IBM ImagePlus VisualInfo Library and Object Servers for MVS/ESA Planning Guide

Document Number GG24-4452-00

December 1994

International Technical Support Organization
Bethesda Center

Take Note!

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

First Edition (December 1994)

This edition applies to Release 1.0 of IBM ImagePlus VisualInfo, Program Number 5655-071 and 5655-072, for use with the MVS/ESA system.

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. FGC Location BME
6710 Rockledge Drive
Bethesda, Maryland 20817-0000

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Abstract

This document is unique in its detailed coverage of VisualInfo Library and Object Servers for MVS/ESA. It focuses on the Planning process that should take place prior to installing and implementing one of these systems. It provides information about plannning activities as well as areas of concern of which customers should be made aware.

This document was written for IBM Systems Engineering Representatives. Some knowledge of VisualInfo, DB2, CICS and DFSMS is assumed.

(64 pages)

Contents

Abstract . . . iii

Special Notices . . . xiii

Preface . . . xv
How This Document is Organized . . . xv
Related Publications . . . xvi
International Technical Support Organization Publications . . . xvii
Acknowledgments . . . xviii

Chapter 1. Introduction to VisualInfo MVS Library and Object Servers . . . 1
1.1 What is Client/Server? . . . 1
1.2 Introduction to IBM ImagePlus VisualInfo . . . 2
1.3 Introduction to VisualInfo MVS/ESA . . . 3
1.4 Positioning VisualInfo MVS/ESA . . . 4
1.4.1 Positioning VisualInfo MVS/ESA versus VisualInfo OS/2 . . . 4
1.4.2 Positioning VisualInfo MVS/ESA versus ImagePlus MVS/ESA . . . 5
1.4.3 Summary . . . 5
1.5 Product Components and Prerequisites . . . 6
1.5.1 VisualInfo Client . . . 6
1.5.2 VisualInfo MVS/ESA Library Server . . . 6
1.5.3 VisualInfo Object Server MVS . . . 7
1.6 VisualInfo Extensions . . . 7
1.6.1 FlowMark . . . 7
1.6.2 ICPF . . . 8
1.6.3 FaxRouter/2 . . . 9
1.7 VisualInfo Planned Additions . . . 10

Chapter 2. Integrating Line of Business Applications . . . 11
2.1 Application Programming Interfaces (APIs) . . . 11
2.1.1 Common Application Programming Interfaces . . . 11
2.1.2 Other Application Programming Interfaces . . . 12
2.2 User Exits . . . 12
2.2.1 Client Application User Exits . . . 12
2.2.2 Library Server User Exits . . . 14
2.2.3 Object Server User Exits . . . 14
2.2.4 Message Processing User Exits . . . 14
2.2.5 Priority for FRNI . . . 15
2.2.6 Other User Exits . . . 15

Chapter 3. Enabling Index Classes in an MVS Environment . . . 17
3.1 Index Class Usage . . . 17
3.2 System Administration Considerations . . . 17
3.3 System Programming Considerations . . . 18
3.4 Creating Index Classes . . . 19
3.5 Assumptions . . . 20
3.6 Case-Sensitive Index Searches . . . 21
3.7 Recommendations . . . 21

Chapter 4. Static Searches in VisualInfo Library Server MVS/ESA . . . 23
4.1 Dynamic Index Class View Searches . . . 23

4.2 Static Index Class View Searches . . . 23
4.3 Customer-Written Static Index Class View Searches . . . 24
4.4 Planning Considerations . . . 24
4.5 Recommendations . . . 25

Chapter 5. Database Considerations . . . 27
5.1 Grouping Database Tables Together . . . 27
5.2 Database Sizes . . . 27
5.2.1 Events Table . . . 27
5.2.2 Notes Storage . . . 28
5.2.3 Index Table . . . 28
5.3 Database Table Placement . . . 29
5.4 Loading Databases by Program . . . 29
5.5 Database Utilities . . . 29
5.6 Database Cleanup . . . 30

Chapter 6. VisualInfo Coexistence/Migration and ImagePlus MVS/ESA . . . 31
6.1 Migration Utility Considerations . . . 31
6.2 VisualInfo Library Server Considerations . . . 33
6.3 VisualInfo Object Server Considerations . . . 33
6.4 VisualInfo Client Considerations . . . 33

Chapter 7. Configurations and Network Planning . . . 35
7.1 Using the System Configuration Utility . . . 35
7.2 Loading Databases by Program . . . 36
7.3 Configuration for Small Environments . . . 36
7.4 Configuration for Large Environments . . . 37
7.5 Configuration for Large Distributed Environments . . . 39
7.6 Configuration for Mixed Environments . . . 40
7.7 Links between Library Servers and Object Servers . . . 41
7.8 Links between Object Servers . . . 42
7.9 Links between Object Servers and Clients . . . 43
7.10 Links between Library Servers and Clients . . . 43
7.11 APPC versus APPN Links . . . 43

Chapter 8. Maintenance Considerations . . . 45
8.1 Customer Control . . . 45
8.2 SMP/E versus C/370 Maintenance Considerations . . . 45

Chapter 9. VisualInfo MVS/ESA Performance and Capacity . . . 47
9.1 Capacity Constraints in CICS . . . 47
9.2 Library Server Usage Statistics . . . 47
9.3 Performance and System Parameters . . . 48
9.3.1 Database 2 MVS/ESA Parameters . . . 48
9.3.2 CICS MVS/ESA Parameters . . . 49
9.4 Performance Tips . . . 49

Chapter 10. The Object Server and Systems Managed Storage . . . 51
10.1 The MVS/OAM/OSMC Environment versus the OS/2 VisualInfo Object Serve . . . 51
10.2 Storage Management in MVS versus OS/2 . . . 51

Chapter 11. Administration . . . 53
11.1 The VisualInfo Administrator's Role . . . 53
11.2 Examples . . . 53

Appendix A. Tables 55

Appendix B. MVS/ESA Server Measurements MVS/ESA 57
B.1 Test Scenario: Simulated Workload 58
B.1.1 Measurement Settings 58
B.1.2 Measurement Results 58
B.1.3 Run 9E: 100 Simulated Workstations 59
B.1.4 Run 9G: 125 Simulated Workstations 60
B.1.5 Run 9J: 200 Simulated Workstations 61
B.2 Conclusion 62

List of Abbreviations 63

Index 65

Figures

1. The Main Components of VisualInfo System 2
2. Index Class Creation Data Model 20
3. ImagePlus MVS/ESA to VisualInfo MVS/ESA 31
4. ImagePlus MVS/ESA to VisualInfo MVS/ESA 32
5. OS/2 Configuration with MVS/ESA as Remote Object Server 36
6. OS/2 Client with MVS/ESA Library and Object Servers 37
7. ICPF Client with MVS/ESA Library and Object Servers 38
8. OS/2 Client and Object Server with MVS/ESA Library Server 39
9. OS/2 Client with OS/2 Servers and MVS/ESA Library Server 40
10. Server Link Definitions 42
11. Client Link Definitions 43
12. Object System-Managed Storage Hierarchy Chart 52
13. Configuration: Combined Library Server and Object Server 57
14. MVS/ESA Server Capacity vs. Response Time 62

Tables

1. DB2 Parameters Used for Performance Measurement 48
2. CICS Parameters Used for Performance Measurement 49
3. VI Journal Record Layout 55
4. Summary of the MVS Servers Measurements 58
5. Processing Environment: Case 9E 59
6. Detail Measurement Results: Case 9E 59
7. Measurement Summary: Case 9E 59
8. Processing Environment: Case 9G 60
9. Detail Measurement Results: Case 9G 60
10. Measurement Summary: Case 9G 60
11. Processing Environment: Case 9G 61
12. Detail Measurement Results: Case 9J 61
13. Measurement Summary: Case 9J 61

Special Notices

This publication is intended to help the IBM mrketing representative or systems engineering Representative plan for a VisualInfo MVS/ESA implementation. The information in this publication is not intended as the specification of any programming interfaces that are provided by VisualInfo MVS/ESA or its related extensions. See the PUBLICATIONS section of the IBM Programming Announcement for VisualInfo MVS/ESA 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 information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. 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.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

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, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM	AD/Cycle
Advanced Peer-to-Peer Networking	APPN

C/370
CICS
CICS/MVS
DATABASE 2
DB2
DB2/2
DFSMS
FFST
FFST/MVS
First Failure Support Technology
First Failure Support Technology/MVS
FlowMark
IBM
ImagePlus
IMS/ESA
MVS/DFP
MVS/ESA
MVS/SP
OS/2
RACF
S/390
SAA
VisualAge
VTAM
xHLLAPI

The following terms, which are denoted by a double asterisk (**) in this publication, are trademarks of other companies:

EASEL
Interactive Images, Inc.

Other trademarks are trademarks of their respective companies.

Preface

This document is intended to be used for planning purposes by IBM personnel responsible for implementing a VisualInfo MVS/ESA system. This document should provide enough planning information to help the IBM Representative prepare the customer for the VisualInfo MVS/ESA system and its extensions.

This document should also help the IBM Representative make an educated decision on when VisualInfo MVS/ESA should be proposed instead of other ImagePlus products. Throughout this document, when we refer to VisualInfo, we refer to all the VisualInfo components, including the Client Application which is supplied with the system. Some of the restrictions mentioned may actually be caused by the Client Application and may not apply to a customer planning to write their own client application using the Library Server APIs or even the Folder Manager APIs.

How This Document is Organized

The document is organized as follows:

- Chapter 1, “Introduction to VisualInfo MVS Library and Object Servers”

  This chapter contains an introductory presentation of VisualInfo family of products and design methodology.

- Chapter 2, “Integrating Line of Business Applications”

  This chapter provides more detail about VisulInfo MVS interfaces. It also gives an overview on user exits.

- Chapter 3, “Enabling Index Classes in an MVS Environment”

  This chapter introduces index class concept and how to administer the index class on a MVS environment.

- Chapter 4, “Static Searches in VisualInfo Library Server MVS/ESA”

  This chapter discusses various VisualInfo query functions.

- Chapter 5, “Database Considerations”

  This chapter presents VisualInfo database structure. Details are given for consideration on database placement.

- Chapter 6, “VisualInfo Coexistence/Migration and ImagePlus MVS/ESA”

  This chapter discusses migration issues.

- Chapter 7, “Configurations and Network Planning”

  This chapter provides samples on how to configure VisualInfo system.

- Chapter 8, “Maintenance Considerations”

  This chapter discusses several maintenance issues.

- Chapter 9, “VisualInfo MVS/ESA Performance and Capacity”

  This chapter presents parameters for performance tuning.

- Chapter 10, “The Object Server and Systems Managed Storage”

  This chapter discusses another important issue on how to manage system storage.

- Chapter 11, “Administration”

  Important administration functions are discussed in this chapter.

- Appendix A, “Tables”

  This appendix contains VisualInfo internal journal record layout.

- Appendix B, “MVS/ESA Server Measurements MVS/ESA”

  This appendix lists some performance measurement data.

Related Publications

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

- *ImagePlus VisualInfo General Information and Planning Guide,* GK2T-1709
- *ImagePlus VisualInfo, Understanding the IBM ImagePlus VisualInfo Solution*, GC31-7672
- *ImagePlus VisualInfo, Managing Your IBM ImagePlus VisualInfo Solution*, GC31-7673
- *ImagePlus VisualInfo, Selecting and Planning for IBM ImagePlus VisualInfo Solution*, GC31-7675
- *ImagePlus VisualInfo, Using the IBM ImagePlus VisualInfo Solution*, GC31-7676
- *ImagePlus VisualInfo, Getting Started with the IBM ImagePlus VisualInfo Solution*, GC31-7677
- *Installing the IBM ImagePlus VisualInfo Library Server for MVS/ESA*, GC31-7687
- *Installing the IBM ImagePlus VisualInfo Object Server for MVS/ESA*, GC31-7688
- *Enabling Index Classes for IBM ImagePlus VisualInfo Library Server for MVS/ESA*, GC31-7689
- *Operating the IBM ImagePlus VisualInfo Library Server for MVS/ESA*, GC31-7699
- *Configuring an IBM ImagePlus VisualInfo Custom System*, GC31-7711
- *Customizing User Exits for the IBM ImagePlus VisualInfo Library Server for MVS/ESA*, GC31-7720
- *ImagePlus VisualInfo Application Programming Guide, Volume 1: Folder Manager Applications and Library Interfaces*, SC31-7662
- *ImagePlus VisualInfo Application Programming Guide, Volume 2: Image Services Interface*, SC31-7664
- *ImagePlus VisualInfo Application Programming Reference, Volume 1: Folder Manager Applications Interfaces*, SC31-7663
- *ImagePlus VisualInfo Application Programming Reference, Volume 2: Image Services Interface*, SC31-7682
- *ImagePlus VisualInfo Application Programming Reference, Volume 3: Common Data Structures and Database Tables*, SC31-7665

- *ImagePlus VisualInfo Application Programming Reference, Volume 4: Library Reference*, SC31-7667
- *ImagePlus VisualInfo Administration and Operation Guide* , SC31-7661
- *ImagePlus VisualInfo User's Guide*, SC31-7670

International Technical Support Organization Publications

- *The Library for Systems Solutions Image Processing: ImagePlus and VisualInfo Client/Server Solutions*, GG24-4109-00
- *VisualInfo Systems Management: CID Installation*, GG24-4415-00
- *A Simple Approach to VisualInfo*, GG24-4444-00
- *VisualInfo Building Blocks for REXX and OS/2 Command Line*, GG24-4500-00

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

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

To get listings of ITSO technical publications (known as "redbooks") online, VNET users may type:

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

How to Order ITSO Technical Publications

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 1-800-284-4721. VISA or MasterCard is accepted. Outside the USA, customers should contact their local IBM office.

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 books on a variety of products.

Acknowledgments

The advisor for this project was:

Scott Chen
International Technical Support Organization, Bethesda Center

The author of this document is:

Jose Goncalves
IBM Canada

Michael A. Ebbers
IBM Education and Training

Jan M. Bester
IBM South Africa

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

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

John Reid
International Technical Support Organization, Bethesda Center

David Rusnak
IBM US

John DiClementi
IBM US

Marty Reed
IBM US

Chapter 1. Introduction to VisualInfo MVS Library and Object Servers

The architectural design of the IBM ImagePlus VisualInfo system is based on a new software engineering methodology called client/server. VisualInfo MVS/ESA is one of the VisualInfo server family of software. The concept of the client/server is different from the traditional large mainframe system development methodology. Before we start, here are a few words about client/server concepts.

1.1 What is Client/Server?

Although client/server is the leading buzzword in today's software industry, there is, actually, no agreed-upon definition of what that term really means. However, most of the experts agree that client/server systems should have the following distinguishing characteristics:

- *Service*: Client/server is primarily a relationship between processes (programs) running on separate (or same) machine(s). The server process is a provider of services. The client is a consumer of services. In essence, client/server provides a clean separation of function based on the idea of service.
- *Shared resources*: A server can service many clients at the same time and regulate their access to shared resources.
- *Asymmetrical protocols*: There is a many-to-one relationship between clients and server. Clients always initiate the dialog by requesting a service. Servers are passively waiting on requests from the clients.
- *Transparency of location*: The server is a process which can reside on the same machine as the client or on a different machine across a network. Client/server software usually masks the location of the server from the clients by redirecting the service calls when needed. A program can be a client, a server, or both.
- *Mix and match*: The ideal client/service software is independent of the hardware or operating system software platform. You should be able to mix and match client and server platforms.
- *Message-based exchanges*: Clients and servers are loosely coupled systems which interact through a message-passing mechanism. The messages deliver service requests and replies.
- *Encapsulation of services*: The server is a "specialist". A message tells a server what service is requested; it is then up to the server to determine how to get the job done. The server can be upgraded without affecting the clients as long as the published message interface is not changed.
- *Scalability*: Client/server systems can be scaled horizontally and/or vertically. Horizontal scaling means adding or removing client workstations with only a slight performance impact. Vertical scaling means migrating to a larger and faster server machine or multiple servers.
- *Integrity*: The server code and server data are centrally maintained, which results in cheaper maintenance and guarding of shared data integrity. At the same time, the clients remain personal and independent.

