels to be hot plugged without disruption.

12.2 3990 Storage Control Functions

IBM 3990 storage controllers provide a fault-tolerant DASD subsystem. The 3990 provides enhanced capabilities for concurrent or nondisruptive maintenance and upgrade. The Model 6 provides higher levels of performance for high availability functions such as fast dual copy and concurrent copy. CMOS technology provides significant improvements to the hardware reliability. Availability is improved through a variety of enhancements in both the hardware and the Licensed Internal Code.

IBM 3990 Storage Controllers (3990 models 3 and 6) provides track caching, DASD fast write, concurrent copy, dual copy, and sequential data striping. See section 11.2, “Alternatives to BatchPipes” on page 117 for more information on sequential data striping. The 3990 model 6 also provides extended remote copy (XRC) and peer-to-peer remote copy (PPRC).

The following 3990 storage control hardware functions contribute to continuous availability:

- Concurrent copy
- Dual copy
- Remote copy

12.2.1 Concurrent Copy

Concurrent copy is a 3990 extended function designed to facilitate the copying of important data. By significantly reducing the time required for backup (from possibly hours of planned unavailability to just minutes in most cases), concurrent copy can greatly increase the scheduling flexibility of online operations and batch processing in an MVS/ESA environment. Concurrent copy is available on all cached models of the 3990.

Concurrent copy can dramatically reduce the amount of time that is required to back up your data, hence increasing the time available for online service. When you use concurrent copy, application processing is interrupted for only a short period while the system initializes the concurrent copy environment. Once

concurrent copy is active, your applications can continue to process the data while it is being backed up using concurrent copy.

You can use concurrent copy to produce the backup copies you now make before and after batch update runs, as well as backups taken after online processing completes for the day. By using concurrent copy, either the available window for online batch processing may be extended to other tasks, or the period of availability of the online systems may overlap with the execution time of concurrent copy. The scenario followed varies considerably, depending on the operational, availability, and performance requirements of your enterprise.

Another potential use is to produce a spin-off copy of a data set or database for use by query applications or analysis programs. You may have to freeze the database so that all queries produce consistent results, while still allowing the online systems to update the working copy of the database. Also, for performance reasons, you may not want to allow complex queries to contend with the production, high-performance copy of the database. With concurrent copy, you can produce such spin-off copies with minimal outage time for the application. It is your responsibility to establish proper operational procedures to make use of the spin-off copy.

When used in conjunction with the extended distance capability of the ESCON-equipped 3990, as well as the ESCON-equipped 3490 tape controller, you can produce backup copies of data sets with much less disruption to applications. These data sets can be written to either DASD or 3490 tape drives located a significant distance from the primary data center. Using the extended distance capabilities enhances your capability to deliver backup copies of critical data to an alternative site for offsite backup retention or disaster recovery. See Chapter 13, “Disaster Recovery” on page 127 for more information.

The *IBM 3990 Storage Control Planning, Installation, and Storage Administration Guide* offers more information on this function.

12.2.2 Dual Copy

The dual copy function allows you to maintain logically identical copies of a DASD volume on two different devices in the same subsystem. The 3990 internally synchronizes the volumes by writing all modified data on both volumes. Thus, dual copy provides:

- Significantly enhanced data availability during a volume outage by automatically directing all accesses to the operational volume without application interruption
- Resynchronization of a duplex pair after repair
- An excellent way to migrate data to allow for device maintenance.

Dual copy is an extended function of the IBM 3990 model 3 or 6 storage control. It allows full synchronization of pairs of DASD volumes in its configuration, provided they are of the same model and geometry. One volume, which is online and accessible by applications, is the primary. The other volume, which is offline and accessed only for updates by the 3990, is the secondary.

Once started by an IDCAMS command or ISMF, synchronization occurs and is maintained automatically by the 3990. When an error occurs that prevents access to the primary volume, the 3990 automatically switches all of the activity

to the secondary volume. This is done without interrupting the ongoing I/O activity. The operator is notified of this situation.

Dual copy protects your data against all types of hardware failure at a volume or HDA level. Because of its characteristics, the dual copy function does not protect your data against corruption or subsystem failure.

You can also use transient dual copy. You can implement it by starting a dual copy pair and later interrupting it in a controlled way so as to replace the primary volume.

You could use the dual copy function on volumes that contain data sets with high availability requirements. Dual copy can simplify the volume and data set backup procedures to protect against a volume media failure but does not provide support for disaster recovery or data corruption. In some cases, you can use transient dual copy to replace a volume, with minimum delay to the ongoing applications.

The *IBM 3990 Storage Control Planning, Installation, and Storage Administration Guide* offers more information on this function.

12.2.3 Remote Copy

Extended remote copy (XRC) is a function of the 3990 model 6 for disaster recovery and workload migration management, and operates with a combination of 3990 LIC and DFSMS/MVS software. Data specified at a volume level can be copied asynchronously from a 3990 model 6 at the primary site to a 3990 model 2, 3, or 6 at a recovery site. The primary 3990 model 6 can be attached to the XRC DFSMS/MVS host at up to 20 kilometers (km) using ESCON links. The recovery site 3990 can be attached to the same host using ESCON links at up to 20 km for a Model 6, or 15 km for Model 3.

Peer-to-peer remote copy (PPRC) is a 3990 model 6 capability that provides critical volume protection across storage subsystems. To implement this continuous data shadowing feature, the primary 3990 model 6 sends updates to the recovery 3990 model 6 in a synchronous cache-to-cache communication using ESCON links.

XRC and PPRC provide recovery at a secondary site with fewer transactions lost as a result of an unexpected outage. Asynchronous copying with XRC provides minimal data loss in the event of a system failure. PPRC is a synchronous copy function designed to prevent any loss of data. These 3990 Model 6 subsystem abilities can greatly improve the data availability of your databases in the unlikely event of a system failure.

See Chapter 13, “Disaster Recovery” on page 127 for details.

12.3 RAMAC DASD

The RAMAC DASD (unless otherwise noted the descriptions are for both the RAMAC DASD and the RAMAC 2 DASD) is a fault tolerant, software transparent device using a high performance RAID 5 implementation. The RAMAC DASD combines the availability features of a RAID 5 DASD with the extended functions of the 3990 model 3 or 6 storage control to offer a high performance, high availability storage subsystem. (RAID 5 is an acronym for redundant array of independent disks, level 5.)

The RAMAC DASD extends the availability of your data storage by providing the following enhancements:

- High performance RAID 5 implementation designed for continuous availability requirements
- Increased data availability through the use of RAID 5 architecture, dynamic sparing, and a fault tolerant design
- Enhanced capabilities for nondisruptive maintenance and upgrade
- Enhanced predictive failure capabilities, designed to predict disk drive failure conditions before they occur.

RAMAC DASD is designed to address continuous availability requirements using features such as dual power cables and the logical data redundancy design of the array. See B.2, " RAMAC DASD" on page 157 for a list of the features and functions that support continuous availability.

12.4 Environmentals

The two main types of environmental hardware failures that can cause an unplanned CICS outage are electricity supply and cooling failure.

In some cases, the CICS outage can be prevented if the appropriate action is taken beforehand. In other cases, all that can be achieved is a reduction in the duration of the outage.

12.4.1 Electricity Supply Failure

It is possible to install uninterruptable power supply (UPS) units, where the power supply from external sources is backed up by alternate sources. Switching from one source to the other can be automatic and sufficiently quick to ensure continuous power supply to the hardware.

The S/390 microprocessor cluster gives you the option to use the battery backup unit. It enables the S/390 microprocessor cluster to withstand virtually any power line disturbance for up to 3.5 minutes, including total loss of utility power.

12.4.2 Cooling Failure

Cooling failures can be divided to air conditioning and water cooling failures.

An air conditioning failure can impact all devices. You should monitor air temperature and humidity levels to ensure they remain within allowed range. Your procedures should specify what action to take if your air conditioning fails.

Water cooling failure affects the water cooled processors, and they will detect the failure and issue a timely warning before initiating an automatic shutdown. Procedures should be in place to take the necessary actions.

Chapter 13. Disaster Recovery

If the normal availability of your CICS system is being extended into the 99 percent range, you should look at your disaster recovery plan. The same pressure that drives high availability drives timely and current disaster recovery.

In this chapter we describe what you need to consider when planning for disaster recovery in a CICS environment:

- Why should you have a disaster recovery plan
- Testing
- Six tiers of solutions
- IMS/ESA 5.1 remote site recovery (RSR)
- Remote Recovery Data Facility (RRDF)
- Peer-to-peer remote copy (PPRC)
- Extended remote copy (XRC).

13.1 Why a Disaster Recovery Plan?

There is one simple question to ask yourself that determines if you need a disaster recovery plan: "*Can my business continue to function without my CICS system?*" An unqualified, "*Yes, my business can continue to function without my CICS system and the data it accesses*" is the only answer that indicates that you do not need a disaster recovery plan. In order to build a disaster recovery plan you have to take into account a number of items unique to disaster recovery. They include:

- What data is vital to my business?
- How long can the data be unavailable?
- How current does the data need to be?
- What type of disaster is possible, or even likely, and how long will it affect my system?
- What is the cost of a disaster to my company?
- What is the cost of my disaster recovery plan?
- Is performance after a disaster a consideration?

Some or all of your CICS applications may be considered vital to the operations of your business. If all applications are vital, you need to recover all the data your CICS systems use: If only some of your applications are vital, you have to determine what data is associated with those applications. The transaction to data set or database documentation referred to in 5.2.3.3, "Transaction Tables" on page 56 will help you to determine the data you need to recover.

The length of time between the disaster and recovery of your vital applications is a key factor. If your business can not continue without access to your CICS data, your disaster recovery plan must take this into account.

The time-sensitive nature of your recovered data can be a overriding factor. If your vital application is a high volume, high change application, recovering week-old data may not be acceptable, an hour old may also be unacceptable. You may need to recover up to the point of the disaster.

The type of disaster you plan to recover from can determine where your disaster recovery site is located. If you only foresee fire and water damage to your computer floor, a disaster recovery site in the building next door may be acceptable. If you are located in an area prone to hurricanes or earthquakes, for example, a disaster recovery site next door would be pointless.

You must consider the cost of not operating your business for a period of time when planning for disaster recovery. You have to determine the number of lost transactions, and the future loss of business as your customers do business with your competitors, as we discuss in 2.1, “Assessing the Cost of Unavailability” on page 5. Your disaster recovery solution should not be more expensive than the loss from the disaster, unless your business would fail as a result of the outage caused by a disaster.

What is the real cost of your disaster recovery plan? Keeping track of the total cost of your disaster recovery procedures allows you to look at different options available and judge the benefits and cost difference of each.

Your disaster recovery plan should include some performance considerations once you have recovered. Unless your disaster site mirrors your production site, you should determine acceptable levels of throughput and transaction times while operating from the disaster recovery site. The length of time to recover your primary site can also determine what your disaster recovery site will support in the interim.

Figure 15 shows that risk, speed of recovery and completeness of recovery have to be balanced against cost.

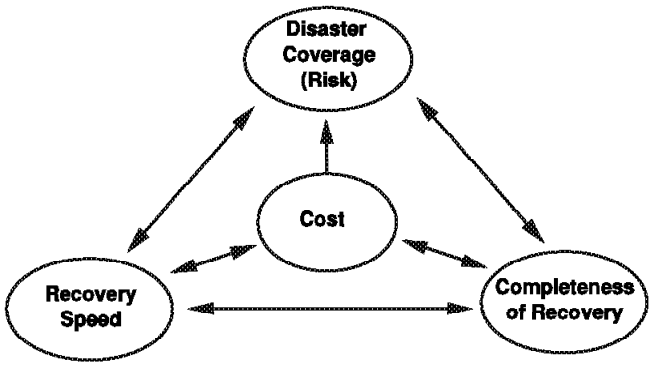


Figure 15. Disaster Recovery: Cost, Completeness, Speed, and Risk

13.2 Disaster Recovery Testing

Testing is an essential part of your disaster recovery planning. All too frequently, just creating a disaster recovery plan results in a false sense of security. If you do not test your disaster recovery plan, it will probably not work when you really need it.

Whenever possible, you should choose a remote site recovery strategy that you can test frequently. Testing your disaster recovery process has the following benefits:

- You know that your recovery plan works.
- You discover problems, mistakes, and errors, and can resolve them before you have to actually use the procedures.
- Your staff will be educated in executing tests and managing disaster recovery situations.
- Your recovery plan becomes a “living” document.
- Members of your IT organization recognize the necessity of such a disaster recovery concept and plan.
- The awareness of your disaster recovery strategy is increased.

