

# Fidelity National Information Systems Payments Reference Architecture for IBM System z



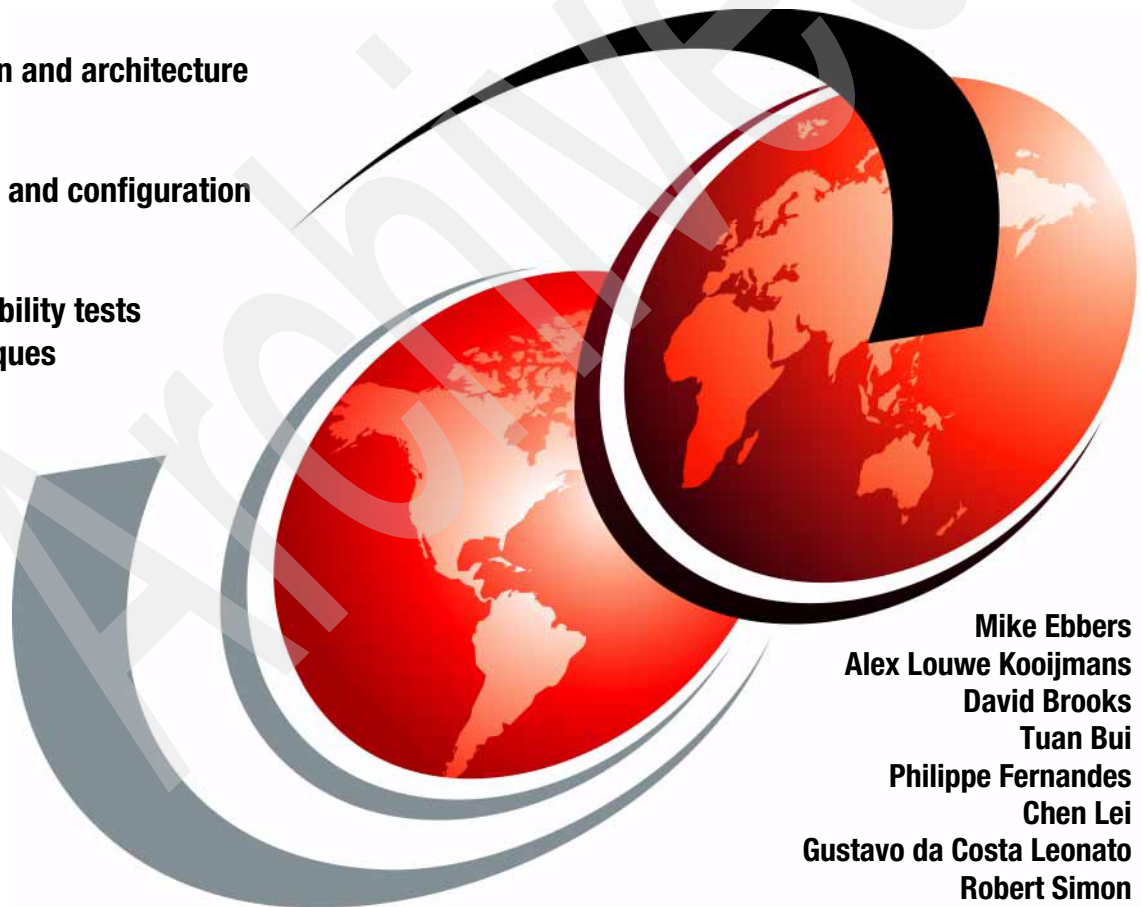
Introduction and architecture



Installation and configuration



High availability tests  
and techniques



Mike Ebbers  
Alex Louwe Kooijmans  
David Brooks  
Tuan Bui  
Philippe Fernandes  
Chen Lei  
Gustavo da Costa Leonato  
Robert Simon





International Technical Support Organization

**Fidelity National Information Systems Payments  
Reference Architecture for IBM System z**

September 2008

Archived

**Note:** Before using this information and the product it supports, read the information in “Notices” on page ix.

## **First Edition (September 2008)**

This edition applies to FIS Payments Applications on IBM z/OS V1R8.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

|  |      |
|--|------|
| <b>Notices</b> .....   | ix   |
| Trademarks .....   | x    |
| <b>Preface</b> .....   | xi   |
| The team that wrote this book .....                                  | xi   |
| Become a published author .....                                      | xiii |
| Comments welcome .....   | xiii |
| <b>Part 1. Overview of products and project</b> .....                | 1    |
| <b>Chapter 1. Enterprise payments</b> .....                          | 3    |
| 1.1 Processing terminology .....                                     | 5    |
| 1.1.1 Issuer and acquirer business models .....                      | 5    |
| 1.1.2 ATM and POS processing terminology .....                       | 5    |
| 1.2 Transaction categories .....                                     | 9    |
| 1.3 Payments landscape - business overview .....                     | 10   |
| 1.4 Banking industry payment challenges .....                        | 10   |
| 1.5 Finance industry requirements .....                              | 11   |
| 1.5.1 The requirement for reliability .....                          | 11   |
| 1.5.2 System availability .....                                      | 12   |
| 1.6 Functional components of a retail payments system .....          | 13   |
| 1.7 Reference architecture for enterprise payments .....             | 14   |
| 1.7.1 Transaction services .....                                     | 15   |
| 1.7.2 Enterprise payments operations .....                           | 15   |
| 1.7.3 Customer services .....  | 16   |
| 1.7.4 Fraud management .....   | 16   |
| 1.8 Bridging payments and risk management for business insight ..... | 16   |
| <b>Chapter 2. FIS reference architecture overview</b> .....          | 19   |
| 2.1 FIS credentials: an evolution in payments .....                  | 20   |
| 2.2 FIS value proposition .....                                      | 20   |
| 2.3 FIS Enterprise Payments Framework .....                          | 21   |
| 2.4 FIS Enterprise Payment system products .....                     | 22   |
| 2.4.1 FIS Connex on IBM .....  | 22   |
| 2.4.2 FIS Fraud Navigator .....                                      | 23   |
| 2.4.3 FIS DataNavigator .....  | 27   |
| 2.4.4 FIS EnterpriseView .....                                       | 30   |
| 2.5 Overview of FIS Enterprise Payments system .....                 | 33   |
| 2.5.1 Logical view .....   | 33   |

|                   |   |           |
|-------------------|---|-----------|
| 2.6               | Integration, applications, data, tools                          | 36        |
| 2.6.1             | Front end   | 37        |
| 2.6.2             | Back end  | 40        |
| 2.6.3             | MQ between DataNavigator and Connex on IBM                      | 40        |
| 2.6.4             | FIS Fraud Navigator integrating with Connex on IBM              | 41        |
| <b>Chapter 3.</b> | <b>Benefits of a mainframe solution for payments</b>            | <b>43</b> |
| 3.1               | FIS and IBM partnership   | 44        |
| 3.2               | An IBM tradition of System z value                              | 44        |
| 3.2.1             | System z architecture   | 45        |
| 3.2.2             | Who uses mainframe computers                                    | 45        |
| 3.2.3             | System z in the finance industry                                | 46        |
| 3.3               | Continuous availability   | 49        |
| 3.3.1             | High availability   | 49        |
| 3.3.2             | Continuous operations   | 49        |
| 3.3.3             | System z features   | 49        |
| 3.3.4             | Parallel Sysplex - a high availability clustering configuration | 50        |
| 3.3.5             | Linux for System z  | 52        |
| <b>Part 2.</b>    | <b>Environment and process</b>                                  | <b>53</b> |
| <b>Chapter 4.</b> | <b>Design decisions</b>   | <b>55</b> |
| 4.1               | Integration planning  | 56        |
| 4.2               | Customer options and needs                                      | 56        |
| 4.3               | Availability planning   | 57        |
| 4.3.1             | Availability considerations                                     | 58        |
| 4.3.2             | FIS Enterprise Payments system considerations                   | 59        |
| 4.3.3             | Avoiding single points of failure                               | 64        |
| 4.4               | FIS configurations  | 66        |
| 4.4.1             | Single FIS LPAR/Single CPC                                      | 66        |
| 4.4.2             | Multiple LPARs/Single CPCs                                      | 67        |
| 4.4.3             | Multiple LPARs/Multiple CPCs                                    | 70        |
| 4.4.4             | Multiple LPARs/multiple CPCs/dual site                          | 72        |
| <b>Chapter 5.</b> | <b>Installation and configuration</b>                           | <b>73</b> |
| 5.1               | FIS Connex on IBM   | 74        |
| 5.1.1             | Skills  | 74        |
| 5.1.2             | Prerequisite software and hardware                              | 74        |
| 5.1.3             | Preinstallation planning for Connex on IBM                      | 74        |
| 5.1.4             | Installation steps  | 77        |
| 5.1.5             | Installation verification                                       | 79        |
| 5.2               | FIS DataNavigator   | 82        |
| 5.2.1             | Skills  | 82        |
| 5.2.2             | Prerequisite software and hardware                              | 83        |

|                   |   |            |
|-------------------|---|------------|
| 5.2.3             | Preinstallation tasks for DataNavigator | 83         |
| 5.2.4             | Installation steps                      | 84         |
| 5.2.5             | Installation verification               | 86         |
| 5.2.6             | FIS DataNavigator user interface        | 86         |
| 5.3               | FIS Fraud Navigator                     | 87         |
| 5.3.1             | Skills                                  | 88         |
| 5.3.2             | Prerequisite software and hardware      | 88         |
| 5.3.3             | Installation steps                      | 88         |
| 5.4               | FIS EnterpriseView                      | 92         |
| 5.4.1             | Installation steps                      | 93         |
| 5.4.2             | Installation verification               | 94         |
| 5.5               | Overall solution verification           | 95         |
| 5.5.1             | DataNavigator Delphi interface          | 95         |
| 5.5.2             | DataNavigator Web interface             | 99         |
| 5.6               | Onsite customization of all products    | 103        |
| 5.6.1             | Connex on IBM                           | 103        |
| 5.6.2             | DataNavigator                           | 106        |
| 5.6.3             | FIS Fraud Navigator                     | 107        |
| 5.6.4             | EnterpriseView                          | 109        |
| 5.7               | Maintenance                             | 109        |
| 5.7.1             | Connex on IBM                           | 109        |
| 5.7.2             | DataNavigator                           | 110        |
| 5.7.3             | FIS Fraud Navigator                     | 111        |
| 5.7.4             | EnterpriseView                          | 111        |
| <b>Chapter 6.</b> | <b>Monitoring</b>                       | <b>113</b> |
| 6.1               | FIS monitoring tools                    | 114        |
| 6.1.1             | System Health Monitor in Connex on IBM  | 114        |
| 6.1.2             | System Health Monitor in DataNavigator  | 116        |
| 6.1.3             | Enterprise Control                      | 117        |
| 6.1.4             | EnterpriseView                          | 117        |
| 6.2               | Resource Measurement Facility           | 118        |
| <b>Chapter 7.</b> | <b>Optimization and best practices</b>  | <b>121</b> |
| 7.1               | Throughput                              | 122        |
| 7.2               | zAAP and FIS Fraud Navigator            | 122        |
| 7.3               | Connex on IBM best practices            | 123        |
| 7.4               | Tuning tips and techniques              | 124        |
| 7.4.1             | DASD                                    | 124        |
| 7.4.2             | DB2                                     | 125        |
| 7.4.3             | VSAM                                    | 125        |
| 7.4.4             | Workload Manager                        | 126        |

|   |     |
|---|-----|
| <b>Chapter 8. Problem determination on z/OS</b>               | 129 |
| 8.1 Performance troubleshooting tips                          | 130 |
| 8.2 FIS component troubleshooting                             | 130 |
| 8.2.1 Connex on IBM: processes to be monitored                | 130 |
| 8.2.2 DataNavigator   | 132 |
| 8.2.3 EnterpriseView  | 133 |
| 8.2.4 FIS Fraud Navigator                                     | 133 |
| 8.2.5 Causes for application delays                           | 134 |
| 8.3 FIS debugging features                                    | 134 |
| 8.3.1 Traces  | 134 |
| 8.3.2 Dumps   | 134 |
| <b>Part 3. Project details</b>                                | 137 |
| <b>Chapter 9. High availability tests</b>                     | 139 |
| 9.1 Introduction  | 140 |
| 9.2 Test environment topology                                 | 140 |
| 9.2.1 Test configuration                                      | 141 |
| 9.2.2 Environment description                                 | 141 |
| 9.2.3 Reference architecture                                  | 143 |
| 9.3 Unplanned outages   | 144 |
| 9.3.1 Scenario 1: losing a non-critical task in Connex on IBM | 145 |
| 9.3.2 Scenario 2: Canceling critical tasks in Connex on IBM   | 146 |
| 9.3.3 Scenario 3: failure of a Connex on IBM image            | 147 |
| 9.3.4 Scenario 4: z/OS system failover                        | 149 |
| 9.4 Planned outages   | 152 |
| 9.4.1 Scenario 5: applying IBM PTFs                           | 152 |
| 9.4.2 Scenario 6: upgrading FIS application system            | 153 |
| 9.5 Test summary  | 155 |
| 9.6 High availability configurations                          | 156 |
| 9.6.1 Standalone z/OS system                                  | 157 |
| 9.6.2 Parallel Sysplex  | 158 |
| 9.6.3 Parallel Sysplex+GDPS/PPRC/HyperSwap                    | 158 |
| 9.6.4 Parallel Sysplex+GDPS/XRC                               | 160 |
| <b>Part 4. Appendixes</b>                                     | 163 |
| <b>Appendix A. Capacity planning tips</b>                     | 165 |
| Introduction  | 166 |
| Balancing resources for capacity planning                     | 166 |
| Managing mixed workloads                                      | 167 |
| Distributed versus central system                             | 168 |
| Managing the system resources                                 | 169 |
| CPU management  | 170 |



|   |     |
|---|-----|
| Disk management .....                                 | 170 |
| Storage management .....                              | 170 |
| System z architecture .....                           | 171 |
| Capacity planning methodology .....                   | 173 |
| Measurements .....                                    | 174 |
| <b>Appendix B. Business continuity concepts</b> ..... | 177 |
| Overview .....  | 178 |
| Business continuity solution methodology .....        | 181 |
| <b>Related publications</b> .....                     | 201 |
| IBM Redbooks .....                                    | 201 |
| Other publications .....                              | 201 |
| Online resources .....                                | 202 |
| How to get Redbooks .....                             | 202 |
| Help from IBM .....                                   | 202 |
| <b>Index</b> .....                                    | 203 |



# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:  
*IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.


## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

# Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

|            |   |                 |
|------------|---|-----------------|
| DB2®       | Geographically Dispersed  | RMF™            |
| DFSMS™     | Parallel Sysplex™   | System p®       |
| DFSMSdftp™ | HyperSwap™  | System Storage™ |
| DRDA®      | IBM®  | System z®       |
| DS6000™    | Maestro™  | Tivoli®         |
| DS8000™    | MVS™  | VSE/ESA™        |
| ESCON®     | NetView®  | VTAM®           |
| eServer™   | Parallel Sysplex®   | WebSphere®      |
| FICON®     | Person to Person™   | z/OS®           |
| FlashCopy® | RACF®   | z/VM®           |
| GDPS®      | Redbooks®   | zSeries®        |
|            | Redbooks (logo)  ® |                 |

The following terms are trademarks of other companies:

Java, JDBC, RSM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Convergence, Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

# Preface

The number of electronic transactions conducted by financial institutions around the world every day is growing astronomically. Today payments can account for a significant portion of a bank's total cost. The ability to move money quickly and safely and to manage the associated data management flows are necessary components of banking. Banks must also gain critical insights from their data to stay steps ahead of Internet-based theft of bank and customer information. These thefts can dangerously undermine the efficiencies of more cost-effective virtual delivery channels. On the positive side, this data can provide key insights into the way their customers behave and the products and services that they are likely to buy.

The FIS approach to enterprise payments puts the consumer account at the core, surrounded by rich functionality that provides access via multiple channels, extensive options for managing customer service, as well as a range of integrated fraud management tools. A complete payments solution for issuing and acquiring, clearing and settlement, FIS Enterprise Payments also contributes unique insight into the customer relationship.

This IBM® Redbooks® publication describes the FIS Enterprise Payments offerings and how to implement them. The book is written for decision makers and financial solutions architects, and assumes a basic knowledge of payments systems.

## The team that wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

**Mike Ebberts** is a Consulting IT Specialist and a Project Leader in the ITSO in Poughkeepsie, NY. He has worked for IBM since 1974 on mainframe systems.

**Alex Louwe Kooijmans** is a project leader with the ITSO in Poughkeepsie, NY, and specializes in WebSphere®, Java™, and service-oriented architecture (SOA) on IBM System z® with a focus on integration, security, high availability, and application development. Previously, he worked as a Client IT Architect in the financial services sector with IBM in the Netherlands, advising financial services companies on such IT issues as software and hardware strategy and on demand. Alex has also worked at the Technical Marketing Competence Center

for IBM System z and Linux® in Boeblingen, Germany, providing support to customers starting with Java and WebSphere on System z.

**David Brooks** has 20 years of experience in payment processing in North America and Europe, including implementing an FIS solution at a large financial institution using IBM System z. He has eight years of IBM experience in financial services, serving as Director of Product Management at FIS for Payment Delivery production suite.

**Tuan Bui** is an IT Senior Developer in Canada. He has over 25 years of experience in the IT industry, including with electronic payment systems during the past 16 years. He works for Fidelity National Information Services (FIS). His areas of expertise include the System z, the development cycle of project works, presentations, and hands-on training.

**Philippe Fernandes** is a System z Field Technical Sales Support Specialist in IBM France. His areas of expertise are WebSphere, Linux, and high-availability architectures in the banking industry.

**Chen Lei** is a senior IT Specialist in China. He is responsible for z/OS® and SYSPLEX maintenance in IBM China. He is also involved in building the Industrial and Commercial Bank of China (ICBC) Disaster Recovery project and implementing the Bank of China (BOC) GDPS/PPRC/HyperSwap™ High availability project. He has over 13 years of experience in mainframes, including as a Senior System Administrator and Database Administrator. He has worked at IBM for six years. His areas of expertise include application development, CICS/PLEX, and DB2® Datasharing performance monitoring and tuning.

**Gustavo da Costa Leonato** is a certified Database Administrator/Data Architect for IBM in São Paulo, Brazil, supporting several large installations of IBM internal and commercial accounts. He worked as a System Analyst on System z in the financial sector before joining IBM. During that time he has designed, implemented, and administered several financial automation systems in Brazil. He has a bachelor's degree in computer science from Universidade Paulista.

**Robert Simon** is a Team Leader in Systems Assurance in the IBM Systems & Technology Group.

Thanks to the following IBMers for their contributions to this project:

Rich Conway  
International Technical Support Organization, Poughkeepsie Center

Michael Hilman  
Jenny Li, for her graphics and planning assistance  
Dave Robertson

Thanks to the following FISers for their contributions to this project:

Chuck Bram  
Tim Cincotta  
Dave Paull  
Andy Stockhausen

## Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- Send your comments in an e-mail to:

[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)

- Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400







# Part 1

## Overview of products and project

This part describes the function of an enterprise payments system. It also describes the features and benefits of the EFP payments system and the benefits of running it on a mainframe.

Archived

# Enterprise payments

This chapter provides a general overview of an enterprise payments system, including its requirements.

The number of electronic transactions conducted by financial institutions around the world every day is growing astronomically. As the global banking industry seeks to consolidate and modernize its operations to deliver better and faster services to customers, these firms are under intense pressure to exploit technology to address business challenges and opportunity.

Several factors are driving the growth of electronic payments. The transition of paper checks in the United States to a card-based payment method is a major factor. The increased number of payment instruments is also driving the growth of electronic payments, such as pre-paid instruments (gift cards) and innovative eCommerce payment mechanisms, along with the growth in electronic commerce. In the United Kingdom, the growth of the Faster Payments initiative will increase the growth of electronic payments in that region, which also positions for growth in wholesale and Person to Person™ transactions.

Optimizing the payments business is increasingly fundamental to a bank's commercial health. Today payments can account for a significant portion of a bank's total cost. The ability to move money quickly and safely and to manage the associated data management flows are necessary components of banking. The combined forces of increased regulatory pressure and the commoditization of the payments business are forcing banks to take a close look at how they provision it.

Adding further complexity, banks must also gain critical insights from their data to stay steps ahead of Internet-based theft of bank and customer information—threats that can dangerously undermine the efficiencies of more cost-effective virtual delivery channels. At the same, this data can provide key insights into the way their customers behave and the products and services that they are likely to buy.

Financial institutions realize that they must strengthen their relationship with the customer in order to have a long-term success and profitability. By focusing on this relationship with the right customers, they can sustain the growth of their business through strategic decisions, opportunities for operational improvement, and investments and marketing initiative.

# 1.1 Processing terminology

Before we describe the products, we present some helpful terms.

## 1.1.1 Issuer and acquirer business models

There are multiple business models operating within the retail payments landscape. At a high level, the major delineation exists between the issuer business model and the acquirer business model.

The issuer business model is related to the lending family, with its focus on credit worthiness and managing risk. Within retail payments, this is the role of *issuing* cards to consumers or corporations. Revenue for issuers is generated by fees charged to the cardholder and interest revenue generated from outstanding balances (loans).

The acquirer business model generates its revenue from the acceptance element of the value chain. These are the companies that establish commercial relationships with retail merchants, enabling the merchants to accept multiple forms of payment from their customers. In addition, acquirers (or their agents) deploy Automated Banking Machines (ABM)/Automated Teller Machines (ATM), which provide access to cash withdrawals.