1.2 Introduction to IBM ImagePlus VisualInfo

The IBM ImagePlus VisualInfo system is a complete client/server document management system. It includes three major components which are:

- Client
  - Client Application
  - System Administration Program
  - Image Services
  - Folder Manager
  - Library Client
- Library Server
- Object Server

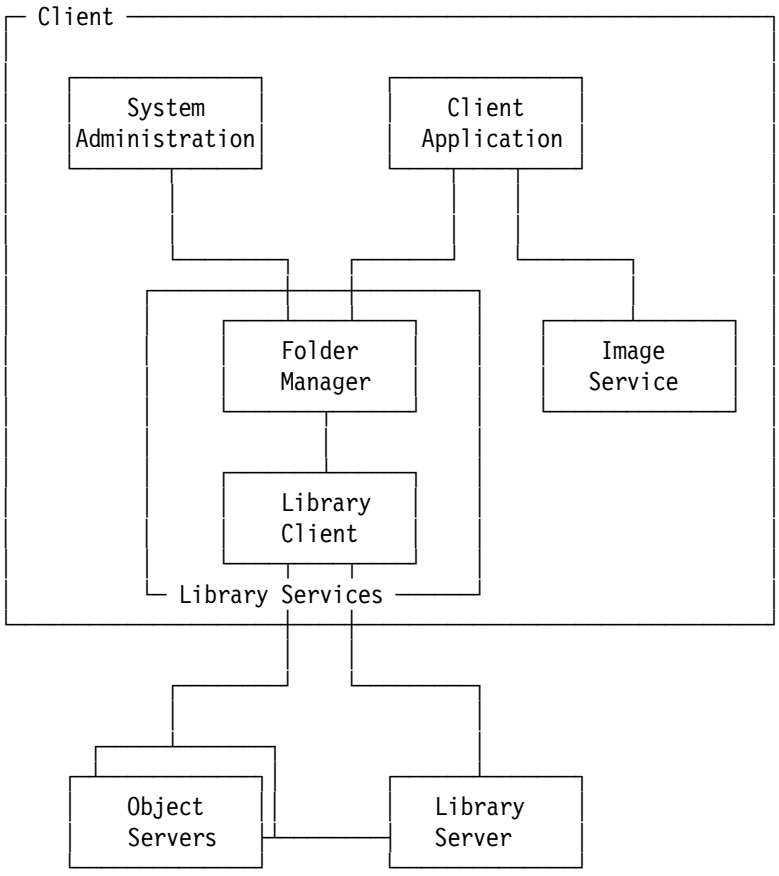


Figure 1. The Main Components of VisualInfo System

Figure 1 on page 2 shows the overall picture of the VisualInfo system. The VisualInfo client presently runs only in an OS/2 environment. However, additional client software development is planned (see listing on Section 1.7, “VisualInfo Planned Additions” on page 10) and will be announced at a later date. The client system includes a flexible and comprehensive set of APIs (Application Programming Interfaces), which provide a powerful tool for customers to develop a document management application that includes library and information-processing capabilities for multiple media types in a client/server environment.

1.3 Introduction to VisualInfo MVS/ESA

The VisualInfo MVS/ESA is a member of *the server system* of the IBM ImagePlus VisualInfo system. The components of VisualInfo which run in the MVS/ESA environment are:

- Library server
- Object server

The MVS library server system creates and maintains the master copy of the library catalog, basically described as the data information about the stored object. It also directs requests from the library client to the appropriate object server to perform the following functions:

- Store, retrieve, and update objects stored in the object servers.
- Update and query the object indexes and descriptive information stored in the library catalog.

The VisualInfo library server runs under MVS transaction processor CICS control and uses MVS Database 2 (DB2) to store and manage library data. It performs basically the same functions as the OS/2 version, even though some processes may be slightly changed to accommodate the MVS/CICS/DB2 environment. These differences are highlighted in later chapters of this book.

The VisualInfo object server maintains, stores, and delivers the objects stored in the system. In the OS/2 environment, the object server also provides the System Managed Storage. In the MVS environment, all the System Managed Storage functions are performed via DFSMS. The MVS VisualInfo object server simply receives requests from the library or the client and communicates with the MVS Object Access Method (OAM) to store or retrieve objects.

The VisualInfo MVS/ESA object server also runs under CICS/ESA control and uses APPC to communicate with the clients. .

1.4 Positioning VisualInfo MVS/ESA

This section will attempt to position VisualInfo MVS/ESA with VisualInfo OS/2 and ImagePlus MVS/ESA. This is not an easy task since some of the decisions may be dictated by corporate direction instead of selecting specific products based on their functions and the requirements of the application.

To start, let's establish the strengths of each product.

1.4.1 Positioning VisualInfo MVS/ESA versus VisualInfo OS/2

Let's compare VisualInfo servers running in the OS/2 Environment with those in the MVS/CICS environment.

1.4.1.1 VisualInfo under MVS/CICS

Let's assume that the library and object servers are running under the MVS environment. The following are reasons why a customer may decide to go with VisualInfo MVS/ESA:

- High vertical growth capability.
- Capability to support large applications (hundreds of users, millions of documents).
- Centrally managed, tight control.
- High degree of recoverability.
- High degree of security and control.
- High reliability for mission-critical applications.
- Systems Managed Storage provided by operating system.
- Faster response time to users.
- Beside MVS, CICS is capable of running on other IBM hardware platforms such as CICS/400, CICS/6000, etc. It's even capable of running on other vendors' hardware (HP/CICS), thus, it is possible to develop an integrated data warehouse client/server information application system.

1.4.1.2 VisualInfo Running on the OS/2 Platform

Let's assume that the library and object servers are running under the OS/2 operating system. Following are reasons why a customer may decide to go with VisualInfo OS/2:

- Small isolated applications.
- High degree of control by end users.
- Easier to modify.
- Customer does not have an MVS machine.
- Fast deployment.
- No involvement required from the information services department.
- Scalability (start small).

1.4.2 Positioning VisualInfo MVS/ESA versus ImagePlus MVS/ESA

The following section will attempt to position VisualInfo MVS/ESA with ImagePlus MVS/ESA.

1.4.2.1 VisualInfo MVS/ESA

- Client/server architecture.
- Highly flexible configuration.
- APIs reside at the workstation level making it easy to port applications.
- Versatility (the architecture supports various kinds of objects).
- GUI turnkey application (standard).
- FlowMark integration.
- Easier integration with other vendors′ components such as OCR and Image Services/2 replacements.

1.4.2.2 ImagePlus MVS/ESA

- Host Print (high volume) as well as workstation print.
- APIs at the host (tighter integration for IMS/CICS applications) or workstation (as a services offering by the Connecticut trading area).
- Batch store capabilities.
- Higher performance/capacity capabilities.
- Multiple level of APIs (IPFAF, IODM, OAM, IWPM/2).
- IMS or CICS support of IPFAF.
- Document versioning.
- Proven and reliable technology.
- Windows 3.1 and DOS support (IWPM/DfW and IWPM/DOS).
- Basic MVS application existed (FWA), no need to develop application system.

1.4.3 Summary

The product′s strengths should be taken into consideration at the time you evaluate the product′s functions. For example, evaluate the strengths of the IPFAF APIs versus the VisualInfo APIs along with the application requirements before making the final decision.

In general, for applications with high volumes, fast response and tight integration with legacy LOB applications, ImagePlus MVS/ESA is probably the best selection at the present time. For applications which are client/server based or have plans to go into a client/server environment in the near future, or applications which require a *high degree of flexibility and modularity,* then ImagePlus VisualInfo should be selected on the platform that it makes sense from a performance and capacity standpoint.

When selecting platforms (OS/2 vs. MVS), the user should also bear in mind the reliability, security, and control that a given platform can provide.

1.5 Product Components and Prerequisites

Let's take a look at the supporting software needed by each of the VisualInfo components.

1.5.1 VisualInfo Client

The VisualInfo Client runs on the OS/2 environment and requires at least the following software levels:

- OS/2 Version 2.1
- Communications Manager/2 (CM/2) Version 1.1
- IBM Network Transport Services for OS/2 (NTS/2) Version 1.1

After installing the prerequisite software, you may also need to apply corrective services diskettes (CSDs) to some products to bring them up to current service levels.

1.5.2 VisualInfo MVS/ESA Library Server

The MVS version of the VisualInfo library server runs under CICS and requires the following MVS software:

- MVS/ESA Version 4, Release 2.2, or later with one of the following:
  - MVS/SP JES2 Version 4, Release 2.2 (5695-047), or later
  - MVS/SP JES3 Version 4, Release 2.2 (5695-048), or later
- DATABASE 2 (DB2) Version 2, Release 3 (5665-DB2), or later
- CICS/ESA Version 3, Release 3 (5685-083), or later
- ACF/VTAM Version 3, Release 4.1 for MVS/ESA (5685-085), or later
- SMP/E Version 1, Release 7 (5668-949), or later
- ISPF/MVS Version 3, Release 3 (5685-054), or later
- ISPF/PDF Version 3, Release 3 (5665-402), or later
- TSO/E Version 2, Release 3.1 (5685-025), or later
- Assembler H Version 2, Release 1 (5668-962)
- FASTService for MVS Version 1, Release 2 (5685-088)
- FFST/MVS Version 1, Release 2 (5695-044)
- C/370 Library Version 2, Release 2 (5688-188)
- IBM SAA AD/CYCLE C/370 Version 1, Release 2.1 (5688-216)
- PL/I Version 2, Release 3 (5668-911), run-time library only

  Note: Although PL/I is required in the first release of the library server, this requirement may be dropped in future releases. Please check with the product owner to see if PL/I is still required at the time of your installation.

1.5.3 VisualInfo Object Server MVS

The MVS version of the VisualInfo library server runs under CICS and requires the following MVS software:

- MVS/ESA Version 4, Release 2.2, or later with one of the following:
  - MVS/SP JES2 Version 4, Release 2.2 (5695-047), or later
  - MVS/SP JES3 Version 4, Release 2.2 (5695-048), or later
- One of the following:
  - MVS/DFP Version 3, Release 3.1 (5665-XA3)
  - DFSMS/MVS Version 1, Release 1 (5695-DF1)
- DATABASE 2 (DB2) Version 2, Release 3 (5665-DB2), or later
- CICS/ESA Version 3, Release 3 (5685-083), or later
- ACF/VTAM Version 3, Release 4.1 for MVS/ESA (5685-085), or later
- SMP/E Version 1, Release 7 (5668-949), or later
- ISPF/MVS Version 3, Release 3 (5685-054), or later
- ISPF/PDF Version 3, Release 3 (5665-402), or later
- TSO/E Version 2, Release 3.1 (5685-025), or later
- Assembler H Version 2, Release 1 (5668-962)
- FASTService for MVS Version 1, Release 2 (5685-088)
- FFST/MVS Version 1, Release 2 (5695-044)
- C/370 Library Version 2, Release 2 (5688-188)
- PL/I Version 2, Release 3 (5668-911), run-time library only

  Note: Although PL/I is required in the first release of the object server, this requirement may be dropped in future releases. Please check with the product owner to see if PL/I is still required at the time you plan to install.

1.6 VisualInfo Extensions

Let's look at some complementary products which can be used with VisualInfo.

1.6.1 FlowMark

FlowMark is a workflow management tool which automates business processes. You can model your business and automate the flow of information and activities between departments. FlowMark can interface with VisualInfo, calling it when document processing is needed. VisualInfo provides document entry, storage, display, and output to complement the workflow automation performed by FlowMark.

Here are the software prerequisites for FlowMark:

- IBM Operating System/2 (OS/2) 2.1
- IBM Operating System/2 Developer's Toolkit 2.1 (optional)

  Note: Needed to display output of Export utility supplied with FlowMark with OS/2 VIEW command.

- **And one of the following:**
  - IBM Extended Services for OS/2 (TM)1.0

  or both of the following:

  - IBM Communications Manager/2
  - IBM DATABASE 2 (TM) OS/2 5622-044 (optional)

    Note: Needed if audit trail information is imported into DB2/2.
- **If communications with MVS host is required:**
  - IBM Application Support Facility 3.1 5655-002
  - IBM Application Support Facility 3.1 Workstation Feature

    Note: Can be used to invoke CICS/MVS, CICS/ESA, IMS/VS, and IMS/ESA transaction from the workstation. For invoking TSO programs through ASF, ASF under either CICS or IMS is required, and NetView Access Services or the equivalent session manager.

  or:

  - IBM CICS OS/2 2.0 5648-036 (optional)

    Note: Can be used to invoke CICS OS/2 transactions from this workstation.
  - IBM TCP/IP 2.0 5622-086 (optional)

    Note: Can be used for communication between FlowMark buildtime client, FlowMark server, and FlowMark OODB server. You can use either TCP/IP or NETBIOS (available with NTS/2).

1.6.2 ICPF

IBM SAA ImagePlus Capture Facility (ICPF) is a LAN-based, high-volume document capture subsystem. It offers flexibility in document capture and the ability to scan and index large document volumes. The following are major ICPF functions:

- Indexes large document volumes
- Stores large document volumes
- Manages batch processing
- Supports barcode recognition
- Performs optical character recognition (OCR)

1.6.2.1 Indexing Feature

- OS/2 Version 2.1 (32 bit compatibility mode)
- SAA ImagePlus Workstation Program/2 (IWPM/2) Version 1.2.1 or later
- OS/2 3.0 LAN Requester (comes with OS.2 LAN Server)
- and either:
  - OS/2 Extended Services 1.0 (OS/2 Communications and Database Managers)

  or both of the following:

  - Communications Manager/2 Version 1.1
  - Database 2/2 Version 1

Note: Using IWPM/2, ICPF also supports barcode and patch code recognition. Using IBM SAA ImagePlus Intelligent Forms Facility/2 (IPFO), ICPF also performs optical character recognition (OCR/ICR).

1.6.2.2 File Server

- OS/2 Version 2.1 (32 bit compatibility mode)
- OS/2 3.0 LAN Server (Entry Level)
- IBM SAA ImagePlus Workstation Program/2 Version 1.2.1

  Note: Only required for the batch maintenance application.
- and either:
  - OS/2 Extended Services 1.0 (OS/2 Communications and Database Managers)

  or both of the following:
  - Communications Manager/2 Version 1.1
  - Database2/2 Version 1

  Note: The above software products are required to install ICPF Version 1.1.1. A complete installation of VisualInfo and IWPM/2 is required if ICPF is to interface with VisualInfo. ICPF interfaces with VisualInfo at the Client level. Even though ICPF supports DOS scanning stations, they are not supported when integrating with VisualInfo since IWPM/DOS does not support the MGDS data stream.

1.6.3 FaxRouter/2

IBM FaxRouter/2 is a high-performance, client/server facsimile (fax) solution for OS/2 workstations on a LAN. It eliminates the time consuming process associated with conventional faxing. Following are its major functions:

- Automatically routes incoming facsimiles to recipients by using a built-in voice prompt or an optical mark recognition (OMR) sheet.
- Automatically generates cover sheets.
- Queues facsimiles to the FaxRouter server prior to sending.
- Retries the number for you, if the phone line is busy, and notifies you when the fax has been sent.
- Keeps a log showing the detail of all incoming and outgoing faxes.
- Stores individual facsimiles to the disk.