After each test, you should use the detailed logs and schedules to identify any errors in your procedures, and eliminate them. You should retest the changed procedures, and then incorporate them into your recovery plan. After changing the recovery plan, you should completely revise all existing disaster recovery documents.

You should make frequent tests early in the implementation of your disaster recovery plan. Once you have removed the major problems, you can reduce the frequency of testing. The frequency will depend on:

- The interval between major changes in your hardware and software environment
- How current you want to keep the recovery plan
- How critical and sensitive your business processes are; the more critical they are, the more frequently testing may be required.

Under normal conditions, we suggest that the recovery plan be tested at least once a year.

13.3 Six Tiers of Solutions

At SHARE in Anaheim California, the automated remote site recovery task force presented a scheme consisting of six tiers of recoverability. The tiers cover a full range of recovery options, ranging from no data moved offsite to full offsite copies with no loss of data. The following figures and text describe them from a CICS perspective.

13.3.1 Tier 0 - No Offsite Data

Figure 16 summarizes the tier 0 solution.

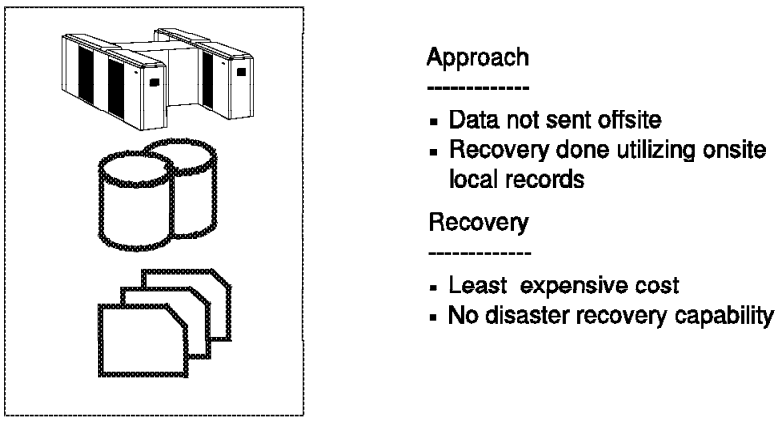


Figure 16. Disaster Recovery Tier 0: No Offsite Backup

Tier 0 is defined as having no requirements to save information offsite, establish a backup hardware platform, or develop a disaster recovery plan. Tier 0 is the no cost disaster recovery recovery solution.

Any disaster recovery capability would be from recovering onsite local records. For most true disasters, such as fire or earthquake, you would not be able to recover your data or systems if you implement a tier 0 solution.

13.3.2 Tier 1 - Physical Removal

Figure 17 summarizes the tier 1 solution.

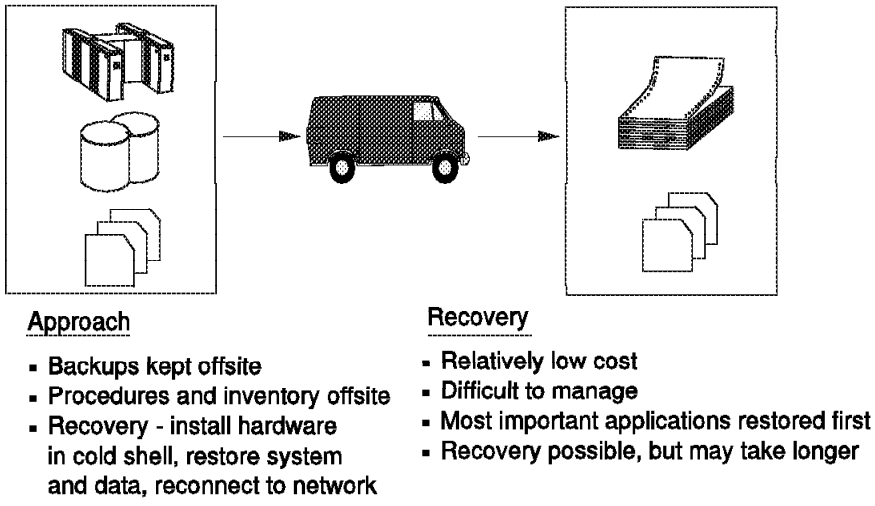


Figure 17. Disaster Recovery Tier 1: Physical Removal

Tier 1 is defined as having:

- A disaster recovery plan
- The required backups are taken and physically removed to an offsite storage facility, usually in a truck or other vehicle.
- Optionally, a backup site without the required hardware currently installed.

Your disaster recovery plan has to include the necessary material to guide the staff responsible for recovering your system, from hardware requirements to day-to-day operations.

The backups required for offsite storage have to be done periodically. That period, daily, weekly, monthly or what ever period you decide, would be as current as your data would be after a disaster, since your recovery action is to restore the backups at the recovery site (when you have one).

This method has some conflicts with the requirement for continuous availability of your online systems:

- If you require data from two or more subsystems to be synchronized, for example, DB2 and VSAM, you would have to stop updates to both, then copy both sets of data.
- DB2 and VSAM would both be unavailable for update until the longest running copy is finished.
- If you require a point-in-time copy for all your data, your application may be unavailable for updates for a considerable time.

The major benefit of tier 1 is the low cost. The major costs are the storage site and the transportation.

The drawbacks are:

- Setting up a computer floor, and obtaining the necessary hardware after a disaster can take a long time.
- Recovery is to the point in time of your last backups. You have no record of any transactions that took place after these backups were taken.
- The process is difficult to manage.
- Your disaster recovery plan is difficult to test.

Tier 1

Tier 1 provides a very basic level of disaster recovery. You will lose data in the disaster, perhaps a considerable amount of data. However, tier 1 allows you to recover and provide some form of service at low cost. You must assess whether the loss of data and the time taken to restore a service will prevent your company from continuing in business.

13.3.3 Tier 2 - Physical Removal with Hot Site

Figure 18 summarizes the tier 2 solution.

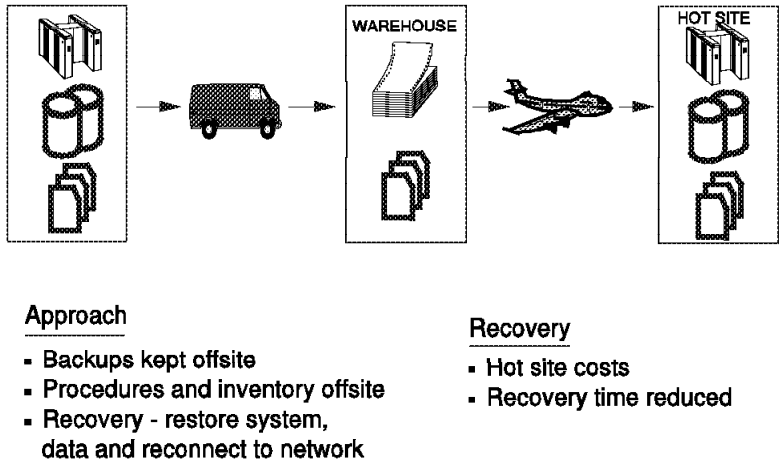


Figure 18. Disaster Recovery Tier 2: Physical Removal to a Hot Site

Tier 2 is similar to tier 1. The difference is that in tier 2, a secondary site already has installed hardware that would be made available to support the vital applications of the primary site. The same process is used to backup and store the vital data, therefore the same availability issues exist at the primary site as do in tier 1.

The benefits of tier 2 are the elimination of time to obtain and setup the hardware at the secondary site, and the ability to test your disaster recovery plan.

The drawback is the expense of providing, or contracting for, a hot site.

Tier 2

Tier 2, like tier 1, provides a very basic level of disaster recovery. You will lose data in the disaster, perhaps a considerable amount of data. However, tier 2 allows you to recover and provide some form of service at low cost and more rapidly than tier 1. You must assess whether the loss of data and the time taken to restore a service will prevent your company from continuing in business.

13.3.4 Tier 3 - Electronic Vaulting

Figure 19 summarizes the tier 3 solution.

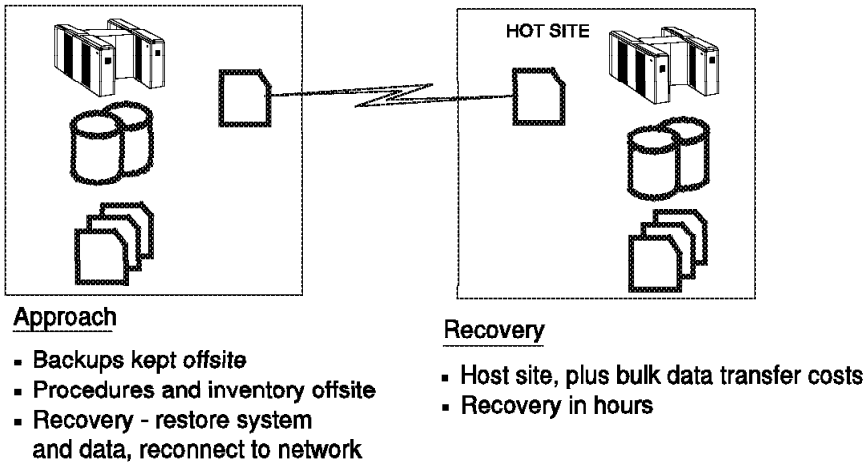


Figure 19. Disaster Recovery Tier 3: Electronic Vaulting

Tier 3 is similar to tier 2. The difference is that data is electronically transmitted to the hot site. This eliminates the truck and the offsite storage warehouse. The same process is used to backup the data, so the same primary site availability issues exist in tier 1.

The benefits of tier 3 are faster recovery, as the data does not have to be retrieved from offsite and downloaded, and the elimination of manually shipping and storing the backups at a warehouse.

The drawbacks are that the DASD at the hot site is reserved for you, and you must have a link to the hot site and software to transfer the data between sites.

Don't forget, procedures and documentation still have to be transported to the hot site, but this can be done electronically.

Tier 3

Tier 3, like tier 1, provides a very basic level of disaster recovery. You will lose data in the disaster, perhaps a considerable amount of data. The advantage of tier 3 is that you should be able to provide service to your users quite rapidly. You must assess whether the loss of data will prevent your company from continuing in business.

13.3.5 Tier 0 - 3 Solutions

Figure 20 summarizes the solutions for tiers 0 through 3, and shows the approximate time required for a recovery with each tier of solution.

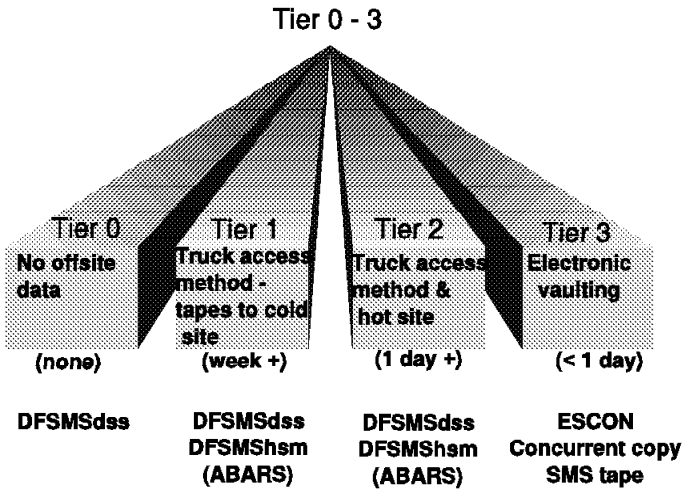


Figure 20. Disaster Recovery Tier 0-3: Summary of Solutions

Tiers 0 to 3 cover the disaster recovery plans of many CICS users. With the exception of tier 0, they employ the same basic design: A point-in-time copy of the necessary data. That data is then moved offsite to be used when required after a disaster.

The advantage of these methods is their low cost.

The disadvantages of these methods are:

- Recovery speed: It can take from days to weeks to recover.
- Any recovery is incomplete: Any updates made after the point-in-time backup are lost.
- Disaster recovery is risky: Difficulties in testing your disaster recovery plan could lead to incomplete recovery.

13.3.6 Tier 4 - Active Secondary Site

Figure 21 summarizes the tier 4 solution.