Financial Institutions often participate in both the issuer business model and the acquirer business model. With the industry's evolution, multiple variations of these business models have developed, based upon the maturity of the market and local regulation.

## 1.1.2 ATM and POS processing terminology

Within the acquiring business model, ATM (or ABM, depending upon your region's terminology) and point-of-sale (POS) use terminology that is outlined in this section.

In many ways, automated teller machine, automated banking machine, point-of-sale processing, and POS card processing are similar and follow similar paths though the network and associated systems.

The retail payment transaction itself has its roots in a two-step process to complete a transaction. The first step, known as the authorization, is done at the point-of-sale, or ATM, which either enables the transaction to proceed (approved) or does not approve the transaction (declined). This collection of

systems and operations is referred to in the industry as front end (FE) transaction processing.

The merchant or ATM in turn aggregated these electronic receipts and then *deposited* them for credit to their bank account. Similar to checks, historically, these drafts were typically posted to their respective accounts overnight. This trailing or clearing activity is referred to in the industry as back-end (BE) transaction processing.

Although the technology had made significant advancements since the time of the initial credit card transactions, much of the terminology continues today. The first portion of the transaction continues to be known as the front end authorization. The second portion is referred to as the back end clearing and settlement portion of the transaction.

Both ATM and POS transactions must traverse three elements of front end authorization through its complete processing: acquisition, authentication, and issuer authorization. (Historically, back-end settlement and reconciliation activities are not a part of the actual real-time transaction.)

In this book we reference the terms *host* and *network* frequently. The term host includes the IBM System z, yet is not limited to it. Host includes the transaction processing infrastructure of either an issuer or an acquirer for the purposes of payment transaction processing. The term network (also known as *scheme*) is not intended to imply the traditional telecommunications network. References to network in this document include the telecommunication network and the applications that switch the transaction from the acquirer to the issuer. Examples of networks include VISA and MasterCard.

**Types of ATM and POS transactions**

Within the front-end authorization itself, the transaction could be categorized in a few different ways. There are several different types of transactions, depending upon where the transaction originated and the transaction’s destination for authorization (issuer).

Four categories of transaction can take place, as described in Table 1-1.

Table 1-1 Transaction switching schemes

| Category | Description  |
|----------|--|
| On Us    | The transaction arrives via a bank-owned ATM or POS device and is never routed outside the bank.<br>Example: A customer of the “SampleA” Credit Union uses an ATM card to withdraw money from the ATM located in the local credit union network. |

|                              |   |
|------------------------------|---|
| Network On Us                | <p>The transaction originates from a sharing network such as STAR, Cirrus, or PLUS, in which both the bank and the ATM or POS-owning institution are members.</p> <p>Example: A local customer travels somewhere away from home and is a member of “SampleA” Credit Union. The customer needs to use an ATM or POS at the “SampleB” Credit Union. Both “SampleA” and “SampleB” are members of the STAR network.</p>     |
| Reciprocal Transaction       | <p>The card holder initiates a transaction at an ATM or POS owned by a bank that is a member of a different regional network.</p> <p>Example: A New York resident attempts to withdraw money from an ATM or POS in Milwaukee, Wisconsin. An agreement between the network in Wisconsin and the network in New York allows the transaction to be switched from one regional network to another.</p>                      |
| National Bridge Transactions | <p>The card holder uses an ATM or POS at a bank not their own, and the two banks belong to different regional networks that do not have any agreement. Both banks <i>must</i> belong to the same national network. The transaction is handed from the ATM or POS regional network to the national network, and finally to the authorizing bank’s regional network. In this case, there are three switches involved.</p> |

Transactions are routed through the originating ATM or POS—either the bank's own network or through some combination of regional or national networks. Figure 1-1 demonstrates, at high level, the possible paths that a transaction might follow to completion.

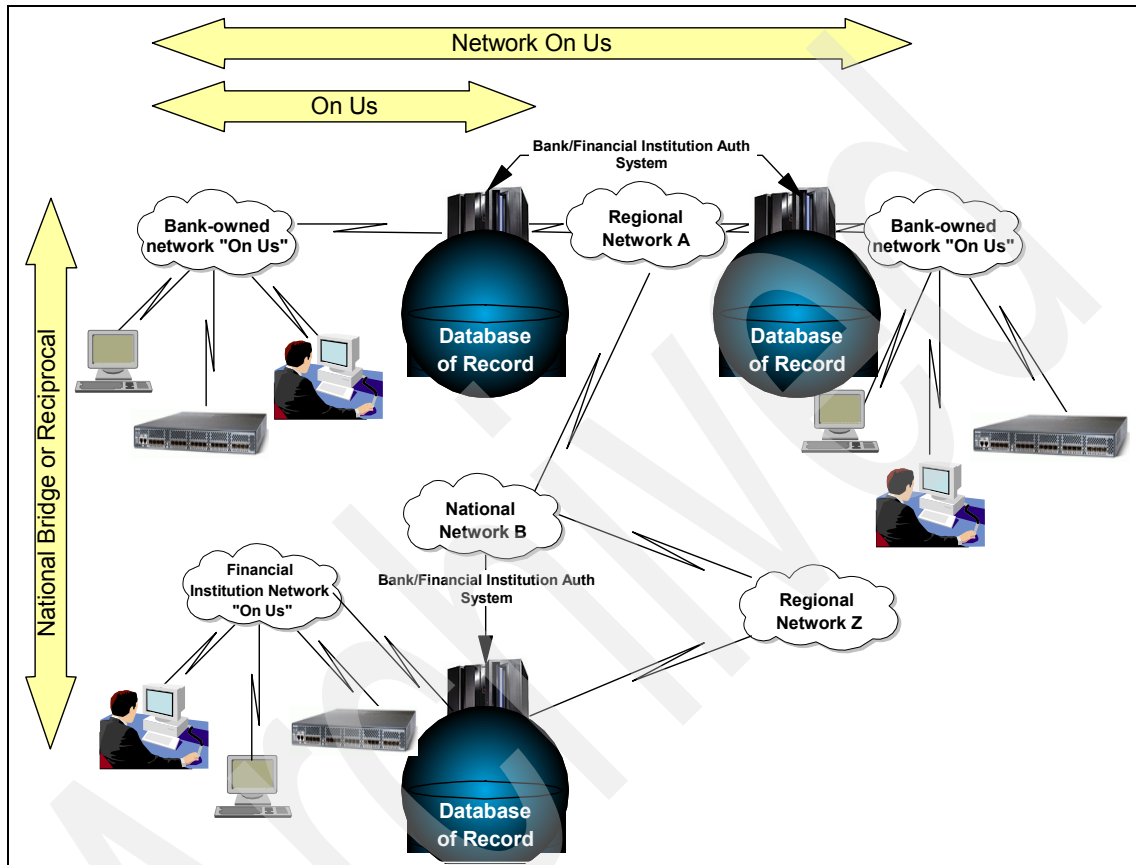


Figure 1-1 Transaction flows

Both the regional and national networks are switching systems that resemble the systems used within the financial institution. The switching systems drive transactions from initiation to destination and back. We use the word *switched* to describe the hand-off of a transaction from host to network to host.

- On Us: An ATM or POS transaction is accepted by the host of the acquiring institution, and processed locally *On Us* if it is acquired by the same institution that issued the card.



- ▶ **Network On Us:** An ATM or POS transaction is accepted by the host of the acquiring institution, and switched to one of the directly connected regional networks, *Network On Us*, to the institution that issued the card.
- ▶ **National Bridge or Reciprocal:** The regional networks switch the transaction to a national network for hand-off to another regional network *National Bridge* or *Reciprocal*.
- ▶ Responses are switched from the card issuer back to the owning host (acquirer) of the ATM or POS using the same path that routed the transaction from the acquirer to the issuer (albeit in the returned direction).

**Note:** An international network is easy to add to this scenario, and would be similar to the National Bridge portion of the diagram.

There are other variations on this theme. Transactions are switched from host to host through network switches until the original acquired transaction request is authorized by the issuer or it times out.

## 1.2 Transaction categories

The previous section discussed how ATM or POS processing can occur across multiple paths, either directly (via a bank's own ATM, typically called *On Us*), or through one of many shared networks where the card holder is using a terminal that is owned and operated by a entity.

Common shared networks do more than provide credit card purchases and ATM withdrawals. They may also provide other services to financial institutions, such as:

- ▶ Debit and ATM network
- ▶ Deposits, deposit sharing program
- ▶ Gateway connections
- ▶ ATM driving
- ▶ PIN-secured debit, signature debit, and stored value card processing
- ▶ Card authorization, activation, and production
- ▶ Merchant acquirer and agent bank programs
- ▶ Bill payment services
- ▶ Risk management and fraud prevention services

Shared ATM networks provide added availability to financial institution customers, although they can also introduce a certain degree of risk. Many networks provide stand-in authorization, which enables a network to authorize

transactions when a card issuer or processor cannot do so. However, this may lead to an increased risk of fraud.

## 1.3 Payments landscape - business overview

Financial institutions face difficult challenges and opportunities when they require a complete view of each customer relationship and how to deliver, more effectively, the correct products and services to each customer. Some of the challenges are:

- ▶ There is a high volume of transaction generated through various channels, each channel with its own infrastructure (credit cards, deposit accounts, commercial, eCommerce).
- ▶ Fraud is becoming more pervasive and complex with each channel and between multiple channels.
- ▶ Consolidation, both within the industry and within individual organizations, brings with it the need to rationalize infrastructure. Often large financial institutions have duplicate infrastructure within the single company. Current solutions provision the rationalization of duplicate infrastructure onto a single platform (for example, wholesale and retail payments).
- ▶ Increased regulatory intervention is driving up compliance expenses.

These challenges must be dealt with if financial institutions want to seize the opportunities as presented by the global growth in payments in order to succeed.

In the search for any strategic advantage, the business need to focus on a complete view of the customer relationship turns rapidly to a need to consolidate the payment delivery and combine it more effectively with the fraud and risk management.

Conventional payment systems provide solutions to address a certain payment type or specific delivery channel. They are normally linked to one or more fraud management systems.

## 1.4 Banking industry payment challenges

There are several challenges facing financial institutions today, namely:

- ▶ Point solutions forcing client-driven integration rather than pre-integrated solution suites

- ▶ Consolidation of information and infrastructure to accommodate multiple banking channels (fraud, data navigator/data mining)
- ▶ Business development along with associated transaction growth ISO 20022:
  - Upstream (wholesale, SWIFT, FedWire, and EBA Step 2)
  - Downstream (person to person)

Expansion of the traditional business-to-consumer market to incorporate the business-to-business marketplace (wholesale banking), along with going further downstream, entering into the consumer-to-consumer market (person-to person-marketplace).
- ▶ Exploiting technology advances (System z and others)
- ▶ Competitive threats

## 1.5 Finance industry requirements

The payments arena is competitive. Business can be lost over just one outage or bad customer experience. Therefore, companies in the finance industry have a number of requirements for their payments system, including:

- ▶ Functionality
- ▶ Reliability
- ▶ High availability
- ▶ Manageability
- ▶ Security
- ▶ Scalability
- ▶ Dynamic workload balancing

### 1.5.1 The requirement for reliability

Few issues impact the reputation of a financial institution, retailer, or customer more than ATM/credit card processing reliability. Trying to explain that a card being declined is the fault of the bank is a difficult task, and failures at the point-of-sale embarrass customers and damage the reputation of the bank itself. As the transition to a cashless society continues, customers will quickly revert to using cash, and—more damagingly—take their business elsewhere, over a single embarrassing event.

Moreover, the brand name of the ATM has to be considered. Maestro™, Star, PLUS, and others value their name brand and specify performance objectives to ensure that they are not a target of poor performance complaints by customers.

## 1.5.2 System availability

Typically, the issuing financial institution or bank has about 10 seconds for domestic transactions to respond before the network begins to stand in. In some instances, such as an overseas banks and gateway transactions, more time is allocated. With improvements in technology and telecommunication, the acceptable transaction time continues to decrease over time.

In the event that a financial institution is unable to respond to a transaction within a specified time period, the ATM or POS network may have the right to stand in for the bank and, following specified rules, then approve or decline the transaction. This could potentially expose the bank to overdraft of the customer's account, and the possibility of approving a fraudulent transaction. The specific rules and charges for stand in are usually the result of a negotiated agreement between the financial institution and the shared network.

A financial institution or card issuer with unsatisfactory or poor reliability or poor response-time performance may have to accept the risk of a shared network standing in for the bank, an activity that is neither free nor without risk.

### **System reliability and data integrity**

A financial transaction may pass through several switches before it is completed on the round-trip journey from acquisition to authorization, and back to its point-of-initiation ATM or POS station. When everything goes according to plan, the transaction completes and is prepared for final settlement, which typically takes place in real-time, or in some form of batch processing scenario. When failures and errors occur, the host issuer authorizing system must keep track of all transactions.

There are several methods available for meeting the response time, availability, and data integrity requirements imposed on financial processing systems. These techniques go beyond simple fault-tolerance. The system must provide continuous availability. Fault tolerance is not a requirement for high availability. It is simply a tool that contributes to the high availability capability.

## 1.6 Functional components of a retail payments system

Retail payments have historically been provisioned through a collection of components, integrated together with operations and sales to deliver the service to the cardholder and merchant. These systems have tended to fulfil the following components of capability:

- ▶ Client solicitation and acquisition/adjudication
- ▶ Card production/personalization (issuers)
- ▶ POS terminal deployment (acquirers)
- ▶ Transaction switching
- ▶ Risk management
- ▶ Clearing and settlement
- ▶ Statement production
- ▶ Customer help desk
- ▶ Dispute management/chargeback processing

One role of the payments management team is to acquire or provision solutions and operational procedures in order to monitor and manage each of the components. Integration between systems is often complex.

Management's challenge is to maintain and grow the business by way of adding new feature functions in order to enable new products and services. For example, adding a new payment card, such as a gift card or person-to-person transaction, could require:

- ▶ Changes to the acquisition system for the new product
- ▶ Changes to the POS terminals to accept the card and identify it on the customer receipt
- ▶ Enabling the transaction system to process the new transaction type
- ▶ Adjusting the fraud management system and any rules associated with it
- ▶ Enabling the clearing and settlement system to financially account for the transaction and its associated billing
- ▶ Including the new transaction type on applicable statements and reports
- ▶ Ensuring that both the customer service systems and the dispute management system are able to accommodate the additional transaction type

This is one simple, seemingly minor, additional transaction type that requires a significant amount of effort to enable the business unit to accept this additional payment type.

In order to simplify the management of the payments business, FIS has created the reference architecture for enterprise payments.

## 1.7 Reference architecture for enterprise payments

The reference architecture for enterprise payments is a fresh, innovative way to think about the payment processing infrastructure. Rather than the traditional component pieces being integrated by the bank or financial institution, the FIS Reference Architecture incorporates the multiple components of payment processing into a single architecture.

Enterprise payments provides solutions to increase the value of every account and transaction.

With emerging payment types and international regulations causing geographic boundaries to effectively disappear, demand for payments solutions that operate across local, national, and regional boundaries is on the rise.

Banks are working to gain strategic insight and competitive advantage by unifying payments within the enterprise, with the goal of integration across all channels. As part of this initiative, organizations are bridging the gap between payments and fraud management.

The FIS approach to enterprise payments puts the client account at the core, surrounded by rich functionality that provides access via multiple channels, extensive options for managing customer service, as well as a range of integrated fraud management tools. A complete payments solution for issuing and acquiring, clearing and settlement, FIS Enterprise Payments also contributes unique insight into the customer relationship.

### **FIS Enterprise Payments framework**

FIS Enterprise Payments is an innovative, robust, and flexible solution, balancing the need for rationalization of platforms and interoperable components with the requirement to accommodate business strategy across different customer segments and in different geographies. Financial institutions have the ability to obtain a more complete and accurate view of the transactional data that identifies opportunities for cross-sell and up-sell with profitable customers.

Similarly, the solution complements existing fraud detection and reduction offerings to identify suspect transactions and reduce fraud losses. Executives also benefit from a consistent view of management information across the entire business, incorporating multiple payment types. That information can be organized to provide insights into what is happening at the local, national,

regional, and international level, giving the bank and financial institution the power to think globally and act locally.

The solution is built using the re-usable components in a three-layered services-oriented architecture such as transaction services, enterprise payment operations, and customer services.

### **1.7.1 Transaction services**

Transaction services is a transaction processing engine that provides payment delivery for Electronic Funds Transfer (EFT) transactions, as follows:

- ▶ Transaction acquiring
- ▶ Transaction switching
- ▶ Authorization services
- ▶ Fraud monitoring

### **1.7.2 Enterprise payments operations**

As the volume of transactions and the number of channels for payment transactions increase, the complexity and cost of operating the back office increases. Enterprise payment operations is a component that provides additional functions and services to enable back office optimization and provide business insight, as follows:

- ▶ Transaction research
- ▶ Exception item and chargeback processing
- ▶ Device and cash management
- ▶ Device problem management
- ▶ Settlement services
- ▶ Fee billing
- ▶ Card management services
- ▶ Liquidity management
- ▶ Management and operational reporting
- ▶ Management dashboard
- ▶ EMV script management

An integral part of an effective payments infrastructure is a productive back office organization that is efficient at its handling of business operations and support of customer requests.

### 1.7.3 Customer services

The customer services layer provides a range of card management (for issuers and acquires) and enhanced customer service options to build more profitable customer relationships, as follows:

- ▶ Cardholder services
  - Hot-carding
  - Voice authorization
  - Customer inquiries
  - Customer change of detail
- ▶ Account life-cycle management services
  - Customer acquisition
  - Marketing cross-sell/up-sell
  - Account retention and rehabilitation

### 1.7.4 Fraud management

For financial institutions, merchants, processors, and networks, controlling preventable payment card fraud remains one of the single biggest challenges—and opportunities—in the industry. Until now, payment businesses have attacked fraud through a combination of tools and technologies that manage the key phases of fraud management: prevention, detection, and reduction. Yet there remains a significant gap in the industry for solutions that take a more holistic approach to the detection phase of the fraud management life cycle.

## 1.8 Bridging payments and risk management for business insight

Payments are key to a growing cashless society, and payments are a key element of banking and financial services. Inherent in those transactions are several levels of risk, but also opportunity. The exposure risk associated with managing a settlement position with a central bank, risk of fraud on the individual transaction, merchant fraud, and abuse of cardholder accounts are all facets of risk and fraud with which banks must contend.

With FIS, financial services companies have the unique ability to bring payments and all three aspects of risk management together on one platform and then perform business-focused analytics on the transaction data to better understand the dynamics of the payment business profit and loss. In so doing, banks can be



better positioned to drive business insight to the next level—capitalizing on opportunities for customer cross-sell and up-sell.

Banks are seeking flexibility and simplicity. Many are consolidating their systems to streamline operations and achieve cost efficiencies in the process. System z provides flexibility of implementation options. Unique workload management capabilities of the System z can help simplify systems management.

Additionally, the deeper levels of security, availability, and information integration that the System z provides can help enable banking firms to achieve better enterprise risk management goals. For more information about this offering, see:

<http://www-03.ibm.com/industries/financialservices/doc/content/partner/931731103.html>

Archived

# FIS reference architecture overview

This chapter describes the components of the Fidelity National Information Systems Enterprise Payments system.

Enterprise payments can provide solutions to increase the value of every account and transaction.

With emerging payment types and international regulations causing geographic boundaries to effectively disappear, demand for payments technology that operates across local, national, and regional boundaries is on the rise.

Banks are working to gain strategic insight and competitive advantage by unifying payments within the enterprise, with the goal of integration across all channels and payment types. Organizations have to bridge the gap between payments and fraud management.

The FIS approach to enterprise payments puts the consumer account at the core, surrounded by rich functionality that provides access via multiple channels, extensive options for managing customer service, as well as a range of integrated fraud management tools. A complete payments solution for issuing and acquiring, clearing and settlement, FIS Enterprise Payments also contributes unique insight into the customer relationship.

## 2.1 FIS credentials: an evolution in payments

The corporate history of payments goes back a long way. It started with the check printing business. After operating in the check printing business for many years, it was a logical extension to evolve the business model with the development of an ATM driving switch initially, followed by driving networks and point-of-sale solutions. The software business evolved with the payments industry, which has lead to the creation of the payments reference architecture discussed in this IBM Redbooks publication.

Some facts about FIS to introduce you to our solution are:

- ▶ The company offers both software and processing solutions.
- ▶ The reference architecture solutions are installed in several of the leading global financial institutions.
- ▶ Software handles more than 30 billion transactions globally every year.
- ▶ There are more than 3,500 EFT processing customers.
- ▶ FIS processes 80% of U.S. debit volume.
- ▶ FIS serves 80,000 businesses worldwide with payment fraud solutions.
- ▶ Unique information intelligence from 3 billion consumer records are contained within DebitBureau.
- ▶ Ninety percent of United States financial institution locations use FIS decisioning solutions.

Transforming enterprise payments and data and decisioning expertise into trusted commerce takes more than technology. It requires business insight. FIS delivers flexible, innovative solutions that enable the world's leading businesses to acquire the right customers, serve them more efficiently, and keep them.

## 2.2 FIS value proposition

The value proposition of package software for payments processing is becoming more apparent to banks, financial institutions, networks, and retailers alike. With the increase in compliance mandates (VISA, MasterCard, PCI, regional networks) increasing and the corresponding costs of maintaining proprietary systems, many payment providers are looking to implement package system solutions. The benefit of package system solutions is justified in that the costs of maintenance and new development are spread over multiple installations.