1.6.3.1 Server

- OS/2 2.1 with REXX support installed (Product Number 61G0900).
- OS/2 LAN Server 2.0 or 3.0 Entry Level (Product Number 96F8400) or Server 3.11 or above. OS/2 LAN Server J2.0 or J3.0.
- Communications Manager with APPC Support and the Migration Utility installed.

  Note: Required, if APPC is selected during server installation (Product Number 20G1575).
- IBM SAA ImagePlus Workstation Program/2 (Product Number 5621-047).

Note: Required for FaxRouter/2 ImagePlus MVS/ESA Integration feature only.

1.6.3.2 Client

- OS/2 2.1 with REXX support installed (Product Number 61G0900) or IBM Disk Operating System (DOS), Version 5.0 or above with Microsoft Windows Version 3.1 or above (Product Number 84F9775).
- OS/2 LAN Requester (Product Number 96F8413) or Novell Requestor.
- Communications Manager with APPC Support (Product Number 20G1575).

  Note: Required, if APPC is selected during server installation.

1.7 VisualInfo Planned Additions

The following VisualInfo products have already been announced on the statement of direction and should be available in the near future.

- AIX Servers
- Microsoft Windows Client

Chapter 2. Integrating Line of Business Applications

Although presently VisualInfo APIs are enabled only at the VisualInfo Client level, there are ways to integrate legacy applications, such as those on IMS or CICS, with VisualInfo Servers running under MVS/ESA. This can be achieved using:

- Screen scraping techniques, which can be automated with products like EASEL or EEHLLAPI under C.
- User exit points which exist at the client and server levels.
- A combination of the previously described techniques with the capabilities of DB2/2 and DDCS to access data which is under the control of LOB databases.

If the customer intends to use VisualAge as a development tool, this will help integrate applications since several VisualInfo functions have been enabled under VisualAge. A customer would have VisualAge host integration capabilities as well as VisualInfo functions.

2.1 Application Programming Interfaces (APIs)

It may be quite confusing for a new user of VisualInfo to understand where and how to invoke the different APIs and for what purpose.

In VisualInfo, there are several sets of APIs. These APIs provide functions for the different servers, but all are invoked at the client level. VisualInfo divides them into Common Application Programming Interfaces (CAPIs) available on each operating system platform supporting VisualInfo and other Application Programming Interfaces.

Here are some examples of when you should consider coding to the API interface when planning an image enabled application:

- The user already has a LOB application and needs to image enable it to add image documents to existing client folders under the control of his LOB.
- The user cannot achieve the functions desired via the standard VisualInfo Client Application.
- The user wants to store LOB objects into the VisualInfo system for consistency and workflow management.

You will find object server APIs in the documentation. However, in the MVS environment they are not applicable, since all storage management is performed by DFSMS.

2.1.1 Common Application Programming Interfaces

These APIs are strategic to the ImagePlus family of products. They follow a consistent set of style and functional standards and should remain stable throughout the various product updates. Currently, the CAPIs consist of the Image Services APIs as well as a subset of Folder Manager APIs. Although you can accomplish many functions through the CAPIs, most applications are likely to need some of the other APIs as well. Since CAPIs are intended to provide cross-product consistency in programming interfaces, product dependent functions, such as OS/2 system administration, do not as yet have CAPIs.

The implications of the lack of CAPIs at certain levels is apparent in the MVS environment when dealing with the SMS administration functions. In the VisualInfo MVS environment, all Storage Management is done outside the VisualInfo APIs and under the control of DFSMS.

2.1.2 Other Application Programming Interfaces

Other sets of APIs, such as the library server APIs, represent a lower and more detailed level of the underlying structure than do the folder manager APIs. You can use these other APIs for:

- Services not available through the folder manager.
- For better performance.
- To represent a different library data model than the one folder manager provides.

2.2 User Exits

User exits are specific points in the VisualInfo system where you can specify your own processing routines to enhance or replace default VisualInfo functions. You can create routines that run as user exits during document or folder processing. In this way you extend the features of the original base application software. You can also integrate the VisualInfo system with existing applications by creating routines as user exits.

A customer planning to write to the folder manager APIs can register one or more user exit functions for a specific session. If the customer registers the user exit, its function s are called at the completion of each VisualInfo API, but before the API returns the results. The user can use the Ip2SetUserExits API to register the users exits. The order of processing will be in the same order as you registered them.

If the customer decides to write to the library server APIs, then all exit control will have to be managed by the application the customer is writing.

Using the System Administration functions on the VisualInfo client, the administrator will be able to fill in the exit information when defining a new index class. Each index class can have a unique set of exits, giving it unique LOB integration capabilities at the client level.

2.2.1 Client Application User Exits

This is only valid when using the Client Application delivered with VisualInfo.

These are the user exits available at the Client Application level. They will be executed as OS/2 DLLs.

- AlternateSearchUserExit: Users may want to replace the search function of the client application with a user-written search routine. This exit can be used to restrict queries for specific users or sets of users, such as restricting wildcard searches.
- DetermineNextWBUserExit: This exit can be used to control the workflow based on specific LOB requirements, for example, guidelines in certain customer accounts may dictate that an item be put into a specific workbasket.

- DetermineWorkflowUserExit: The VisualInfo system automatically provides the user exit with the default workflow for the index class, as specified by the system administrator. This user exit can specify that the item should be started in a different workflow, the default workflow, or no workflow. At this point the the customer could inquire into LOB databases to make this determination.
- OverloadTriggerUserExit: This user exit is called every time a document or folder is added to a workbasket that has reached an overloaded condition, except when added as a method of satisfying suspension criteria. This exit could be used to alert supervisors to redistribute the work or assign more people to work the overloaded workbaskets.
- ChangeSMSUserExit: This user exit is called whenever the index class is changed for an item. When running the object server in the MVS environment, this exit has very little applicability because of the difficulties of changing the SMS in MVS from a workstation program.
- QuerySortUserExit: This user exit is called when a folder or workbasket is opened that contains documents or folders in a class for which the user exit is defined. This exit could be used to filter out certain documents or folders to prevent end user display access to these objects.
- SaveRecordUserExit: This user exit is called when a user chooses to save changes to the user-defined attributes (key fields) of a document or folder. You could validate the fields by matching them against existing LOB files.
- Ip2UtAlertExit: This exit provides a means of filtering out certain messages or a way to produce alerts into NetView. This exit is always linked by the VisualInfo Client.

The VisualInfo Client Application also provides sample code to support the conversion of key field data to uppercase for the SaveRecordUserExit and the AlternateSearchUserExit.

In addition, the VisualInfo system provides source code and creates files that the Client Application uses for Image Services user exits. Therefore, you can customize your own Client Application functions. The following is the source code listing:

- SimOpsStartClass
- SimOpsEndClass
- SimOpsSelectClass
- SimOpsCloseWin
- SimOpsInvokeMenu

You can replace the standard Image Service user exit FIWSXCC.DLL by modifying and compiling one or more of the first three source codes above. The last two are for the Image Service standard user exit FIWSXCLS.DLL and FIWSMNU.DLL.

2.2.2 Library Server User Exits

These exits, while called at the client, are executed at the library server level. In VisualInfo MVS/ESA, these exits are linked in the MVS/CICS environment and executed as CICS programs.

- LibACUserExitOne: This user exit is one of two that are used by the internal reference monitor during Access Control processing. It is called prior to processing the access control lists. This exit can be used to control user access to data based on certain criteria, for example, only a manager can view profiles of his/her employees. There is also a possibility that some interface with RACF may be achieved at this level.
- LibACUserExitTwo: This is the second of two user exits which are used by the internal reference monitor during Access Control processing. The access control algorithm calls this exit after processing the access control lists.

  Note: These two user exits are controlled in the MVS environment by the CICS transaction FRN5. For more details on this transaction, please refer to *Module C13, Operating the IBM ImagePlus VisualInfo Library Server for MVS/ESA*.
- Ip2UtAlertExit: This exit is used to filter out certain messages or as a way to introduce alerts into NetView and is always linked via the library server.
- LibUserQueryExit: This exit is used to specify application specific static queries. It could be invoked by a LOB application to define static queries unique to the LOB. Even if these queries cannot be executed by the VisualInfo client, they can be made to work via the client AlternateSearchUserExit.

2.2.3 Object Server User Exits

In the MVS environment, only the Ip2UtAlertExit applies. Any manipulation of SMS must be done via ACS routines or OAM interfaces.

- Ip2UtAlertExit: This exit is used as a means of filtering out certain messages or as a way to produce alerts into NetView. It is always linked via the Object server.
- Ip2LBOSExit: This exit is processed each time the LAN-based object server selects a volume for storing. It allows the user to have some control over the volume selection.

2.2.4 Message Processing User Exits

The following exit is linked via every component of the VisualInfo system prior to logging to First Failure Support Technology (FFST). If the customer wants to filter out or create alerts at the various component levels, then changes to the default exit should be made and the module link edited with the appropriate component.

- Ip2UtAlertExit: This user exit is called by every component of the VisualInfo system whenever it detects an error. In some cases, it lets the user control whether an error is logged and if it should generate an alert.

2.2.5 Priority for FRNI

The LibSelectLibraryTransPriority exit is used to set the Customer Information Control System (CICS) transaction priority for host library server requests. This exit allows the FRNI transactions to run at a certain priority.

This user exit is unique since it is linked at the client daemon level, but affects the VisualInfo MVS/ESA performance. This exit lets you specify the priority at which you would like the server to run a specific request, but affects only the priority of the library server functions (mainly queries). All functions which deal with the storage and retrieval of objects will not be affected because the object server and OAM do not support this mechanism.

2.2.6 Other User Exits

There are other user exits which can be invoked by the different VisualInfo subcomponents. For example, interchange user exits, administration program user exits, and language support user exits. These exits have no applicability in integrating LOB applications in the MVS environment.

For further information on these and other exits, please refer to Appendix B of *ImagePlus VisualInfo Application Programming Guide Volume 1: Folder Manager Application, and Library Interfaces*.

Chapter 3. Enabling Index Classes in an MVS Environment

One of the advantages of VisualInfo versus other IBM ImagePlus products is its richness in APIs and its flexibility. In the MVS environment, however, flexibility can have implications regarding security and controls. An example of this is the automatic generation of index classes in the MVS/DB2/CICS environment. The following sections will describe the mechanism used to define the index classes, including what elements can be controlled by the database administrator and system programmer.

3.1 Index Class Usage

An index class consists of one or more key fields which you assign to specific types of documents and folders. An index class is used to group similar types of objects together. For example, you can store personnel folders in a personnel folder index class. An index class will:

- Further limit access to folders and documents using access control lists.
- Provide key fields that the applications should use when storing and retrieving objects for a specific index class.
- Enable specific user exit capability for an index class.
- Provide separate storage for the different object parts such as document, note, and history.
- Provide automatic workflow processing.
- Provide automatic placement of objects in folders.

Index classes enable you to group the items in your database. Each item in your database has unique attributes, or characteristics, associated with it. An index class is a logical grouping of items that have similar attributes. When you define an index class, the VisualInfo Library Server for MVS/ESA dynamically creates a DATABASE2 (DB2) table, index, view, and access program. This way, you can associate items without needing to know the physical layout of your database.

3.2 System Administration Considerations

When planning for index classes, you should take the following into consideration:

- What type of documents do you currently process or store for a specific application? Are all these documents of the same type? Do all of them have to be scanned?
- What information do you use to identify the folders and documents?
- How do you store the folders and documents? Do you keep the document in a file drawer, a cabinet, or a warehouse?
- How do you retrieve the folders and documents?
- How do you organize the folders and documents? Do you file documents alphabetically by company name, last name, numerically by purchase order number, or by social security number?
- Do you group documents together and place them in a file folder? If so, how do you organize the folders? Do you cross-reference the documents in one folder to documents in other folders?

- How do you process the documents? For example, do you have to retrieve certain type of documents frequently once they have been filed? Do many people process these documents? Does the processing include one step or multiple steps?
- Do you have multiple security levels for different types of folders and documents?

By answering these questions, you should have a good idea on how to organize the folders and documents in a VisualInfo application system. For more information on how to plan index classes and index class subsets, please refer to the *ImagePlus VisualInfo Administration and Operation Guide*.

The VisualInfo system administrator analyzes business requirements and creates index classes using the VisualInfo system administration functions. However, before the index classes can be created, the DB2 system programmer must prepare for index class generation. In some IS centers, the security administrator also needs to prepare for setting up database access at the MVS system level.

The VisualInfo system administrator responsible for creating index classes and the MVS system programmer responsible for enabling index classes (normally a DB administrator) must consult with each other to plan the necessary steps required to accomplish this task. For more information, please refer to Chapter 11, “Administration” on page 53.

3.3 System Programming Considerations

Before index classes can be created, a job to enable them must be prepared by the MVS system programmer. The VisualInfo system administrator should gather the following information and pass it along to the system programmer.

- The number and size of the database attributes.
- The type and number of indexes.
- The appropriate number of rows in the index class table.
- Frequency of database updates, deletes, inserts, and selects.

The system programmer uses this information to organize DB2 tables and to determine which DASD volumes to place tables and indexes on. These decisions about DASD placement can affect system performance as well as backup and recovery. However, in some large IS centers, production database placement is in the performance/tuning group instead of system programmers. For more information and database considerations, please refer to Chapter 5, “Database Considerations” on page 27.

The systems programmer (or security administrator) should also:

- Ensure that the user ID specified in the batch job running the index class preparation has access to the required data sets.
- Ensure that the CICS FRNI transaction has SYSADM authority.
- Give the user ID specified in DFLTUSR bind package authority if the batch job contains a bind package step.
- Give the ability to issue refresh program copies in the CICS environment (CEMT capability).

3.4 Creating Index Classes

After the VisualInfo MVS/ESA system has been enabled for index classes, they can be created. This is done either by logging on to the system administrator function on a VisualInfo client and using the menu-driven Index Class interface, or by writing a program which uses the LibDefineIndexClass API. In either case, the VisualInfo Library server MVS/ESA takes the input and submits a batch job to JES. This job prepares a program which will access the data in the index class. The JCL for these jobs is under customer control and should be modified to satisfy specific site rules and procedures. Figure 2 on page 20 shows the overall processing model of index class creation. Following is a summary.

- The client issues LibDefineIndexClass API, label (1) in Figure 2 on page 20.
- The Library server requires DB2 to create all necessary data management (DB2) tables by executing the following DB2 SQL commands, label (2) in Figure 2 on page 20.
  - Create Table (create a new AVTnnnnn library server DB2 database)
  - Create Primary Index (IXTnnnnn)
  - Create View (ICVnnnnn)
- The Library server creates the Index Class access source program (C language) and writes it to a CICS TSQ (temporary storage queue) (3).
- The Library server reads required JCL for compiling this newly created program from its system database (4) and writes it to a CICS TDQ (transient data queue).
- The Library server writes the source program to the TDQ (5).
- The Library reads additional required JCL, linkage, copy to production load library, if necessary, DB2 bind, etc., from its database and write s to TDQ to form a complete job stream JCL (6).
- The TDQ is directed to an MVS Internal Reader (job is submitted for batch processing).

For more information, please refer to the manual, *Enabling Index Classes for the IBM ImagePlus VisualInfo Library Server for MVS/ESA*.