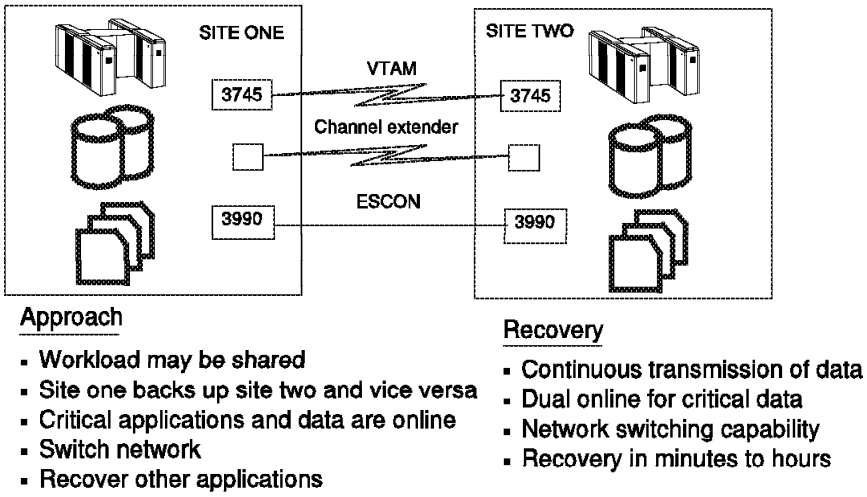


Figure 21. Disaster Recovery Tier 4: Active Secondary Site

Tier 4 closes the gap between the point-in-time backups and current online processing recovery. Here site one acts as a backup to site two, and site two acts as a backup to site one.

Tier 4 duplicates the vital data of each system at the other's site. You must transmit image copies of data to the alternate site on a regular basis. You must also transmit CICS logs and journals, once they have been archived. Similarly you must transmit logs for IMS and DB2 subsystems. Your recovery action is to perform a forward recovery of the data at the alternate site. This allows recovery up to the point of the latest closed log for each subsystem.

You must also copy other vital data that is necessary to run your system to the alternate site. For example, you must copy your load libraries and JCL. You can do this on a regular basis, or when the libraries and JCL change.

The benefits of tier 4 are:

- Recovery is fast: The required hardware and software is already in place at the secondary site.
- Recovery is more complete than in the tier 1 to 3 solutions. You can recover all data to the end of the log for each of your data subsystems.
- Recovery risk is low, because you can easily test your procedures.

The drawbacks are:

- Recovery is more difficult to manage, as you have to ensure that all the logs and copies are transmitted to the other system.
- This solution introduces synchronization problems. Logs for different data subsystems are transmitted at different times. When you complete your recovery at the secondary site, you may find that your VSAM data is complete up to 30 minutes before the disaster, whereas your DB2 data is complete up to 15 minutes before the disaster. If your data must be

synchronized, you may have to develop further procedures to resynchronize data in different subsystems.

- Cost is higher than for the tier 1 to 3 solutions, because we need dedicated hardware, software, and communication links.

Tier 4

Tier 4 provides a more advanced level of disaster recovery. You will lose data in the disaster, but only a few hours or minutes worth. You must assess whether the loss of data will prevent your company from continuing in business, and what the cost of lost data will be.

13.3.7 Tier 5 - Two Site, Two Phase Commit

Figure 22 summarizes the tier 5 solution.

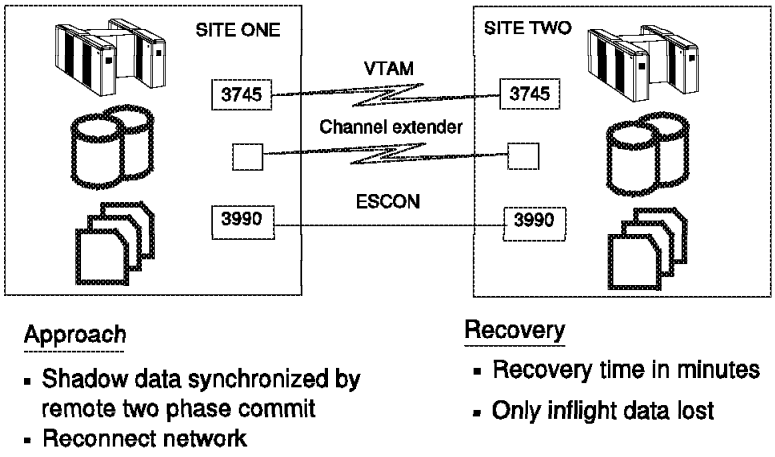


Figure 22. Disaster Recovery Tier 5: Two Site, Two Phase Commit

Tier 5, remote two phase commit, is an application solution to high currency of data at a remote site. This requires partially or fully dedicated hardware at the remote site to keep the vital data in image format and perform the two phase commit. The vital data at the remote site and the primary site will be updated or backed out as a single unit of work (UOW). This ensures that the only vital data lost would be from transactions that were in process when the disaster occurred.

Other data required to run your vital application will have to be sent to the secondary site as well. For example current load libraries and documentation would have to be kept up to date on the secondary site.

The benefits of tier 5 are high currency of vital data and fast recovery of vital data. The drawbacks are:

- The cost of maintaining and running two sites
- The solution is application program dependent. You must write your applications so that they write out data both locally and remotely, by transmitting data to a partner application at the remote site which writes out the data there.
- This solution increases transaction response time. Your application program waits every time it needs to transmit data to the remote site.

Tier 5

Tier 5 solutions are for custom designed solutions with special applications. Because of the requirement for applications to be designed to use this solution, it cannot be used at most CICS sites.

13.3.8 Tier 6 - Minimal to Zero Data Loss

Figure 23 summarizes the tier 6 solution.

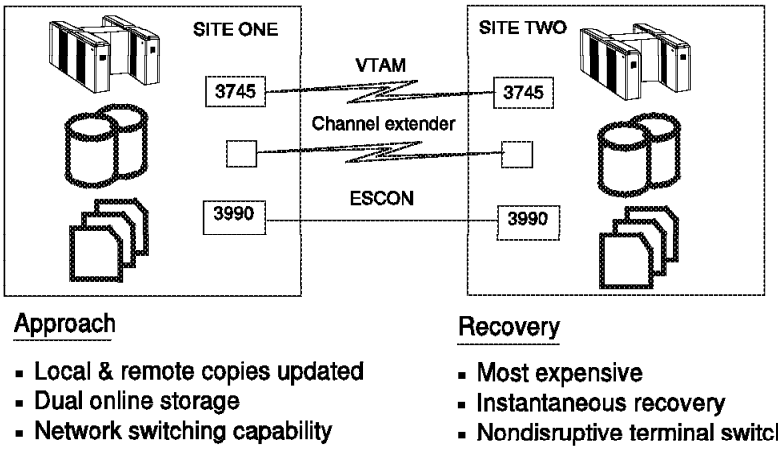


Figure 23. Disaster Recovery Tier 6: Minimal to Zero Data Loss

Tier 6, minimal to zero data loss, is the ultimate level of disaster recovery.

There are two tier 6 solutions: hardware-based and software-based. The hardware solution involves the use of 3990-6 DASD controllers with remote and local copies of vital data. There are two flavors of the hardware solution: peer-to-peer remote copy (PPRC) and or extended remote copy (XRC). We discuss PPRC and XRC in more detail in section 13.4.1, “Peer-to-Peer Remote Copy and Extended Remote Copy” on page 142. There are two flavors of the software solution: IMS remote site recovery (RSR) and Remote Recovery Data Facility (RRDF). RSR applies only to IMS data. It is a separately orderable feature of IMS/ESA DB 5.1. RRDF applies to data sets managed by CICS file control and to the DB2, IMS, IDMS, CPCS, ADABAS, and SuperMICR database management systems, collecting real-time log and journal data from them. RRDF is supplied by E-Net Corporation and is available from IBM as part of the IBM Cooperative Software Program.

The benefits of tier 6 are:

- No loss of data
- Recovery in a very short period of time
- Emergency restart at remote site should be possible.

The drawbacks are the cost of running two sites and the communication overhead. If you are using the hardware solution based on 3990-6, you are limited in how far away your recovery site can be. If you use PPRC, updates are sent from the primary 3990-6 directly to the 3990-6 at your recovery site using ESCON links between the two 3990-6s. The 3990-6s can be up to 43 km (26.7 miles) apart (with an RPQ).

If you use XRC, the 3990-6s at the primary and recovery sites can be attached to the XRC DFSMS/MVS host at up to 43 km (26.7 miles) using ESCON links (with an RPQ). If you use three sites, one for the primary 3990, one to support the XRC DFSMS/MVS host, and one for the recovery 3990, this allows a total of 86

km (53.4 miles) between the 3990s. If you use channel extenders with XRC, there is no limit on the distance between your primary and remote site.

For RSR and RRDF there is no limit to the distance between the primary and secondary sites.

Tier 6

Tier 6 provides a very complete level of disaster recovery. You must assess whether the cost of achieving this level of disaster recovery is justified for your company.

13.3.9 Tier 4 - 6 Solutions

Figure 24 summarizes the solutions for tiers 4 through 6, and shows the approximate time required for a recovery with each tier of solution.

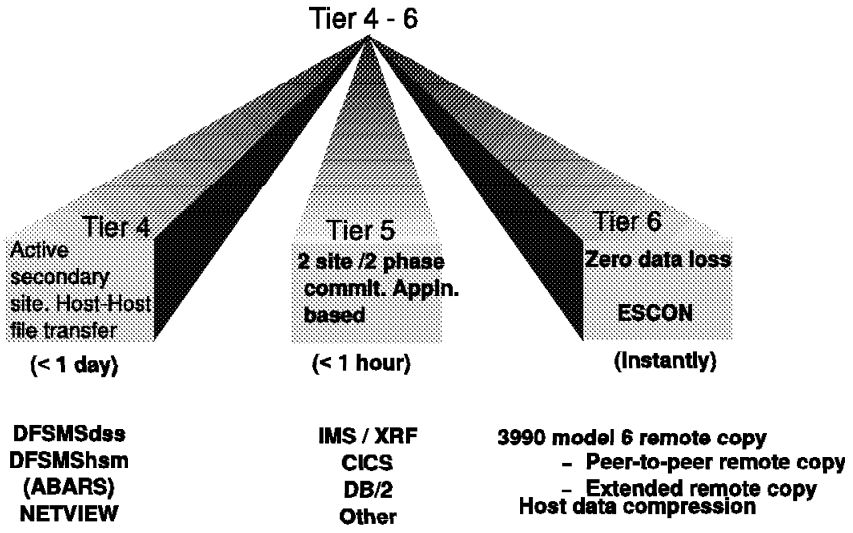


Figure 24. Disaster Recovery Tier 4-6: Summary of Solutions

This summary shows the three tiers and the various tools for each that can assist you to reach your required level of disaster recovery.

Tier 4 will have to rely on automation to control the sending of backups to the remote site. NetView provides the ability to schedule work in order to maintain recoverability at the remote site.

Tier 5 relies on the two phase commit processing supported by various database products and your application programs use of these features. Tier 5 requires additional backup processing to ensure that vital data, other than databases, is copied to the remote system.

Tier 6 is divided into two sections: software solutions for specific access methods and database management systems, and hardware solutions for any data.

RSR can provide you very high currency and recoverability of your IMS data. However your other CICS files and databases are not included, neither are IMS system files. Only IMS databases are copied to the remote site.

RRDF can provide very high currency and recoverability for a wide range of data. It does not cover all of the data in which you may be interested, however. For example, RRDF does not support load module libraries.

The 3990-6 hardware solution is independent of the data being stored on the DASD. PPRC and XRC can be used for databases, CICS files, logs, and any other data sets that you need to ensure complete recovery on the remote system.

13.4 Disaster Recovery and High Availability

In this section we describe the tier 6 solutions to high availability and data currency in a disaster recovery situation.

13.4.1 Peer-to-Peer Remote Copy and Extended Remote Copy

PPRC and XRC are both 3990-6 hardware solutions that provide data currency to secondary, remote volumes. Updates made to secondary DASD are kept in time sequence. This ensures that updates are applied consistently across volumes. PPRC and XRC also ensure that updates are applied in time sequence across control units as well. This sequencing offers a very high degree of data integrity across volumes behind different control units.

Since PPRC and XRC are hardware solutions, they are application and data independent. The data can be DB2, VSAM, IMS, or any other data set type. All your vital data on DASD can be duplicated offsite. This reduces the complexity of recovery. These solutions can also make use of RAMAC DASD to deliver the highest levels of data integrity and availability.

PPRC synchronously shadows updates from the primary to the secondary site. This ensures that no data is lost between the data committed on the primary and secondary DASD. The time taken for the synchronous write to the secondary unit has an impact on your application, increasing response time. This additional time (required for each write operation) is approximately equivalent to a DASD fastwrite operation. Since the implementation of PPRC is almost entirely in the 3990-6, you must perform capacity planning for cache and non-volatile storage (NVS) to ensure optimum performance.

XRC is an asynchronous implementation of remote copy. The application updates the primary data as usual, XRC then manages the passing of the updates to the secondary site. The currency of the secondary site will lag slightly behind the primary site due to updates in transit. As part of XRC data management, updates to the secondary site are performed in the same sequence as the primary site. This ensures data integrity across controllers and devices. Since XRC does not wait for updates to be made at the secondary site, the application′s performance is not directly affected. XRC uses cache and NVS, so you must perform appropriate capacity planning to ensure optimum performance.