FIS offers a comprehensive solution suite that offers robust availability that leverages the bank's IT investment in System z infrastructure and support.

By way of reference, many of the world's leading payment providers depend upon FIS for their payment processing needs. In addition to an impressive list of reference clients, FIS also operates its own payments processing software in an outsourced service offering.

FIS is a leading provider of financial payments switching and services to retail institutions in the U.S., flowing over 75% of U.S. retail payments transactions through its switches. Operating worldwide, FIS provides financial payments offerings including transaction switching, clearing, settlement, account opening, and so on, and delivers these offerings either as services in its data centers or through software the it markets to financial, government, and merchant institutions.

- ▶ Lower back-office costs by taking a cross-business line view of payments data.
- ▶ Consolidate payments infrastructure on one cost-effective IBM System z platform.
- ▶ Convergence™ of customer disputes across credit, debit, and prepaid can save money.
- ▶ Enable branches and call centers to benefit from transaction data insight.
- ▶ Move from operational integrity to business insight that drives revenue.

## 2.3 FIS Enterprise Payments Framework

FIS Enterprise Payments is an innovative, robust, and flexible solution, balancing the need for rationalization of platforms and interoperable components with the requirement to accommodate business strategy across different customer segments and in different locations. Financial institutions have the ability to obtain a more complete and accurate view of the transactional data that identifies opportunities for cross-sell and up-sell with profitable customers.

Similarly, the solution complements existing fraud detection and reduction offerings to identify suspect transactions and reduce fraud losses. Executives also benefit from a consistent view of management information across the entire business, including all payment types. That information can be organized to provide insights into what is happening at the local, national, regional, and international level, giving the bank the power to think locally and act globally.

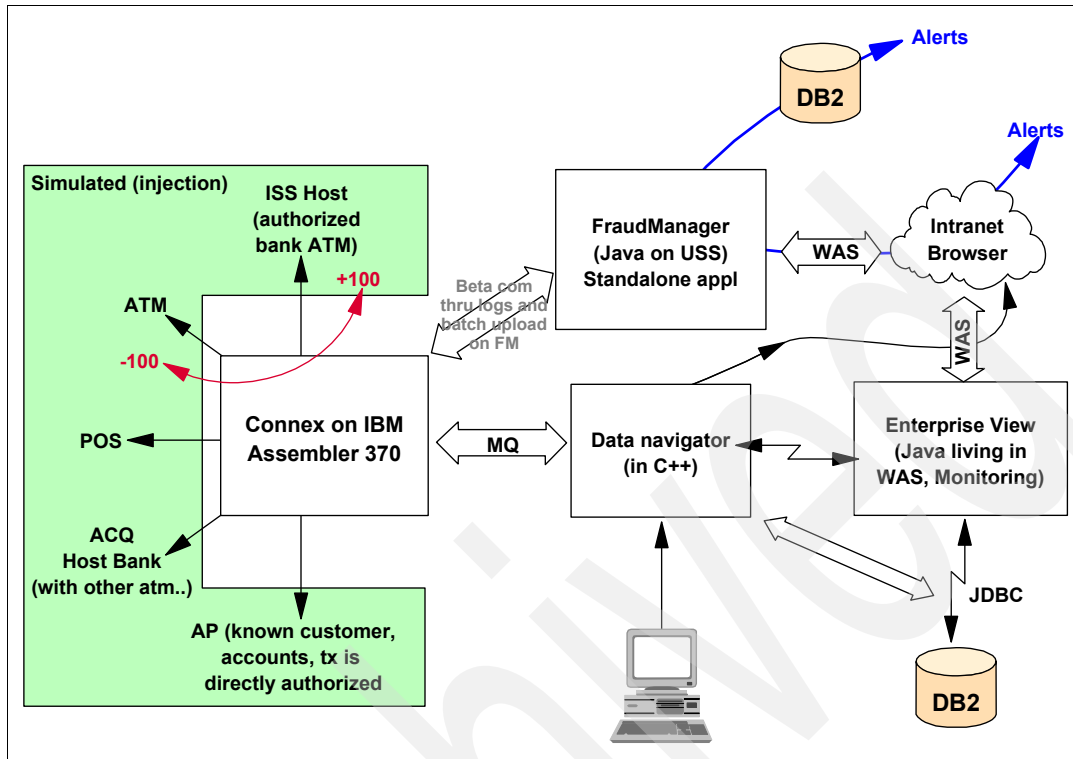


Figure 2-1 Schematic of our testing environment

## 2.4 FIS Enterprise Payment system products

FIS Enterprise Payment System on z/OS comprises the following FIS products:

- ▶ FIS Connex on IBM
- ▶ FIS Fraud Navigator
- ▶ FIS DataNavigator
- ▶ FIS EnterpriseView

### 2.4.1 FIS Connex on IBM

Connex on IBM is FIS's global electronic payment solution. Every year, 80% of U.S. debit transactions flow through it and more than 26 billion transactions are processed by it, totaling \$720 billion in settlements. It supports more than 26 billion transactions every year. Connex on IBM processes the full spectrum of

transaction activities, including EFT, EBT, and POS, while providing 100% uptime.

FIS Connex on IBM software offers a single, integrated hardware and software solution for transaction acquisition, processing, settlement, and reporting. Connex on IBM provides everything that you need to operate a full-service, in-house EFT processing center, including:

- ▶ ATM and POS device driving
- ▶ Transaction switching
- ▶ Authorization
- ▶ Settlement and reconciliation
- ▶ Advanced systems monitoring and management
- ▶ ATM self-service banking
- ▶ Device problem management
- ▶ Data capture and transaction analysis via DataNavigator
- ▶ Network access
- ▶ eCommerce

The Connex on IBM high volume processing solution features continuous availability, 24 x 7 monitoring of vitals (such as network status and the availability of terminals and processors), dynamic system configuration, and linear scalability.

Additionally, Connex on IBM features continuous processing that enables you to distribute transaction workloads across multiple sites or nodes and minimize the risk of system failure. Continuous processing provides non-interrupted service while one node is taken down, usually for scheduled maintenance, saving you the cost of implementing more expensive back-up systems. Connex on IBM Continuous Processing allows you to temporarily move core processes from your primary node to an alternate node located anywhere in the country—without any interruption of ATM and POS uptime. Now you can schedule maintenance at any time of day. With continuous processing you can:

- ▶ Limit outages to one node while others continue processing transactions.
- ▶ Transfer processing from one node to another prior to a scheduled outage.
- ▶ Balance processing loads between nodes during busy processing periods.

## 2.4.2 FIS Fraud Navigator

For financial institutions, merchants, processors, and networks, controlling preventable payment card fraud remains one of the single biggest challenges—and opportunities—in the industry. Until now, payment businesses have attacked fraud through a combination of tools and technologies that manage the key phases of fraud management: prevention, detection, and reduction. Yet there remains a significant gap in the industry for solutions that

take a more holistic approach to the detection phase of the fraud management life cycle.

To combat the proliferation of payment card fraud, FIS Fraud Navigator offers a robust, integrated fraud management solution that enables issuers and acquirers to detect preventable fraud as early as possible to avoid potential loss. FIS Fraud Navigator analyzes payment transactions that occur at the issuer, acquirer, or transaction switch in real time as part of the inline authorization path, or post-authorization for subsequent follow-up. As an integrated solution, FIS Fraud Navigator combines flexible rules creation and management, extensive research and decisioning features, system-generated alerts, and a wide range of off-the-shelf reports.

### **FIS Fraud Navigator highlights**

The highlights of FIS Fraud Navigator are:

- ▶ It enables quick response to fraud, including card disablement, fraud alert generation, and cardholder notification.
- ▶ It features a wide range of packaged rules to enable rapid, out-of-the-box detection.
- ▶ Rules are easy to add, modify, or delete—with no service outage.
- ▶ Rule changes can be audited and reversed—FIS Fraud Navigators can validate rule operation before turning it on in production mode.
- ▶ Extensive research and decisioning capabilities, including alert management, enable fraud analysts to research potential fraud and take timely action.
- ▶ Standard reports, including totals on transactions processed, alerts generated, and activity indicators; fraud analyst productivity based on alerts generated, case management load, false positive analysis, and more.

The Rules Configuration screen enables fraud officers or managers to create, modify, and schedule transaction analysis rules. FIS Fraud Navigators can create and edit rules quickly in the Current Rule List panel, fully define rules in the Editor panel, and exercise a high degree of scheduling control.

### **Flexible rules creation and management**

FIS Fraud Navigator's rules management capabilities are designed for maximum flexibility and advanced rules creation, but also feature extensive packaged rules so that businesses can get their fraud detection solutions up and running quickly. FIS Fraud Navigator rules are triggered based on configurable conditions and thresholds. Each rule has one or more actions associated with it, which are executed when the rule is triggered. Examples of these actions include the ability to disable a card, generate a fraud alert, or notify an FI. Rules are domain based (for multi-institution implementations), with secure distributed access ensured.



Rules can be easily added, modified, or deleted by an FIS Fraud Navigator analyst using an advanced graphical configuration system.

Changes to rules can be audited and are completely reversible. FIS Fraud Navigator also makes rule changes quick and easy, with no service outage. Changes can also be conveniently previewed before implementation by installing them in an audit mode, which enables fraud analysts to validate the operation of a rule before turning it on in a production capacity. Depending on how fraud rules have been configured by the business owner, possible actions can include:

- ▶ Sending a pass or fail response to the switch (for inline)
- ▶ Creating a fraud alert
- ▶ Disabling or suspending a card, BIN, terminal, or merchant

Rules can be defined to check and compare any data values in the transaction or the logical data store (aggregated information/statistics related to previous transaction history). Each rule is made up of conditions, actions, and commands. There can be one or more conditions, one or more actions, and one or more commands for each rule:

- ▶ Condition: This is the check that needs to be made.
- ▶ Action: An example is *create an alert*.
- ▶ Command: An example would be *disable merchant*.

### **Packaged rules for immediate fraud detection**

FIS Fraud Navigator comes with a set of packaged rules that can be implemented as part of the installation to begin rapid fraud detection capabilities. Examples of some of the packaged rules include:

- ▶ Card exceeds three withdrawal attempts within five minutes.
- ▶ Card exceeds \$1,000 in total withdrawal amount within two days.
- ▶ Card performs ATM transactions at more than two unique devices within two days.
- ▶ Card performs more than five transactions within five minutes.
- ▶ ATM withdrawal amount exceeds 50% deviation from card's average withdrawal amount (based on at least three samples).
- ▶ Deposit amount is less than \$1.
- ▶ Card performs more than two balance inquiries within one day.
- ▶ Card performs more than two transactions from high risk MCC (list) within 1 day.
- ▶ ATM exceeds four PIN rejects within 15 minutes.
- ▶ Returns exceed purchases within one day.

- ▶ ATM exceeds two captured cards within one day.
- ▶ POS performs more than four transactions where card not swiped within one day.

These rules can be extended, modified, appended, or deleted to fit your specific fraud detection needs. In addition to packaged rules, FIS Fraud Navigator enables the fast creation, testing, and deployment of an unlimited number of custom rules designed to address the ever-changing patterns of fraudsters.

## **Fraud alerts**

FIS Fraud Navigator generates fraud alerts based on parameters specified in the fraud rules. A fraud alert consists of information about the transaction, the rule that was triggered, actions to be taken based on the triggered rule, and recommendations for resolution.

The fraud alert is put into a work queue as part of the Research & Decisioning component of FIS Fraud Navigator or, alternatively, can be sent to an external system or application for processing.

## **Research and decisioning**

Research and decisioning provides alert management functionality for a fraud analyst to be able to research potentially fraudulent transactions and take appropriate actions. Some of the functions provided by the Research & Decisioning component include the following: A fraud analyst is able to view all information about the alert including transaction details and the list of rules that were executed and the results. They are able to make notes, close the alert identifying resolution, cause, and so on.

## **Reports**

FIS Fraud Navigator will provide two primary reports (both available on demand to authorized users). The first report provides information about the number of transactions processed, the number of alerts generated, and the activity indicators. Data included on this report includes:

- ▶ Total number of transactions analyzed
- ▶ Number of transactions analyzed that generated an alert
- ▶ Total number of fraud alerts triggered
- ▶ Number of fraud alerts triggered and by the rule that triggered them
- ▶ Top five merchants with the highest aggregate fraud alerts and the number of alerts
- ▶ Top five terminals/devices with the highest aggregate fraud alerts and the number of alerts

- ▶ Top five card numbers with the highest aggregate fraud alerts and the number of alerts
- ▶ Top five BINs with the highest aggregate number of fraud alerts and the number of alerts

A second report is provided to manage fraud analysts and their productivity. This report provides data that includes:

- ▶ Number of fraud alerts total
- ▶ Number of fraud alerts in queue (assigned to a Research & Decisioning analyst)
- ▶ Number of fraud alerts assigned to each Research & Decisioning analyst
- ▶ Fraud alerts outstanding (assigned to a Research & Decisioning analyst but not closed)
- ▶ Fraud alerts closed (assigned to a Research & Decisioning analyst and closed)
- ▶ Percentage of closed fraud alerts proven false positive

### **FIS Fraud Navigator alert workstation**

The alert workstation helps fraud officers or managers research suspicious fraud activity based on alerts triggered by a transaction analysis rule. As part of the Research & Decisioning component of FIS Fraud Navigator, fraud alerts can be put into a work queue.

## **2.4.3 FIS DataNavigator**

With the benefits of modern payment processing technology, reliability and efficiency are taken for granted as billions of payment transactions circulate around the globe. To maintain a profitable payments business today, having the correct infrastructure to drive ATM and POS devices and manage switching effectively is the starting point for all financial institutions. However, with banking margins under pressure and costs cut as far back as possible, companies are now asking themselves how to obtain more value from the payment processing platform, as they seek to deepen customer relationships and generate increased revenue from the retail channel.

FIS DataNavigator is a comprehensive solution that collects and connects transactional data. Vast amounts of data are handled in the payments environment every day, providing a rich source of information about customers and their activities. Whether it helps improve customer service levels, track fraud patterns, or understand customer behavior in order to manage cash availability, insight into the data behind each transaction can provide the basis for faster

responsiveness, better decision-making, and improved operations within the payment processing business.

### **New challenges**

As the number of transactions generated by EFT systems grows, databases become more complex and data management costs increase. With more organizations seeking to consolidate payment processing operations across subsidiaries, countries, and continents, effective data management is necessary to control costs.

Meanwhile, in the face of intense competition, customer service levels have to be consistent across all parts of the retail banking operation.

### **Collecting and connecting data**

FIS DataNavigator is the back-office transaction management component of the FIS Enterprise Payments Platform, aggregating data from all the disparate acquisition channels and processing systems. As part of the central payments engine in a hub and spoke architecture, FIS DataNavigator allows banks to obtain a real-time enterprise-wide view of all transactions and consumer interactions flowing through the system. This endless stream of information has to be stored, aggregated, and analyzed. FIS DataNavigator optimizes the transaction research, exception management, ATM management, settlement, and reporting functions in one central location.

### **Simplified transaction research**

Transactions are fed into FIS DataNavigator as they occur and can be accessed immediately by customer service staff using either a PC or Web-based user interface. This means that bank representatives can retrieve all relevant information when a customer phones to query a transaction and handle that inquiry on the spot, leading to high levels of customer satisfaction and improved efficiency in the customer service team.

The system can be customized to extract specific data required by an individual bank to monitor its own business. It can also be configured to provide various levels of access to customer service staff, dependent on need and authorization. Transaction research can be as simple or complex as the institution deems appropriate.

### **Round-the-clock settlement**

As many organizations consider the benefits of a continuous processing environment, and with compliance mandates demanding shorter settlement windows, the ability to view the current settlement position at any time is critical. Built to offer customers a wider choice of configuration options and facilitate the

rapid deployment of new payment technologies, FIS DataNavigator will support any processor's settlement needs, regardless of the front-end switching platform.

FIS DataNavigator monitors settlement totals in real-time as transactions are loaded into the database. It provides summary and drill-down information about net funds flow and suspense account positions, providing detailed management reports and generating funds movement files. Funds may be moved by a variety of methods, including ACH, wire transfer, e-mail notification, manual transfer, or a mixture of these.

If multiple methods are used, the Auto Reconciliation feature is available to balance totals for the settlement period by automatically comparing transactions from any external endpoint against those stored in its data repository. Back office staff can identify exceptions and opt to send them directly to FIS DataNavigator's Exception Management system.

### **Meaningful management reporting**

On demand and scheduled reports generated by FIS DataNavigator are available in issuer and acquirer packages. They can be configured using current and historical data and totals can be aggregated on an hourly, daily, monthly, quarterly, and annual basis. Similarly, data can be broken down to global, regional, national, or branch level. Giving access to a broad range of transaction and interchange data, FIS DataNavigator allows business managers to have the most comprehensive overview of transaction activity available.

### **Faster exception management**

On any given business day, a percentage of transactions will become exception items, which must be handled in accordance with network and bank rules.

Manual processing of exceptions is time consuming and expensive. FIS DataNavigator simplifies the management process for PIN-based and signature transactions. Its exception management system is an intelligent, rules-based engine, supporting the entire life cycle of an exception from retrieval requests to adjustments, charge-backs, re-presentment, rejection, or arbitration.

Visa and MasterCard fraud reporting and miscellaneous fee handling are integrated into the system. To eliminate user errors, all reason codes, next actions, and required documents are presented to the user based on transaction attributes.

FIS DataNavigator reduces losses and improves customer service by handling disputes, adjustments, and fraud reporting more effectively. Revenues are increased by automating the collection of appropriate fees, while staff training and servicing costs are reduced.

FIS DataNavigator can take data feeds from FIS systems, as well as from third parties and other internal systems.

### **Smarter ATM management**

The ATM fleet is a significant overhead that requires precise management to control costs. Getting value for money out of ATMs is increasingly difficult and can only be achieved with instant access to device information and the ability to quickly troubleshoot problems. FIS DataNavigator has a full suite of features and functions to increase customer satisfaction and make device management more profitable and hassle-free:

- ▶ Device inquiry provides a full business view of device information, showing real-time status as well as maintaining a complete device history with user comments.
- ▶ Cash management provides a real-time view of the current cash position for each ATM, per currency and totalled by individual canister, optimizing cash levels and enhancing ATM profitability.
- ▶ Auto balance is an online tool for balancing ATM devices. Integrated with the exception management functionality, auto balance is an automated reconciliation between system online totals and totals provided by branch staff from their ATMs.
- ▶ Deposit verification tracks all deposits received at an ATM. Integrated with other FIS DataNavigator features, it provides clear audit and tracking capabilities to streamline the back-office process for validating deposits.

## **2.4.4 FIS EnterpriseView**

In this section we discuss FIS EnterpriseView.

### **A better view of your payments enterprise**

As the cost of running multiple back-office operations continues to skyrocket, financial institutions and processors are looking for innovative solutions to help them seize control of the vast amounts of transactional data. In parallel with convergence in back office, the appointment of *payment czars* at many progressive institutions is helping to provide enterprise-wide payment oversight across multiple business lines. This in turn helps to identify opportunities for both operational efficiencies and new revenue growth.

FIS's EnterpriseView is a revolutionary transaction data management solution that enables financial institutions and processors to lower their overall cost of back-office processing operations, accelerate convergence efficiencies, and provide new business insight into growing volumes of payment transaction data.

In today's electronic payment processing environment, financial institutions and processors typically run multiple in-house and third-party vendor applications to manage issuing/acquiring businesses, monitor the ATM fleet, and track settlement positions across the networks.

For back-office operators, there is an ongoing challenge to collect management-level transaction data within a single, user-friendly application that connects seamlessly to multiple data repositories and systems. For years, FIS software and processing customers have leveraged FIS DataNavigator to take advantage of powerful transaction analysis and settlement features. Now, with FIS EnterpriseView, customers can take their back-office management to a new level.

FIS EnterpriseView leverages multiple transaction data sources to collect and display key management-level payment processing information through a highly interactive, easy-to-use *dashboard*. Enterprises can choose from predefined packages of summary data (called view packs) that represent segments of the overall payments enterprise.

### **EnterpriseView benefits**

The EnterpriseView benefits are:

- ▶ Easy-to-use interface
- ▶ Connects multiple transaction data systems and repositories
- ▶ Leverages power of FIS DataNavigator and the FIS Open Enterprise architecture
- ▶ Provides at-a-glance health of entire payment processing operation
- ▶ Reduces back-office management expenses and resources
- ▶ Flexible user profiles and access controls

### **Accessing EnterpriseView**

From a security perspective, users must log on to EnterpriseView, and each view pack is secured through the use of a security profile. The profile will control what view packs the user has access to. In addition, data maintained within the system is secured. This helps to ensure that users with a certain set of access privileges are restricted to specific view packs.

### **Dashboard refresh**

Dashboard information can be refreshed manually or automatically. Automatic refresh can be set through user preferences. This includes turning automatic refresh on or off, as well as the time interval for automatic refresh.

## The main view packs

EnterpriseView ships with three standard view packs: ATM Status, Acquirer/Issuer Activity, and Settlement Positions.

- ▶ ATM Status displays a graphical representation of how many ATMs are currently open, closed, closed for maintenance, or wounded.
- ▶ Acquirer/Issuer Activity displays the average transaction per second rate for the last hour of processing, along with the current day's transaction count and approval rate for on us/at us, on us/at foreign, and not on us/at us activity (through the time of the last refresh).
- ▶ Settlement Position shows the previous day's settled financial activity along with the current settlement position (through the time of the last refresh).