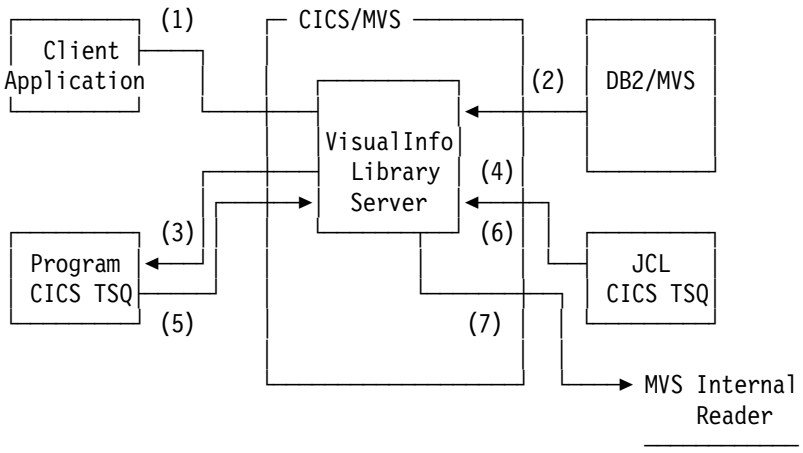


Figure 2. Index Class Creation Data Model

3.5 Assumptions

The above mentioned index classes creation process is based on the following assumptions.

Assumption 1: All of the jobs supplied assume that a compiler is available in the same MVS image as the library server CICS address space. It also assumes that the jobs are able to update the status columns in the library server DB2 tables. For a large IS center, this environment might not be the case and further customization would be needed.

Assumption 2: It is *important* to note that the CICS FRNI transaction must run with DB2 SYSADM authority. Otherwise, index creations would fail if attempted by the VisualInfo administrator. If this is not allowed for security reasons, then other methods of updating the VisualInfo databases must be considered. Since the process of defining index classes will probably occur only occasionally after the initial system definition, this should not cause great inconvenience.
Following are alternate procedures:

- Create a test system which has the same database definitions as your production system. Install the new index class on the test system. After testing, copy the definition tables to the production system. This method should be used only if the test system has duplicate database definitions from the production system.
- Test the index class definitions on a test system. Then, during an off-shift, give SYSADM authority to the FRNI transaction and create the index classes on your production system. If using System Administration from a client, you would re-enter the fields that you used in the test system. If using an API approach, you would rerun the API application copied from your test system. After the index classes are created, remove SYSADM authority.

3.6 Case-Sensitive Index Searches

When using VisualInfo MVS/ESA library server, the user should pay particular attention to the attribute fields associated with a particular index class. This is because the VisualInfo library server running in MVS *does not support mixed case index fields.* Let′s assume that a character field was defined as an index field to a specific index class, for instance, last name, and at index time the indexer wrote ″Smith″ as the last name. If somebody later searches for last name = ″smith″ or ″SMITH″ or ″SMith″, no documents will be found.

3.7 Recommendations

Consider the following recommendations when creating index classes:

- Create index classes at off-peak hours to reduce the possibility of DB2 deadlocks when creating the AVTs.
- To reduce the risk of DB2 contention when creating index classes, do not create user-defined index class tables or indexes in the same data space as the VisualInfo library server control tables.
- For more efficient placement of your data sets, use IDCAMS to allocate the VSAM clusters for the Index Class Table and Index. Use SQL to create the DB2 table spaces in the database.
- For each unique library server, always use the same DB2 table qualifier for all index class AVTs and associated indexes.

Chapter 4. Static Searches in VisualInfo Library Server MVS/ESA

Static queries are stored programs which contain SQL statements for all the queries for a particular view of the index class. If a static query is not stored and available for a given set of search arguments, a dynamic query will be created on the spot. Dynamic queries will regenerate SQL statements every time the search is performed. Any client application using the LibItemSearch request to search an index class will force the library server to perform a dynamic query against the specific index class tables.

In order to improve performance of the system, the VisualInfo Library Server has implemented a mechanism to allow the generation of static queries for searches which are executed frequently. This involves a dynamic query threshold, which is set up by the VisualInfo system administrator.

The following sections will attempt to describe the mechanism used to automatically generate these programs in an MVS/DB2/CICS/C370 environment and at the same time allow the application programmer and system programmer to have a certain degree of control.

4.1 Dynamic Index Class View Searches

The VisualInfo Library Server processes each query dynamically. This means that for every LibItemSearch API, the library server generates SQL statements to search the database. This process is done at run time and has a severe effect on system performance because of database contention and the overhead with the DB2 bind process. In order to improve performance, a mechanism was put in place to allow for the optimization of searches that are performed frequently. This is controlled via the Optimization parameter of the LibItemSearch API (Optimize_never, Optimize_now or Optimize_nopreference).

If Optimize_nopreference is received with the LibItemSearch API, then the decision to optimize the query by generating a static query is based on the value of the threshold fields in the database (Querythreshold or Dllthreshold in the FRCNTL table).

In general, you will probably want to avoid dynamic queries on your system as they use a lot of resources. The system administrator can turn off this capability by a setting in the library server configuration. See 4.5, “Recommendations” on page 25 for alternate ways to allow your users to search.

4.2 Static Index Class View Searches

These static queries are generated by the Library Server whenever the LibItemSearch API is accompanied by the parameter Optimize_now or Optimize_nopreference and the threshold has been reached. This threshold is adjusted by the administrator using the maximum queries before regeneration parameter in the library server configuration.

Whenever the VisualInfo Library Server MVS/ESA decides that it is time to generate a static query, it simply generates the C language programs with the code necessary to execute the query and submits an MVS Batch Job through JES. The JCL of this batch job is under the control of the customer and can be

customized to run according to the specific site requirements. This is adjusted by the administrator using the maximum queries before becoming a static query parameter in the library server configuration. For more information on this process, please refer to *Enabling Index Classes for the IBM ImagePlus VisualInfo Library Server MVS/ESA*.

4.3 Customer-Written Static Index Class View Searches

These queries are totally controlled by the user. The code is generated by the user and the execution of these queries is also controlled by the user. Although the VisualInfo Client Application does perform static queries, the user can force customer written static queries to be executed by coding the AlternateSearchUserExit exit and including a LibStatQuery API, specifying the query which the user wants to execute. For more information, see the *ImagePlus VisualInfo Application Programming Reference, Volume 4.*

4.4 Planning Considerations

Prior to implementing a VisualInfo MVS/ESA Library Server, the customer should carefully plan the environment in which it will run. Because the library server needs a C language development environment in order to compile and link the code generated by the server, the customer must have the following:

- A machine running the appropriate C/370 language environment. Please refer to 1.5, “Product Components and Prerequisites” on page 6. This could be in either the same machine or on a different machine connected by RJE to the VisualInfo server machine.
- A procedure to migrate the load modules from a development environment to a production environment in a speedy fashion. Most large MVS customers have an elaborate procedure to control the introduction of new programs into a production environment. This is done to prevent the system from receiving a partially prepared module which may compromise operations.
- A way to avoid or disable wildcard view searches, as they have a severe impact in system performance because of DB2 optimization. One way to disable them is to code a user exit at the client.
- Ability to schedule batch jobs to update DB2 rows in the production VisualInfo Library Server machine. If your development environment is separate from the production environment, then certain VisualInfo control tables have to be updated in order to indicate the availability of a specific static query program.

Your enterprise must determine how, when, and where to generate the optimized view search programs. The VisualInfo Library Server MVS/ESA submits a batch job to JES for program preparation. The JCL for this job is under customer control and should be modified in order to satisfy specific site rules and procedures. For more information, please refer to *Enabling Index Classes for the IBM ImagePlus VisualInfo Library Server for MVS/ESA*.

4.5 Recommendations

Consider the following recommendations when generating static query programs:

- Avoid generating static queries ″on the fly″ altogether, for performance reasons. Instead, at index class generation time, the administrator should log on to the system alone, set the threshold for dynamic queries to 1, then execute all possible queries against the data. This should generate all the static queries for the index class.

  Note: The following is an example:

  Assume you are creating an index class with two search fields.

  – Name

  – Account Number

  If the customer were to issue a search on name alone, this would generate a query. If name and account number are searched for, a different query would be generated. A search on a name like JO* would still generate another query.

  As you can see, the administrator should be aware of the different types of searches that the users might issue against a specific index class.

  After all the programs have been successfully compiled and linked into production, then set the value of Maximum queries before regeneration high enough to avoid regeneration.

  Another approach is to code and call your own static queries, as mentioned in 4.3, “Customer-Written Static Index Class View Searches” on page 24. This process can give the customer more control than leaving the generation of static queries to the VisualInfo Library Server. But it requires a much more detailed knowledge of the static query structure, as well as programming expertise.

- Consider coding the AlternateSearchUserExit exit to control the type of queries performed; for example, to avoid wildcard searches or to call your own static queries.

Chapter 5. Database Considerations

Many users who choose VisualInfo MVS/ESA servers may have done so to obtain large capacity and good performance. Capacity should be planned very carefully because of the size restrictions of DB2 tables. VisualInfo Library Server presently does not have the capability to span across multiple tables (items, parts, properties, links, events). This limits the number of documents that a user can manage using a single library server. Although the capacity of VisualInfo MVS/ESA is considerably higher than that of the OS/2 LAN version (because of DB2/2 table size restrictions), customers should still pay particular attention to this item, especially for applications which require documents to be kept under control of VisualInfo for long periods of time.

The ability for a VisualInfo administrator to create index classes interactively may pose some concern to certain customers planning to use a common DB2 subsystem to store VisualInfo tables as well as other LOB tables. The reason is that the VisualInfo CICS transaction requires SYSADM authority. A transaction with this authority could allow coding errors to corrupt the databases; therefore, customers may be reluctant to implement this. For more information on this topic, please refer to Chapter 3, “Enabling Index Classes in an MVS Environment” on page 17.

Presently, VisualInfo MVS/ESA Library and Object Servers do not have a way of synchronizing OAM activity with library parts and items. Although OAM has deleted an object because the Management class indicated that the object should be deleted, a pointer to this object will remain in the library Parts table. The customer is responsible for updating the Library Parts table.

5.1 Grouping Database Tables Together

The VisualInfo MVS/ESA default is to define each table in a unique table space. You may want to consider grouping various tables into a single table space, taking into consideration the various table sizes and dependencies. Grouping various tables into one table space may provide faster backup/recovery and better synchronization.

5.2 Database Sizes

One of the first activities in planning a database implementation is determining how large it might become. Please refer to the topic “Estimating Table Sizes for the Library Server” in the *ImagePlus VisualInfo Application Programming Reference, Volume 3: Common Data Structures and Database Tables*.

5.2.1 Events Table

One of the tables which should be carefully looked at is the Events table. It is listed as having a capacity similar to the Parts table, but because of the way the Events table is used, this may not be the case.

If your application puts every item into workflow, you will generate several events per document. Each event is on the average about 100 bytes long. It is not uncommon to have five events per routed document.

To minimize the changes of running out of table space in the Events table, VisualInfo provides a utility which scans the table for events attached to documents which are no longer in workflow queues. These events can either be deleted or migrated to optical media. Plan the migration option carefully using the tips below. The utility is started at the client and issues API requests to the library to scan the databases using dynamic queries. It passes the events data to the client, which in turn makes an object out of the data and sends it to the object server for optical storage.

The following are issues that a VisualInfo MVS/ESA user should plan for, if choosing to migrate rather than delete:

- The migration of Events into OAM where the event data is small (few events per document) may cause a larger amount of DASD to be used for object indexing purposes than if the events remained in the Events table.
- For each Event group (all events for the same document) migrating to optical, a Parts table row is used. Depending on the size of the Events being migrated, this can use quite a bit of DASD storage. Also, this will have an impact on the capacity of the Parts table.
- Since the engine that controls this migration is the client, the migration utility may run for a long time. It may be preferable to schedule the utility to run for a short time at regular intervals.

5.2.2 Notes Storage

One of the features of VisualInfo client is the ability to add notes to documents. These notes are stored as an object pointed to by a PART in the library server. Every time a user updates the notes, a new object will be created to replace the old one and the appropriate changes to the PARTS table takes place. Using the VisualInfo Client, every time a user opens a document, two objects will always be read:

- The object containing the document
- The object containing the notes

Since these objects will always be opened together, the user should try to put them into the same optical platter, whenever possible. This may be difficult to accomplish, especially if modifications to the notes don't occur at the same time the document is stored. Whenever the latter case occurs, the object containing the document and the object containing the note will be migrated to optical during different cycles. If this is likely to occur in your installation, you may want to define the note storage to be permanently on DASD or at least with a DASD life to satisfy the majority of the requests.

For further information about object placement, please refer to 7.5, "Configuration for Large Distributed Environments" on page 39.

5.2.3 Index Table

Another thing you should look at when planning database sizes is the size of the index table for index classes defined with many large variable character length index attributes, since the maximum length is set aside for the index.

Up to Version 3, Release 1 of DB2, a table space can contain at most 64 GB of data. This is the limitation of VisualInfo MVS/ESA. There are ways an application can define multiple table sets, but they have so far not been implemented by

VisualInfo MVS/ESA. Therefore, you should plan carefully when large numbers of documents have to be stored for long periods of time. You should also bear in mind that the deletion of VisualInfo library PARTS entries (for objects which have been deleted by OAM) is the responsibility of the customer.

5.3 Database Table Placement

For better performance of the customer DB2 subsystem, proper placement of tables is essential. Large and highly accessed database tables should be separated from the others. They should be placed on a separate volume, and, if possible, a separate channel and control unit. Also, the indexes should be separated from the data. Here is a list of highly used tables/indexes:

- FRNITEMS
- FRNLINKS
- FRNWIPITEMS
- FRNCHECKEDOUT
- FRNPARTS
- FRNEVENTS
- AVT00006 (No-index class)

In addition, the library server databases (tables) should be put on a separate volume and channel from the OAM system databases.

For further information on table access frequency, please refer to Table 1 in *Installing the IBM ImagePlus VisualInfo Library Server for MVS/ESA*.

For additional performance tips, see 9.4, “Performance Tips” on page 49.

5.4 Loading Databases by Program

Under some conditions, you may want to load data into VisualInfo databases using a program. For instance, instead of using the panel-driven System Administration function, you could write an API program to define VisualInfo resources. An example would be adding many user IDs to the system. A program can do this faster and easier than entering them by hand.

5.5 Database Utilities

The ability to run database utilities against the library server is a feature of VisualInfo System Administration. However, the implementation of these utilities depends on the environment VisualInfo is running under. When running VisualInfo MVS/ESA, the administrator will only have access to the history log utility and the expired time utility. For more information on these utilities, please refer to the *ImagePlus VisualInfo Administration and Operation Guide.* All VisualInfo MVS/ESA database utilities, such as backup or reorganization, must be run as DB2 utilities by the MVS database administrator. To ensure synchronization of data among distributed databases or between library and object servers, proper backup and recovery procedures are required.

The history log migration utility will scan the events database for documents which are no longer in workflow and will either delete or migrate the events into

the object server (via OAM for MVS/ESA servers). This process may take quite a long time since it is driven by the client application.

Because of this, we recommend that you run this utility off-shift unless the customer expects to have capacity problems with the EVENTS table. Since the history log migration utility will delete a good portion of the events table, a reorganization should be run soon afterwards to reclaim the space.

5.6 Database Cleanup

For every object passed to the Object server for storage, the library server creates a Parts table entry pointing to this object. When the object server is on MVS/ESA, it uses OAM to store and perform storage management functions on the object.

OAM will delete any object which expires (this is specified in the Storage Management class) and reports this activity by logging the collection name and the objects deleted under a specific collection.

In order to synchronize the library server Parts database, the customer will need to write a DB2 program which reads the object name as logged by OAM and parses from the object name the keys necessary to delete the appropriate entry in the parts table. It is the customer′s responsibility to maintain the synchronization of the Parts table to OAM.