In the event of a disaster you need to check the state of all secondary volumes to ensure data consistency against the shadowed log data sets. This ensures that the same sequence of updates is maintained on the secondary volumes as was on the primary up to the point of the disaster. Since PPRC and XRC do not require restores and forward recovery of data, your restart procedures on the secondary system may be the same as a short term outage at the primary site, a power outage for example.

When running with PPRC or XRC the data you need to replicate along with the databases includes:

- CICS logs and journals
- CICS CSDs, SYSIN and load libraries
- RECON and RDS for IMS
- WADS and OLDS for IMS

- ACBLIB for IMS
- BSDS catalog and directory for DB2
- DB2 logs
- Any essential non-database volumes.

CICS applications can make use of non-DASD storage for processing data. If your application depends on this type of data, be aware that PPRC and XRC does not handle it.

For more information on PPRC and XRC, see *Planning for IBM Remote Copy*.

13.4.1.1 PPRC or XRC?

You need to choose between PPRC and XRC for transmitting data to your backup site. In this section we contrast the two methods to help you make your choice.

PPRC is the remote copy facility of choice for you if you:

- Require data currency at the secondary site
- Have your recovery site within ESCON distance
- Can accept some performance degradation
- Have a duplicate DASD configuration available at the remote site.

The synchronous nature of PPRC ensures that if you have a disaster at your main site, you only lose inflight transactions. The committed data recorded at the remote site is the same as that at the primary site.

Use PPRC for High Value Transactions

You should consider PPRC if you deal with high value transactions, and data integrity in a disaster is more important to you than day-to-day performance. PPRC is more likely to be the solution for you if you characterize your business as being low volume, high value transactions; a system supporting payments of thousands, or even millions, of dollars, for example.

XRC is the remote copy facility of choice for you if you:

- Can accept that your data at the secondary site will be a few seconds behind the primary
- Have your secondary site outside ESCON distance
- Require high performance at the primary site.

The asynchronous nature of XRC means that the remote site may have no knowledge of transactions that ran at the primary site, or does not know that they completed successfully. XRC ensures that the data recorded at the remote site is consistent (that is, it looks like a snapshot of the data at the primary site, but the snapshot may be several seconds old).

Use XRC for High Volume Transactions

You should consider XRC if you deal with low value transactions, and data integrity in a disaster is less important to you than day-to-day performance. XRC is more likely to be the solution for you if you characterize your business as being high volume, low value transactions; a system supporting a network of ATMs, for example, where there is a high volume of transactions, but each transaction is typically for less than 200 dollars.

13.4.1.2 Other Benefits of PPRC and XRC

PPRC or XRC may eliminate the need for disaster recovery backups to be taken at the primary site, or to be taken at all. PPRC allows you to temporarily suspend the copying of updates to the secondary site. This allows you to effectively suspend updates at the secondary site so that you can make image copies or backups of the data there. Once the backups are complete, you can reestablish the pairing of data sets on the primary and secondary sites. Updates to the primary that have been recorded by the 3990-6 are applied to the secondary to resynchronize the pair.

XRC supports the running of concurrent copy sessions to its secondary volumes. This enables you to create a point-in-time copy of the data.

PPRC and XRC also allow you to migrate data to the same or larger DASD of similar geometry, behind the same or different control units at the secondary site. This can be done for workload management or DASD maintenance, for example.

13.4.1.3 Forward Recovery

Whether you use PPRC or XRC, you have two fundamental choices: You can pair volumes containing both the data and the log records, or you can pair only the volumes containing the log records. In the first case you should be able to perform an emergency restart of your systems and restore service very rapidly. In the latter case you would need to use the log records, along with an image copy transmitted separately, to perform a forward recovery of your data, followed by an emergency restart.

Pairing the data volumes, as well as the log volumes, costs more (you have more data flowing between the sites, and need a greater bandwidth to support the flow). In theory you can restart much faster than if you have to perform a forward recovery. When deciding which to use, you must ask yourself whether this method is significantly faster, and whether you think it is worth the additional costs.

13.4.2 Remote Site Recovery

The RSR features of IMS/ESA 5.1 allow you to recover quickly from an interruption of computer services at your active (primary) site. RSR supports recovery of IMS/ESA DB full-function databases and IMS/ESA DB Fast Path data entry databases (DEDBs).

IMS database and online transaction information on the IMS log is continuously transmitted to a tracking (remote, secondary) site. The tracking site is continually ready to become the active site (and resume the active workload) if service is interrupted at the active site.

RSR provides these recovery features:

- It provides a remote copy of the necessary IMS/ESA DB log records for database and message queue recovery at the tracking site.
- It supports the HDAM, HIDAM, HISAM, and SHISAM access methods. It also supports Fast Path DEDBs.
- It supports IMS/ESA DB, IMS DBCTL, and batch workloads.
- It maintains remote copies of full-function databases and Fast Path DEDBs.
- It recognizes that IMS database recovery control (DBRC) is operating at the active site and, separately, at the tracking site.
- It supports data sharing at the active site.
- It coexists with the IMS extended recovery facility (XRF).
- It lets you filter out log records that are not needed to support the defined critical environment. An added benefit with this is reduced line traffic.
- It continues to operate when the active site, the tracking site, or the RSR transmission facility becomes temporarily unavailable and provides a way to resynchronize the sites.
- It provides transaction consistency between the active and tracking sites.
- It supports standard VTAM communication protocols, so new technology is not required for data transmission.

If an unplanned remote takeover occurs, such as a disaster, log records may need to be removed from the tracking system log data set (SLDS) so that invalid data is not applied to the databases once the remote takeover is complete. This process is called log truncation.

Log truncation occurs:

- If some log records are missing at the tracking site for an active subsystem being tracked, the records beyond the missing data must be truncated for that active subsystem
- When log streams must be merged in a creation-time sequence before routing to the tracking components, routing for all of the streams being merged stops at the point that a gap is encountered on a log stream or when there is no more log data for a stream. Thus, log truncation must occur for all of the streams (in the case of a gap), or for all but one of the streams.
- When an unplanned remote takeover occurs in the above situations, the subsystems′ logs at the tracking site are truncated at the point in the logs where log routing has stopped. This ensures that the sequence of the logs is correct, and data integrity is ensured. The log router does not delete the truncated data, however, because you may be able to use it, even though IMS cannot.

The benefits of RSR are high currency of vital data and fast recovery of vital IMS data, and the ability to independently operate the remote site while it automatically tracks the status of the active site.

The drawback of RSR is that only IMS data is recoverable.

13.4.3 Remote Recovery Data Facility

RRDF Version 2 Release 1 minimizes data loss and service outage time in the event of a disaster by providing a real-time remote logging function. Real-time remote logging provides data currency at the remote site, enabling the remote site to recover databases to within seconds of the outage—typically in less than 1 sec.

RRDF runs in its own address space. It provides programs that run in the CICS or database management system address space. These programs are invoked through standard interfaces, for example, at exit points associated with writing out log records.

The programs that run in the CICS or database management system address space use MVS cross-memory services to move log data to the RRDF address space. The RRDF address space maintains a virtual storage queue at the primary site for records awaiting transmission, with provision for spill files if communication between the primary and secondary sites is interrupted. Remote logging is only as effective as the currency of the data that is sent offsite. RRDF transports log and journal data to a remote location in real-time, within seconds of the log operation at the primary site.

When the RRDF address space at the remote site receives the log and journal data, it formats it into archived log data sets. Once data has been stored at the remote site, you can use it as needed to meet business requirements. The recovery process uses standard recovery utilities. For most data formats you must first use the log and journal data transmitted by RRDF in conjunction with a recent image copy of the data sets and databases that you have to recover. Then you perform a forward recovery. If you are using DB2 or IDMS, you have the option of applying log records to the remote copies of your databases as RRDF receives the log records.

If you use DB2, you can use the optional RRDF log apply feature. With this feature you can maintain a logically consistent “shadow” copy of a DB2 database at the remote site. The RRDF log apply feature updates the shadow database at selected intervals, using log data transmitted from the primary site. Thus restart time is shortened because the work to do after a disaster is minimal. The currency of the data depends on the log data transmitted by RRDF and how frequently you run the RRDF log apply feature. The RRDF log apply feature also enhances data availability, as you have read access to the shadow copy through a remote site DB2 subsystem. RRDF supports DB2 remote logging for all environments, including TSO, IMS, CICS, batch, and call attach.

If you use IDMS, you can use the optional E-NET2 - IDMS Shadow Database Facility software product. With E-NET2 you can shadow database updates, using IDMS journal data. This journal data can be the output of RRDF′s reformat utility or standard IDMS archive journal data. You can shadow any desired subset of IDMS databases, subject to the usual restrictions on interarea dependencies.

At least two RRDF licenses are required to support the remote site recovery function, one for the primary site and one for the remote site.

13.4.4 Choosing among RSR, RRDF, and 3990-6 Solutions

Table 3 summarizes the characteristics of the products you can use to implement a tier 6 solution. You must decide which solution or solutions is most appropriate for your environment.

Table 3. Selecting a Tier 6 Implementation. This table compares the strengths of the tier 6 solutions.

	RSR	**RRDF**	**3990-6**
Data type supported	IMS/ESA DB 5.1 or later only	Various data sets[1]	Any on DASD
Database shadowing	Yes	Optional. Available for DB2 and IDMS only	Optional
Forward recovery required	No	Yes	Depends on implementation
Distance limitation	None	None	About 40 km for ESCON. Unlimited for XRC with channel extenders.

[1] Data sets managed by CICS file control and the DB2, IMS, IDMS, CPCS, ADABAS, and SuperMICR database management systems.

13.5 More Thoughts on Disaster Recovery

Your disaster recovery plan will only be truly tested at a very difficult time for your business, during a disaster. Careful planning and thorough testing may mean the difference between a temporary inconvenience and going out of business.

The goal of your disaster recovery plan is to get your CICS system back online. The currency of the data and the time it takes to get back online is a function of which disaster recovery tier you use. Unless legal requirements for disaster recovery dictate the type of disaster recovery you must have, the choice of tiers is usually a business decision.

Making your disaster recovery work requires a good plan, up-to-date documentation, and regular testing.

13.5.1 Disaster Recovery Personnel Considerations

When planning for disaster recovery, you need to consider personnel issues.

You should ensure that a senior manager is designated as the disaster recovery manager. They must make the final decision as to whether you need to switch to a remote site, or try to rebuild the local system (this is especially true if you have adopted a solution that does not have a warm or hot standby site).

You must decide who will run the remote site, especially during the early hours of the disaster. If your recovery site is a long way from the primary site, many of your staff will be in the wrong place.

Finally, and to show the seriousness of disaster recovery, it is quite likely that some of your key staff will be severely injured, or even dead. Your plans need to identify backups for all your key staff.

13.5.2 Returning to Your Primary Site

One aspect of disaster recovery planning which many people forget is creating a plan for returning operations from the recovery site back to the primary site (or to a new primary site if the disaster was bad enough). Do build this as part of your plan. The worst possible time to create a plan for moving back to your primary site is ***after*** a disaster. You will probably be far too busy to spent time building a plan. As a result, the return to your primary site may be delayed, and may even not work properly.

13.5.3 Disaster Recovery Further Information

If you require further information regarding the recover of specific types of data please see:

- *IBM DATABASE 2 Administration Guide* for DB2 database recovery
- *IMS/ESA Release Planning Guide* for information on RSR
- *IMS/ESA Operations Guide* for IMS database recovery
- *CICS/ESA Recovery and Restart Guide* for CICS and VSAM data set recovery.

The redbook *Disaster Recovery Library: Planning Guide* is also a valuable reference to assist you in setting up a detailed disaster recovery plan if you use a combinations of databases, such as DB2 and IMS.

Chapter 14. Unavailability Cause and Solution Checklist

This chapter provides a checklist of possible causes of unavailability, and where in this book and in other publications you can find information about solutions to these problems.