## Main EnterpriseView interface

On the left-hand side of the dashboard, a drill-down button for each of the view packs appears. This is a scrollable list of available view packs. Below the list of buttons is a summary button, which will always bring the user to this initial display. The right-hand pane (labeled Summary) shows a high-level summary for each of the view packs. View pack summaries will be shown for the view pack buttons displayed in the navigation pane.

## ATM status

This view pack provides a list of the ATMs that this user has authorization to view. Information provided for each ATM includes:

- ▶ The device ID
- ▶ Current status of the device (open, closed, closed for maintenance, or wounded)
- ▶ Date/time of the last transaction from this device
- ▶ Current cash balance in the ATM
- ▶ Approval rate (current day from midnight through the time of the last refresh)

The list of ATMs can be sorted by clicking any column heading (in ascending or descending order). In addition, if the user clicks an individual device, additional information is displayed in the lower pane. This information includes the device address and the last ten status messages from the device.



### **Acquirer/issuer activity**

The acquirer/issuer activity view pack displays four types of acquirer/issuer activity:

- ▶ Transaction totals (amounts)
- ▶ Transaction counts
- ▶ Surcharge amounts
- ▶ Surcharge fee amounts

Activity information can be displayed for today (hourly), for the current month (daily), for the current quarter (weekly), or for the current year (monthly). In addition, for each of these periods the previous period can be displayed (yesterday, last month, last quarter, or last year).

### **Settlement position**

The settlement position view pack displays a list of each network that the organization participates in, displaying the network ID, total transaction amount, total transaction count, and the total adjustment amount and count for each network (today and yesterday). In addition, a pie chart on the right-hand side of the pane displays this same information graphically.

## **2.5 Overview of FIS Enterprise Payments system**

This section discusses the logical view and shows the operational view.

### **2.5.1 Logical view**

The FIS Enterprise Payments System comprises four major components that can run on the mainframe: UNIX®, z/OS UNIX (formerly UNIX System Services), and WebSphere Application Server (WAS). If the chosen operating system is z/OS, then Connex on IBM is used. Otherwise, if it is Linux, then IST will be used instead.

Figure 2-2 illustrates the logical view of the FIS Enterprise Payments System.

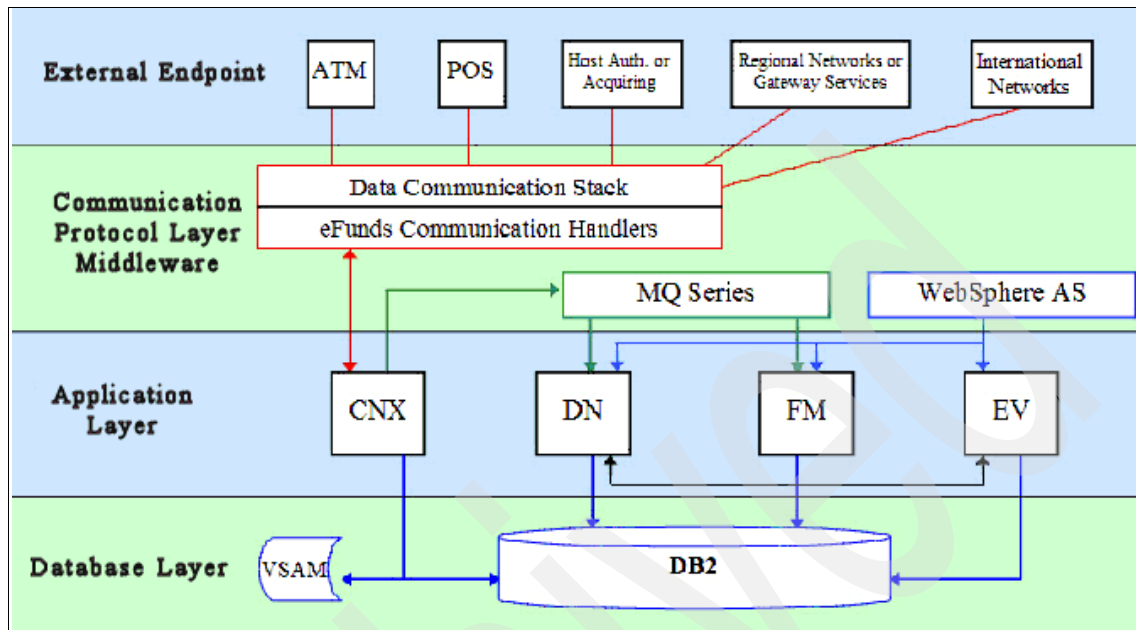


Figure 2-2 FIS Enterprise Payments logical architecture

### Acquiring transactions from endpoints

The Connex on IBM system acquires transactions from terminals and other computer systems through communications and acquiring tasks. Acquiring tasks ensure that each transaction completes properly by providing the requested services or by handling the appropriate error condition. Even if the system is interrupted, the acquiring tasks ensure that transactions being processed at the time of interruption are completed in some way. Acquiring tasks also deal with different formats among incoming transactions, and with the need to record the result of each transaction for later offline processing and analysis.

### Communication handler

Connex on IBM Foundation Services interact directly with the IBM control software, such as MVS™, socket-base TCP/IP communications stack, SNA, MQ Series, and Virtual Telecommunications Access Method (VTAM®). Ultimately, all Connex on IBM software uses the standard system services provided by the IBM system control software to work with the underlying hardware.

### Sending transactions to external processors

The system exchanges transactions with other processors through communications and processor interface tasks. These tasks ensure that each

transaction keeps moving by keeping track of how long external processors take to respond with a reply, and by notifying the system when a processor is unavailable or responding slowly. Processor interface tasks also manage connection, logon, handshake, and security requirements for external processors.

## **Applications**

The FIS Enterprise Payments System is the collection of the following applications:

- ▶ Connex on IBM, which is also known as a transaction service layer, is a generalized transaction processing system that acquires transactions from an external source, processes them, and sends a reply back to the original source. The components of the online system are individual programs that perform specific function. The programs, which are also called Connex on IBM tasks, communicate with each other through the Message Delivery System (MDS), which is a Connex on IBM Foundation Service that moves messages between programs.
- ▶ DataNavigator, which is also known as a Transaction Management Layer, is the back-office transaction management component of the FIS Enterprise Payments platform.
- ▶ FIS Fraud Navigator, a customer interaction layer, is a robust and integrated fraud management solution that enables issuers and acquirers to detect preventable fraud as early as possible to avoid potential loss.
- ▶ EnterpriseView, another customer interaction layer, is a transaction data management solution that enables financial institutions and processors to lower their overall cost of back-office processing operations, and at the same time accelerate convergence efficiencies. This solution can also provide new business insight into growing volumes of payment transaction data.

## **Databases**

FIS products support both DB2 and VSAM file systems.

Figure 2-3 shows an operational view.

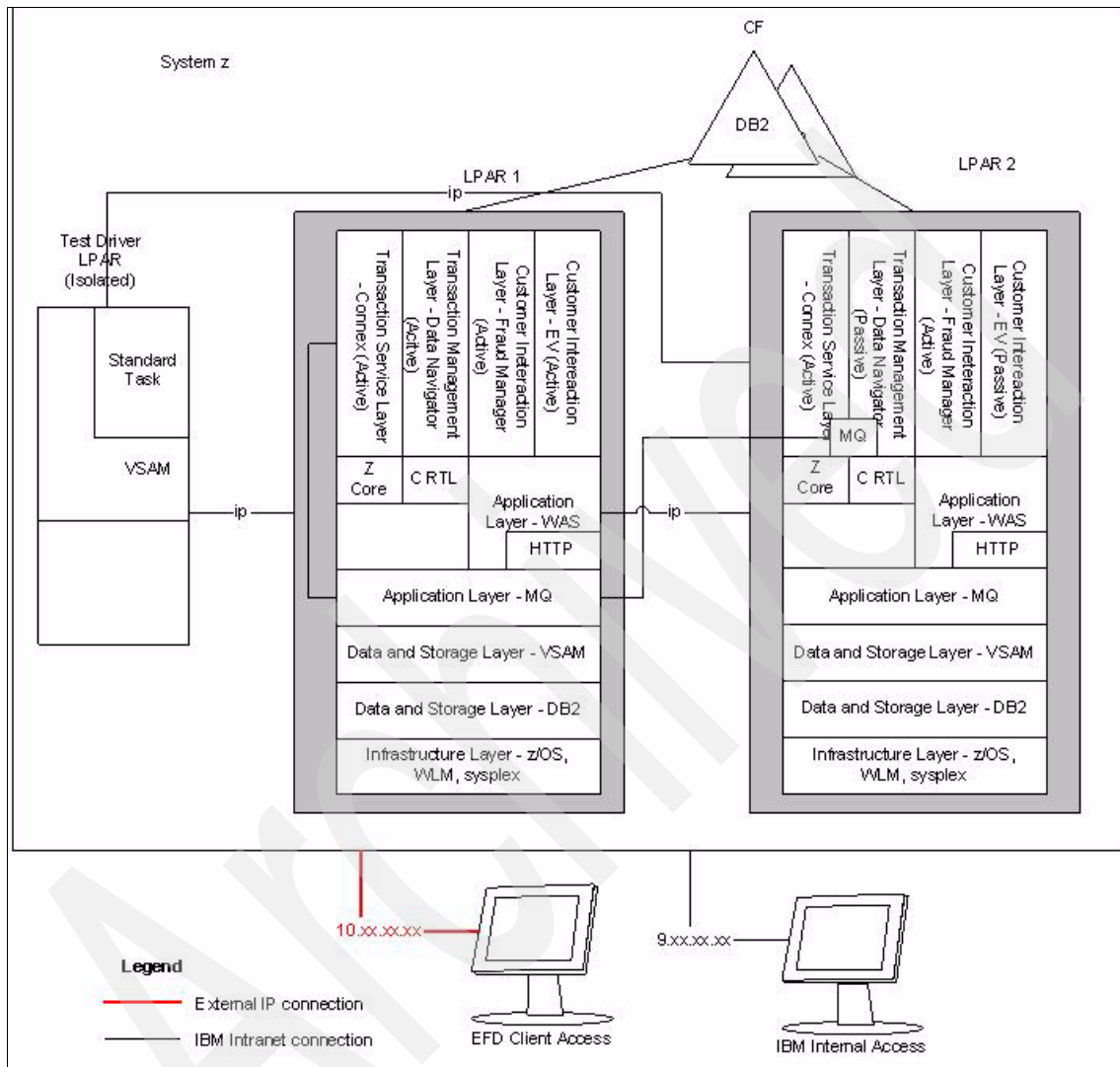


Figure 2-3 Operational view

## 2.6 Integration, applications, data, tools

This section discusses the front end, back end, and points in between.

## 2.6.1 Front end

The FIS ISO 8583 Processor Interface is an optional component of the Connex on IBM application. The messages are based on the International Standards Organization (ISO) 8583-1987 standards. The messages are variable in length and use a bit map structure. All 192 bits in the ISO message are supported to the extent that any data element can be present in any message. If the receiver of the message does not need the data element, it is bypassed. This allows the flexibility to introduce new data elements and allow the endpoints to utilize the data or not.

The FIS ISO PI allows the data within the message to be structured in many different options, as follows:

- ▶ The message can be EBCDIC or ASCII characters. This is controlled by a configuration repository (CR) option found in the PI task Extension book, (Detail primary tab and Sessions subtopic tab). The default is EBCDIC.
- ▶ The bit map in the message can be either expanded or compressed (unsigned packed) characters. This is controlled by a custom configuration table entry (DINFOU) in the CR. The default is compressed.
- ▶ The numeric data can be packed or unpacked data. This refers to the numeric length fields of variable data and the numeric data. This is controlled by a Custom Configuration Table entry (DINFOU) in the CR. The default is packed.
- ▶ The variable data fields can be fixed or variable in length. This implies that the variable length fields can be padded with blanks to fill the full size of the field or the blanks can be excluded. This is controlled by a Custom Configuration Table entry (DINFOU) in the CR. The default is variable.

This supports many different types of message flows, from authorizations to Authorization Services (AS)/Authorization Processor (AP) hot card maintenance. A brief summary of the supported messages follows:

- ▶ Authorization/financial transactions - This includes withdrawals, balance inquiries, deposits, payments, transfers, purchases, funds inquiries, merchandise returns, check verifications, and check guarantees. Check verification and check guarantee messages currently are only supported using the source-based routing options of Connex on IBM Routing Services (CRS).
- ▶ Reversal messages - This includes both full and partial reversals.
- ▶ File update messages - This includes AS negative file updates and AP file update messages.

- ▶ Reconciliation messages - This includes receiving reconciliation messages, checking them with online collected totals, and sending collected online totals to the host.
- ▶ Administrative messages (electronic mail) - Messages can be sent between two different endpoints within the Connex on IBM System or special instruction messages can be sent from a console to an endpoint, and from an endpoint to a console.
- ▶ Network management message (signon, signoff, key changes, cutoff and echo tests) - These can be a single network message for issuer and acquirer or separate messages for the issuer and for the acquirer.
- ▶ Confirmation messages - The financial, authorization, and reversal advice messages may optionally require response messages. If the acquirer requires financial advice responses, then the Intercept (Acquirer) book, (Main primary tab and Options2 sub topic tab), Confirmation Required field must be set to Y.
- ▶ The Advice Response Required field controls whether the financial advices are required from the issuer.

The following processing functions are supported:

- ▶ Address verification support - The Connex on IBM system supports the transmission of all the required address verification data between hosts, but does not perform the actual verification function. The only PIs that currently pass address verification data through the Connex on IBM System are the Format 8.2 PI and the FIS ISO PI.
- ▶ VISA PS2000 support - The Connex on IBM system supports this program. If the FIS ISO PI receives additional data that is required to support PS2000, the Connex on IBM system logs it. The data is only available to the offline system and is not passed on to the hosts.
- ▶ VISA CVV support - In the Connex on IBM product, the Connex on IBM system may perform actual card verification using the proper hardware security box, or the host may perform the verification. In either case, a CVV failure causes a transaction failure. The Connex on IBM system does not allow a transaction to be approved if the CVV process failed.
- ▶ Interlink support - The Connex on IBM system logs an additional fee type indicator to support interlink. The FIS ISO PI receives this indicator from transactions that originate at Interlink. This indicator is not passed on to other hosts.
- ▶ Dynamic key support - Dynamic keys are supported.
- ▶ Message authentication code (MAC) - MACing is not supported in the Connex on IBM FIS ISO PI at this time.

- ▶ Currency conversion facility - Currency conversion is supported by the FIS ISO PI. On transactions acquired by Connex on IBM, amount fields can be converted by the FIS ISO PI to other currencies when sent on to issuers.
- ▶ Statement print facility - The FIS ISO PI supports the statement print function as both acquirer and issuer. The FIS ISO and Format 8 PIs or NCR and Diebold THs can acquire transactions and then route them to either FIS ISO or Format 8 PIs for processing.
- ▶ Surcharging/rebate fees - The Connex on IBM system supports all fee processing except in a partial reversal, where partial fees can exist. If partial fees are encountered, a log record (C0000) is generated.
- ▶ Advanced ATM functions - The FIS ISO PI supports all of the advanced ATM functions as both acquirer and issuer. This includes stop pay, check reorder, check inquiry, mail statement, information inquiry, notification to band, and quasi cash (gift certificates, money orders, script, and prepaid cards) transactions. The FIS ISO and Format 8 PIs or NCR and Diebold THs can acquire these transactions and then route them to either the FIS ISO or Format 8 PIs for processing.
- ▶ Duplicate financial advice transactions - A user exit is available that allows the PI to check for duplicate financial advice transactions. To enable this user exit, add the CXPSFS0H module to the module list for the PI task.
- ▶ EMV chip support - Networks are implementing rules to replace magnetic stripe authentication with EuroPay MasterCard Visa (EMV) chip card authentication, which offers a more secure method of validating cardholders and is less susceptible to fraud schemes such as skimming. Support for EMV chip card authentication is an optional function of this processor interface. For specifics on implementing EMV chip card support, see the FIS ISO 8583 Processor Interface Message Specifications Guide.

In addition to the bit map supported for the ISO 8583, the following bit maps are also supported:

- ▶ ADV/REV CODE (bit 60) - This field contains a byte map to control which reversal and advice sub fields follow in the field. For more information about what the values mean, refer to the *FIS ISO 8583 Processor Interface Specifications Manual* (Message Specifications bookshelf). If one of these values is specified, then the ADV/REV CODE data is included on outbound messages. Entering a value also makes Adv/Rev code required on inbound messages. The default setting is "C0" for all advices. This sends both a reversal (00 if not a reversal) and an advice (00 if not an advice) code.
- ▶ FIS DATA (bit 63) - This field contains a byte map to control which sub fields follow in this field. For more information about what values can be specified, refer to the *FIS ISO 8583 Processor Interface Specifications Manual* (Message Specifications bookshelf). If a non-blank value is specified, then the

FIS data is included on outbound messages. Entering a value also makes this field required on inbound messages. The default setting is blanks on all messages.

The FIS ISO 8583 PI can be used with the SNA or TCP/IP communications protocols. Follow the steps in this book to define your FIS ISO 8583 PI. Then set up communications for your FIS ISO 8583 PI as explained in the manual for the communications protocol that you are using.

- ▶ For a TCP/IP HCH, refer to the *TCP/IP HCH Installation Guide*.
- ▶ For an SNA HCH, refer to one of the following manuals:
  - *VTAM Host Communication Handler General Installation Guide*
  - *VTAM Host Communications Handler Logical Unit Type 6.2 Installation Guide*

## 2.6.2 Back end

The enterprise payments applications can connect to various external systems such as financial institution, authorization host, regional or gateway services, and international networks. Some of these systems may be considered older or corporate services, hence, some form of integration or message brokering will be necessary. Others are more directly related to the support of Internet-like applications.

## 2.6.3 MQ between DataNavigator and Connex on IBM

The Connex on IBM uses IBM WebSphere MQ as the mechanism to continuously send the log data to the DataNavigator. To use this MQ feed, there are three new tasks added to the Connex on IBM system. The first task is a Log Control (LOGCNTL) task that acts as the primary point of logging. Its function is to distribute the log messages to each of the log tasks as well as MQ. This task also sends back the acknowledgement to the acquiring task (for example, terminal handler or processor interface) when the logging of the message is complete. The next task is a MQ Control (MQCNTL) task. Its function is to keep track of what has been sent and what needs to be sent. It also checkpoints the log record if MQ is not available. The final task is the MQ Server (MQSTASK) task, which does the actual interface with the MQ sub-system. Its function is to write individual log messages to the queue and notify the MQCNTL task of success or failure.

When DataNavigator is down and the MQ sub-system is still available, Connex on IBM will continue writing messages to the MQ queue until the queue is full. The size (for example, maximum number of records that can be on the queue) of



this MQ queue is defined in the configuration repository (CR). When the queue is full, the MQCNTL task will checkpoint the information about the log records so that they can be sent once the MQ sub-system is up and running again. The actual place where the records are stored is on the log file, so it is important to have enough log files to prevent them from being reused for the duration of time that you think the MQ sub-system can be down.

When the DataNavigator application is down and the MQ sub-system is also down, the MQCNTL task in the Connex on IBM application will checkpoint the information about the log records so that they can be sent once the MQ sub-system is up and running again.

However, if the DataNavigator application is down beyond the amount of time that we can store the log records on the queue and checkpoint, then a batch feed of transactions from the log files will be needed when DataNavigator is restarted. It will detect duplicates so that it does not care whether the log contains more than just the transactions during the outage.

#### **2.6.4 FIS Fraud Navigator integrating with Connex on IBM**

Currently, the Connex on IBM data logs are unloaded and being sent to the FIS Fraud Navigator application using the File Transfer Protocol (FTP). This process can be done manually or done under the automated process.

Archived

## Benefits of a mainframe solution for payments

Optimizing the payments business is increasingly foundational to a bank's commercial health. Today payments can account for a significant portion of a bank's total cost. The combined forces of increased regulatory pressure and increased costs of maintaining fragmented systems on multiple platforms are making it difficult for banks and processors to remain competitive and capitalize on emerging growth opportunities.

In the search to reduce costs while providing more insight into customer needs, banks are focusing on enterprise payments, transaction data insights, and fraud management, among others. This chapter describes a partnership between FIS and IBM that offers a payments solution with these System z benefits:

- ▶ Centralized data storage
- ▶ Workload management
- ▶ Performance management
- ▶ Scalability
- ▶ Encryption
- ▶ High availability and security
- ▶ Recovery

## 3.1 FIS and IBM partnership

Through a program called *the ISV Advantage for Industries Alliance*, FIS and IBM are helping global financial services customers transform their retail banking and payments environments, and capitalize on emerging opportunities in a rapidly changing payments market.

With FIS applications, financial services companies have the unique ability to bring end-to-end payments processing, transaction data management, and fraud management together in an open industry standards architecture using DB2, Tivoli® and WebSphere—whether they choose to run System z, System p®, or a mixed platform environment.

The partnership has created a proven reference architecture that reduces customer risk and helps to ensure implementation success. There is already a joint customer base of 70+ successful FIS payments processing implementations around the world, on IBM System p or on System z with DB2, Tivoli, and WebSphere.

FIS products for z/OS take full advantage of the features provided by System z and z/OS Parallel Sysplex® clustering technology to achieve continuous availability. Connex on IBM and DataNavigator both incorporate a code design to take advantage of z/OS strengths. This design allows for software errors to be managed so that the software errors are detected, isolated, and corrected to avoid or minimize any unplanned outages.