Chapter 6. VisualInfo Coexistence/Migration and ImagePlus MVS/ESA

In some cases, VisualInfo MVS servers may be installed side-by-side with ImagePlus MVS/ESA IPFAF and IODM. From the same workstation, a user can access both the IPFAF and VisualInfo Client for OS/2 or any other VisualInfo application. You may continue using IPFAF as you are for as long as you want. However, customers may want to migrate some or all applications from an ImagePlus MVS/ESA environment to a VisualInfo environment. The following section will describe some of the issues you may want to consider before embarking on a migration project.

6.1 Migration Utility Considerations

The IBM ImagePlus VisualInfo MVS/ESA Migration feature is a component of VisualInfo that helps the customer move the ImagePlus Folder Application Facility (IPFAF) from your current host environment to a client/server solution. For the library server, you can migrate to either VisualInfo MVS/ESA library or LAN (OS/2) VisualInfo library server. You can only migrate to the MVS/ESA object server from IODM. Figure 3 and Figure 4 on page 32 show the possible migration paths.

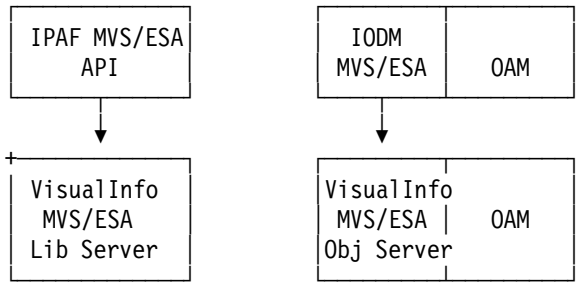


Figure 3. ImagePlus MVS/ESA to VisualInfo MVS/ESA

In either case, the MVS/ESA Migration feature allows you to:

- Migrate immediately and completely.

or

- Coexist your present ImagePlus MVS/ESA system with VisualInfo and migrate incrementally within a planned period.

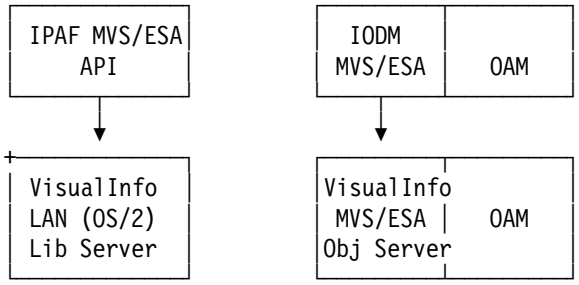


Figure 4. ImagePlus MVS/ESA to VisualInfo MVS/ESA

There are several factors which have to be taken into consideration when planning to migrate an ImagePlus IPFAF applications to VisualInfo or to coexist with VisualInfo.

- First of all, the customer must carefully decide which applications are suitable for migration/coexistence. Since the functionality between these two systems is not a perfect match, the target system may not have the same level of function.
- Next, the VisualInfo client has a new user interface requiring user retraining.
- A very important factor when planning the migration is the time required to migrate a certain application. Large applications may require a customer to be off-line for longer periods of time than their window allows, so there is an option for a partial migration. This creates a new set of problems since the user is forced to either migrate a prearranged set of folders from one environment to another (thus splitting the application), or migrate/coexist portions in such a way that the FROM system will have access to all the data until the full migration takes place. This latter option has to be carefully controlled since the potential exists for changes to the migrated data in the ImagePlus ESA system. This can be controlled by coding in the IPFAF user exits.
- One should also take into consideration that the Coexistence/Migration feature will not migrate all the components of ImagePlus IPFAF into VisualInfo and the format of the VisualInfo index classes are predetermined by the migration utility. The following are such examples:
  - Only the latest version of document is migrated (VisualInfo does not presently support document versions).
  - Folder notes are not migrated.
  - Events table will not be migrated.
  - Only high-resolution master object indexes will be migrated (IBX).
  - IODM/2 objects are not supported unless they can be routed to IODM/OAM before migration/coexistence.

For planning purposes, we strongly suggest that you review the *IBM ImagePlus VisualInfo General Information and Planning Guide* for detailed information.

6.2 VisualInfo Library Server Considerations

The VisualInfo Coexistence/Migration utility will allow a user to migrate the ImagePlus MVS/ESA folders and documents into a VisualInfo system running on either MVS or OS/2.

Before migrating an ImagePlus IPFAF application to VisualInfo, the following things should be considered:

- What environment will the target library run on (MVS/CICS or OS/2)?
- What LOB interfaces to legacy systems are required?
- What is the growth rate for this application?
- Does this application have a relationship to others which remain in the ImagePlus IPFAF environment?

The migration utility works in only one direction (IPFAF to VisualInfo), so once you run the utility and start using VisualInfo, there is no turning back. In addition, the migration works only on the latest IPFAF release (Version 2, Release 2) only.

6.3 VisualInfo Object Server Considerations

When migrating from ImagePlus MVS/ESA to VisualInfo, the objects have to remain under the control of OAM. SMS does not provide for moving objects from OAM to a LAN-based object server either. Therefore, the customer has no flexibility to move the objects from OAM to an LBOS. This may be viewed as a problem by customers who are aggressively rightsizing their applications.

6.4 VisualInfo Client Considerations

The VisualInfo Client is a OS/2 workstation application which communicates with the VisualInfo library and object servers. This client application is totally different from the FAF turnkey or any other CICS/IMS 3270 type of client application. Special consideration should be given to the end user when planning this conversion, since major retraining may have to take place.

Chapter 7. Configurations and Network Planning

One of the topics which you should consider when planning to install VisualInfo MVS/ESA is the layout of your network. Because of the flexibility to deploy library servers and object servers on different CPUs in different locations, careful planning of the connections between servers and clients will ensure reliability of service and performance.

You need to carefully analyze your document management requirement. The following questions help to start your VisualInfo planning.

- How are documents used (memos, work order, etc)?
- What are the document dimensions?
- What information contained in the documents will be used for application?
- What business processes use this information?
- What is the active life of a document?
- How long do you retain documents once they are no longer active?
- How many people handle the information?
- How many locations will store information?
- How many forms do you process each day?
- Is there any additional information associated with the document?
- How much information do you print daily?

The answers to these questions will help to make decisions about the type of VisualInfo configuration system, such as number of servers, server location, client location, and number of client station, etc. The following configurations will give the user several examples on how to run VisualInfo in mixed environments. Different configurations are possible.

7.1 Using the System Configuration Utility

This utility lets the user create or change one or more system configuration files for a custom system. The installation program needs the information in these files to create the network table for each node on your system. The network table is an ASCII file named FRNOLINT.TBL that records the following information:

- VisualInfo servers
- Protocol specific addressing information for each VisualInfo server

It is found in the VisualInfo subdirectory (currently FRNV1R0) on each LAN-based workstation. For more information on the configuration utility, refer to *Configuring an IBM ImagePlus VisualInfo Custom System*.

When installing the VisualInfo MVS/ESA library and object servers, this utility may be cumbersome, for example, in defining hundreds of clients or restricting functions which are legitimate in an MVS/ESA environment, such as attaching multiple library servers to one VisualInfo MVS/ESA object server. Here are some possible solutions:

- If your network has many clients and you are using APPN, you can define one client to the system configuration utility on each server and all clients will use that connection.
- If you want to define a connection which VisualInfo MVS will support but the system configuration utility will not (such as an object server connected to two library servers), you can run the utility and then edit the resulting network table with a standard OS/2 editor.

7.2 Loading Databases by Program

Under some conditions, you may want to load information into VisualInfo MVS/ESA with a program at its initialization time. That is, instead of using the panel-driven System Administration function, you could write an API program to define VisualInfo resources.

One example would be adding many userids to the system. A program can do this faster and easier than entering them by hand.

7.3 Configuration for Small Environments

This configuration will apply to customers which have applications with a small number of users and a small number of documents to be scanned in daily. Figure 5 illustrates a possible configuration.

In this scenario, a customer may decide to use a VisualInfo MVS/ESA remote object server for long-term retention of documents.

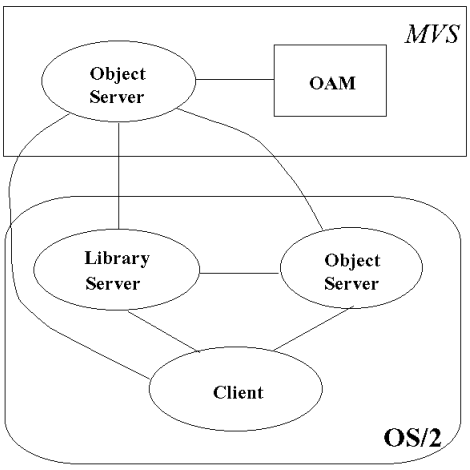


Figure 5. OS/2 Configuration with MVS/ESA as Remote Object Server

A customer will probably define the storage of documents in the OS/2 object server (DASD) with a migration into the VisualInfo MVS/ESA object server after a predetermined period of time. This will allow the customer to maintain slow speed links between their LAN environment and the host environment and still store the documents in MVS for long term archival.

It is suggested that the migration of documents from the OS/2 object server to the MVS/ESA object server be done during the night when the activity on the OS/2 object server is light.

If slow speed links are used, then you would be planning very little production activity against the MVS object server.

7.4 Configuration for Large Environments

This configuration will apply to customers who require hundreds of users to be attached to the production VisualInfo system, and who have large amounts of documents to be scanned daily. Figure 6 represents such a configuration model. The high-speed scanning configuration is shown in Figure 7 on page 38.

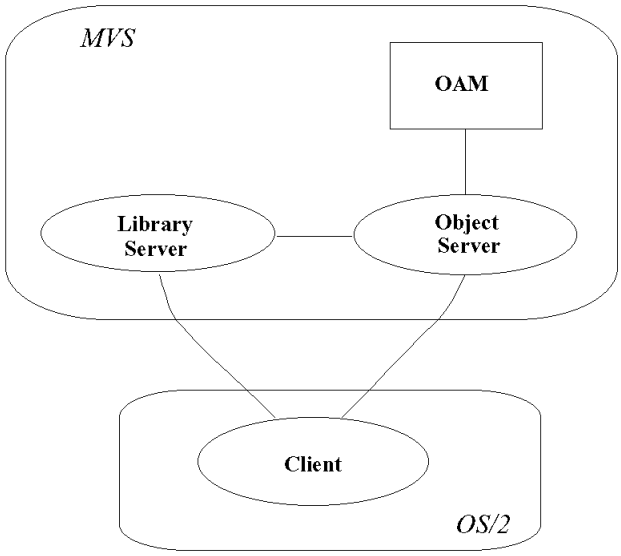


Figure 6. OS/2 Client with MVS/ESA Library and Object Servers

This scenario requires that the links between the client and the object server be high speed links in order to guarantee good performance.

The links between the library server and the object server do not have to be high speed links since the amount of data travelling between the two servers is small.

The following configuration shows the attachment of ICPF for customers that require high speed scanning into VisualInfo MVS/ESA.

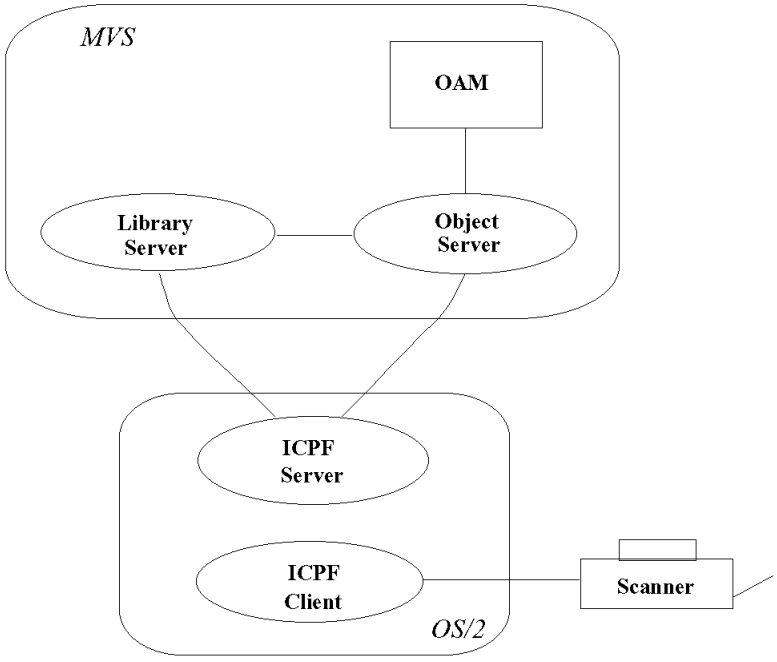


Figure 7. ICPF Client with MVS/ESA Library and Object Servers

ICPF allows a user to scan documents into VisualInfo MVS/ESA at higher speeds than that of a normal VisualInfo client. ICPF uses IWPM/2 as the scanner driver and as the user interface. ICPF is also used as the OCR/ICR interface to IPFO and IPFA for users with auto indexing requirements.

ICPF is also the vehicle to process document batches by the use of patch codes and bar codes.

Although ICPF is viewed as a high speed scanning solution for VisualInfo MVS/ESA, this will not compare (from a throughput point of view) with the High Speed Scanning feature of ImagePlus MVS/ESA.

Because of ICPF's OS/2 server dependencies, it is not expected that ICPF will be able to drive the high speed scanners (such as the KODAK 900) at rated speed.

7.5 Configuration for Large Distributed Environments

This configuration allows a customer to attach several hundred users to a central VisualInfo MVS/ESA library server and distribute the object servers.

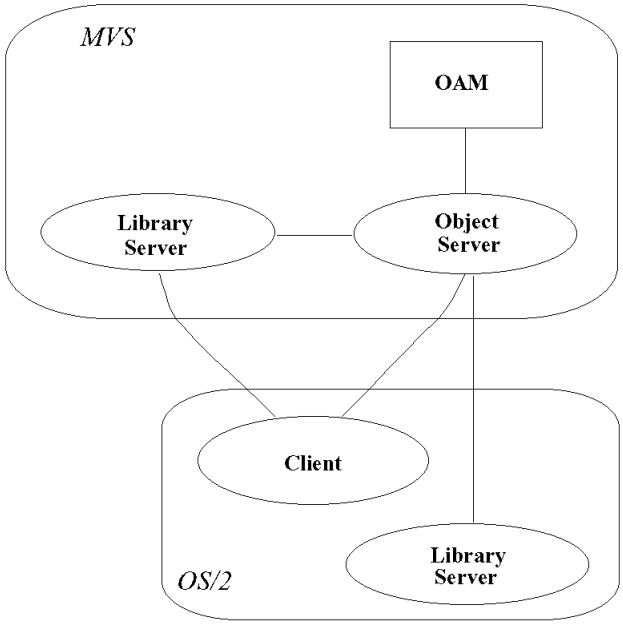


Figure 8. OS/2 Client and Object Server with MVS/ESA Library Server

The users will have the flexibility to choose which object server to place the documents in. In VisualInfo, this assignment can happen at the user ID level or at the index class level. When the object server is assigned at the user ID level, this allows the objects to be located at the user-specified location. When assigning the object server based on index class, this will allow all documents of the same index class to be placed together on the same object server.

The user ID defaults will take precedence over the index class defaults if you are storing and indexing at the same time.

When using the VisualInfo Client application, the following happens:

- At scan time, all documents are stored under the NoIndex index class. At this time, the client looks at the user defaults and if the user ID has defaults specified, then they are used to store the document. The notes, however, will use the index class defaults, therefore having two related objects in two separate object servers.
- Because VisualInfo will not move objects at reindexing time, the initial location of the objects indexed under the noindex index class will not change when the document is properly indexed.

Note: In summary, controlling the distribution of objects into collections via the client application is very difficult.

The links between the LAN and the MVS/ESA host environment do not have to be high speed links since the bulk of the data remains at the LAN.

7.6 Configuration for Mixed Environments

This configuration is geared to customers which have users who need to access applications that are departmental in nature (small) and corporate in nature (large).