Table 4 (Page 1 of 2). Problems and Possible Solutions

Symptom or Problem	Read This Section or Book
MVS System Unavailability	
General Solutions	3.1.3, “MVS Sysplex” on page 26
CICS Region Unavailability	
General Solutions	4.1, “The CICSplex” on page 37
	4.2, “Dynamic Transaction Routing” on page 38
	CICS/ESA Dynamic Transaction Routing in a CICSplex
Making Changes	7.1.1.1, “Resource Definition Online” on page 75
	5.1.1, “Running Different CICS Releases on the Same MVS System Image” on page 45
	5.1.2, “Adjusting for Daylight Savings Time” on page 46
	5.1.3, “Introducing New Versions of Application Programs” on page 46
	7.1.1.2, “Autoinstall” on page 76
Errors in CICS	5.2.1, “Maintenance Strategy” on page 48
	5.2.2.2, “Regression Testing” on page 52
	7.2.1, “Use the Latest Version of CICS” on page 78
Programming Errors	5.2.2.1, “Functional Testing” on page 51
	5.2.2.2, “Regression Testing” on page 52
	6.1, “Dealing with Errors” on page 63
Operational Errors	5.2.4, “Automation” on page 57
	Automating CICS/ESA Operations With CICSPlex SM and NetView
Storage Violations	7.2.2, “Minimize Storage Addressing Errors” on page 79
Overstressed Systems	5.2.2.3, “Stress Testing” on page 52
	5.3, “Removing Stress Points” on page 60
	6.2, “Avoiding Stress Points” on page 67
	7.3.1, “Storage Management” on page 79
	7.3.2, “Intersystem Session Queue Management” on page 80
	7.3.4, “Limiting System Usage” on page 82
Frequent SOS	7.3.1, “Storage Management” on page 79
	7.3.4, “Limiting System Usage” on page 82
Stalls	6.1.2, “Soft Errors” on page 64
	7.3.1, “Storage Management” on page 79
	7.3.2, “Intersystem Session Queue Management” on page 80
Single Points of Unavailability	7.4.1, “VTAM Generic Resource Registration” on page 83
	7.4.2, “Failure of a FOR in a CICSplex” on page 83
	7.4.3, “CICS/ESA Data Sets” on page 83

Table 4 (Page 2 of 2). Problems and Possible Solutions	
Symptom or Problem	**Read This Section or Book**
Slow Restart Time	5.2.4.4, “Automatic Restart Manager” on page 59
	7.5.2, “Persistent Session Support” on page 86
	7.5.3, “Using Autoinstall to Improve Restart Times” on page 87
	7.5.4, “Extended Recovery Facility” on page 87
	CICS/ESA Performance Guide
Data Unavailability	
General Solutions	4.3, “Data Sharing” on page 40
Making Changes	8.1, “Changing Your System” on page 89
	9.1, “Changing Your System” on page 99
Making Backups	8.4.2, “Copying Databases” on page 92
	9.4.2, “Copying Databases” on page 103
	10.2.2, “Copying Data Sets” on page 112
Batch Processing	Chapter 11, “Batch Processing Considerations” on page 115
Reorganization	8.4.3, “Reorganizing Databases” on page 93
	9.4.3, “Reorganizing Databases” on page 103
	10.2.3, “Reorganizing Data Sets” on page 112
Hardware Unavailability	
General Solutions	3.1.3, “MVS Sysplex” on page 26
	3.1.7, “The I/O Subsystem” on page 30
	Chapter 12, “Hardware That Supports Continuous Availability” on page 119

Appendix A. Service Level Management

Service level management is the process of planning and delivering sufficient computing capability and related services to meet the service needs of all users. This process must, however, be performed in a cost efficient manner.

Furthermore, there is a fundamental conviction that the provision of information systems services should be managed by the same basic principles as any other business process. It is, therefore, mandatory that a series of Information Systems Management (ISM) disciplines be established that constitute methods, approaches for planning, setting objectives, organizing, managing and controlling processes to create an effective service environment. Specific ISM disciplines that influence the delivery of user service include application development procedures, support center facilities, problem management, change management, security procedures, operations control and capacity planning.

In order to monitor the effectiveness of this inter-related set of ISM disciplines, management needs to be provided with formal reporting and feedback mechanisms (which include exception reports and out-of-line escalation procedures).

Figure 25 shows the basic relationships between the major ISM disciplines. The objective of the disciplines when they are all integrated together is to provide user service, and at the same time, apply business principles to increase the efficiency of the service provider (Information Systems department).

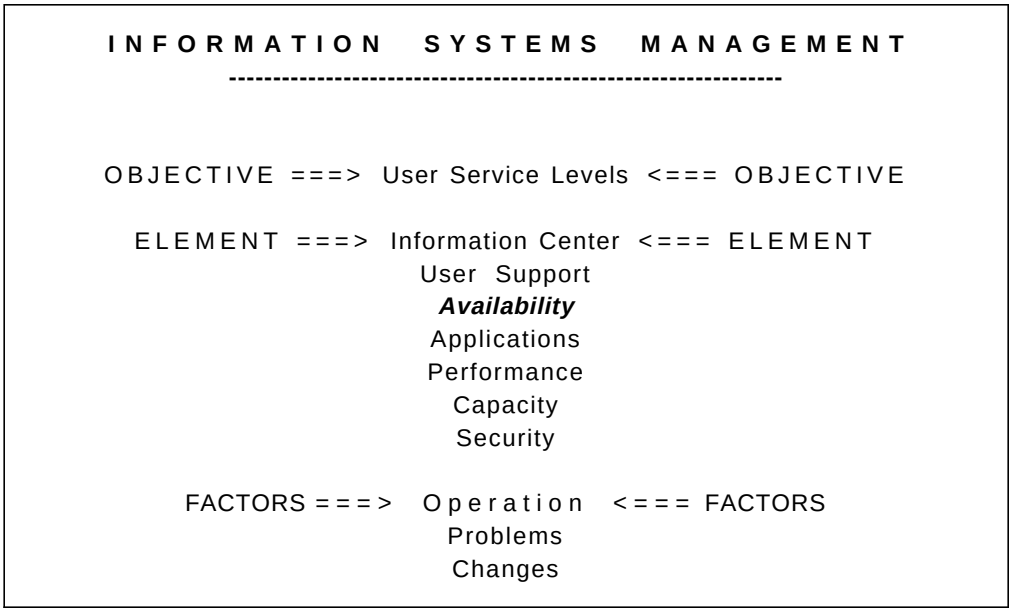


Figure 25. Information Systems Management Discipline Inter-relationships

The ultimate objective should be to have Service Level Agreements that tie all these processes together by formalizing the commitment of both the user departments and the I/S department as the service provider, and to utilize a suitable charging system as a mechanism for users to justify their requested service. This does, of course, require planning and commitment. In the interim, at least service level objectives (internal to the I/S department) should be established.

Once service level objectives are in place, the remaining discipline is Performance Management. This is the process of monitoring and reporting on the service actually provided. Management reporting quantifies whether or not the service objectives are being met and also indicates the effectiveness of the information systems management procedures.

A.1 An Overall Service Model

Figure 26 provides a basic model that depicts processes associated with the provision of service to users. It concentrates on the response time aspects of service to online application users (and batch turnaround), which is directly influenced by capacity and performance management processes.

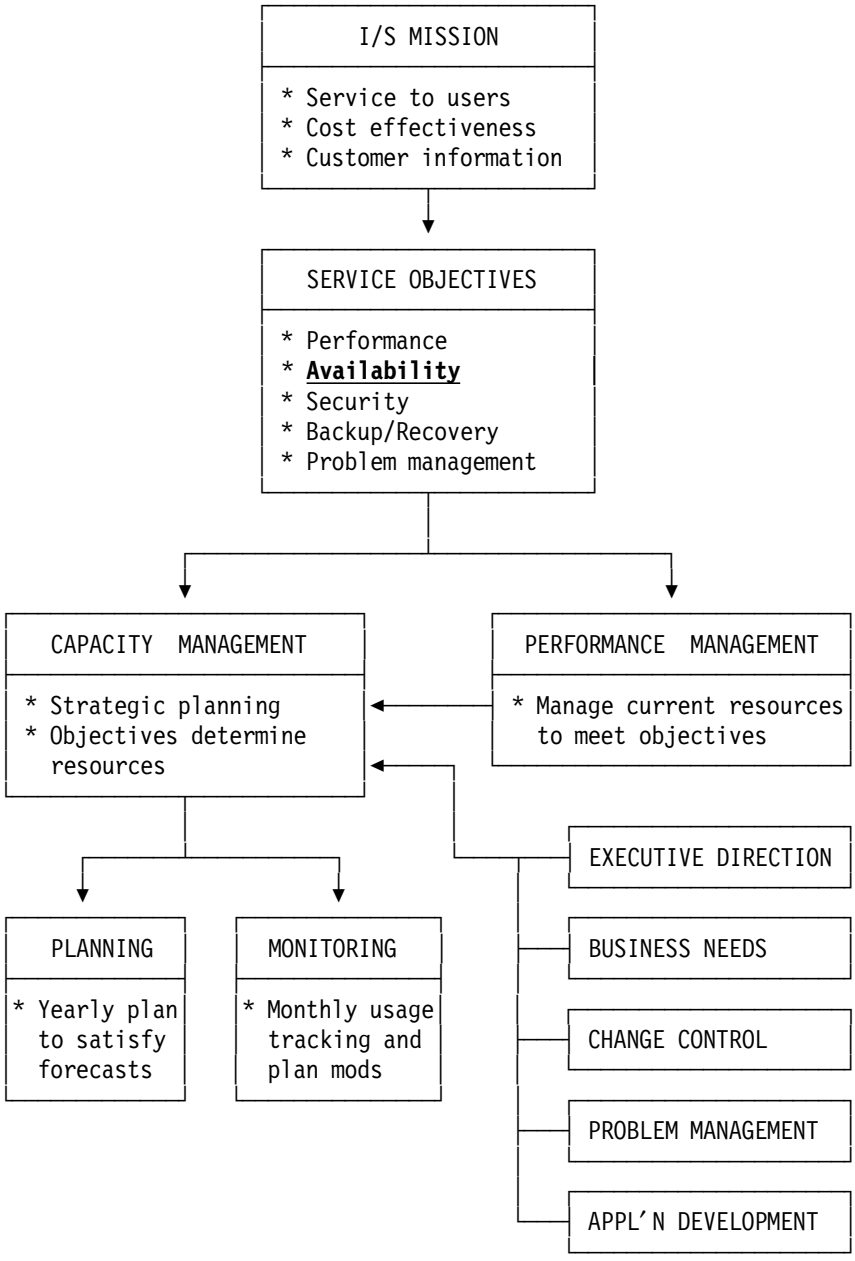


Figure 26. Overall Service Model

With reference to Figure 26, the following points are relevant:

- All day-to-day processes that are implemented within the organization should relate back to the organization′s basic business objectives. In turn, the Information Systems (I/S) department′s mission should reflect these business objectives. As a user service department, I/S must have disciplines in place that reflect this service orientation. There needs to be a consistent focus on the delivery of service.
- In an organization of reasonable size, the only effective way of ensuring that user service levels meet user requirements is to establish formal service level objectives and to put in place management reporting processes that objectively track actual service delivered. The service objectives should cover all aspects of service, including performance, *availability,* security, backup and recovery, and problem management. These objectives should be set out as clear definitions of the service levels that are to be provided and the conditions under which the service commitments are made. They should be established to reflect the user′s specific justified business needs.

  Since it takes time to define and establish effective service objectives for all categories of system usage (especially in the performance area), it is recommended that you define broad objectives first and then progressively define more detailed objectives over time. In addition, the service objectives can initially be internal to the I/S department until such time as there is sufficient confidence in being able to meet objectives, and until such time as executive management wishes to negotiate actual formal service level agreements with user departments. This should be the ultimate goal. But again, all aspects of service need not be negotiated at once. *Availability* agreements could be established before addressing the more complex issues of performance commitments.
- Performance management should be done daily to ensure that service objectives are being met. Tuning is part of performance management and is concerned with adjusting control parameters or configuration components to satisfy present performance objectives for the current system under actual workload volumes, mix and characteristics. With the introduction of workload management into MVS/ESA 5.1 and complimentary function in subsystems such as CICS/ESA 4.1, the system itself can manage performance and thus reduce or eliminate tuning activity by technical staff.
- The provision of user service relies on the processes of capacity and performance management. In return, these two processes should be driven by a constant focus on levels of service to be delivered.

A.2 Service Level Agreements

By combining the experience gained from monitoring internal objectives with the requirements of end users, formal service level agreements (SLAs) can be drawn up and agreed to by all parties involved. The SLAs should be made up of external service objectives in terms that relate directly to performance as seen by the user. For example, response at the user′s workstation or the *availability* of applications to the end users. From the users′ perspective this is all they are interested in.

Since the establishment of any objective will only be of use if actual performance can be measured against this objective, you need to be able to measure external performance. Often, measurements of response or the monitoring of availability

at the screen on remotely attached workstations are not available. Should this be the case, you should consider developing techniques for measuring actual response times and application availability at the workstations.

You should monitor service objectives using daily management reports. These reports should be in graphic form and should allow direct monitoring of actual internal service against the defined service objectives. No translations should be needed. These management reports should clearly indicate whether or not the service objectives are being met.