**Note:** In addition to z/OS, Linux on System z is another platform that supports FIS products.

## 3.2 An IBM tradition of System z value

As the world's largest provider of financial services business solutions, IBM can help customers of all sizes transform their business and capitalize on modern applications—all built on an optimized technology infrastructure with exceptional levels of availability, performance, and cost control. Two thirds of the world's business transactions are processed on IBM System z mainframes, and have been for four decades.

### 3.2.1 System z architecture

Starting with the first large machines, which arrived on the scene in the 1960s, each new generation of mainframe computers has included improvements in the following areas of the architecture:

- ▶ More and faster processors
- ▶ More physical memory and greater memory addressing capability
- ▶ Enhanced devices for storage of data
- ▶ More and faster channels between storage devices and processors
- ▶ Dynamic capabilities for upgrading both hardware and software
- ▶ Increased automation of hardware error checking and recovery
- ▶ A greater ability to divide the resources of one machine into multiple, logically independent and isolated systems, each running its own operating system
- ▶ Advanced clustering technologies, such as Parallel Sysplex, and the ability to share data among multiple systems.

Mainframe computers remain the most stable, secure, and compatible of all computing platforms. The latest models can handle the most advanced and demanding customer workloads, yet continue to run applications that were written decades ago.

With the expanded functions and added tiers of data processing capabilities such as Web-serving, autonomics, disaster recovery, and grid computing, the mainframe computer is riding the next wave of growth in the IT industry.

### 3.2.2 Who uses mainframe computers

Just about *everyone* has used a mainframe computer at one point or another. For example, many automated teller machines are connected to mainframes. In banking, finance, health care, insurance, utilities, government, and a multitude of other public and private enterprises, the mainframe computer continues to be the foundation of modern business.

Until the mid-1990s, mainframes provided the *only* acceptable means of handling the data processing requirements of a large business. These requirements were then (and are often now) based on large and complex batch jobs, such as payroll and general ledger processing.

The mainframe owes much of its popularity and longevity to its inherent reliability and stability, a result of careful and steady technological advances. No other computer architecture can claim as much continuous, evolutionary improvement, while maintaining compatibility with previous releases.

Because of these design strengths, the mainframe is often used by IT organizations to host the most important, *mission-critical* applications. These applications typically include customer order processing, financial transactions, production and inventory control, payroll, as well as many other types of work.

For more information about the value of the IBM System z platform, see *IBM System z Strengths and Values*, SG24-7333.<sup>1</sup>

### 3.2.3 System z in the finance industry

Financial institutions cite the following reasons for choosing the IBM System z. We discuss them in more detail in this chapter.

- ▶ Centralized data storage
- ▶ Workload management
- ▶ Performance management
- ▶ Scalability
- ▶ Encryption
- ▶ High availability and security
- ▶ Recovery

#### Centralized data storage

Centralized data storage provides banks with IBM System z highly regarded database server technology, configuration options, support, services, and financial incentives for high availability, security-rich database serving and business value.

System z provides the strategic, centralized access point for customer and corporate data used for marketing and analytics. Multiple applications running in individual logical partitions on System z access the centralized data repository. This approach provides very high systems availability with attractive total cost of ownership for customer insight solutions.

#### Workload management

One of the strengths of the System z platform and the z/OS operating system is the ability to run multiple workloads at the same time, either within one z/OS image or across multiple images. The function that makes this possible is dynamic workload management, which is implemented in the Workload Manager (WLM) component of z/OS. It prioritizes production jobs and user response time to ensure that the bank's business objectives are met.

---

<sup>1</sup> <http://www.redbooks.ibm.com/abstracts/sg247333.html>

WLM allows banks to balance their needs for line reporting, transaction management, and payments operations all on the same system, meeting customer peak performance needs and balancing the priorities.

## **Recovery**

z/OS and DB2 integrated backup and recovery allow back office functions to stay up and running.

### ***Resource Recovery Services (RRS)***

With the increasing number of resource managers available on z/OS, there was a need for a general sync point manager that any resource manager could exploit. RRS enables transactions to update protected resources managed by many resource managers.

### ***Automatic Restart Manager (ARM)***

The Automatic Restart Manager enables fast recovery of the subsystems that might hold critical resources at the time of failure. If other instances of the subsystem in a Parallel Sysplex need any of these critical resources, fast recovery will make these resources available more quickly. Even though automation packages are used today to restart the subsystem to resolve such deadlocks, ARM can be activated closer to the time of failure. It provides automatic restart after a started task or job failure, as well as automatic redistribution of work to an appropriate system following a system failure.

## **Performance management**

Financial institutions need to provide fast response time to their customers. z/OS provides the ability to share resources and direct them dynamically and virtually, whenever and wherever they are needed, according to priorities and objectives set by the bank. System z helps enable financial institutions to consolidate and tightly integrate multiple workloads on a single server with centralized systems management to reduce costs.

## **Scalability**

Scalability tries to ensure that no performance limitation is due to hardware, software, or the size of the computing problem. IBM System z has excellent scalability. This is significantly influenced by the hardware configuration, software configuration, and workload.

Scalable systems (that scale up and scale out) can scale on more than one application or measurement. Increasing performance by adding more processors is commonly referred to as *scaling up*. Increasing performance by adding additional systems is referred to as *scaling out*. System z can do both.

A scalable system adjusts the performance of applications in a predictable manner when the configuration is either increased or decreased.

## **Encryption**

Encryption is a vital part of today's information systems. Transactions sent across networks must be protected from eavesdropping and alteration. Data files on Internet-connected servers must be protected from malicious hackers. Secure Sockets Layer (SSL) traffic must be encrypted at high speeds. The list of areas that benefit from encryption grows every year.

The Encryption Facility for z/OS applies the powerful encryption capabilities of the IBM mainframe to allow banks to encrypt sensitive information, further reducing their risk. With its leading-edge security technologies, including high-performance cryptography and supporting middleware, System z enables the efficient processing of large volumes of retail payments transactions, including card management and ATM point-of-sale transactions in a continuously available and resilient environment.

IBM continuously adds support for new customer requirements, and this generation of System z adds two important improvements to existing cryptographic facilities based on such requirements. The first improvement provides a flexible way to configure cryptographic hardware. The second provides improved cryptographic key management, targeted principally at loading of encryption keys in remotely located automatic teller machines (ATMs).

## **High availability**

There are some FIS products that require near-continuous availability. Availability in this context means the ability of all or some of the users and applications to access the production databases. The question of what percentage of work can be processed on a continuous basis is an economic decision based on cost and the value of continuous availability to the organization.

Although there is no standard definition of near-continuous availability, many large companies have adopted a goal of no more than five to 10 hours per year of planned and unplanned outages. This translates to an overall systems availability of 99.9%. This contrasts with the historical paradigm of a maintenance window of eight hours every weekend, or 400 hours of outage per year. Other companies are adopting service-level objectives slightly less demanding but still challenging. For example, an objective of a quarterly four-hour maintenance window (16 hours per year of planned outage) requires many of the same design principles as the near-continuous availability objective.

Let us discuss this concept in more detail.



## 3.3 Continuous availability

One popular definition of continuous availability is *high availability plus continuous operations*. There are four essential building blocks for a continuously available system:

- ▶ A design with no single points of failure
- ▶ Highly reliable hardware and software components
- ▶ Ability to avoid planned outages
- ▶ Seamless mechanisms for relocating work loads

### 3.3.1 High availability

There is much confusion in the industry over the use of the terms *reliability* and *availability*. Reliability is the resilience of a system or component to unplanned outages. This is typically achieved through quality components, internal redundancy, and sophisticated recovery or correction mechanisms. This applies to software as well as hardware. Z/OS has millions of lines of code devoted to correction and recovery. This reliability coupled with a design that eliminates single points of failure at the component level provides what is known in the industry as high availability. In fact, this is high reliability, or the ability to avoid unplanned incidents. To achieve continuous availability, we strive for a design that has 99.999% reliability.

### 3.3.2 Continuous operations

The other component of continuous availability is continuous operations. This is the ability to minimize planned outages such as administrative or maintenance work. This includes the ability to upgrade and maintain the central processors, the operating systems, FIS products, DB2, and coupling facilities without disrupting the cardholder's ability to complete a given transaction.

### 3.3.3 System z features

The System z server architecture is designed for continuous availability. It includes self-healing capabilities to avoid downtime caused by system crashes.

The latest System z strategy to provide reliability, availability, and serviceability (RAS) is a building-block approach. It was developed to meet customers' stringent requirements for achieving continuous reliable operation (CRO). The building blocks are:

- ▶ Error prevention
- ▶ Error detection
- ▶ Recovery
- ▶ Problem determination
- ▶ Service structure
- ▶ Change management
- ▶ Measurement and analysis

A primary focus is on preventing failures from occurring in the first place. This is accomplished by using high reliability components and employing screening, sorting, burn-in, and run-in techniques. Failures are also eliminated through rigorous design rules, design walkthroughs, peer reviews, simulations, and extensive engineering and manufacturing testing.

For more information about the latest System z hardware availability and scalability items, see *IBM eServer zSeries 990 Technical Guide*, SG24-6947.

### **3.3.4 Parallel Sysplex - a high availability clustering configuration**

The z/OS operating system can be configured into a Parallel Sysplex, which is the clustering technology for the mainframes. A sysplex refers to a tightly coupled cluster of independent instances of the z/OS operating system. The main objective of a Parallel Sysplex is continuous availability without compromising perceived client performance.

A sysplex can be either basic or parallel. A basic sysplex can communicate using channel-to-channel (CTC) connections between LPARs. Parallel Sysplex uses a processor called a Coupling Facility (CF). Information such as workload, status, and data transmission occurs through the Coupling Facility. The information sharing is constant and continuous, allowing the independent z/OS images to know detailed information about the current status of all images within the sysplex.

Parallel Sysplex architecture is designed to integrate up to 32 systems in one cluster. Each of these systems can handle multiple different workloads and access databases using data-sharing technology. A properly configured Parallel Sysplex cluster is designed to remain available to its users and applications with minimal downtime. Here are some examples:

- ▶ Hardware and software components provide for concurrency to facilitate nondisruptive maintenance. For example, Capacity Upgrade on Demand

allows processing or coupling capacity to be added, one engine at a time, without disruption to running workloads.

- ▶ DASD subsystems employ disk mirroring or RAID technologies to help protect against data loss. They exploit technologies to enable point-in-time backup, without the need to shut down applications.
- ▶ Networking technologies deliver functions such as VTAM Generic Resources, Multi-Node Persistent Sessions, Virtual IP Addressing, and Sysplex Distributor to provide fault-tolerant network connections.
- ▶ I/O subsystems support multiple paths and dynamic switching to prevent loss of data access and improved throughput.
- ▶ z/OS software components allow new software releases to coexist with previous levels of those software components to facilitate rolling maintenance.
- ▶ Business applications are *data sharing-enabled* and cloned across servers to allow workload balancing. This also maintains application availability in the event of an outage.
- ▶ Operational and recovery processes are fully automated and transparent to users. They reduce or eliminate the need for human intervention.

High availability requires at least two servers that provide the same service to their clients, so they can allow for the recovery of service when failures occur. That is, servers perform a backup function for each other within the cluster. High availability minimizes unplanned outages.

Continuous availability is achieved with the System z and z/OS Parallel Sysplex clustering technology. This technology implements a data-sharing design that allows a database to be concurrently read and updated by application clones running on multiple z/OS images on one or more physical servers. Continuous availability avoids or minimizes unplanned outages (high availability) and reduces or eliminates planned outages.

The Workload Manager balances application workloads across the systems in the Parallel Sysplex. If there is a failure or a planned outage on one system, other systems within the Parallel Sysplex take over the full workload. By implementing Geographically Dispersed Parallel Sysplex™ (GDPS®), the servers can be as far as 40 km apart from each other, thus avoiding a total site-wide disaster.

Using the Parallel Sysplex clustering architecture, you can achieve a near-continuous availability of 99.999%, or 5 minutes of downtime a year.

### 3.3.5 Linux for System z

Most Linux for System z environments are run on the z/VM® operating system, although they could run directly on the System z processor or LPAR.

Linux for System z inherits the System z hardware reliability. This is done by configuring a highly available e-business infrastructure to take advantage of the z/VM availability items such as hardware error recovery, automation of starting and stopping Linux guests, heartbeat mechanisms between two z/VM systems for takeover functions, clustering, and DASD sharing.



## Part 2

# Environment and process

This part describes our activities and the environment that we used while writing this IBM publication.

Archived

## Design decisions

This chapter provides the information needed to plan for an FIS installation on z/OS. It discusses:

- ▶ Overall payments system planning
- ▶ Planning steps for each component:
  - Gather requirements: ATMs, links, the environment.
  - Design the solution: selecting the components.
  - Develop the solution.
  - List the installation skill requirements.

## 4.1 Integration planning

Before an installation begins, FIS collects the requirements from the customer to find out how FIS products fit their environment. FIS then works with the customer to select the FIS solution that satisfies the requirement. Once the solution has been determined, FIS develops the solution. During this time, FIS provides a pre-installation checklist. After it is reviewed, FIS schedules a date for the onsite installation.

Since the installation of the FIS products requires a number of different skills, roles and responsibilities must be defined for each participant. For example, a database administrator (DBA) is responsible for creating and managing DB2 databases and their table spaces. A software developer is responsible for installing and building the FIS system environment. A pre-installation checklist shows roles and responsibilities, as documented in the installation guide for each product.

### Customer variables

Typical activities for each installation are list below. Integration planning shows who is responsible for which activity during the installation process.

- ▶ Set up configuration repository (CR).
- ▶ Create the configuration repository database.
- ▶ Load the configuration repository database.
- ▶ Change the endpoint names, file information, and logmodes.
- ▶ Create a configuration repository extract file.
- ▶ Install the configuration repository extract file.
- ▶ Allocate and initialize online and image-specific files.
- ▶ Allocate dedicated DASD volumes for critical files.
- ▶ Customize operations JCL (for example, logon procedures and started tasks).
- ▶ Set up access authority.
- ▶ Certify system generation.
- ▶ Review the installation output.

Some form of integration or message brokering might be necessary for FIS products to successfully connect to the corporate back-end services.

## 4.2 Customer options and needs

You can pick and choose options for each product, and these will impact performance and capacity.



You can use system-managed storage (SMS) to manage online data sets. However, you should be aware of the following:

- ▶ Data sets might be archived, and thus unavailable for online tasks to use.
- ▶ Online tasks might not know the space allocation of data sets being managed by SMS.

You may need additional systems to accommodate your particular testing needs. For example, if you routinely do certification testing of new terminals for merchants or financial institutions, or new acquirer/issuer host links, you may want to define a separate certification (CERT) system with a copy of your production software and a test set of application data files.

Although you can run a production online system environment on a single image system, a multi-image provides higher availability.

You can install all the FIS products yourself using the provided FIS installation documentation. An option is that you can work with the FIS installation team.

## 4.3 Availability planning

Availability is always a trade-off between business needs and cost. Higher availability costs more. At some point the cost of high availability will outweigh its benefits. This analysis is different for every customer. Only the customer can decide the outcome of the cost/benefit analysis.

Although the software is highly available, FIS products do not impose any specific level of required availability. You can run them in a single machine environment with lots of single points of hardware failure or you can run them in a multi-node environment (for example, Parallel Sysplex) where all components are replicated with no single points of failure, or any point in between. You get to choose.

Connex on IBM can also take advantage of GDPS. This allows you to have Connex on IBM images running in multiple data centers, further enhancing availability. But there are distance restrictions. What do you do if your data centers are not close enough together? You can still configure Connex on IBM to run in this environment. Without GDPS some automated recovery functionality may be lost (for example, use of ARM), but manual actions can be used to compensate. In this mode, a single Connex on IBM system can be spread around the world.

Note also that availability should be measured at the user (that is, consumer) interface. Many failures can make the system seem unavailable even though the

mainframe hardware and software are operating flawlessly. Recoverability and availability must be considered at all levels. Some of these factors include:

- ▶ Endpoint device (such as ATM, POS, cash register)
- ▶ Modems
- ▶ Communication lines
- ▶ Communication switching points
- ▶ Routers
- ▶ HSMS (for example, encryption devices)

### 4.3.1 Availability considerations

Consider these concepts when determining system and application availability:

- ▶ Availability requirement

Consider the goal of your application regarding planned hours of operation.

- ▶ Recovery time objective (RTO)

Consider the amount of time that can be allowed to recover an application and put it back online. This could be minutes, hours, or simply a long, medium, or short amount of time to recover.

- ▶ Tolerance of recovery time

Consider what the impact might be of extended recovery times needed to resynchronize data, restore transactions, and so on. This is mostly tied into the time-to-recover, but can vary widely depending on who the users of the system are.

- ▶ Recovery point objective (RPO)

Consider how much data you are willing to lose, and whether it needs to be kept intact.

- ▶ Application resiliency

Consider the application availability that you are seeking. In other words, should your applications be able to be restarted on other machines without the user having to reconnect?

- ▶ Degree of distributed access/synchronization

For systems that are geographically distributed, consider how tightly or loosely coupled the application is with the data. If the application and data are very loosely coupled and can stand on their own for periods of time, then they can be resynchronized at a later date.

- ▶ **Planned outage schedules**  
Consider the anticipated rate of scheduled maintenance required for the box, z/OS, network, middleware, application software, and other components in the system stack.
- ▶ **Performance and scalability**  
Consider the system performance and scalability needed for this application.
- ▶ **Cost of downtime**  
Consider the cost for every minute of downtime. Typically, short downtimes have lower costs, and the costs grow exponentially for longer downtimes.
- ▶ **Cost to build and maintain a high availability solution**  
Finally, consider the costs associated with the design, implementation, and maintenance of the high availability system.

### 4.3.2 FIS Enterprise Payments system considerations

In this section we list considerations to keep in mind to enable the FIS Enterprise Payment system solution to achieve a high level of availability.

The FIS Enterprise Payment System on z/OS includes Connex on IBM, FIS Fraud Navigator, DataNavigator, and EnterpriseView. We discuss each product separately and consider what is needed for each to achieve a high level of availability.

#### **FIS Connex on IBM**

Connex on IBM is FIS's global electronic payment solution and features continuous processing availability. The hardware and software components should be set up to avoid a single point of failure (SPOF).

- ▶ **System z hardware**
  - **Multiple central processor complexes (CPCs):** Configure a minimum of two CPCs, regardless of the fact that a single CPC might be able to provide sufficient processing power to handle the entire workload. This is because planned changes require a shutdown or Power-On-Reset of an entire CPC.
  - **Multiple Coupling Facilities (CFs):** We recommend that every Parallel Sysplex has at least two CFs. If using DB2 data sharing, our recommendation is three CFs. This provides greater capacity to cope with unexpected workload spikes and also ensures that there will be no single point of failure, even if one CF becomes unavailable.

- Sysplex timer: The recommended configuration in a Parallel Sysplex environment is the 9037 Expanded Availability configuration. This configuration is fault-tolerant to single points of failure and minimizes the possibility that a failure can cause a loss of time synchronization information to the attached CPCs.
- Production LPAR: Configure two images from the sysplex on each server. Depending on your LPAR definitions, this may give you the ability to continue to utilize all the available MIPS on the processor even if one of the images has a planned or unplanned outage.
- Hardware configuration: From a *logical* perspective, these best practices can also have a positive impact on availability:
  - Single IODF containing all the hardware in the installation.
  - Establish a meaningful naming convention for CPCs, LPARs, and OS configurations.
  - Use the same device number for the same physical device across all images.
  - Maintain I/O symmetry across all systems within the sysplex, so that all I/O appears as one I/O pool that is shared among all images. Doing this will simplify operations and make it much easier to move workloads between different systems as required.
  - Configure *at least* two paths to every device, regardless of the bandwidth requirement. Heavily used devices should have more paths to provide the required level of performance. Configuring paths through ESCON® directors and FICON® switches provides much greater flexibility and utilization of the available bandwidth, especially for FICON channels. Make sure that the paths are configured through different directors and switches.
  - Share the channels between LPARs and provide greater redundancy.

► Database server (AP databases only)

DB2 data sharing effectively eliminates the database server as a single point of failure. In addition, using DB2 data sharing with Connex on IBM can increase the transaction volume per second and provide high availability for data access.

► Disk subsystem

Although disk subsystems have achieved very high levels of availability through internal redundancy and RAID technologies, some companies want complete protection against a disk subsystem failure. Peer-to-peer Remote Copy (PPRC) and Extended Remote Copy (XRC) are two technologies that can provide this protection. Some additional considerations are:

- When setting up DFSMS™ Storage Groups for dual logging, there should be sufficient granularity in the design to enable COPY1 and COPY2 to be placed on different disk subsystems.
- We need to separate high I/O data sets into different volumes. This can reduce system I/O time and reduce interlock opportunity.
- We need to separate very important system data sets in two different CUs or different DASDs. This can ensure application data recovery if DASD fails.

► Application server (Connex on IBM)

You can replicate Connex on IBM over multiple z/OS images, increase the number of address spaces, and logically spread out the TCH and TH tasks. This will improve the capacity to handle more of the workload and provide high availability.

The Connex on IBM application can run as one or more images. Each image is made up of a group of address spaces (jobs) and each address space has one or more dispatchable processes (TCBs). These address spaces can all exist in a single z/OS image or be spread across multiple z/OS images. Each image is capable of running the entire application. When you configure the system, you indicate which processes are active in which address spaces of which images. There are several options depending on the process:

- A process can be running in one image with other images having a backup copy.
- A process can be concurrently running in multiple images.
- Some processes are concurrently running in all images.