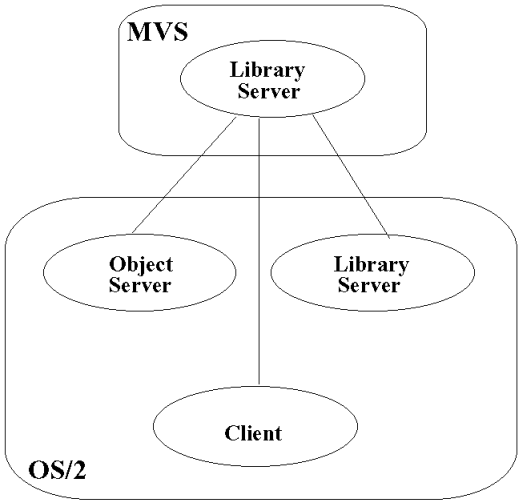


Figure 9. OS/2 Client with OS/2 Servers and MVS/ESA Library Server

Since this customer needs to run an object server at the host to support its corporate applications, it may be advantageous to connect all the LAN library servers to a central object server, since an investment in optical libraries has already been made at the host.

Unique collection names should be assigned in each library server to guarantee uniqueness within OAM. This may also be a way of identifying which library you should run the cleanup utility against. This cleanup utility is required for cleaning the Parts table for objects deleted by OAM.

When running VisualInfo MVS/ESA object servers, it is possible to attach multiple library servers to one object server. To define more than one library server per object server, the customer must bypass the configuration utility and edit the LAN workstations′ network tables manually.

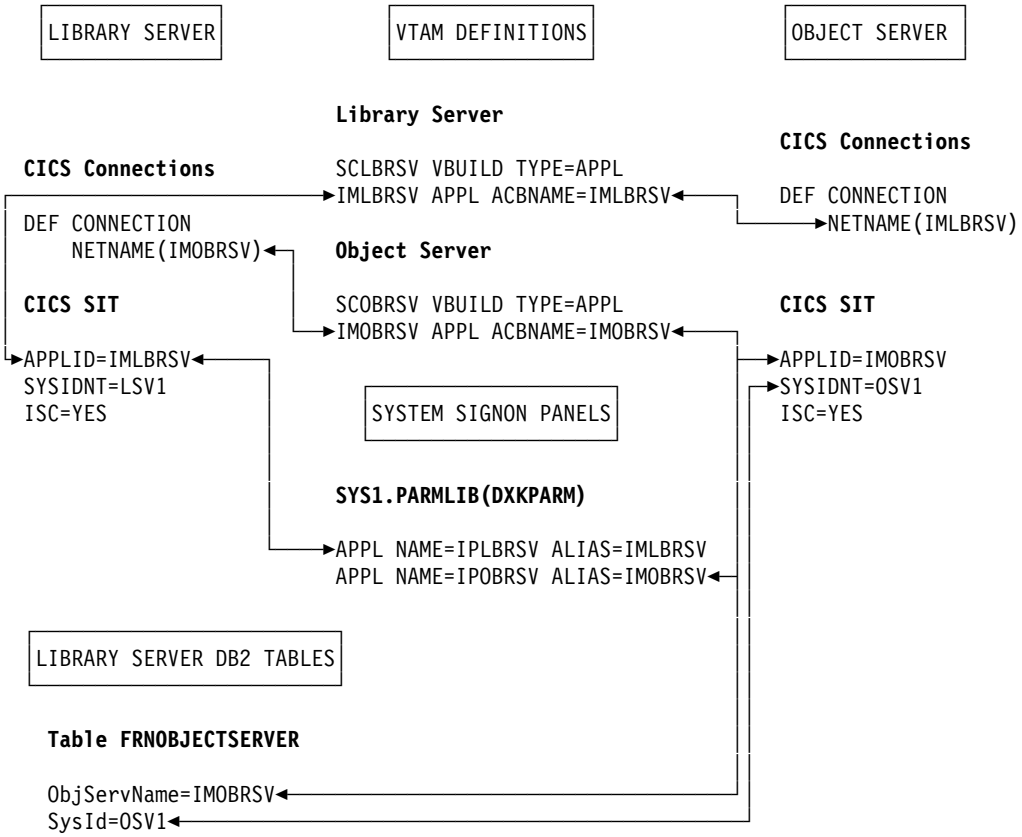
Note: Because of the capability of the OS/2 object servers to perform storage management functions such as object migration, multiple library servers to one object server would pose a synchronization problem in OS/2. The Configuration Manager does not allow users to configure VisualInfo in this way.

7.7 Links between Library Servers and Object Servers

The library server needs to communicate to the object server any time a request is received from the client which requires access to data which is under the control of OAM. VisualInfo MVS/ESA will store documents, notes and events as objects under OAM.

The object server will dialogue with the library server any time a client requests storage management changes against an object, such as migrating an object from management class X to Y. It will also inform the library server that a certain library request has completed successfully. In MVS environment, some of these links are system parameters of the transaction processor CICS subsystem, some belong to communication subsystem VTAM system parameters, and some go as database management (DB2) system parameters.

In order for the library server CICS region to communicate with the object server CICS region, the connection and session parameters of these two regions must match each other. Figure 10 on page 42 shows the summary of the different components and their relationship.



Note: The System Signon Panels (SAMON screen) may not be present in your system environment

.

Figure 10. Server Link Definitions

7.8 Links between Object Servers

An object server will talk to another object server whenever SMS requests that an object be moved from one object server to another, or an application invokes the LibMoveBlobs API.

The SMS conversations will always be initiated from the OS/2 based object server. The VisualInfo MVS/ESA object server can only receive documents from other object servers, but cannot send, except when the LibMoveBlobs API is invoked. The reason for this is that all the SMS in MVS/ESA is done externally to the VisualInfo object server.

7.9 Links between Object Servers and Clients

These links are used to send the objects from the client to the object server and vice versa. These links need to be high speed links. See Figure 11 for how to define these links.

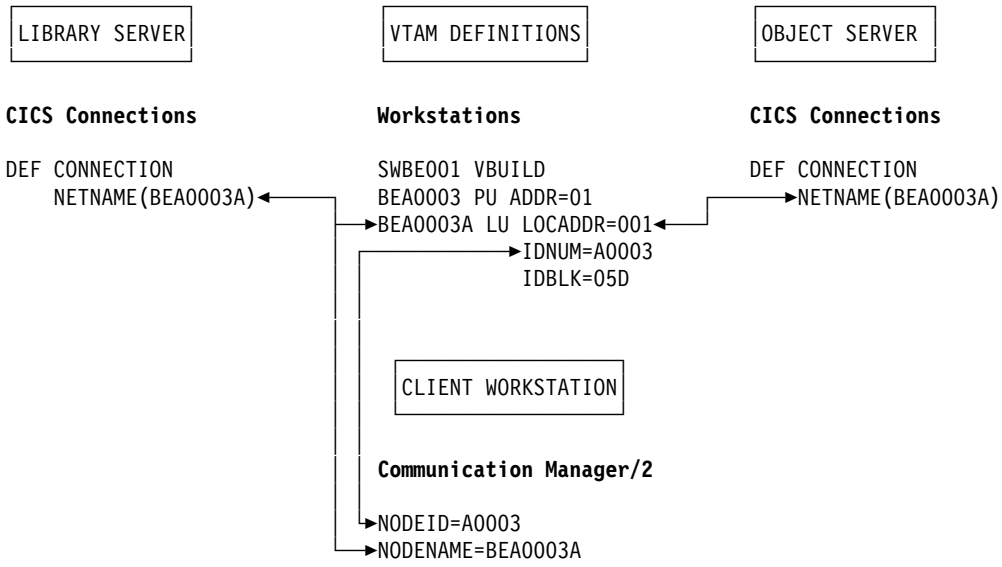


Figure 11. Client Link Definitions

7.10 Links between Library Servers and Clients

These links are used to send all the library client and folder API command requests. Even if there is high activity through these links, the bursts of data are small in nature. This may not be the case for customers writing their own client applications and issuing queries resulting in large amounts of data. See Figure 11 for how to define these links.

7.11 APPC versus APPN Links

There are certain advantages in defining your client and server connections using APPN: they are much simpler to define and they have a lot more flexibility. The problem with an APPN definition is that Communications Manager (CM/2) will release the connection after 40 seconds of inactivity. This is OK for connections which are used constantly, but in an image environment it is very likely that the time between conversations of the client and the object server may exceed 40 seconds. In that case, the response time will deteriorate due to the overhead of having to re-establish a connection every time the user wants to do something.

The recommended solution for high activity users is to define the links as APPC links. APPC links have to be hard-coded and are not as flexible as APPN definitions. They will remain active as long as the user is logged on and will be automatically acquired upon recovery from a network failure.

This is an area which should be thoroughly discussed with the person responsible for telecommunications. For more information, please read *Creating Network Tables Manually in an IBM ImagePlus VisualInfo System*.

Chapter 8. Maintenance Considerations

Because of the potential of losing track of maintenance in a complex client/server environment, software maintenance is a major issue in the MVS/ESA environment. Whoever maintains VisualInfo application software should carefully plan the maintenance strategy for VisualInfo MVS/ESA, especially when implemented on a large scale (multiple library and object servers).

8.1 Customer Control

There are two major reasons why an MVS/ESA customer wants to have strict control over software maintenance.

- System failures impact other applications and users.

  Because of the size of an MVS/ESA processor, it is normal to run many applications in the same machine, sharing common subsystems like CICS, DB2, and VTAM. A failure in any of these common subsystems will impact all of the applications using it. Although the applications, in theory, run in a separate layer, it is often the case that an application error may cause one of the subsystem layers to fail, which will affect all users.

- Application failures affect many users.

  The prime reason a customer chooses to run VisualInfo MVS/ESA is capacity. This means that the customer needs to run an application with lots of users. The impact of a failure in this situation is a lot more severe than in a small LAN environment with a few users.

There is no dialogue among clients and servers to indicate the maintenance levels under which each is running. Therefore, incompatibilities of code levels can be difficult to resolve.

8.2 SMP/E versus C/370 Maintenance Considerations

The nature of C language allows programmers to create C source files, compile them into separate programs and then link all these programs into a load module. The combination of writable static storage, long name support, etc. and lack of dynamics fetching facility such in PL/I or Cobol at execution time makes it impractical to maintain the C load modules using SMP/E.

This means that the changes delivered via SMP/E are cumulative. The latest change to a C module will have to include all the previous changes, and, therefore, will not allow a customer to pick and choose the maintenance based on application requirements.

It is common practice for customers to go through the PTF list of an application, choose the maintenance applicable to the specific problems they encounter, and then wait for a certain period of time until all other fixes have matured before they install them.

As an example of how a PTF for C includes all the previous fixes, let's take a look at the main VisualInfo MVS/ESA component FRNMHBCH, which is comprised of about 80 different programs. Whenever a problem is encountered

in any of the 80 programs, because of the requirement to pre-link them all together to produce a load module, all the maintenance is dragged in. This method does not give the user much control over maintenance selection.

Because the customer must install all the maintenance up to the PTF that the customer needs, this may require that the customer upgrade the client code as well as other server codes, if these servers are talking to each other or sharing clients.

After installing a new maintenance release for MVS/ESA servers, the customer should thoroughly test all aspects of his application. Exercising all the components fully is a requirement to ensure that all functions are still operational after installation of software maintenance.

The maintenance issues are not caused by VisualInfo MVS/ESA, but by C/370 and SMP/E. They will go away whenever the C/370 language environment and SMP/E resolve the issues that presently are causing these problems.

Chapter 9. VisualInfo MVS/ESA Performance and Capacity

This chapter deals with capacity planning and performance expectations when running VisualInfo servers in an MVS/ESA environment. Some of the capacity issues were previously mentioned in Chapter 5, “Database Considerations” on page 27, but in this section we will expand capacity issues from a CICS and VisualInfo perspective.

9.1 Capacity Constraints in CICS

A CICS region can exploit only a single processor of a multiple processor system. This means that when doing CICS work, the upper boundary of a VisualInfo MVS/ESA server is the capacity of a single processor in the 9000 system. Of course, multiple processors are exploited by DB2 and VTAM through work performed by different TCBs on behalf of the VisualInfo CICS region.

The customer has the capability of splitting the object server from the library server, but cannot run multiple library servers for the same application because of CICS memory tables synchronization.

The splitting of VisualInfo library servers into different applications is a legitimate thing to do whenever customers require processing capacity, but this should be planned before the system runs into capacity problems. The splitting of a library server into two in midstream is complex and may cause problems.

9.2 Library Server Usage Statistics

The VisualInfo MVS/ESA Library Server will allow the user to log all usage activity against the server. This logging capability will allow users to track system performance throughout the day as well as the ability to charge back users per function executed.

This utility is the equivalent of the OBJECT and LIBRARY server processor activity and performance information utility.

The usage data is written to a CICS journal. You can either use the CICS system journal, an existing journal, or define a unique journal for usage data. For information on how to define a CICS journal, please refer to the *CICS/ESA System Definition Guide*. Customer can also uses MVS RMF to monitor overall system resources performance and makes adjustment accordingly.

The customer can control the usage journal via VisualInfo MVS Library Server Global Control table entries. The keywords relevant to the usage journal are:

- JOURNAL_USAGE default(0)

  Note: Zero (0) will not record library server usage. Any value other than zero will record library usage.

- USAGE_JNL_NUM default(1)

  Note: This indicates the journal number to which the data is written.

- USAGE_JNL_TYPE default(VI)

Note: This indicates the journal record type used to differentiate one VisualInfo library server instance record from others.

- USAGE_JNL_FORMAT default(S)

  Note: Instructs the library server to create the usage journal record in one of two formats:

  - S - Sorted and summarized. Multiple occurrences of the same library server order will be grouped together.
  - O - in order sequence. The journal record will contain a list of library server orders as they are processed by the library server.

  The defaults can be overridden by either of the following:

  - Modify the contents of the data set attached to FRNCONFG DD statement in the VisualInfo library server CICS startup JCL.
  - Use FRN5 operator transaction to alter the keyword values.

    Note: These changes are effective only until the library server is restarted or changed by another instance of the FRN5 transaction.

The size of the journal record is dependent upon several factors:

- The number of orders per library server request block
- The number of unique library server orders (in case of ″S″ format)
- The choice of format ″S″ vs ″O″

For record format layout, please refer to Appendix A, “Tables” on page 55.

9.3 Performance and System Parameters

The performance group of VisualInfo development has been studied as well as the performance characteristics of VisualInfo servers both on LAN base and MVS/ESA. Appendix B, “MVS/ESA Server Measurements MVS/ESA” on page 57 includes the latest measurement on MVS servers performance. In addition, the following parameters value were used by the group to do their measurement. If you are not familiar with the number, this is a good place to start.

9.3.1 Database 2 MVS/ESA Parameters

Table 1. DB2 Parameters Used for Performance Measurement

Parameter	Setting
DB2 threads to FRNI	8
DB2 threads to FRNO	8

9.3.2 CICS MVS/ESA Parameters

Table 2. CICS Parameters Used for Performance Measurement

Parameter	Library Server Setting	Object Server Setting
MXT	120	50
AMXT	60	20
CDSA	2MB	4MB
UDSA	2MB	4MB
ERDSA	8MB	4MB
ECDSA	8MB	8MB
EUDSA	20MB	8MB

9.4 Performance Tips

Here are some tips which may increase your system performance or keep it from degrading.