If service objectives are being met, then there is no need to examine the system further or to perform any tuning tasks. On the other hand, if service objectives are not being met, or if there is a trend that suggests that even though objectives are currently being met this will not continue, you must take action.

To minimize the time required to correct service problems, you need to have a well-defined analysis and tuning methodology. Part of this methodology is the ability to recognize “out of line” situations that may be causing degraded service (either on their own or in combination). A set of system and subsystem component usage indicators are required, each with a defined “trigger” level that should, on an exception basis, initiate a review action to determine if it is impacting or is likely to impact service.

You can use the Real-Time Analysis (RTA) functions provided by CICSPlex SM to monitor both CICS and non-CICS resource status and to generate external notification if the status changes or if thresholds are reached.

The performance model is based on a set of management reports that compare actual performance against the defined objectives. The production of these reports should be a normal “production” job controlled by operators, who should also be responsible for their review. In theory, if performance objectives are being met, then no actions are required. It is not productive use of technical staff time to have them doing any tuning when objectives are being satisfied. There are always tasks that could be done to improve performance, but if the objectives are currently what the users require, then this would be totally unproductive effort.

In practice, since it is probably not possible in most environments to establish and measure service for every user and every application, the performance model will not cover every aspect of performance management. Consequently, there will be some situations (but these should be the exception not the rule) where a user experiences specific performance problems even though all monitored service objectives are being met.

Once performance objectives have been defined and performance management procedures have been established, and the process is fully operational (say after 6-12 months), user departments should be encouraged to view daily performance management graphs, or at least to view weekly and monthly summary reports. This helps to educate users, remove any misconceptions, and enhance the I/S department′s image as a service provider.

Appendix B. Hardware Features

This appendix contains additional information on the major hardware contributors to continuous availability, the parallel transaction server (PTS) and RAMAC DASD.

B.1 Parallel Transaction Servers

Use of the IBM S/390 Parallel Transaction Server 9672 and the 9674 Coupling Facility result in increased availability of data, and provide easily expandable S/390 processing power. Continuous availability is provided by allowing processors to be installed, operated, expanded, and maintained independently.

The S/390 Parallel Transaction Server is a concept combining multiple ESA/390 processors into a single functioning computer system. The individual processors are commonly combined in multiples into a Central Processing Complex (CPC). In turn, the CPCs are physically packaged together in a multiframe system. The entire group of CPCs is perceived by the user as a single computer system.

With all processors running in parallel, a PTS can provide a significant increase in performance and capacity over previous offerings in an air-cooled package. In addition, there are inherent advantages in incremental horizontal growth without impacting availability.

A PTS can also be a building block for use in larger system environments or configurations, for example, as part of a sysplex or parallel sysplex.

The PTS is composed of a number of Electronics Industry Association (EIA) frames. Each frame contains one or two CPCs. Each CPC contains two or more processors. If a CPC has two processors it is called a “Dyadic,” or in more general terms an “N-Way” processor. The number of processors in a PTS is found by multiplying the number of CPCs by the number of processors in each CPC. Thus, a 32-CPC system with six-way processors has a total of 192 processors. The PTS and coupling facility products are built using CMOS (complementary metal oxide semiconductor) technology.

The coupling facility technology, which makes high-performance data sharing possible in the parallel sysplex, is delivered through a combination of hardware and software functions. Coupling Links are high bandwidth fiber optic links that connect the PTS to a Coupling Facility. These links provide high-speed access to a Coupling Facility from MVS/ESA systems running on a PTS in a parallel sysplex. The Coupling Links require direct connection between the PTS and the Coupling Facility.

Features and functions of a PTS that support high availability:

- In an n-way processor CPC, a processor failure is contained at the L2-cache level and deconfigured (degraded CPC operation). The next IML will decide, by using the CPC′s self-test capability, to call for service (solid error), or to configure the processor back into operation (intermittent error). If a system assist processor (SAP) fails with a solid error, a re-IML will bring in an operational processing unit (the highest numbered one) as a SAP, and the CPC will operate with one less CPU.

- Intermittent errors detected during the execution of processing unit Licensed Internal Code are resolved by a reload of writeable control store and an automatic retry.
- Command retry allows a sub-channel to retry a command without causing an I/O interruption. This retry is initiated by the control unit and handled by the system assist processor.
- Arrays, storage, and storage path failures have a large impact on the functioning of the system. Therefore, measures are taken to tolerate errors that can subsequently be recovered. Techniques such as error correction code checking and complement/recomplement are used extensively in these areas. Some examples that apply to a S/390 Microprocessor CPC are:
  - Processor storage (PS) uses error detection and correction (ECC). All single-bit errors are detected and corrected, double-bit errors are detected and some are corrected through complement/recomplement logic, while multiple-bit errors are detected and flagged for further action by the operating system.
  - Processor storage uses redundant DRAMs to implement chip sparing (some chips are spare initially, if more than one bit within a DRAM fails, the spare chip can be used at the next POR).
  - Memory scrubbing is performed in processor storage on a periodic basis to prevent soft-bit errors caused by alpha particle contamination.
  - L2-caches are also protected by ECC. ECC on L1-caches is not required as L1-cache is store-through and hence a copy of the data is always available from L2-cache or processor storage.
  - Storage protect keys are stored in duplex storage protect key arrays within the storage controller (STC) to provide active parallel redundancy. Key storage is also parity checked.
- Console integration allows MVS/ESA to use a hardware console for software communications. On a S/390 microprocessor CPC, MVS/ESA messages can be directed to the support element (SE) associated with that CPC. In turn, these messages can be viewed on the hardware management console (HMC). Console integration increases MVS/ESA console availability.
- The S/390 microprocessor power system has an “N+1” design, optional battery backup units (BBU), plus “hot-pluggable” components to provide for continuous availability. The BBUs provide a short-term (minute or two) override of power supply outages without the need for the installation of an uninterruptable power supply. Studies have found that over 90% of power outages last less than 10 seconds. In addition, a power-save mode allows a CPC being used as a CF to remain partially powered (storage cards, processor card and ETR card) for at least 80 minutes, thus preserving the non-volatility of storage.
- At the system level, many functions are implemented that prevent or tolerate errors. Examples of these functions are:
  - Subsystem storage protection (SSSP) - This is a generic protection capability. It implements storage protection for multiple components within a single address space. It is exploited by CICS/ESA Version 3 (or later) running under MVS/ESA Version 4 (or later) to protect CICS control blocks and data from user applications.
  - CICS transaction isolation - This facility extends the protection provided under SSSP to protect one CICS user application from other user

applications within the same CICS address space. The subspace group facility is used to implement transaction isolation to build upon CICS storage protection functions previously available, such as subsystem storage protection.

- Dynamic reconfiguration management (DRM) - DRM allows an installation to modify its hardware and software I/O configurations dynamically for devices, control units and channel paths without requiring a power-on reset (POR) of the hardware or IPL of the MVS/ESA operating system.
- LPAR logical CP vary on/off - This facility allows for the reconfiguration of dedicated CP resources between partitions without partition deactivation.

B.2 RAMAC DASD

The RAMAC DASD consists of a rack (IBM 9391, model A10) and drawers (IBM 9392, model B13 and model B23). The minimum number of drawers in a rack is two; the maximum number is 16.

Each drawer emulates two (model B13) or four (model B23) 3390 model 3 volumes depending on the drawer model. It is a logical emulation with the actual hardware being transparent to the host system.

The RAMAC DASD, fully configured with 16 drawers, provides a maximum of 90.8GB (model B13 drawers) or a maximum of 181.6GB (model B23 drawers) of useful data capacity.

Features and functions for continuous availability:

- RAID 5

  The data striping and parity architecture of RAID 5 are used to ensure access to data in the event of a single disk drive failure. In the RAMAC DASD, each drawer is an independent RAID 5 array.

- Separate disk drive power

  Each disk drive has its own DC power supply card. If the power supply card for one disk drive fails, power is maintained to the other disk drives.

- Fault tolerant power and logic

  The DC power to the drawer is supplied by two DC buses. All of the drawer logic is powered from a pair of DC power cards. In the unlikely event of a power card failure, the second power card provides continuous power to the drawer logic.

- Nonvolatile drawer cache

  Each drawer contains batteries for power backup. If the primary power is interrupted, these backup batteries can power the drawer for a sufficient amount of time to allow changed data and parity in the drawer cache to be written to the disk drives.

- Compensating cooling fans

  Both the rack and the drawers have compensating cooling systems. In the rack, two cooling fans are interconnected so that if one fails, the other increases speed to maintain the required cooling for the device adapter (DA) cards and the power subsystem in the rack. In the drawer, the two fans are

also interconnected so that if one fails, the speed of the other one increases to provide the required cooling.

- Multiple paths to data

  The RAMAC DASD uses the device level selection enhanced (DLSE) architecture of the 3990-3390 subsystems. Each drawer has 4-path connection to the 3990.

- Predictive failure analysis

  The RAMAC DASD disk drive performs a set of processes called predictive failure analysis (PFA). PFA is designed to predict disk failure conditions before they occur. When one of the predictive functions detects an anomaly, the disk drive notifies the drawer of the problem. This information is then sent to the 3990, where the appropriate service information message (SIM) is generated and sent to the system.

- Dual mainline power cables

  Two AC mainline power cables are standard in the RAMAC DASD. When connected to separate power sources, the RAMAC DASD has continuous power as long as one mainline power cable has power. In addition to providing protection against a single power source failure, this capability can allow maintenance of data center power distribution systems without disrupting access to the RAMAC DASD.

- Redundant (2N) power subsystem

  A fault tolerant redundant power subsystem is standard with the RAMAC DASD. Full power to the unit is maintained in the event of a failure of a power cable or a power supply.

- Concurrent licensed internal code installation

  A service representative can add or delete licensed internal code in the 3990 Storage Control without disrupting data access. Installation is concurrent with your operation. Activating modified LIC may result in the momentary interruption of data access.

- Nondisruptive drawer installation and removal

  The RAMAC DASD is designed to allow continuous access to your data while allowing the addition or removal of array drawers.

  A service representative can install or remove array drawers without interrupting availability to other array drawers in the RAMAC DASD. Access is retained to the data on previously installed array drawers.

- Automatic media maintenance

  There are no user media maintenance requirements for the RAMAC DASD for recoverable data checks. The handling of defective surfaces on the disk drives that make up the RAMAC DASD is done automatically.

- Dynamic sparing for drawer logic and disk drive failures

  Dynamic sparing, when enabled by the service representative, allows a drawer to notify the subsystem of an error and automatically copies the data to a predefined spare drawer, without customer involvement. The process, similar to the dual copy function of the 3990 storage control, maintains full access to the data while the copy operation is in progress. Within a 3990 storage control, you can have up to four spare drawers defined.

Initiation of the dynamic sparing function is a parameter that is selected by the service representative at the service panel as either an automatic process or a process that can be invoked by the service representative at the time of the repair. Dynamic sparing requires that 3990 cache and nonvolatile storage (NVS) are both active.

If the drawer failure involves a disk drive, access to data is maintained through the RAID 5 architecture of the drawer array.

Appendix C. Products and Tools

In this appendix we describe products and tools from IBM that help you build a continuously available CICS environment.

C.1 IBM CICSPlex System Manager for MVS/ESA

Program No. 5695-081

IBM CICSPlex System Manager for MVS/ESA (CICSPlex SM) is a product in the CICS family that provides a real-time, single system image of the multiple CICS/ESA and CICS/MVS 2.1.2 systems that make up the transaction processing environment in many enterprises. From a single point of control, CPSM can be used to monitor and operate the CICS address spaces and resources throughout an enterprise. It also:

- Provides management by exception, automatically warning staff or software of deviations from intended behavior
- Performs workload management, allowing dynamic workload balancing across CICS systems.

Functions and Benefits

- Dynamically keeps track of all CICS resources and resource changes
- Provides a single point of control for all tasks. All CICSplexes can be controlled from a single logon
- Manages multiple systems with little or no effort on the part of the user
- Single System Image reduces the risk of operator errors, with the result that CICS end users see improved reliability and higher availability of their systems.

CICSPlex SM is unique in offering a true single system image. The whole CICSplex can be managed as if it were a single CICS system. This improves your ability to manage, enterprise wide, a CICSplex by reducing the complexity of operations and systems programming.

Minimum Requirements for Management System

- Hardware: Runs on any processor that supports CICS/ESA.
- Software:
  - MVS/SP 3.1.3 or later
  - CICS/ESA 3.3 or later, for the managing address space
  - CICS/ESA 4.1 or later if goal-oriented WLM is used, or if the XCF is to be used when there are CICSPlex SM instances on different MVS images
  - ISPF 3.3 or later
  - DFP 2.3 or later
  - TSO/E 2.1 or later
  - ACF/VTAM 3.4 or later.