Based on the capabilities of your hardware/OS configuration (for example, single machine, simple plex, Parallel Sysplex) you can configure Connex on IBM for maximum availability. If a component (such as a CPU) fails, its workload can be automatically restarted in place, a backup copy can be started, or it can be transferred to an already active copy running elsewhere. This is done quickly and automatically without operator intervention, thereby maximizing the consumer's view of availability.

Availability can also be improved for many other failures. The Connex on IBM software is aware of the status of what it talks to. If a failure is detected, automatic retry procedures is invoked. If they fail, the operator is notified so

that manual intervention can be initiated. During this time, automatic retry continues, but probably at a slower rate. This eliminates the need for and the time wasted by a technician calling operations to notify them that the problem has been fixed and asking them to restart the component. As soon as recovery is completed, the failed component is automatically brought back into service and the problem is cleared.

► Network

Availability is enhanced by the ability to dynamically move IP addresses using dynamic VIPA. Sysplex distributor combines dynamic VIPA, Workload Manager, and autonomic computing to create the highest possible availability of an IP host. Besides availability, DVIPA also provides additional benefits:

- Single image: This allows multiple LPARs to logically be a single, highly available network host. Since DVIPA movement is automatic, users and clients might never know that a DVIPA address move has occurred.
- Application movement: With DVIPA, applications can be seamlessly moved from one LPAR to another, allowing a virtualization of the application itself.

► WebSphere MQ queue sharing

Use MQ queue sharing groups for Connex on IBM. This can provide high availability for data translation from one system to another system.

► Automation

Use automatic restart manager (ARM) or other automatic control tools to monitor the FIS address spaces and restart an address space if it fails. This reduces the time of the system outage.

**Note:** Refer to *Achieving the Highest Levels of Parallel Sysplex Availability*, SG24-6061, for additional information.

## FIS Fraud Navigator

The FIS Fraud Navigator application runs on z/OS UNIX. At the time that data was being gathered for this book, the Connex on IBM data logs were unloaded and sent to the FIS Fraud Navigator application using the File Transfer Protocol (FTP). This process can be done manually or under an automated process. The application needs to be highly available, but currently, not continuously, available.

FIS Fraud Navigator has the capability of running in three modes. One of these modes allows it to participate in the online real-time transaction flow. If fraud is suspected, transactions can be denied and alerts generated. In some cases, you may want to block the card or device from future transactions. When used in this way, high or continuous availability is important.

Many of the availability considerations for Connex on IBM can apply to FIS Fraud Navigator (FN), since they are system availability measures and can benefit any application in general. Specifically, in order for FN to be highly available, it requires all subsystems connected to the WebSphere Application Server infrastructure to also be configured with redundancy. This can be achieved by creating redundant subsystem servers on different LPARs or by configuring the subsystems as sysplex-aware.

Sysplex-aware subsystems, such as DB2 in datasharing mode and WebSphere MQ using shared queues, can take advantage of z/OS advanced workload management concepts, as well as provide availability during failover situations.

WebSphere Application Server (WAS), when configured in Network Deployment mode, can take full advantage of the underlying z/OS Parallel Sysplex and Workload Manager capabilities to provide for fully available and scalable infrastructures.

Like Connex on IBM, the FIS Fraud Navigator application server code can be configured in many ways depending on the desired level of availability. Running in UNIX System Services (z/OS UNIX), it has the benefits of the highly available System z hardware and the z/OS operating system. It runs as a series of processes, each with multiple threads. In a z/OS UNIX environment, each LPAR can be thought of as a node. The application can run on a single node or be configured as a multi-node application with processes existing in one or more LPARs that are on one or more machines (both local or remote). Workload can be assumed if processes/threads fail. Multiple nodes can be active and sharing workload. In case of error, workload is moved dynamically and automatically. Depending on your configuration, there may be no perceived outage.

**Note:** Refer to *Architecting High Availability Using WebSphere V6 on z/OS*, SG24-6850, for additional information.

## **FIS DataNavigator**

FIS DataNavigator is a comprehensive solution that collects and connects transactional data across the enterprise. Vast amounts of data are handled in the payments environment every day, providing a rich source of information about customers and their activities. Connex on IBM uses WebSphere MQ as the mechanism to continuously send the log data to the DataNavigator. If the DataNavigator becomes unavailable, Connex on IBM will restart sending the log data via MQ or initiate a batch feed of transaction from the log data. Currently, the DataNavigator must be highly available.

Many of the availability considerations for Connex on IBM can apply to DataNavigator. However, DataNavigator currently runs on one LPAR at a time

and cannot be replicated across multiple LPARS. But by using DB2 in datasharing mode and automation mechanisms like ARM, DataNavigator can be highly available in a system already made continuously available for Connex on IBM.

### **EnterpriseView**

FIS EnterpriseView is a transaction data management solution. It is a WebSphere application that accesses the DB2 database collected by the DataNavigator. This application needs to be highly available.

Since this is a WAS application, the same availability considerations that apply to the FIS Fraud Navigator (WAS and DB2) also apply to EnterpriseView.

## **4.3.3 Avoiding single points of failure**

A single point of failure exists when a critical function is provided by a single component. If that component fails, the system has no other way to provide that function, and essential services, such as the FIS Enterprise Payment system, become unavailable.

### **Where are the single points of failure**

Consider an example where a System z server has several LPARs running z/OS. You have installed the FIS Enterprise Payment solution applications on a single z/OS server. Where are the points of failure? There are several:

- ▶ The System z hardware could experience a planned or unplanned outage.
- ▶ The disk subsystem could fail.
- ▶ The network could fail.
- ▶ The LPAR microcode could fail.
- ▶ z/OS could fail.
- ▶ Middleware could fail (MQ, DB2, or WAS).
- ▶ One or more of the FIS applications could fail.

The odds of each failure are different. In this case, the probability of an application failure is highest, while the probability of the System z hardware failure is lowest. The others fall on a continuum between those extremes.



## Probability and cost of fixing SPOFs

How do we eliminate these single points of failure? An easy and effective method is to eliminate them by duplicating them. Duplicating the application is usually easy, but duplicating the System z hardware can be expensive, with the cost and difficulty of the other components or subsystems falling on a continuum between these extremes. Table 4-1 illustrates the probability of a failure with the cost of fixing the exposure.

Table 4-1 *Single points of failure*

| Single point of failure | Probability of failure | Cost to fix SPoF |
|-------------------------|------------------------|------------------|
| System z hardware       | Very low               | High             |
| Disk subsystem          | Very low               | Medium           |
| LPAR                    | Very low               | Low              |
| z/OS                    | Low                    | Low              |
| Middleware              | Low                    | Very low         |
| Application             | High                   | Very low         |

## Design principles to avoid SPOFs

In the late 1970s and the 1980s, significant effort was applied to developing configurations that had no single points of failure. Frequently, a methodology called Component Failure Impact Analysis (CFIA) was used to systematically analyze the relationship of infrastructure components to application elements. Over time, the lessons of CFIA were institutionalized in configuration design, best practices, and systems assurance reviews. As a result, it became routine for a system that was being designed for high availability to include redundancy in system elements, such as:

- ▶ Central processor complexes (CPCs)
- ▶ Channel paths
- ▶ Coupling facilities (CFs)
- ▶ CF links
- ▶ DB2 dual logging
- ▶ DB2 dual archiving
- ▶ DB2 dual image copies with one for on-site table space recovery and one for off-site disaster recovery storage
- ▶ CF structures through duplexing or fast rebuild
- ▶ Application servers

- Network paths between application servers and the database server (for example, OSA-E adapters with VIPA)

## 4.4 FIS configurations

IBM supports many different ways to configure FIS applications. However, both technical configuration issues and business considerations must be taken into account. Among those business considerations is impact to users, card holders, and account holders when the system is unavailable.

In this section we examine the options to configure when running under z/OS:

- Single LPAR
- Single CPC with two LPARS
- Three CPCs with one LPAR each
- Dual sites with three CPCs having one LPAR each

Several of these are illustrated with actual customer configurations.

These options are not meant to be all-encompassing. Instead, they highlight details to consider when planning an FIS configuration. There are many other considerations (cost of hardware, costs associated with ongoing maintenance and support of the configuration, requirements of other applications, and so on) to be aware of when planning such a configuration, but they are beyond the scope of this book.

These configuration options progress from the most simple to the more complex. While the technical issues are a consideration when determining the appropriate configuration for FIS, the business requirements cannot be ignored. Customers consider the implementation of an e-payment system to be a mission-critical application, so availability is a very significant business requirement. Therefore, the primary focus when choosing options is the availability of FIS applications to process transactions.

Each option includes a description of the configuration and a discussion of the single points of failure associated with the option.

### 4.4.1 Single FIS LPAR/Single CPC

In this section we discuss Single FIS LPAR/Single CPC.

## Customer configuration one

In this situation, all the FIS components run in one LPAR or perhaps more, but continuous availability is not a consideration.

### Single points of failure

This configuration has several points of failure and is not typically recommended for production use. This option is suited for a test environment.

Software is generally considered less reliable than hardware. The System z hardware contains redundant components, making its *mean time between failures* (MTBF) in the range of years. Because the System z hardware is so reliable, we allow the System z server to be a single point of failure in this architecture.

The component most likely to fail is the application or middleware. When this happens, the application can be restarted and recovery time will be a few minutes. Or perhaps the LPAR has to be rebooted and recovery time could be around 5 minutes. Recovery requires that some manual or automatic method be implemented to detect the failure and initiate recovery.

If this recovery time is sufficient, then there is nothing more that needs to be done. If not, then higher availability is needed.

### Single CPC

The single CPC is a possible point of failure. While a CPC failure is rare, planned maintenance will cause a system outage for the duration of the maintenance.

### Single LPAR

The first point of failure is the single LPAR. If there is a planned or unplanned outage of the LPAR, the entire system becomes unavailable for the duration of the outage.

### FIS Enterprise Payments System

All the FIS components in this configuration require some manual or automatic method be implemented to detect the failure and initiate recovery.

## 4.4.2 Multiple LPARs/Single CPCs

This option duplicated the software environments, since software is generally considered less reliable than hardware. The System z hardware contains redundant components, making its mean time between failure (MTBF) in the range of years.

Based on the fact that the System z hardware is so reliable, we allow the System z server to be a single point of failure in this architecture.

We duplicate all the critical software environments (LPAR, Connex on IBM, WebSphere, DB2) so that none of them is a single point of failure.

The architecture also leverages the high availability capabilities and features of z/OS Parallel Sysplex.

## **Customer configuration two**

In this situation, the production and test environment utilizes two Connex on IBM images (image01 and image02) residing on the same System z server (CPC). Image01 and image02 operate on separate LPARs, with each having multiple CPUs. Each LPAR is dedicated to the Connex on IBM application (and some associated applications that help monitor the network). All tasks are run on both image01 and image02 with the workload shared across both images. They share common tasks operating on both images (AP, AS, Security, CRS, logging).

They have about 12–13 started tasks (address spaces) per image. They assign all major tasks to their own address spaces to minimize impact on the network upon failure of any given task. For instance, each AP task is in its own address space, whereas a PI has its associated HCH and TT in one address space.

DataNavigator is initially started on images01 and has a defined Parallel Sysplex ARM policy to restart in an available z/OS image.

WebSphere Application Server (WAS) is configured in Network Deployment mode across LPARS. WebSphere MQ clustering is used to set up a full DB2 data-sharing group and a queue-sharing group across production LPARs.

## ***Communication between the LPARs***

Communication between the LPARs is:

- ▶ IP communication among independent operating systems running in logical partitions (LPARs) on the same machine.
- ▶ Communications among a tightly coupled cluster of mainframe LPARs (called a Parallel Sysplex).
- ▶ The database volumes are configured for multipathing, so that if one path fails, access to the device is maintained through the surviving paths.
- ▶ Dynamic virtual IP addressing (DVIPA) is used to balance workload and minimize network failure on the user.

### **Single points of failure**

This option provides greater availability than the previous option. It has some of the same points of failure, but nevertheless may satisfy the availability needs of many FIS customers.

#### ***Single CPC***

The single CPC is still a point of failure. While a CPC failure is rare, planned maintenance causes a system outage for the duration of the maintenance.

#### ***Single copy of data***

The final point of failure in this model is the file subsystem. Because there is only one copy of the data, a catastrophic failure of the DASD subsystem results in a system outage.

### 4.4.3 Multiple LPARs/Multiple CPCs

In this option we duplicated the hardware and software environments to eliminate almost all single points of failure. These critical hardware and software components include the CPC, LPARs, Connex on IBM, WebSphere, and DB2. The architecture also leverages the high-availability capabilities and features of z/OS Parallel Sysplex. See Figure 4-1.

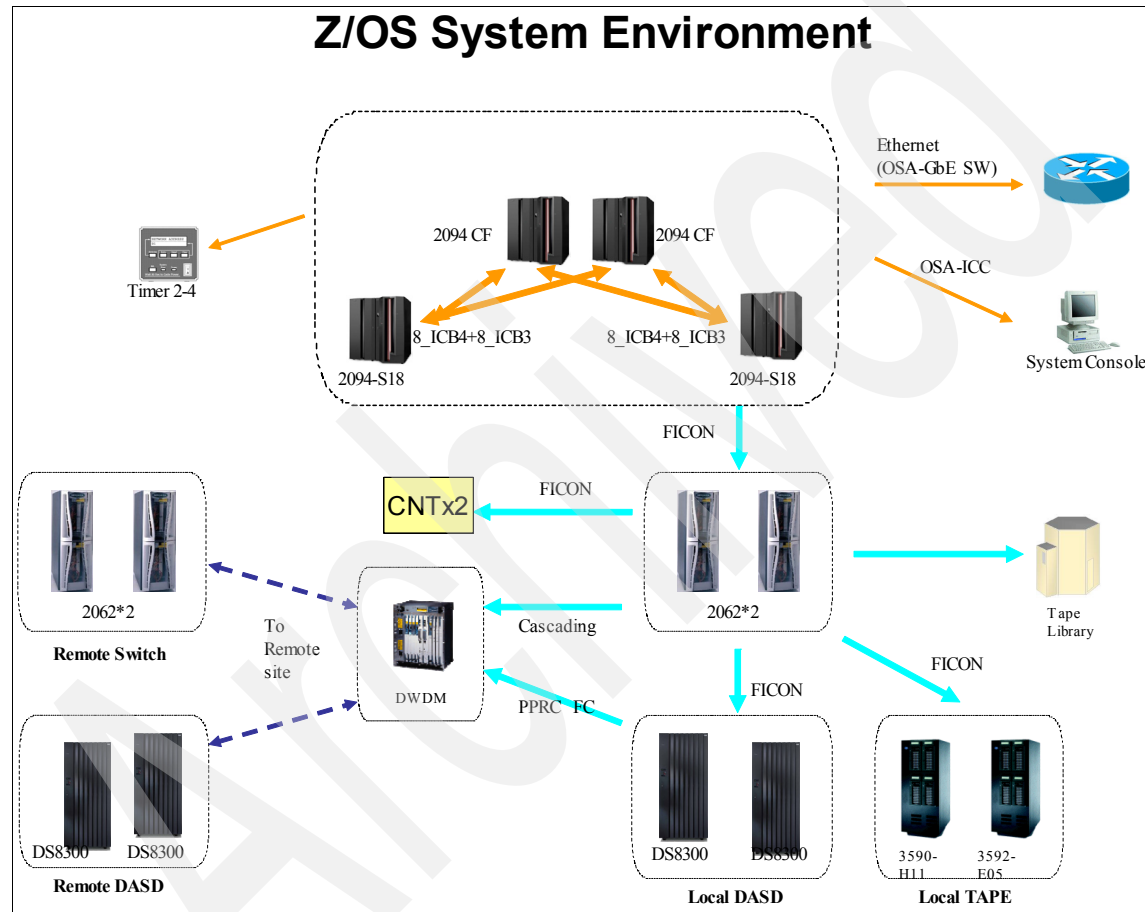


Figure 4-1 Hardware duplication

### Third customer configuration

The production and test environment each utilize two Connex on IBM images (image01 and image02) residing on different System z servers (CPCs). There is also a third CPC for use as a backup during planned and unplanned outages.

The CPCs are side by side in the same location. Image01 and image02 operate on separate LPARs, with each having multiple CPUs.

Each LPAR is dedicated to the Connex on IBM application (and some associated applications that help monitor the network). All tasks are run on both image01 and image02, with the workload shared across both images. They share common tasks operating on both images (AP, AS, security, CRS, logging). They have about 260 started tasks (address spaces) per image. They give major tasks their own address spaces to minimize impact on the network upon failure of any given task. For instance, each AP task is in its own address space, whereas a PI will have its associated HCH and TT in one address space.

VSAM Record Level Sharing (RLS) is used on the AP and AS tasks so that the files can be shared across images. DataNavigator is initially started on images01 and has a defined Parallel Sysplex ARM policy to restart in another z/OS image.

WebSphere Application Server (WAS) is configured in Network Deployment mode across LPARS. WebSphere MQ clustering is used to help set up a full DB2 data-sharing group and a queue-sharing group across production LPARs.

### ***Communication between the LPARs***

Communication between the LPARs is:

- ▶ IP communication among independent operating systems running in logical partitions (LPARs) on the same or different machines.
- ▶ IP communications among a tightly coupled cluster of mainframe LPARs (called a Parallel Sysplex).
- ▶ The database volumes are configured for multipathing, so that if one path fails, access to the device is maintained through the surviving paths.
- ▶ DVIPA is used to balance workload and minimize network failure for the user.

Parallel Sysplex is implemented with the GDPS/PPRC/HyperSwap solution to eliminate a single copy of data as a point of failure.

### **Single points of failure**

This option provides the highest availability that can be achieved at one site. A well-maintained and well-operated system using this configuration should be able to withstand any event, short of a catastrophic site failure.

#### **4.4.4 Multiple LPARs/multiple CPCs/dual site**

This option duplicates the configuration (from the previous section) at an alternate site. We added a Parallel Sysplex+GDPS/XRC solution to eliminate all single points of failure.

For additional tips on high availability (also called Business Continuity), see Appendix B, “Business continuity concepts” on page 177.



# Installation and configuration

This chapter provides customers with the overall steps needed to install FIS products on z/OS. It is a supplement to the product documentation, and we provide tips from our experiences. We also provide a task to verify the installation of each product. We answer these questions:

- ▶ What steps will be needed in order to install these products?
- ▶ What skills must be available?

For each product, we discuss:

- ▶ A general sequence of activities
- ▶ Installation verification
- ▶ Customization hints and tips

**Note:** This suite of payments processing software was written by a company called eFunds (EFD). In 2007, eFunds was acquired by Fidelity National Information Systems (FIS).

## 5.1 FIS Connex on IBM

The first product the we cover is Connex on IBM. We list the skills needed to install and maintain Connex on IBM, the hardware and software needed, preinstallation planning tips, and installation steps.

**Note:** This section should be used in conjunction with the standard product documentation.

### 5.1.1 Skills

For this project we need:

- ▶ z/OS system programmer
- ▶ DB2 database administrator
- ▶ Storage management administrator
- ▶ Security administrator
- ▶ Network administrator
- ▶ WebSphere MQ administrator
- ▶ Connex on IBM administrator (or input from FIS consultant)

### 5.1.2 Prerequisite software and hardware

Review Chapter 5 of the *Connex on IBM System Installation Guide*.

Although system managed storage (SMS) volumes can be used to receive the product files, the installation process (and tuning parameters) dictate that the installer select the specific volumes rather than letting SMS choose them. Some volumes (such as those containing log files) must be dedicated for performance reasons.

FIS software can utilize multiple CPs, so the system can run as many tasks as required by defining more CPs in the LPARs. Transactions are time-sensitive, so allocate enough CPs to support the desired response time.

### 5.1.3 Preinstallation planning for Connex on IBM

One of the first steps is to fill out the FIS preinstallation activities questionnaire (Chapter 5 of the *FIS Connex on IBM Installation Guide*).

**Note:** We highly recommend that an FIS expert be onsite to help the customer install a test environment.

## **DASD requirements**

The customer needs to estimate how much data they will keep on disk. One consideration is how much business data they will have for both the online and batch jobs for two to three years. This information will help the customer plan their DASD layout. Make sure that there is enough space available and that the data sets are placed for optimum performance (see installation guide). FIS can prepare a hardware sizing that will provide both DASD and CPU utilization estimates.

As you place your data sets in their desired location and a SAN is available, consider the advantages and disadvantages of using a SAN rather than a local DASD. Some of these include:

- ▶ Data access on a SAN is typically faster than local DASD. Your data is spread across more actuators and access arms.
- ▶ SANs can be accessed from both remote and local locations. Note the remote bandwidth implications for time-critical access.
- ▶ The SAN typically has replication options that can be implemented without the use of mainframe cycles and with no impact on the application.
- ▶ Operating costs of a SAN are typically lower.

Both SAN and local storage work fine. Each installation needs to evaluate its environment.

## **Enterprise naming conventions**

When the customer installs Connex on IBM, they need to follow their enterprise naming convention for system data sets, catalogs, databases, tablespaces, and indexspace names. These include LOADLIBs, customization JCL, ISPF data sets, start procedure, and RACF®-protected data set names.

## **System log backup strategy**

If the Connex on IBM system log is not archived for a long period of time, our suggestion is to define a GDG. If it is archived for a long time, the customer can use IBM Hierarchical Storage Manager (HSM) to archive their log data sets. Think about legal requirements for data retention, as well as consumer protection requirements such as encryption.

### ***Connex on IBM log considerations***

Connex on IBM system log, store and forward data sets, and checkpoint data sets are very important. These data sets should be placed on a separate DASD volume for recovery purposes.