- Give dispatching priority to the following address spaces in this order: VTAM, IRLM, DB2, then CICS.
- Some workloads may load up your DB2 system, for example, one which has lots of workflow. If DB2 performance on your MVS system becomes a consideration, limit the number of active DB2 threads. Start with eight dedicated threads to the Library Server FRNI transaction and eight to the Object Server FRNO transaction. Increase the threads gradually to balance transaction response time against deadlocks.
- Define your highly used databases with lots of free space. See 5.3, “Database Table Placement” on page 29 for a list. Then, after the databases and indexes are initially loaded with data, run the reorganization utility. Perform a reorg on a regular basis, especially if you often add or delete items from your VisualInfo system.
- Set the Deadlock Detection cycle in DB2 to a short time period since some tables are accessed by multiple users. This will enable the deadlocks to be detected and resolved quickly.
- Where possible, restrict the number of users who need to access a workbasket simultaneously to 10 or fewer. Give each department or worker their own workbasket if the application allows it. This will help prevent database deadlocking situations.
- In high usage situations, consider defining the links between object servers and clients using APPC rather than APPN connections. An APPN dynamic link can take 10-15 seconds to connect each time it is needed.
- Place the DB2 tables of the library server and OAM on different volumes.
- Isolate the DB2 log onto a single volume.
- Monitor the utilization of CICS DSAs and set the values of MXT and AMXT accordingly.
- Reduce the arrival rate of transactions to see if throughput increases versus queueing them up.
- If possible, design your item key fields with random values rather than sequential values.

In summary, each customer environment is different, so you should monitor any changes to ensure a positive result.

Chapter 10. The Object Server and Systems Managed Storage

Storage management in an MVS environment is a function of DFSMS and is therefore handled outside the VisualInfo product set. VisualInfo Object Server for OS/2 has storage management integrated into the object server. This section will describe the differences and issues when using a combination of MVS and OS/2 object servers, especially when the customer is planning to use system management strategies which include the movement of data from an OS/2 object server to an MVS object server.

10.1 The MVS/OAM/OSMC Environment versus the OS/2 VisualInfo Object Serve

When planning for a VisualInfo MVS/ESA object server, the user should have a storage management policy. This defines the migration of objects between the storage media and the expiration criteria of objects. Using the VisualInfo library server in the OS/2 platform, a user can define a remote object server and migrate objects to it based on management class definitions. However, the MVS host-based object server (HBOS) participates in the remote object server architecture as a recipient of objects only. DFSMS migrates objects between different storage media but cannot move objects between servers.

When planning to use a remote VisualInfo MVS/ESA object server for long term archival, we recommend that you define the storage group as having an one-day DASD retention. This will help performance as well as optical platter utilization.

10.2 Storage Management in MVS versus OS/2

The SMS process implemented through the LAB based object server environment is basically an emulation of the SMS/OAM MVS environment with a couple of exceptions:

- In MVS there is no staging area. Throughput in MVS is achieved via caching in the 3390 controllers.
- In MVS, when SMS manages objects, only optical volumes are attached to storage groups. DASD is used in placing objects based on the management class, but all DASD allocation is controlled under DB/2.
- In MVS there is no correlation of Storage Class to Device Manager. There are no Device Manager definition requirements in the DFSMS environment.

The following chart depicts all the components of SMS and its relationships. For more information, please refer to ″Planning System-Managed Storage″ in the *VisualInfo Administration and Operations Guide*.

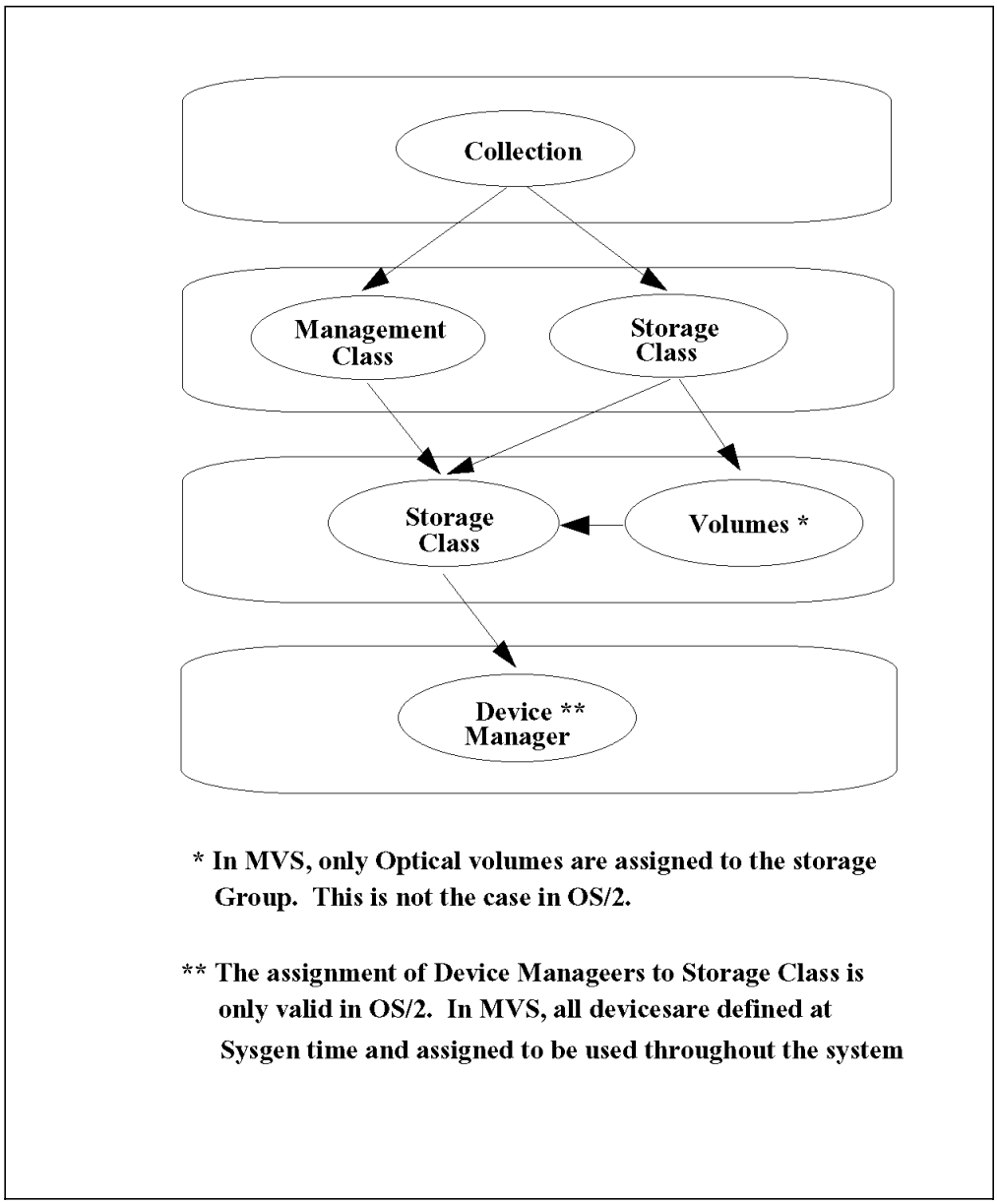


Figure 12. Object System-Managed Storage Hierarchy Chart

Chapter 11. Administration

This chapter will help to define the administrator's role in the MVS VisualInfo environment.

11.1 The VisualInfo Administrator's Role

The VisualInfo Administrator is mentioned throughout this document and most of the VisualInfo documents. Normally, most users view an application administrator as someone who adds and deletes users to the system as well as controls access authority.

A VisualInfo administrator should know the various applications running in VisualInfo and understand the impact of certain user requirements in the VisualInfo system.

The administrator has to be able to dialogue with the database administrator as well as with the network planner, the storage planner (if running in MVS) and development people.

Because of the client/server nature of VisualInfo, the administrator (from the client) has the power to affect the various components of the system (library server and object server). In an OS/2 environment, these systems are probably all under the control of the administrator, but under MVS the responsibility falls under a specialist (DB2, Communications, CICS, DFSMS, etc.). The administrator of VisualInfo MVS/ESA will have to be able to coordinate application changes with the appropriate people.

11.2 Examples

These examples demonstrate the dependence of the VisualInfo Administrator on other professionals when running in an MVS environment.

- Let's assume the administrator wants to define a new index class under VisualInfo. This will have the following results in VisualInfo MVS/ESA:
  - DB2 table spaces and tables have to be allocated and defined.
  - MVS access authority needs to be established.
  - Access programs have to be compiled/linked/bound.
  - Possible MVS JCL changes may be required.
  - Certain CICS transactions (and possible security access) have to be run to refresh in-memory tables.
- Let's suppose that the administrator decides that a certain management class should be migrated from an OS/2 object server to MVS. The following planning should take place:
  - A collection name and storage group will have to be defined separately by the person responsible for storage management in MVS.
  - Proper links have to be installed and defined to the MVS network.

Appendix A. Tables

Table 3. VI Journal Record Layout

Name	Type	Length	Description
JRECUSAG	DSECT		
JRECTSKN	DS	PL4	CICS EIB Task #
JRECUSER	DS	F	VisualInfo patron USERID
JRECTS	DS	CL26	Transaction Timestamp
JRECUSRN	DS	CL26	Patron name
JRECWSID	DS	CL17	Workstation ID
JRECFRMT	DS	CL1	Journal Format
JRECNUMO	DS	H	Number of orders that follow
JRECFSIZ	EQU	(*-JRECUSAG)	Length of fixed portion
JRECORG	DS	0C	Org point for array(s)
JRECSORG	DS	H	Summarized order code
JRECNORG	DS	H	Number of occurances of order
JRECSSIZ	EQU	(*-JRECSORD)	Size of summarized entry
	ORG	JRECORG)	Org back to array start
JRECORDR	DS	H	Order code(s) in order sequence
JRECOSIZ	EQU	(*-JRECORDR)	Size of order sequence entry

Appendix B. MVS/ESA Server Measurements MVS/ESA

The performance group of VisualInfo development has measured the various components of VisualInfo and published their results as the Performance Newsletter for the user community. Enclosed herein is their latest measurement on MVS/ESA servers. Thanks to the performance group of the BPL for allowing us to use their measurements. Figure 13 is the environment of the measurement. The group was using TPNS (IBM product) to simulate workstations. However, a word of warning. The results achieved in the lab may not be achievable for extended periods of time in a less controlled environment. Many factors can adversely affect performance of the MVS/ESA Object Server store operations. For example, free space in key VisualInfo tables, the number of DB2 threads defined between the CICS regions, and DB2 will have a significant effect on the number of database deadlocks, user response time, and the server throughput capacity.

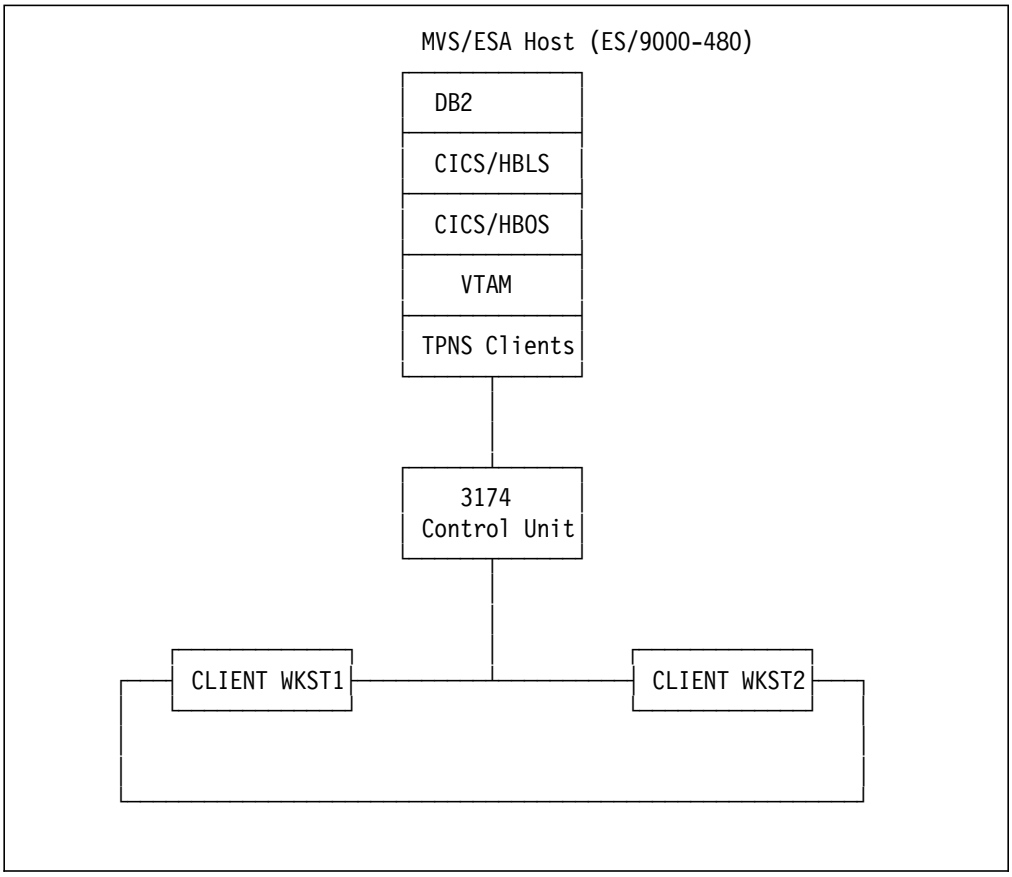


Figure 13. Configuration: Combined Library Server and Object Server

B.1 Test Scenario: Simulated Workload

This test simulates a workplace environment using the workload scenario.

B.1.1 Measurement Settings

Ten 60-minute runs were made, each varying on one or more of the following parameters:

- Number of TPNS clients
- Think time between iterations
- Think time between functions
- EUDSA (Extended User Defined Storage Area) for the HBLS/CICS region.