C.2 IBM CICS VSAM Recovery MVS/ESA

Program No. 5695-010

IBM CICS VSAM Recovery MVS/ESA (CICSVR) is an automated recovery tool that helps you avoid loss of CICS VSAM data in the event of hardware, human or software errors. CICSVR

- Uses CICS journals and backup copies of lost data sets
- Allows you to select the data sets you want to recover and to select the function you need
- Allows you to choose among forward recovery, backout, or complete recovery
- Invokes an archive utility each time a CICS journal is archived. It reads the CICS journal and stores information in the CICSVR recovery control data set (RCDS). CICSVR uses this information when creating recovery jobs.

Functions and Benefits

- Guides you through the creation of your recovery job. An extensive help function ensures that you always know what to do next
- Has an ISPF dialog interface that complies with Common User Access (CUA)
- Automates recovery job construction
- Supports manually constructed recovery jobs
- Lets you manage the contents of the recovery control data set using the dialogue interface.

CICSVR is ideal when you require an easy-to-use, fast, reliable, and customizable data recovery system with automation capability. CICSVR provides data security and reduces the chance of future data loss.

Minimum Requirements for Management System

- Hardware: Any IBM S/370 or S/390 processor that supports MVS/XA 2.2 or MVS/ESA; maximum of 4Mb of virtual storage below the 16 MB line
- Software:
  - One of these MVS environments - MVS/SP Version 2 Release 2 (MVS/XA); MVS/SP V3 R1.1; MVS/ESA SP V4 R2
  - One of these products - MVS/XA Data Facility Product (MVS/XA DFP) V2 R3; MVS/ESA Data Facility Product (MVS/DFP) V3
  - One of these CICS environments - CICS/MVS 2.1.2; CICS/ESA 3.3; CICS/ESA 4.1
  - Also required - System Modification Program/Extended (SMP/E) Release 5 Modification Level 1
  - Interactive System Productivity Facility (ISPF) V3 R3
  - Interactive System Productivity Facility/Program Development Facility (ISPF/PDF) V3 R3.

C.3 NetView for MVS/ESA

Program No. 5655-007

IBM NetView for MVS/ESA is a key part of the SystemView program for managing networks and systems through automation. Netview for MVS/ESA is a comprehensive network management solution that lets you monitor your client/server devices on Transmission Control Protocol/Internet Protocol (TCP/IP) networks:

- Manage TCP/IP, Token Ring, Novell, and LAN Management Utilities (LMU) managed workstations using MultiSystem Manager, AS/400s (using NetView ROM)
- Manage Systems Network Architecture (SNA) subarea and Advanced Peer-to-Peer Networking (APPN) resources with NetView′s connections to VTAM
- Manage your multivendor networking system with dynamically built graphical displays showing status and topology information
- Configuration, fault, and performance management functions
- Network and system management tool that provides distributed and centralized management of your entire world-wide enterprise.

Functions and Benefits

- Management of heterogeneous, multivendor networks, virtually any device managed through a NetView service point
- Resources stored as objects in a high speed data cache
- Dynamic Discovery and automatic correlation of workstation protocols including TCP/IP, APPN, IPX, Token Ring, and NetBios
- Integrated performance data using NetView Performance Monitor

IBM Netview for MVS/ESA will benefit users who need central management capability for their entire I/T business. NetView for MVS/ESA provides the only systems management server capable of managing thousands of multi-vendor resources. It provides easy to use graphics and administration facilities to help you create a client/server environment that boosts productivity for your help desk and operations staff.

Minimum Requirements for Management System

NetView for MVS/ESA runs in a virtual storage environment on any System/370 or S/390 processor or configuration with sufficient storage that supports the corresponding system environment.

C.4 Enterprise Performance Data Manager/MVS

Program No. 5695-101

SystemView Enterprise Performance Data Manager is a systems management tool that collects and transforms detailed measurement data into reports that focus on such key business areas as availability and performance management and service level reporting:

- Provides performance data relating to business-oriented workloads and assists in managing service levels efficiently
- Collects, stores, refines, analyzes, and reports systems measurements data at the enterprise level using DB2

Functions and Benefits

- Use of DB2 provides easy access to the stored performance data and makes access to distributed data across multiple SAA platforms possible
- S/390 Parallel Sysplex support with parallel processing and data sharing across systems
- User-friendly environment enables high end-user productivity and easy downloading of EPDM historical data
- MVS Workload Management Reporting
- Provides the information needed to maintain a high level of service with high-quality graphic reports for management presentations

SystemView Enterprise Performance Data Manager/MVS provides the information that is needed to maintain a high level of service. It provides centralized reporting of a distributed network of AS/400 and RISC System/6000 computers on an MVS system.

Minimum Requirements for Management System

Runs on any IBM processor (or equivalent) with the MVS/ESA operating system.

C.5 IBM Service Level Reporter for MVS

Program No. 5665-397

Service Level Reporter (SLR) is a systems management tool that collects and transforms detailed measurement data into reports that focus on key business areas, such as availability and performance management, and service level reporting in a non-DB2 MVS environment. Performance data can easily be collected and analyzed on an MVS system from the following areas:

- MVS and VM systems and major subsystems
- Network
- CICS
- IMS

Functions and Benefits

- Provides the ability to process a mixture of compatibility and goal mode records
- Supplies consolidated reports, at the sysplex level, on shared direct access storage devices (DASD) and channels and coupling facility structures
- Provides the ability to process a mixture of new and old RMF workload data on the same input log data set
- Mapping of compatibility-mode and goal-mode workloads into common user-specified names; this makes it easier to report on the workloads across mode changes.

Service Level Reporter is the preferred solution for managing and reporting of systems measurements data in a non-DB2 environment. The availability of prompt, standardized performance reports, when required, improves the efficiency with which the users can monitor and manage computer resources, thus increasing productivity.

Minimum Requirements for Management System

- Hardware:

  SLR runs on any S/370 or S/390 processor that supports MVS/ESA or MVS/XA; minimum 2.5 Mb of virtual storage in the private area under 16 Mb. When running data collection, more than 4 Mb of virtual storage may be needed.

- Software:

  MVS/SP Version 2 or 3; or MVS/ESA Version 4 or 5. One of the following Data Facility Products is required: MVS/XA DFP Version 2; MVS/DFP Version 3; DFSMS/MVS Version 1.

C.6 IBM Automated Operations Control/MVS

Program No. 5685-151

Automated Operations Control provides functions to simplify the management of multiple systems connected as a sysplex or parallel sysplex. It provides the operator with a single point of control and simplifies enterprise systems management. Automation options (CICS, IMS, and OPC) are integrated at a command level:

- Status is recorded as changes to the appropriate fields in RODM and displayed through icon color changes on the AOC graphical interface.
- Gives the operator the capability of viewing the health of all important system resources from a single point.

Functions and Benefits

- S/390 Parallel Sysplex support
- Provides policy dialogue enhancements which support a sysplex or parallel sysplex environment
- Automation options for CICS, IMS and OPC packaged as AOC/MVS with integrated commands
- Provides the capability of building the automation control file in the background, thus reducing the time required to dedicate a TSO session to build the policy information
- The batch build capability significantly reduces the amount of time a system operator is involved with the policy build operation. This frees the operator to handle other tasks within the data center.

Automated Operations Control/MVS provides the system operator with the ability to manage a large enterprise, simplifying the management of multiple systems connected as a sysplex or parallel sysplex.

Minimum Requirements for Management System

AOC/MVS supports any IBM processor capable of running MVS/ESA. Release 3 runs as an application under NetView V2 R4. The AOC/MVS operator interface requires a personal workstation capable of running OS/2 2.0 or higher.

C.7 Information/Management for MVS/ESA

Program No. 5695-171

IBM Information/Management is an MVS problem and change management and reporting system with additional functions for managing configuration and inventory data:

- Remote Site Recovery Feature automates transport storage and processing of log data and recovery information
- Optional replication of databases is automated, including the capability to recover from broken replica databases without disrupting active site processing while continuing to replicate other databases.

Functions and Benefits

- Data sharing facilities enable block level data sharing with more than two Central Processing Complex (CPC) units
- Automated Operations Facilities facilitates simplifying operations and systems management
- Object Oriented Technology enables you to create and use objects in the IMS TM environments; this minimizes your cost in terms of utilizing existing systems
- Easy to set up and use with NetView for MVS.

The IBM Information Management System (INFO/MAN) provides software tools for systems management functions including problem, change, configuration and inventory management.

Minimum Requirements for Management System

INFO/MAN/ESA executes on all IBM processors capable of running MVS/ESA Version 4 Release 2 or later.

C.8 Teleprocessing Network Simulator

Program No. 5688-121

Teleprocessing Network Simulator (TPNS) enables you to test and evaluate telecommunications application programs (such as CICS/ESA, DB2, IMS/ESA, and TSO applications), communications access methods, control programs, subsystems, and networks. It allows you to examine system performance using multiple scenarios. By testing applications and tuning networks in simulation environments, you can ensure user productivity when they use these applications on the job.

Functions and Benefits

- Simulate various terminals, terminal features, and terminal operator actions
- Provide repeatable test conditions

- Provide a simple method for describing terminal, line, and network activity
- Generate message traffic over communication lines
- Capture and time stamp messages and responses
- Report results in a meaningful manner
- Eliminate the need to install a real network before application testing
- Simulate heavy message traffic
- Expose potential problems
- Reduce the resource required for testing
- Improve testing thoroughness.

TPNS can be used by personnel in development, quality control, and capacity planning functions.

Minimum Requirements for Management System

- Hardware:

  TPNS Version 3 Release 4 runs in a virtual storage environment in any IBM system configuration that supports MVS/370, MVS/ESA, MVS/XA, or VM/SP.

  For the simulation of remote terminals or cross-domain communication over SDLC links, TPNS Version 3 Release 4 requires an IBM 3745, 3720, or 3725 communication controller, which executes a TPNS control program.

- Software:

  TPNS Version 3 Release 4 runs on the following operating system releases:

  – MVS/370 (MVS/SP Version 1; 5752-VS2)
  – MVS/XA (MVS/SP Version 2; 5740-XC6 for JES2 or 5665-291 for JES3)
  – MVS/ESA (MVS/SP Version 3; 5685-001 for JES2 or 5685-002 for JES3)
  – VM/SP 370 (5664-167; Release 4 or later)
  – VM/SP HPO (5664-174; Release 4 or later)
  – VM/SP XA (5664-308; Release 2)
  – VM/ESA (5684-112; Release 1 or later).

  TPNS Version 3 Release 4 also runs on subsequent releases or modification levels of the above operating systems, unless otherwise stated in the announcement documentation of the future release or modification of the operating system or of TPNS.

C.9 IBM CICS Transaction Affinities Utility MVS/ESA

Program No. 5696-582

The IBM CICS Transaction Affinities Utility MVS/ESA is designed to help identify potential difficulties when exploiting a CICSplex, or dynamic transaction routing environment. It is also intended to help identify the interrelationships between various transactions on a CICS system, and to identify those transactions that might cause difficulties when using the transaction isolation feature of CICS/ESA 4.1. The Transaction Affinities Utility MVS/ESA works by detecting the

CICS application programs issuing EXEC CICS commands that may cause transaction affinity.

CICS transactions use many different techniques to pass data from one to another. Some of these techniques require that the transactions involved must execute in the same CICS AOR, and, therefore, impose a restriction on dynamic routing. Such transactions are said to have affinity between them. It is important that the affinities within a CICS workload be understood before dynamic routing can be implemented to balance a workload across a CICSplex.

In order to detect as many potential affinities as possible, IBM CICS Transaction Affinities Utility MVS/ESA should be used against all parts of the workload, including rarely-used transactions and out-of-normal situations.

Functions and Benefits

- Detects inter-transaction and transaction-system affinities
- Assists migration by detecting affinities before migrating to a CICS/ESA dynamic transaction routing environment
- Detects affinities introduced by new or changed application suites or packages
- Runs against production CICS regions as well as test environments
- Supports CICS/MVS 2.1.2, CICS/ESA 3.2.1, CICS/ESA 3.3 and CICS/ESA 4.1.
- Builds a file of affinity transaction group definitions for CICSPlex System Manager/ESA (CICSPlex SM/ESA)
- Benefits CICS/ESA 4.1 Transaction Isolation support users by identifying some of the transactions that share task-local storage.

If you use or are planning to use the CICS transaction routing facility in a CICSplex, then the IBM CICS Transaction Affinities Utility MVS/ESA is for you. It is designed to detect potential causes of inter-transaction affinity and transaction-system affinity.