### **Security**

Secure your resources, including data sets and DB2 data. Utilize ICSF. Use cryptographic keys and APF authorization. Make sure that the crypto processors are turned on and usable. Validate that the crypto hardware is capable of meeting your needs.

Modify user logon procedures according to the sample in Appendix F of the *Connex on IBM Installation Guide*.

Arrange a level of access for each operator (see Figure 5-1 on page 79).

### **Network**

For the network:

- ▶ VTAM definitions (APPLIDs)
- ▶ Firewall definitions
- ▶ 3270-type terminals for operations consoles

### **Subsystems**

For the subsystems:

- ▶ DB2
- ▶ WLM policy
  - Velocity goals only: AS spaces and started tasks fitting with the customer current policy
  - Report classes
  - RMF™ for monitoring production environment
- ▶ JES (Dedicated initiators must be allocated for production unless you will use started tasks)
- ▶ Every CNX switch requires ENQ support
- ▶ MQ configuration and sizing for DataNavigator communication
- ▶ OMVS MAX parameters customization

### ***Coupling facility***

For the coupling facility:

- ▶ Separate XCF path for communication between Connex on IBM switches.
- ▶ We recommend data sharing across LPARs if using DB2 in a Parallel Sysplex.

## **5.1.4 Installation steps**

In this section we discuss the installation steps.

### **Sample JCL**

To begin, we run the sample JCL in the install menu to unload the Connex on IBM installation data sets and system data sets from the tape. The unloaded PDSs contain all JCL for Connex on IBM system installation. We need to modify the JCL according to the installation plan.

Note that if the RACF PROTECTALL option is active, we must define in advance all the data set profiles before doing the Connex on IBM installation. Provide full access to the user IDs that are doing the installation.

### **Installation JCL customization**

Some data sets cannot be allocated with secondary extents, such as LOADLIBs. The reason that many of the PDSs are not defined with secondary extents is because a running program will not be able to find members who are added or changed. These data sets must be defined as PDS rather than PDSE, because of the way the update process works.

MACLIB is not for the compiler. It contains panels, skeletons, and control cards.

### **Modify system parameters**

Two of the Connex on IBM system programs (CXBPOS54 and TCKSSS12) need APF authorization. The administrator must modify the PROGxx in the system parameter data set, adding Connex on IBM data sets and taking the system parameter into effect. The customer needs to issue D PROG,APF(LNKLST) to check whether the data set has been added successfully.

### **Logon procedure**

The customer should add a new TSO logon procedure or modify the old one. It must include the Connex on IBM data set and list LLQ name. We recommend that you create a new logon procedure for the Connex on IBM system. This new procedure needs to be defined in the RACF database.

### **Modify the startup procedure**

The customer may want to customize Connex on IBM startup procedure customization according to the installation plan. The Connex on IBM startup procedure needs to be defined in the RACF database, according to the security policy, with the necessary authorizations.

### **Application node definition in VTAM**

The Connex on IBM system must be able to access VTAM. We can use the samples in Appendix B in the *Connex on IBM Installation Guide*. We should activate the VTAM major node, test it, and add it to the VTAM startup to ensure that VTAM can automatically activate it on startup.

### **Modify Workload Manager policy**

The Connex on IBM system startup procedure must be defined in WLM policy, with velocity goals only for Connex on IBM started tasks. The velocity initialization value can start out at 50. The report class has to be set up for each Connex on IBM system monitoring tool. Tuning can be done on WLM by adjusting the velocity goals and importance value of service classes according to the customer environment requirements.

Figure 5-1 refers to the access security installation guide.

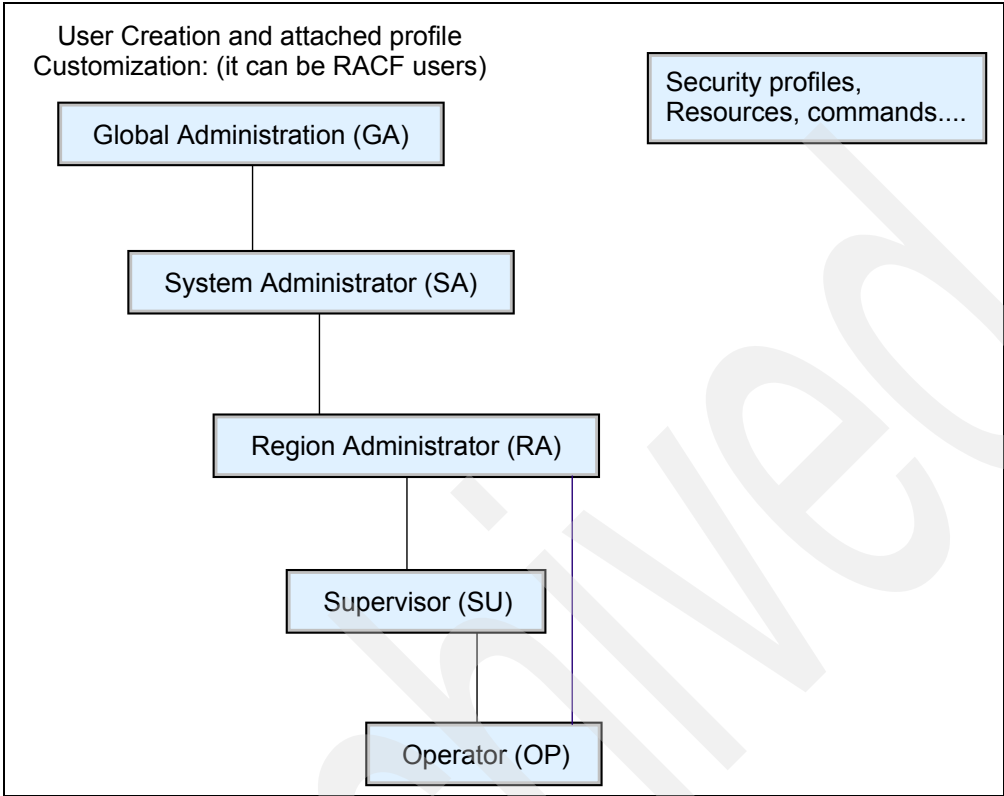


Figure 5-1 System architecture

### 5.1.5 Installation verification

After we run the Connex on IBM startup procedure, we verify the system logs and procedure outputs to check for error messages. In the sysout from the startup procedure, we look for messages indicating that the installation has completed successfully.

After Connex on IBM installation, we can see the following address spaces started in z/OS (Example 5-1).

Example 5-1 Connex on IBM jobs

---

|  |         |       |       |      |       |                |     |      |             |
|--|---------|-------|-------|------|-------|----------------|-----|------|-------------|
| Display Filter View Print Options Help |         |       |       |      |       |                |     |      |             |
| -----                                  |         |       |       |      |       |                |     |      |             |
| SDSF STATUS DISPLAY ALL CLASSES        |         |       |       |      |       | LINE 1-12 (12) |     |      |             |
| NP                                     | JOBNAME | JobID | Owner | PrtY | Queue | C              | Pos | SAff | ASys Status |

|                |          |        |    |           |      |      |
|----------------|----------|--------|----|-----------|------|------|
| Connex on IBMS | STC01244 | START2 | 15 | EXECUTION | PRD1 | PRD1 |
| Connex on IBMS | STC01245 | START2 | 15 | EXECUTION | PRD1 | PRD1 |
| Connex on IBMS | STC01246 | START2 | 15 | EXECUTION | PRD1 | PRD1 |

---

We find the following messages in the Connex on IBM address space (Example 5-2).

*Example 5-2 Connex on IBM startup messages*

---

```

---- TUESDAY, 11 DEC 2007 ----
IEF695I START Connex on IBMS WITH JOBNAME Connex on IBMS IS ASSIGNED TO USER
START2 , GROU
$HASP373 Connex on IBMS STARTED
I01 PMTASK PC100 12/11 08:26:07 STARTUP IN PROGRESS
...
I02 PMTASK PC119 12/11 08:26:55 TTN560B HAS STARTED
I02 A01 PC115 12/11 08:26:55 THD910B ATTACHED SUCCESSFULLY
I02 A01 PC115 12/11 08:26:55 THN560B ATTACHED SUCCESSFULLY
I02 THD910B OF000 12/11 08:26:56 THCS91B OPEN SUCCESSFUL
I02 THN560B OF000 12/11 08:26:56 THCSNCRB OPEN SUCCESSFUL
I01 PMTASK PC119 12/11 08:26:56 APIC01 HAS STARTED
I01 PMTASK PC119 12/11 08:26:56 APLB01 HAS STARTED
I01 PMTASK PC119 12/11 08:26:56 APCI01 HAS STARTED
I01 PMTASK PC119 12/11 08:26:56 CRQ01 HAS STARTED
I01 PMTASK NS000 12/11 08:26:56 **READY TO ACTIVATE COMMUNICATIONS
I01 PMTASK PC121 12/11 08:26:56 EDBTASK HAS BEEN SENT A QUIESCE MESSAGE
I01 PMTASK PC101 12/11 08:26:56 IMAGE I01 ACTIVE
I01 PMTASK PC125 12/11 08:26:56 EDBTASK HAS QUIESCED
I01 PMTASK PC141 12/11 08:26:56 EDBTASK HAS BEEN SENT A SHUTDOWN MESSAGE
I01 A01 PC145 12/11 08:26:56 EDBTASK HAS TERMINATED
I01 A01 PC172 12/11 08:26:56 EDBTASK WILL NOT BE RESTARTED
I01 HCI1001 TX021 12/11 08:26:56 THE TCP/IP API INTERFACE IS NOW ACTIVE
I01 HCI1000 TX021 12/11 08:26:56 THE TCP/IP API INTERFACE IS NOW ACTIVE
...

```

---

Message **IMAGE I01 ACTIVE** in Example 5-2 shows that the Connex on IBM address space has been started successfully. Also, we can check the Connex on IBM console, as follows:

1. Type your user ID and password into the screen to log into the console.
2. On the next screen (Example 5-3), select the Tranid CNXCON.

*Example 5-3 Selecting CNXCON*

---

```

12:47:43 Connex on IBM V3.4 - CNXV341 - Session Manager
Command ==>

```



|               | Tranid        |  | Status      | Started | Ended |
|---------------|---------------|--|-------------|---------|-------|
| 000001        | ACCESS        | ACCESS SECURITY DIALOG                   | IDLE        |         |       |
| <b>000002</b> | <b>CNXCON</b> | <b>Connex on IBM APPLICATION MONITOR</b> | <b>IDLE</b> |         |       |
| 000003        | CRS           | CRS TEST DIALOG                          | IDLE        |         |       |
| 000004        | C0000         | C0000 DIALOG                             | IDLE        |         |       |
| 000005        | DIAGFAC       | Diagnostic Facility Dialog               | IDLE        |         |       |
| 000006        | APM           | AP Maintenance Dialogue                  | IDLE        |         |       |
| 000007        | FTI           | FINANCIAL TRANSACTION INQUIRY            | IDLE        |         |       |
| 000008        | HEALTH        | SYSTEM HEALTH MONITOR DIALOG             | IDLE        |         |       |
| 000009        | NEGMAINT      | NEGATIVE FILE MAINTENANCE DIAL           | IDLE        |         |       |

Type M (Connex on IBM System Messages) and press Enter on the Main menu.  
See Example 5-4.

#### *Example 5-4 Connex on IBM console menu*

17:17:55 REDBOOK - Connex on IBM Console - Main Menu  
Command ==> M

Terminal: SCOTCP15

- M - MESSAGE** - Connex on IBM System Messages
- H - HIGH SEVERITY - Connex on IBM System High-Severity Messages
- A - ALARM - Connex on IBM System Alarms
- D - DOMAIN - Connex on IBM System Domain Counts
- C - CONSOLE - Connex on IBM System Console Settings
- S - SESSION - Connex on IBM System Active Operator Sessions

F1 =Help F3 =End F12=Recall

We must pay attention to any error messages that may appear. Example 5-5 shows an example of Connex on IBM system messages.

#### *Example 5-5 Startup messages*

17:27:01 REDBOOK - Connex on IBM Console - List Messages  
Command ==>

```

Img Entity   Time  Message Text
I01 PMTASK   17:26 AP01TSK IS STILL IN QUIESCE
I01 PMTASK   17:25 AP01TSK IS STILL IN QUIESCE
I01 SHMTASK   17:25 PR1003 ,STATUSPI: COMMAND ISSUED= STATUS *
CMD SHMTASK   17:25 STATUS PR1003
I01 SHMTASK   17:25 PR1002 ,STATUSPI: COMMAND ISSUED= STATUS *
CMD SHMTASK   17:25 STATUS PR1002
I01 PMTASK   17:24 AP01TSK IS STILL IN QUIESCE
I01 PMTASK   17:23 AP01TSK IS STILL IN QUIESCE
I01 PIA1002   17:21 NO RESPONSE TO A0200 FROM PII1003
I01 PIA1001   17:21 NO RESPONSE TO A0200 FROM PII1002
I01 PMTASK   17:21 AP01TSK IS STILL IN QUIESCE
I01 PIA1004   17:21 NO RESPONSE TO A0200 FROM PII1004
I01 PIA1000   17:21 NO RESPONSE TO A0200 FROM PII1004
I01 PIA1003   17:21 NO RESPONSE TO A0200 FROM PII1004
I01 PIA1002   17:20 NO RESPONSE TO A0200 FROM PII1001
I01 PIA1001   17:20 NO RESPONSE TO A0200 FROM PII1000

```

F1 =Help

F3 =End

F5 =Find

F7 =Backward F8 =Forward

F12=Recall

## 5.2 FIS DataNavigator

The next product that we cover is DataNavigator. In this section we discuss skills, hardware and software, preinstallation planning, installation, and verification.

### 5.2.1 Skills

For this project we need:

- ▶ z/OS system programmer
  - RACF Security administrator
  - VTAM administrator
  - WebSphere MQ administrator
- ▶ DB2 database administrator
- ▶ Storage management administrator
- ▶ Network administrator
- ▶ DataNavigator administrator (or advice from FIS consultant)
- ▶ ViewDirect archiving tool person

## 5.2.2 Prerequisite software and hardware

Review Chapter 2 in the FIS *DataNavigator Server Installation Guide*.

Although SMS volumes can be used to receive the product files, the installation process (and tuning parameters) dictate that the installer select the specific volumes rather than let SMS choose them. Some volumes (such as those containing log files) must be dedicated for performance reasons.

## 5.2.3 Preinstallation tasks for DataNavigator

These activities must be done before the installation:

- ▶ Fill out the site specification form that contains information unique to the customer's environment.
- ▶ Install client/server components for DB2 DRDA® remote access.

We highly recommend that an FIS expert be onsite to help install a test environment.

### ***DASD requirements***

Estimate how much data will be kept on DASD and how much business data there will be for online and batch jobs for two to three years. Plan the DASD layout for performance and availability. To estimate the DASD required, FIS can prepare a hardware sizing that will provide both DASD and CPU utilization estimates. See Chapter 2 of the *DataNavigator Server Installation Guide* for more instructions.

### ***Enterprise naming conventions***

During the DataNavigator installation process, enterprise naming conventions must be followed. This applies to system data sets, catalogs, databases, tablespaces, and indexspace names. Examples include LOADLIB, customization JCL, ISPF data sets, start procedures, and RACF-protected data set names.

- ▶ Security
  - Resources (data sets, DB2, ICSF, cryptographic keys, APF authorization)  
Make sure that the crypto processors are turned on and usable. Validate that the crypto HW is capable of meeting their needs.
  - Modify user logon procedures according to the sample in Appendix F of the installation guide.
  - Level of access for each operator (see Figure 5-1 on page 79).

- ▶ Network
  - VTAM definitions (applprod, appltest, applvd, and applvdm)
  - 3270-type terminals for operation consoles
- ▶ Subsystems
  - DB2
  - MQ configuration and sizing

## 5.2.4 Installation steps

In this section we discuss the installation steps.

### Sample JCL

To begin, we run the sample JCL in the install menu to unload the DataNavigator installation data sets and system data sets from the tape. The unloaded PDSs contain all JCL for DataNavigator system installation. We need to modify the JCL according the installation plan.

Note that if the RACF PROTECTALL option is active, we must define in advance all the data set profiles before doing the DataNavigator installation. Provide full access to the user IDs that are doing the installation.

### Set up DB2 subsystem ZPARMS and buffer pools

The ZPARMS values of your system should be determined by an evaluation of the environment and the transaction volume it is intended to support. Six buffer pools need to be configured in addition to the DB2 system default (BP0). More details on setting up those buffer pools can be found in the *DataNavigator Server Installation Guide*.

### Configuration repository (CR)

For the configuration repository:

1. Establish a logon procedure.
2. Create a CR IBM DB2 Database Storage Group.
3. Create a CR profile.
4. Update the CR control card.
5. Create CR base tables (total of 44 tables).
6. Create a DN CR table (total of 143 tables).
7. Bind the CR IBM DB2 Plan.
8. Install the CR/Editor GUI tools (IBM DB2 connect must be installed).

## Loading configuration tables

Follow the procedures described in the *DataNavigator Installation Guide* to populate both configuration repository tables and client configuration tables (used by DataNavigator clients Web and Delphi)

## Customer libraries and database

Allocate customer libraries and customize the JCL to create customer DB2 objects, such as storage groups, databases, tablespaces, tables, indexes, and plans. Some of the DataNavigator tablespaces are created using the DB2 table space compression feature, where a dictionary creation is vital. DB2 privileges must be granted as well. Procedures are described in the *DataNavigator Server Installation Guide*.

## Configure Mobius ViewDirect

ViewDirect is a z/OS product from Mobius Management Systems, Inc. It is a tool that allows you to move DataNavigator transactions to archive storage (such as tape), thereby reducing the amount of data on direct access storage. The product is distributed with DataNavigator and all installation instructions are in the *DataNavigator Server Installation Guide*. Installation includes:

- ▶ VTAM configuration
- ▶ JCL customization
- ▶ Configuration of the ViewDirect product.
- ▶ Adding data to the common database
- ▶ Data migration

## Configuration repository extract files

Generate the PDSs containing information derived from configuration repository and install them using configuration repository interface menus.

## Allocate server data sets

Some data sets are shared between multiple images in a Sysplex environment. Others should be allocated multiple times (once for each image).

## Set up input log files

This does not need to be done if you are using MQ and not batch feed. If needed, you can refer to the *DataNavigator Server Installation Guide*.

1. Set up procedures on SYS1.PROCLIB.
2. Set up startup and shutdown scripts.
3. Install the Delphi client on your Microsoft® Windows® PC.

## 5.2.5 Installation verification

After installing the DataNavigator system, we can see the following address spaces started in z/OS (Example 5-6).

*Example 5-6 DataNavigator jobs started*

---

|  |          |          |        |         |           |   |     |      |      |           |
|--|----------|----------|--------|---------|-----------|---|-----|------|------|-----------|
| Display Filter View Print Options Help |          |          |        |         |           |   |     |      |      |           |
| -----                                  |          |          |        |         |           |   |     |      |      |           |
| SDSF                                   | STATUS   | DISPLAY  | ALL    | CLASSES |           |   |     | DATA | SET  | DISPLAYED |
| NP                                     | JOBNAME  | JobID    | Owner  | PrtY    | Queue     | C | Pos | SAff | ASys | Status    |
|  | CNXPGM   | STC01155 | START2 | 15      | EXECUTION |   |     | PRD1 | PRD1 | ARMELEM   |
|  | CNXPDS   | STC01158 | START2 | 15      | EXECUTION |   |     | PRD1 | PRD1 |           |
|  | CNXPBU   | STC01157 | START2 | 15      | EXECUTION |   |     | PRD1 | PRD1 |           |
|  | CNXPFS   | STC01159 | START2 | 15      | EXECUTION |   |     | PRD1 | PRD1 |           |
|  | CNXPRBCI | STC01160 | START2 | 15      | EXECUTION |   |     | PRD1 | PRD1 |           |
|  | CNXPRBAE | STC01161 | START2 | 15      | EXECUTION |   |     | PRD1 | PRD1 |           |
|  | CNXPRBCU | STC01162 | START2 | 15      | EXECUTION |   |     | PRD1 | PRD1 |           |
|  | CNXPRBL1 | STC01163 | START2 | 15      | EXECUTION |   |     | PRD1 | PRD1 |           |
|  | CNXPRBL2 | STC01164 | START2 | 15      | EXECUTION |   |     | PRD1 | PRD1 |           |
|  | CNXPRBTA | STC01165 | START2 | 15      | EXECUTION |   |     | PRD1 | PRD1 |           |
|  | CNXV341  | STC01243 | START2 | 15      | EXECUTION |   |     | PRD1 | PRD1 | ARMELEM   |
|  | CNXV342  | STC01247 | START2 | 15      | EXECUTION |   |     | PRD2 | PRD2 | ARMELEM   |

---

We can see the messages shown in Example 5-7.

*Example 5-7 Startup messages*

---

|     |     |       |                |           |                                 |
|-----|-----|-------|----------------|-----------|---------------------------------|
| ... |     |       |                |           |                                 |
| I01 | PM  | PC119 | 12/10 08:55:29 | RBGM      | HAS STARTED                     |
| I01 | PM  | PC119 | 12/10 08:55:29 | RBAF      | HAS STARTED                     |
| I01 | A05 | PC115 | 12/10 08:55:29 | RBCH01    | ATTACHED SUCCESSFULLY           |
| I01 | PM  | PC121 | 12/10 08:55:29 | EDBTASK   | HAS BEEN SENT A QUIESCE MESSAGE |
| I01 | PM  | PC101 | 12/10 08:55:29 | IMAGE I01 | ACTIVE                          |
| ... |     |       |                |           |                                 |

---

Message IMAGE I01 ACTIVE shows that the DataNavigator address space has been started successfully.