Along with the response time of each function in the workload scenario, several key metrics were captured for each run:

- CPU Utilization (including CICS/HBLS, CICS/HBOS, TPNS, DB2, and TSO)
- Total Library Server Function Requests (LSFRs) processed
- Library Server Connect Time (Average Response Time of Library Server Order Code 1034, as reported by the FRN6 transaction)
- Number of DB2 deadlocks experienced by the Library Server database

B.1.2 Measurement Results

A summary of those 10 runs follows:

Table 4. Summary of the MVS Servers Measurements

Run number	Number of TPNS Clients	Average Think Time Between Iterations (secs)	Average Think Time Between Functions (secs)	EUDSA (MB)	CPU Utilization (%)	LSFRs	Connect Time (secs)	Deadlocks
9A	50	30	2	16	75.4	68,976	0.290	6
9B	65	30	2	20	79.1	71,028	0.910	14
9C	75	30	2	20	76.1	69,516	1.484	14
9D	75	60	4	16	69.3	62,425	0.214	7
9E	100	60	4	16	78.8	71,273	0.800	7
9F	125	60	4	20	78.2	70,798	2.036	6
9G	125	75	5	16	75.3	69,331	1.307	15
9H	125	75	5	20	34.1	69,331	0.255	1
9I	150	180	12	20	52.1	35,317	0.046	2
9J	200	180	12	32	67.9	62,353	0.370	6

Details on selected runs 9E, 9G, and 9J (corresponding to the low, medium, and high activity workloads previously defined) are as follows:

B.1.3 Run 9E: 100 Simulated Workstations

Table 5. Processing Environment: Case 9E

Number of TPNS Clients	EUDSA (MB)	Elapsed Time (Min.secs)	Minimum Think Time Between Iterations (secs)	Maximum Think Time Between Iterations (secs)	Minimum Think Time Between Functions (secs)	Maximum Think Time Between Functions (secs)
100	16	60:00	30	90	1	7

Table 6. Detail Measurement Results: Case 9E

Function	Low Response Time (secs)	High Response Time (secs)	Average Response Time (w/o Attach, secs)	Average Response Time (w/ Attach, secs)	Number Process-ed
Create Folder	0.05	2.02	0.93	0.94	1373
Create Item	0.05	2.02	0.93	0.94	4118
Add Item to Folder	0.07	2.15	1.02	1.03	4113
Store 100KB Object	0.31	4.92	2.25	2.26	4113
Retrieve 100KB Object	0.25	2.53	1.32	1.33	4113
Store 1KB Object	0.21	4.19	1.86	1.87	4113
Retrieve 1KB Object	0.17	2.32	1.12	1.13	4113
Start Workflow	0.09	4.07	1.81	1.82	4113
Get Next Workbasket Item	0.02	1.76	0.70	0.71	4112
Index Item	0.15	4.45	2.17	2.18	4111
Route Item	0.06	3.88	1.65	1.66	4109
Complete Workflow	0.07	3.73	1.60	1.61	4107

Table 7. Measurement Summary: Case 9E

Functional Requests Processed by the Library Server	71,273
Total Number of Attach Requests Processed	50,819
Low Response Time for Attach Requests (secs)	0.00
High Response Time for Attach Requests (secs)	0.04
Average Response Time for Attach Requests (secs)	0.01
Average Connect Order 1034 Response Time (secs)	0.800
Number of DB2 Deadlocks from HBLS	7
CPU Utilization (%)	78.8

B.1.4 Run 9G: 125 Simulated Workstations

Table 8. Processing Environment: Case 9G

Number of TPNS Clients	EUDSA (MB)	Elapsed Time (Min.secs)	Minimum Think Time Between Iterations (secs)	Maximum Think Time Between Iterations (secs)	Minimum Think Time Between Functions (secs)	Maximum Think Time Between Functions (secs)
125	16	60:00	40	110	1.25	8.75

Table 9. Detail Measurement Results: Case 9G

Function	Low Response Time (secs)	High Response Time (secs)	Average Response Time (w/o Attach, secs)	Average Response Time (w/ Attach, secs)	Number Process-ed
Create Folder	0.11	2.63	1.20	1.21	1245
Create Item	0.11	2.63	1.20	1.21	3735
Add Item to Folder	0.08	2.75	1.32	1.33	3763
Store 100KB Object	0.38	5.98	2.93	2.94	3104
Retrieve 100KB Object	0.27	3.23	1.63	1.64	3433
Store 1KB Object	0.24	5.23	2.45	2.46	3521
Retrieve 1KB Object	0.17	2.93	1.42	1.43	3528
Start Workflow	0.11	5.45	2.47	2.48	3293
Get Next Workbasket Item	0.02	2.42	0.99	1.00	3897
Index Item	0.17	5.78	2.84	2.85	3281
Route Item	0.06	5.20	2.27	2.28	3465
Complete Workflow	0.08	5.12	2.23	2.24	3839

Table 10. Measurement Summary: Case 9G

Functional Requests Processed by the Library Server	69,331
Total Number of Attach Requests Processed	49,952
Low Response Time for Attach Requests (secs)	0.00
High Response Time for Attach Requests (secs)	0.04
Average Response Time for Attach Requests (secs)	0.01
Average Connect Order 1034 Response Time (secs)	1.307
Number of DB2 Deadlocks from HBLS	15
CPU Utilization (%)	75.3

B.1.5 Run 9J: 200 Simulated Workstations

Table 11. Processing Environment: Case 9G

Number of TPNS Clients	EUDSA (MB)	Elapsed Time (Min.secs)	Minimum Think Time Between Iterations (secs)	Maximum Think Time Between Iterations (secs)	Minimum Think Time Between Functions (secs)	Maximum Think Time Between Functions (secs)
200	32	60:00	90	270	6	18

Table 12. Detail Measurement Results: Case 9J

Function	Low Response Time (secs)	High Response Time (secs)	Average Response Time (w/o Attach, secs)	Average Response Time (w/ Attach, secs)	Number Process-ed
Create Folder	0.05	1.27	0.41	0.42	1201
Create Item	0.05	1.27	0.41	0.42	3603
Add Item to Folder	0.06	1.32	0.45	0.46	3600
Store 100KB Object	0.32	3.52	1.18	1.19	3600
Retrieve 100KB Object	0.21	1.74	0.73	0.74	3600
Store 1KB Object	0.20	2.62	0.79	0.80	3600
Retrieve 1KB Object	0.14	1.56	0.53	0.54	3600
Start Workflow	0.07	2.50	0.73	0.74	3600
Get Next Workbasket Item	0.02	0.91	0.21	0.22	3598
Index Item	0.14	3.07	1.01	1.02	3595
Route Item	0.06	2.32	0.59	0.60	3590
Complete Workflow	0.06	2.16	0.54	0.55	3589

Table 13. Measurement Summary: Case 9J

Functional Requests Processed by the Library Server	62,353
Total Number of Attach Requests Processed	44,577
Low Response Time for Attach Requests (secs)	0.00
High Response Time for Attach Requests (secs)	0.03
Average Response Time for Attach Requests (secs)	0.01
Average Connect Order 1034 Response Time (secs)	0.370
Number of DB2 Deadlocks from HBLS	6
CPU Utilization (%)	67.9

B.2 Conclusion

The Figure 14 illustrates how, at constant think times, the end-user response time will increase as the number of workstations increases. Consequently, library server throughput does not increase as more users are added, since each request will wait longer to connect to the library. The three bars in the graph above are based on MVS/ESA runs 9D, 9E, and 9F.

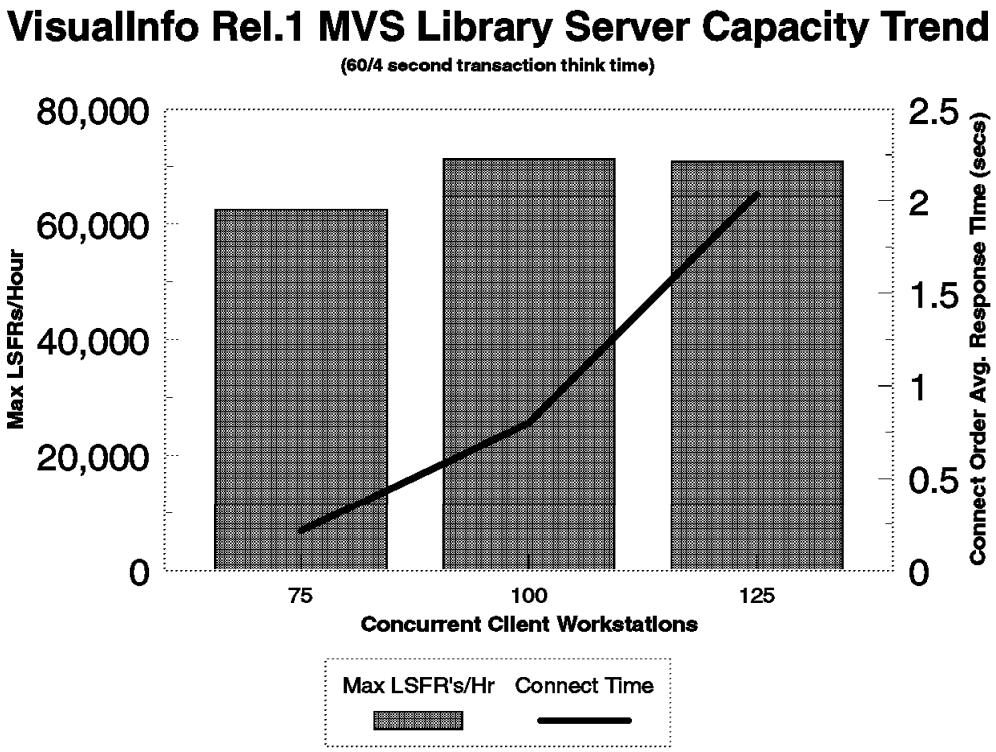


Figure 14. MVS/ESA Server Capacity vs. Response Time

List of Abbreviations

ACF/VTAM	Advanced Communications Function / Virtual Telecommunications Access Method
ACS	Automatic Class Selection
AD/CYCLE	Application Development Cycle
API	Application Programming Interface
APPC	Advanced Program to Program Communication
APPN	Advanced Peer to Peer Networking
ASF	Application Support Facility
AVT	Attributes Value Table
CAPI	Common Application Programing Interface
CICS	Customer Information Control System
CM/2	Communications Manager/2
CSD	Corrective Service Diskette
DASD	Direct Access Storage Device
DB/2	Database 2
DB2/2	Database 2 for Operating System 2
DDCS	Distributed Database Connection Services
DFSMS	Data Facility Storage Management Subsystem
DLL	Dynamic Link Library
EEHLLAPI	Entry Emulator High Level Language Application Programming Interface
ESA	Enterprise Systems Architecture
FFST/MVS	First Failure Support Technology / Multiple Virtual Storage
HBLS	Host-based Library Server
HBOS	Host-based Object Server
IBM	International Business Machines Corporation
ICPF	ImagePlus capture facility
ICR	Intelligent Character Recognition
IMS	Information Management System
IODM	ImagePlus Object Distibution Manager
IPFA	ImagePlus Intelligent Forms Assist
IPFAF	ImagePlus Folder Application Facility
IPFO	ImagePlus Intelligent Forms Facility
ISPF/PDF	Interactive System Productivity Facility / Program Development Facility
ITSO	International Technical Support Organization
IWPM/2	ImagePlus Workstation Program/2 Architecture
JES2	Job Entry System/2
JES3	Job Entry System/3
LAN	Local Area Network
LBLS	LAN-based Library Server
LBOS	LAN-based Object Server
LOB	Line Of Business
MGDS	Machine generated data structure
MVS	Multiple Virtual Storage (IBM System 370 & 390)
OAM	Object Access Method
OODB	Object Oriented Data Base
OCR	Optical Character Recognition
OS/2	Operating System 2
PL/I	Programming Language One
PTF	Program Temporary Fix
RACF	Records Access and Control Facility
RJE	Remote Job Entry
SAA	Systems Application Architecture
SMS	System Managed Storage (OS/2) or Storage Management Subsystem (MVS)
SMP/E	System Modification Program Extended
SQL	Structured Query Language

SYSADM	Systems's Administrator
TCP/IP	Transmission Control Protocol / Internet Protocol
TSO/E	Time Sharing Option Extended
VI/OS2	VisualInfo for Operating System/2
VI	VisualInfo
VI/MVS	VisualInfo MVS
VSAM	Virtual Storage Access Method

Index

A

administration 53
  administrator 53
APPC 43
Application Programming Interfaces (API) 11
APPN 43

C

capacity 47
  constraints 47
client/server 1
coexistence 31
Common Application Programming Interfaces (CAPI) 11
complementary products 7
  FaxRouter/2 9
    APPC 9
    LAN requester 10
    LAN Server 9
    REXX 9, 10
  FlowMark 7
    ASF (Application Support Facility) 8
    CICS 8
    CM/2 8
    DATABASE 2 OS/2 8
    Developer's Toolkit 7
    Extended Services 8
    IMS 8
    NETBIOS 8
    NetView 8
    OODB 8
    OS/2 7
    TCP/IP 8
    TSO 8
  ICPF 8
    File Server 9
    ICR 9
    Indexing feature 8
    IPFO 9
    IWPM/2 8
    LAN Requester 8
    OCR 9

D

database 27
  capacity 27
  cleanup 30
    necessary user-written routine 30
  considerations 27
  events table migration 28
  grouping tables 27
  index table 28
    table sets 28
database *(continued)*
  index table *(continued)*
    variable character length attributes 28
  loading databases by program 29
  necessary user-written routine 30
  note storage 28
  OAM vs Parts table synchronization 27
  performance 29
  placement 29
  sizes 27
    events table 27
    parts table 28
  SYSADM 27
  utilities 29
    Expired time 29
    History log 29
DB2 queries 23
  dynamic
    Dllthreshold 23
    LibItemSearch 23
    Querythreshold 23
  dynamic queries 23
  optimized view search programs 24
  planning for queries 24
  queries, customer-written 24
    AlternateSearchUserExit 24
    LibStatQuery 24
  recommendations 25
    Maximum queries before regeneration parameter 25
  static
    batch job 24
    JES 24
    Optimize_Nopreference 23
    Optimize_Now 23
  static queries 23
DFSMS 11

I

ImagePlus MVS/ESA 5
index classes 17
  assumptions 20
    FRNI 20
    SYSADM 20
  attributes 21
  creating 19
    access program 19
  enabling 18
  index field considerations 21
  recommendations 21
    IDCAMS 21
    VSAM 21
  searches, case-sensitive 21
  system administration 17

index classes *(continued)*
system programming 18
DFLTUSR 18
FRNI 18
SYSADM 18
introduction 1

L

Line of Business integration 11
links 41
APPC vs APPN 43
between object servers 42
library server and object server 41
to clients 43

M

maintenance 45
control of 45
PTFs, cumulative 45
SMP/E vs C/370 45
migration 31
considerations 33
rightsizing 33
utility 31
FAF user exits 32
IODM/2 32
IODM/OAM 32
MVS/ESA measurements 57

N

network planning 35
configuration utility 35
large distributed environments 39
NoIndex index class 39
large environments 37
high speed scanning 38
ICPF 38
IPFA 38
IPFO 38
KODAK 900 38
loading databases by program 36
mixed environments 40
small environments 36

O

object server 51
staging area 51

P

performance 47
performance data 48
performance tips 49
platforms, planned additions 10
prerequisites 6
client 6
CM/2 6
CSDs 6
NTS/2 6
library server MVS 6
ACF/VTAM 6
Assembler H 6
C/370 6
CICS/ESA 6
DATABASE 2 6
FASTService 6
FFST/MVS 6
ISPF/MVS 6
ISPF/PDF 6
JES2 6
JES3 6
PL/I 6
SMP/E 6
TSO/E 6
object server MVS 7
ACF/VTAM 7
Assembler H 7
C/370 7
CICS/ESA 7
DATABASE 2 7
DFSMS 7
FASTService 7
FFST/MVS 7
ISPF/MVS 7
ISPF/PDF 7
JES2 7
JES3 7
MVS/DFP 7
PL/I 7
SMP/E 7
TSO/E 7

Q

queries 23
considerations 24
dynamic 23
recommendations 25
static 23

S

static queries, customer-written 25
AlternateSearchUserExit 25
AlternateSearchUserExit 25
statistics 47
charge-back 47
usage data 47
system managed storage 51
OAM 51
OSMC 51
system parameters 48
CICS parameters 49

system parameters *(continued)*
DB2 parameters 48

T

TPNS 57

U

user exits 12
AlternateSearchUserExit 12
ChangeSMSUserExit 13
DetermineNextWBUserExit 12
DetermineWorkflowUserExit 12
FIWSMNU.DLL 13
FIWSXCC.DLL 13
FIWSXCLS.DLL 13
Ip2LBOSExit 14
Ip2SetUserExits 12
Ip2UtAlertExit 13, 14
LibACUserExitOne 14
LibACUserExitTwo 14
LibACUserQueryExit 14
OverloadTriggerUserExit 13
QuerySortUserExit 13
SaveRecordUserExit 13
SimOpsCloseWin 13
SimOpsEndClass 13
SimOpsInvokeMenu 13
SimOpsSelectClass 13
SimOpsStartClass 13

V

VisualAge 11
VisualInfo extensions 7
VisualInfo MVS/ESA 3
introduction 3
positioning 4
versus ImagePlus MVS/ESA 5
versus OS/2 4

ITSO Technical Bulletin Evaluation RED000

IBM ImagePlus VisualInfo
Library and Object Servers for MVS/ESA
Planning Guide

Publication No. GG24-4452-00

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. _____

Cut or Fold
Along Line

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 FGC, Location BME
6710 Rockledge Drive
Bethesda, MD
USA 20817-1824

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.

GG24-4452-00