The IBM CICS Transaction Affinities Utility MVS/ESA is also of use to those planning to implement asynchronous processing using CICS function shipping, and to those planning to use the transaction isolation facility of CICS/ESA 4.1.

This program is useful not only to run against production CICS regions, but also in test environments and to detect possible affinities introduced by new or changed application suites or packages.

Minimum Requirements for Management System

- Hardware: IBM CICS Transaction Affinities Utility MVS/ESA runs on any IBM S/370 or S/390 processor that supports MVS/ESA with enough processor storage to meet the combined requirements of this program, the host operating system, CICS, and the access methods and application programs.
- Software:

  IBM CICS Transaction Affinities Utility MVS/ESA is designed to run with any of the following operating system releases:

  – MVS/System Product-JES2 Version 3 Release 1.3 (5685-001)

- MVS/System Product-JES3 Version 3 Release 1.3 (5685-002) and MVS/Data Facility Product (MVS/DFP) Version 3 Release 1 (5665-XA3) (Lower levels of JES2/JES3 can be used with MVS/SP Version 3)
- MVS/ESA System Product-JES2 Version 4 Release 1 (5695-047)
- MVS/ESA System Product-JES3 Version 4 Release 1 (5695-048) and MVS/Data Facility Product (MVS/DFP) Version 3 Release 2 (5665-XA3) or a later release.

IBM CICS Transaction Affinities Utility MVS/ESA is designed to run with any of the following CICS releases:

- CICS/MVS Version 2 (5665-403) with PTF UN43826 applied
- CICS/ESA Version 3 Release 2.1 (5685-083)
- CICS/ESA Version 3 Release 3 (5685-083)
- CICS/ESA Version 4 Release 1 (5655-018)

IBM CICS Transaction Affinities Utility MVS/ESA also requires System Modification Program Extended (SMP/E) Version 1 Release 5 (5668-949) or later.

The following optional programs are available, and may improve IBM CICS Transaction Affinities Utility MVS/ESA usability:

- Resource Access Control Facility (RACF) Version 1 Release 9 (5740-XXH);
- CICSPlex System Manager/ESA Version 1 Release 1 (5695-081) (when available)
- MVS/TSO Extensions (TSO/E) Version 2 Modification 2 Release 1 (5685-025)

List of Abbreviations

ACEE	access control environment element
AOR	application-owning region
APF	authorized program facility
APAR	approved program analysis report
API	application program interface
ARM	automatic restart manager
ATM	automatic teller machine
BLSR	batch local shared resources
BMP	batch message processing
BMS	basic mapping support
BSAM	basic sequential access method
BSDS	bootstrap data set
BWO	backup while open
CAF	call attachment facility
CBIPO	custom built installation productivity option
CBPDO	custom built product delivery option
CCT	communication control table
CPC	central processing complex
CICS	Customer Information Control System
CICSVR	CICS VSAM Recovery
CMF	CICS monitoring facility
CPSM	CICSPlex System Manager/ESA
CPU	central processing unit
CRC	command recognition character
CSD	CICS system definition file
CI	control interval
CT	cursor table
CWA	common work area
DASD	direct access storage device
DBRM	database request module
DBCTL	Database Control
DBD	data base descriptor
DBRC	database recovery control
DCT	destination control table
DDL	data description language
DEDB	data entry database
DLL	data link library
DMB	data management block
DML	data manipulating language
DOR	data owning region
DPL	distributed program link
DSA	dynamic storage area
DTP	distributed transaction processing
ECB	event control block
EDF	execution diagnostic facility
EDM	environmental description manager
ENQ	enqueue
EOT	end-of-task
EXCI	external CICS interface
FLPA	fixed link pack area
FOR	file owning region
GTF	generalized trace facility
HDAM	hierarchical database access method
HIDAM	hierarchical indexed database access method
HISAM	hierarchical indexed sequential access method
IBM	International Business Machines Corporation
ITSO	International Technical Support Organization
IRC	interregion communication
IRLM	IMS Resource Lock Manager
ISO	integrated services offering
IVP	installation verification procedure
JCL	job control language
JCT	journal control table
LPA	link pack area
LU	logical unit
LUW	logical unit of work
MASS	multiple address space subsystem
MLPA	modified link pack area
MQM	Message Queue Manager
MRO	multi-region operation
MSU	management service units
NEP	node error program
NVS	non-volatile storage
PCB	program communication block
PCT	program control table
PEP	program error program
PI	program isolation
PLPA	pageable link pack area

PLT	program list table
PLTPI	program list table for post-initialization
PLTSD	program list table for shutdown
PPT	processing program table
PT	package table
PTF	program temporary fix
PUT	program update tape
QSAM	queued sequential access method
RACF	Resource Access Control Facility
RCT	resource control table
RDO	resource definition online
RMF	resource measurement facility
RMI	resource manager interface
SASS	single address space subsystem
SIT	system initialization table
SKCT	skeleton cursor table
SKPT	skeleton package table
SLR	Service Level Reporter
SMF	system monitoring facility
SMP	System Modification Program
SMP/E	System Modification Program Extended
SNA	system network architecture
SNT	sign-on table
SPE	small programming enhancement
SPI	system programming interface
SQL	Structured Query Language
SSI	Subsystem Interface
TAF	terminal access facility
TCAM	Telecommunications Access Method
TCB	task control block
TCT	terminal control table
TCTUA	terminal control table user area
TEP	terminal error program
TPCP	two-phase commit protocol
TOR	terminal-owning region
TRUE	task related user exit
TS	table space
TSO	Time Sharing Option
TSO/E	Time Sharing Option/Extended
UR	unit of recovery
VIO	virtual input/output
VSAM	Virtual Sequential Access Method
VTAM	Virtual Telecommunications Access Method
WITT	workstation interactive test tool
XRF	extended recovery facility

Index

A

AOC/MVS
  product data 165
AOR 37
APAR 48
application-owning region
  See AOR
autoinstall
  APPC connections 76
  dynamic transactions 76
  programs, map sets, partition sets 76
Automated Operations Control/MVS (AOC/MVS) 58
automatic restart manager 59
automation 57
availability 3
  history reporting 61

B

backup while open
  See BWO
batch local shared resources
  See BLSR
batch message processing program
  See BMP
batch processing
  overview 61
BatchPipes
  alternatives to 117
  overview 115
BLSR 117
BMP
  data sharing 42
BWO 112

C

CBIPO 49
CBPDO 49
CEMT 78
change control 45
change management 45
changing the CICS system configuration 78
CICS/ESA
  overview 33
  XZIQUE 34
CICSplex
  description 37
CICSPlex SM 59
  overview 34
  product data 161
CICSVR
  product data 162
  recommendations 35
concurrent copy 122
continuous availability
  definition 3
  improving 18
continuous operation
  definition 2
  dynamic transaction routing 39
controllers
  recommendations 36
coupling facility 121
coupling facility channels 122
CustomPac
  ProductPac 49
  ServicePac 49
  SystemPac 49

D

DASD
  recommendations 36
data sharing 29, 100
  BMP 42
  DB2 41
  DBCTL 41
  IMS/ESA 41
  local DL/I 42
  scheduled outages 40
  VSAM 43
data subsystems
  overview 29
data-owning region
  See DOR
DATABASE 2
  See DB2
database design
  DB2 93
  DB2 index 95
  DB2 table 94
  DB2 table space 94
  disk allocation 95
  impact on availability 93
  recovery 98
  space allocation 96
Database Recovery Control
  See DBRC
daylight savings time 46
DB2
  characteristics 89
  data sharing 41
  logical design 94
  partitioning 97
  physical design 95
  recommendations 35
  segmenting 97

DBCTL
  data sharing 41
  IMS/ESA Version 4 42
  IMS/ESA Version 5 41
DBRC
  data sharing 100
  local DL/I 42
  types of recovery 106
defining resources to CICS 75
disaster recovery
  considerations 127
  high availability 142
  PPRC and XRC 142
  RRDF 146
  RSR 144
  testing 129
  tiered solutions 130
DL/I
  block level sharing 42
  data sharing 42
DOR 38
dual copy 123
dynamic transaction routing
  CICSPlex SM 40
  description 38
  enhancements 34
  program 38
  storage protection 34
  XISCONA 34

E

EPDM/MVS
  history reporting 61
  product data 163
ESCON director 121
exceptional condition handling
  HANDLE CONDITION 63
  overview 63
  RESP and RESP2 64
EXCI
  overview 71
  refresh a load module 46
EXEC CICS commands
  RESP option 64
extended recovery facility
  See XRF
extended remote copy
  See XRC
external CICS interface
  See EXCI

F

file-owning region
  See FOR
FOR
  data sharing in a sysplex 43
  description 38
forward recovery, VSAM 114
function shipping 38
functional testing 51

G

GON/GOT processing options 101

H

HANDLE ABEND 65
HANDLE CONDITION 63
hardware
  overview 36, 119
high availability
  definition 3
  dynamic transaction routing 39
hiperbatch 117

I

improving availability 18
IMS data sharing 29, 100
  block level 101
  database level 101
IMS database design to avoid reorganization 103
IMS online change utility 99
  monitoring 100
IMS remote site recovery 139
IMS/ESA
  database organization 104
  recommendations 35
IMS/ESA DB processing options 66
  GON 66
  GOT 66
InfoMan
  product data 166
irlm. 102

M

maintenance
  application software 48
maintenance strategy
  overview 48
  preventive maintenance 50
modified link pack area 45
monitoring
  history reporting 61
  performance 60
MVS Sysplex
  overview 26
MVS/ESA
  recommendations 33

N

NEP 67

NetView 57
product data 163
node error program
See NEP

P

parallel sysplex 119
parallel transaction server
See PTS
peer-to-peer remote copy
See PPRC
PEP 66
performance testing 52
persistent session support 86
personnel for disaster recovery 147
POR 38
PPRC 124
printer-owning region
See POR
problem management 56
procedures 55
processing options GON/GOT 101
Processors
recommendations 36
program error program
See PEP
PTF 48
PTS
description 155
PUT 49

Q

QOR 38
queue-owning region
See QOR
queueing
sympathy sickness 80

R

RAMAC DASD
9391 157
9392 157
description 157
features and functions 157
overview 124
PFA 158
predictive failure analysis 158
RAID 5 157
RDO 75
real time analysis
See also RTA
description 60
record level sharing
See RLS
regression testing 52
Remote Recovery Data Facility 139
resource definition online
See RDO
resource-owning region
See ROR
RESP option on EXEC CICS commands 64
RLS 43
ROR 38
RRDF 139
RSR 139
RTA
running different releases on the same MVS image 45

S

sequential data striping 118
service level agreement
See SLA
service level management
single points of failure
CICS/ESA data sets 83
FOR 83
TOR 82
SLA
description 153
SLR
history reporting 61
product data 164
software
overview 32
software failure
application code 64
storage management
dynamic DSA 80
overview 79
program compression 80
sysplex timer 120
system managed storage
use with CICS/ESA data sets 82
system topology
overview 23

T

Teleprocessing Network Simulator 11
TEP 67
terminal error program
See TEP
terminal-owning region
See TOR
test system availability 5
third party software 73
TOR
description 37
TPNS
See also Teleprocessing Network Simulator
implementation requirements 54
overview 53

TPNS *(continued)*
product data 166
transaction affinities
intertransaction affinity 68
overview 68
transaction-to-system affinity 69
transaction affinities utility
product data 167
transaction routing
description 37
dynamic 39
routing program 38
static 39

U

unavailability
definition 3
unavailability of testing systems 5
uninterruptable power supply 125
UPS 125

V

VIO to expanded storage 117
VSAM
data sharing 43
forward recovery 114
recommendations 35
record level sharing 43
VTAM
persistent session support 86
recommendations 33
VTAM generic resources
description 83

X

XRC 124
XRF
CICS/ESA availability 87

ITSO Technical Bulletin Evaluation RED000

International Technical Support Organization
Planning for CICS Continuous Availability
in an MVS/ESA Environment
November 1995

Publication No. SG24-4593-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction	____		
Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:

Do you provide billable services for 20% or more of your time? Yes____ No____

Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____

If no, please explain:

_____

_____

What other topics would you like to see in this Bulletin?

_____

_____

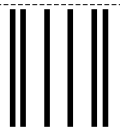
What other Technical Bulletins would you like to see published?

_____

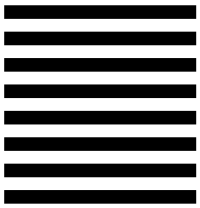
Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

Name	Address
Company or Organization	
Phone No.	

Fold and Tape **Please do not staple** Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department 471/E2
650 Harry Road
San Jose, CA
USA 95120-6099

Fold and Tape **Please do not staple** Fold and Tape

Printed in U.S.A.