## 5.2.6 FIS DataNavigator user interface

In this section we discuss the FIS DataNavigator user interface.

## Prerequisites

The DataNavigator server must be installed and accessible through the network.

## Assumptions

We assume that:

- ▶ WebSphere Application Server with Deployment manager is installed.
- ▶ UNIX System Services and RACF are correctly configured.
- ▶ You completed the FIS checklist questions for the server installation.

## Recommendations

We recommend that:

- ▶ Shared HFS should be used for high-availability concerns and ease of maintenance.
- ▶ zFS should be implemented for performance.
- ▶ WebSphere MultiNode architecture be set up for DataNavigator Web client resilience.

## Overview of steps

The steps are:

1. Deploy the DataNavigator Web client war file on an Application Server through the WAS Admin Console.
2. Configure the class loader to be *Application First* in **Detail Properties** → **Class Loading and Update Detection** and in the modules general properties *manage modules*.
3. Start the application and access the DN Web Client.

## 5.3 FIS Fraud Navigator

The next product that we cover is FIS Fraud Navigator.

### 5.3.1 Skills

For this product, you need:

- ▶ z/OS UNIX system administrator
- ▶ WebSphere system administrator
- ▶ RACF administrator
- ▶ Database administrator

### 5.3.2 Prerequisite software and hardware

Review the FIS *Fraud Navigator System Installation Guide*.

### 5.3.3 Installation steps

In this section we discuss the installation steps.

#### Assumptions

We assume that:

- ▶ z/OS UNIX and RACF are configured.
- ▶ A database administrator is available.
- ▶ You turned on Z/OS UNIX AUTOCVT for ASCII/EBCDIC automatic conversion.
- ▶ You completed the FIS checklist questions for the server installation.

#### Recommendations

We recommend that:

- ▶ zFS or shared HFS file systems be used for high-availability concerns and ease of maintenance.
- ▶ zFS should be implemented for performance.

#### Overview of steps

The steps are:

1. Save previous installation configuration files if available.
2. Create the file system structure (directories).
3. Unzip the tar files.
4. Tag .sh files as ASCII under USS.



5. Get all .sh files customized.
  - a. Set up key-based security and encryption.
  - b. Configure the database.
6. Load the database with the reload.sh file.
7. Start the node agent, monitor, logger, and UI process server with a single .sh file.
8. Deploy and configure four .war files on an application server through the WAS administrative console:
  - JSF UI (AlertWorkstation, and so on)
  - Domain Editor (business rule editor, and so on)
  - Reporting
  - Security
9. Start the user interface.

## Installation verification

Before starting FIS Fraud Navigator, we modified the jobname by exporting the variable `_BPX_JOBNAME='FN'`. This way we can set a WLM policy up for monitoring and workload management, and observe the processes under SDSF.

*Example 5-8 Modifying the jobnames to FN*

| Display Filter View Print Options Help |         |          |          |             |        |           |     |           |      |                |      |  |
|--|---------|----------|----------|-------------|--------|-----------|-----|-----------|------|----------------|------|--|
| -----                                  |         |          |          |             |        |           |     |           |      |                |      |  |
| SDSF                                   | DA      | PRD1     | (ALL)    | PAG         | 0      | CPU/L/Z   | 1/  | 1/        | 0    | LINE 1-10 (10) |      |  |
| NP                                     | JOBNAME | StepName | ProcStep | JobID       | Owner  | C         | Pos | DP        | Real | Paging         | SIO  |  |
|  | FN      | STEP1    |          | STC01456    | D94070 |           | LO  | FF        | 27T  | 0.00           | 0.00 |  |
|  | FN      | STEP1    |          | STC01429    | D94070 |           | LO  | FF        | 370  | 0.00           | 0.00 |  |
|  | FN      | STEP1    |          | STC01427    | D94070 |           | LO  | FF        | 356  | 0.00           | 0.00 |  |
|  | FN      | STEP1    |          | STC01424    | D94070 |           | IN  | F2        | 30T  | 0.00           | 0.00 |  |
|  | FN      | STEP1    |          | STC01454    | D94070 |           | LO  | FF        | 345  | 0.00           | 0.00 |  |
|  | FN      | STEP1    |          | STC01433    | D94070 |           | IN  | F2        | 31T  | 0.00           | 0.00 |  |
|  | FN      | STEP1    |          | STC01435    | D94070 |           | LO  | FF        | 12T  | 0.00           | 0.00 |  |
|  | FN      | STEP1    |          | STC01455    | D94070 |           | LO  | FF        | 343  | 0.00           | 0.00 |  |
|  | FN      | STEP1    |          | STC01437    | D94070 |           | IN  | F2        | 27T  | 0.00           | 0.00 |  |
|  | FN      | STEP1    |          | STC01441    | D94070 |           | LO  | FF        | 365  | 0.00           | 0.00 |  |
| COMMAND INPUT ==>                      |         |          |          |             |        |           |     |           |      |                |      |  |
|  |         |          |          |             |        |           |     |           |      | SCROLL ==> CSR |      |  |
| F1=HELP                                |         | F2=SPLIT |          | F3=END      |        | F4=RETURN |     | F5=IFIND  |      | F6=BOOK        |      |  |
| F7=UP                                  |         | F8=DOWN  |          | F9=SWAP LIS |        | F10=LEFT  |     | F11=RIGHT |      | F12=RETRIEVE   |      |  |

We issue a **ps** command within the OMVS shell to ensure that the five main processes are started (Example 5-9).

*Example 5-9 Checking that the processes are started*

---

```
{OMVSKERN@PRD1.FIS.com}/:  
::ps -ef | grep -i java  
OMVSKERN      65642   83951717 - 15:55:38 ttyp0000  0:59 /usr/lpp/java/J1.4/jre  
/bin/java -DLOGNAME=mon1 -Xdebug -Xrunjdp:transport=dt_  
OMVSKERN      33620075 16842857 - 15:56:37 ttyp0000  0:54 /usr/lpp/java/J1.4/jre  
/bin/java -DLOGNAME=na1 -Dport=7300 -Xdebug -Xrunjdp:tr  
OMVSKERN      33620080   65646 - 15:57:23 ttyp0000  0:56 /usr/lpp/java/J1.4/jre  
/bin/java -DLOGNAME=log -Xdebug -Xrunjdp:transport=dt_s  
OMVSKERN      50397297 16842863 - 15:57:23 ttyp0000  2:25 /usr/lpp/java/J1.4/jre  
/bin/java -DLOGNAME=aws -Xnoargsconversion -Dfile.encodi  
OMVSKERN      33620083   65650 - 15:57:25 ttyp0000  2:41 /usr/lpp/java/J1.4/jre  
/bin/java -DLOGNAME=fnapp1 -Xdebug -Xrunjdp:transport=d  
OMVSKERN      65655   50397278 - 16:02:35 ttyp0000  0:00 grep -i java
```

---

Example 5-10 shows the FIS Fraud Navigator startup messages on the OMVS console.

*Example 5-10 FIS Fraud Navigator startup messages*

---

```
set FN_HOME to /u/d94070/webdocs/apps/fn  
16:06:08,471 INFO - 16:06:08.229 main 484 0 RootElement loaded  
jar:file:/u/d94070/webdocs/apps/fn/builds/build-1.jar!/MONbase.xml  
16:06:08,519 INFO - 16:06:08.286 main 484 0 RootElement loaded  
jar:file:/u/d94070/webdocs/apps/fn/builds/build-1.jar!/NABase.xml  
16:06:09,221 INFO - 16:06:09.187 main 484 0 RootElement loaded  
jar:file:/u/d94070/webdocs/apps/fn/builds/build-1.jar!/FNSystem.xml  
16:06:09,260 INFO - 16:06:09.226 main 484 0 RootElement loaded  
jar:file:/u/d94070/webdocs/apps/fn/builds/build-1.jar!/FNSystem.xml  
16:06:13,349 INFO - 16:06:13.308 main 484 0 RootElement loaded  
jar:file:/u/d94070/webdocs/apps/fn/builds/build-1.jar!/Axis_Name_Service.xml  
16:06:13,969 INFO - 16:06:13.936 main 484 0 RootElement loaded  
jar:file:/u/d94070/webdocs/apps/fn/builds/build-1.jar!/Name_Service.xml  
16:06:20,016 INFO - 16:06:19.964 main 340 0 DBRE DBRE licensed for evaluation  
purposes only.  
16:06:20,060 INFO - 16:06:20.019 main 340 0 DBRE DBRE licensed for evaluation  
purposes only.  
16:06:20,154 INFO - 16:06:20.121 main 484 0 RootElement loaded  
jar:file:/u/d94070/webdocs/apps/fn/builds/build-1.jar!/Tools.xml  
16:06:20,206 INFO - 16:06:20.165 main 340 0 DBRE DBRE licensed for evaluation  
purposes only.
```

```

16:06:20,242 INFO - 16:06:20.200 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:20,247 INFO - 16:06:20.210 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:20,282 INFO - 16:06:20.245 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:20,361 INFO - 16:06:20.327 main 484 0 RootElement loaded
jar:file:/u/d94070/webdocs/apps/fn/builds/build-1.jar!/Tools.xml
16:06:20,416 INFO - 16:06:20.371 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:20,456 INFO - 16:06:20.420 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:24,899 INFO - 16:06:24.862 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:25,417 INFO - 16:06:25.372 main 340 0 DBRE Scheduler licensed for evaluation
purposes only.
16:06:25,458 INFO - 16:06:25.421 main 340 0 DBRE Scheduler licensed for evaluation
purposes only.
16:06:25,501 INFO - 16:06:25.464 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:25,556 INFO - 16:06:25.514 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:26,238 INFO - 16:06:26.204 main 484 0 RootElement loaded
jar:file:/u/d94070/webdocs/apps/fn/builds/build-1.jar!/Process_Monitor.xml
16:06:26,926 INFO - 16:06:26.571 jobs1 1580 0 MemoryCheck(memCheck) total 67566080
used 35646992 free 31919088 change 35646992
16:06:27,172 INFO - 16:06:27.127 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:27,222 INFO - 16:06:27.175 main 340 0 DBRE Scheduler licensed for evaluation
purposes only.
16:06:27,280 INFO - 16:06:27.242 main 340 0 DBRE Scheduler licensed for evaluation
purposes only.
16:06:27,332 INFO - 16:06:27.290 main 340 0 DBRE Scheduler licensed for evaluation
purposes only.
16:06:27,376 INFO - 16:06:27.338 main 340 0 DBRE Scheduler licensed for evaluation
purposes only.
16:06:27,422 INFO - 16:06:27.385 main 340 0 DBRE Scheduler licensed for evaluation
purposes only.
16:06:27,465 INFO - 16:06:27.428 main 340 0 DBRE Scheduler licensed for evaluation
purposes only.
16:06:27,564 INFO - 16:06:27.526 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:28,258 INFO - 16:06:28.220 main 5 0 Running MONbase -process=mon1
-category=mon -system=FNSystem -dump=0 -log=4 -javaLog=1 -env=0
-properties=fn.properties

```

```

16:06:29,345 INFO - 16:06:28.836 pm_workers1 201 0
RaiseEvent(Process_Monitor/logStart) user log
ProcessnamefnapplnodeFNPrimary0fnapplcom.FIS.oe.dbre.config.db.persistence.Entity0000
0.
16:06:31,323 INFO - 16:06:30.863 pm_workers4 201 0
RaiseEvent(Process_Monitor/logStart) user log
ProcessnameloglnodeFNPrimary0loglcom.FIS.oe.dbre.config.db.persistence.Entity00000.
16:06:32,194 INFO - 16:06:32.067 pm_workers3 201 0
RaiseEvent(Process_Monitor/logStart) user log
ProcessnamewslnodeFNPrimary0ws1com.FIS.oe.dbre.config.db.persistence.Entity00000.
16:06:32,447 INFO - 16:06:32.395 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:32,497 INFO - 16:06:32.457 main 340 0 DBRE Scheduler licensed for evaluation
purposes only.
16:06:32,538 INFO - 16:06:32.500 main 340 0 DBRE Scheduler licensed for evaluation
purposes only.
16:06:32,589 INFO - 16:06:32.544 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:32,664 INFO - 16:06:32.600 main 340 0 DBRE DBRE licensed for evaluation
purposes only.
16:06:32,710 INFO - 16:06:32.668 main 5 0 Running NABase -process=nal_PRD1
-category=na -system=FNSystem -dump=0 -log=4 -javaLog=1 -env=0
-properties=fn.properties
16:06:33,789 INFO - 16:06:33.621 jobs1 1580 0 MemoryCheck(memCheck) total 11467264
used 9943472 free 1523792 change 9943472
16:06:34,741 INFO - 16:06:34.670 ios4 714 0 Command
/u/d94070/webdocs/apps/fn/fnapp1/dbreStart.sh in /u/d94070/webdocs/apps/fn/fnapp1
16:06:38,024 WARN - 16:06:37.941 ios5 434 0 broadcast-100 thread no longer
associated with session mode 1
16:06:47,298 INFO - 16:06:47.261 ios5 714 0 Command
/u/d94070/webdocs/apps/fn/ws1/dbreStart.sh in /u/d94070/webdocs/apps/fn/ws1
16:06:51,426 INFO - 16:06:51.390 ios5 714 0 Command
/u/d94070/webdocs/apps/fn/log1/dbreStart.sh in /u/d94070/webdocs/apps/fn/log1
16:07:06,815 INFO - 16:07:06.781 jobs1 484 0 RootElement loaded
jar:file:/u/d94070/webdocs/apps/fn/builds/build-1.jar!/JMX_All.xml
16:11:25,642 INFO - 16:11:25.532 jobs1 1580 0 MemoryCheck(memCheck) total 67566080
used 19826464 free 47739616 change 19826464
16:11:32,732 INFO - 16:11:32.618 jobs1 1580 0 MemoryCheck(memCheck) total 13302272

```

---

## 5.4 FIS EnterpriseView

The final product that we cover is EnterpriseView.

## Skills

For this project you need:

- ▶ WebSphere system administrator
- ▶ Database administrator
- ▶ RACF administrator or someone with database administration authority

## Prerequisite software and hardware

Review the FIS *EnterpriseView System Installation Guide*.

The DataNavigator Server has to be installed with an online transaction database.

A DataNavigator user and a DB2 user must be able to access the transaction database directly.

### 5.4.1 Installation steps

In this section we discuss the installation steps.

#### Assumptions

We assume that:

- ▶ WebSphere Application Server with Deployment manager is installed.
- ▶ z/OS UNIX and RACF are configured.
- ▶ A database administrator is available.
- ▶ You completed the FIS checklist questions for the server installation.

#### Recommendations

We recommend that:

- ▶ zFS or shared HFS file systems be used for high availability concerns and ease of maintenance.
- ▶ zFS should be implemented for performance.
- ▶ WebSphere Cluster might be set up for EnterpriseView Web client resilience.
- ▶ JDBC™ Type 4 driver is recommended if you plan to connect to DB2 v8.

#### Overview of steps

The steps are:

1. Deploy the EnterpriseView war file on an Application Server through the WAS Admin Console.

2. Configure the class loader to be *Application First* in **detail properties** → **Class Loading and update detection** menu and in the modules general properties *manage modules*.
3. Customize the DataNavigator.properties and database.properties file.
4. Start the application and access the EnterpriseView Web client.

## 5.4.2 Installation verification

After installing EnterpriseView, use the following URL to access it:

<http://172.28.86.206:9548/managementPortlets/html/start.faces>

The IP address is your mainframe home IP (or VIPA) address. The port is your definition in the WebSphere configuration. We can see the Web interface in Figure 5-2.

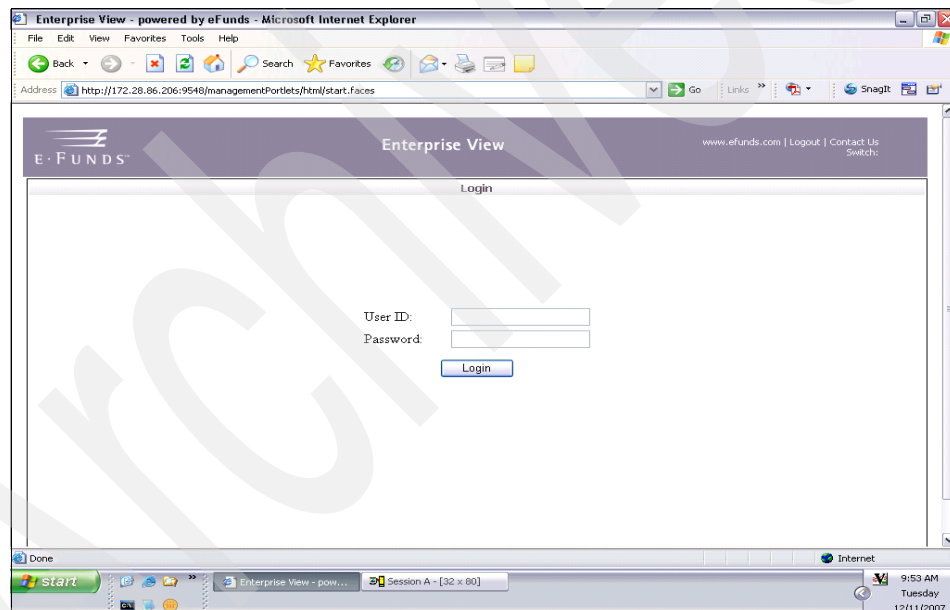


Figure 5-2 EnterpriseView Web interface

The Web interface indicates that you have installed EnterpriseView successfully.

## 5.5 Overall solution verification

This section demonstrates all the components working together, simulating transactions to be processed by the FIS overall solution. CrossCheck, another FIS product, is the simulation software that generated the transactions during our verification section.

We show both DataNavigator Delphi and Web user interfaces that enable us to check that the transactions were processed properly.

### 5.5.1 DataNavigator Delphi interface

When CrossCheck has simulated transactions and sent data to be processed, the next step is to check whether the transactions have been stored on DataNavigator. Figure 5-3 shows the DataNavigator Delphi User Interface logon screen.

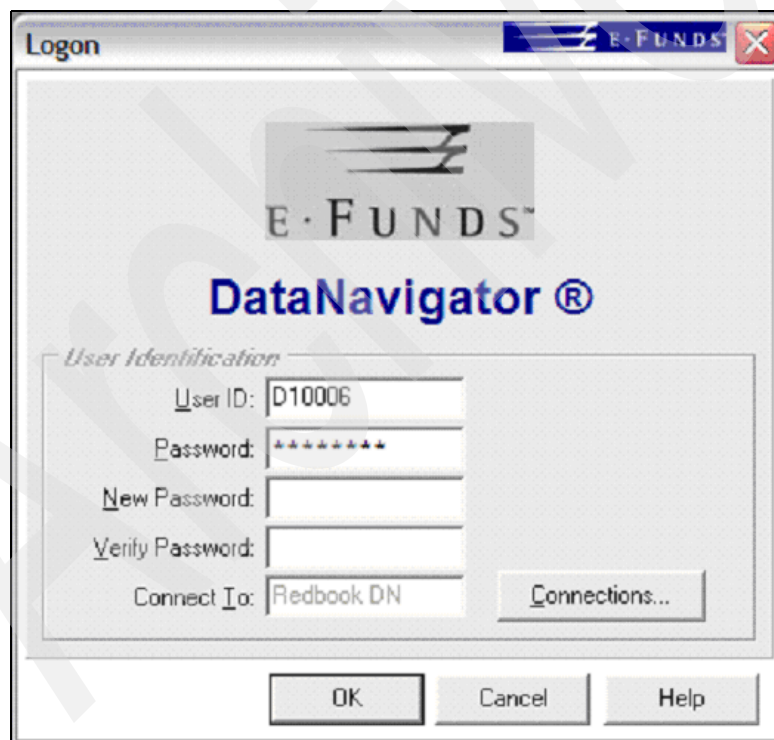


Figure 5-3 Delphi client logon screen

The DataNavigator Delphi client main screen is shown in Figure 5-4.

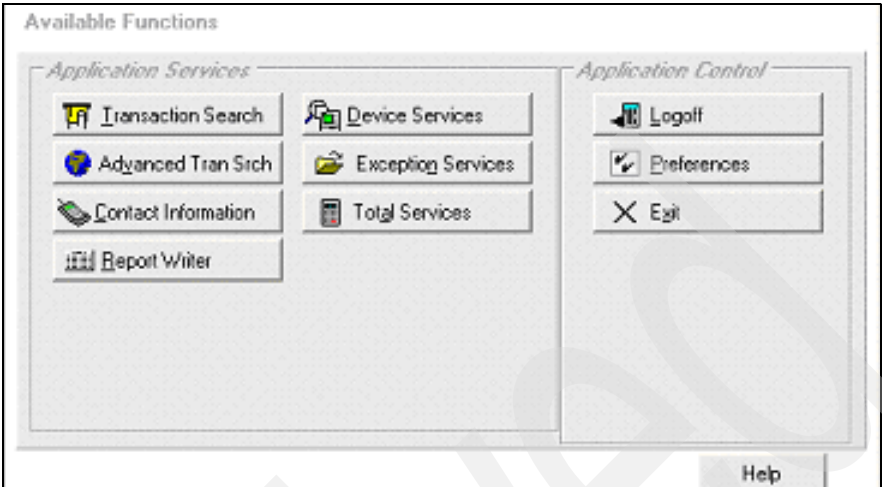
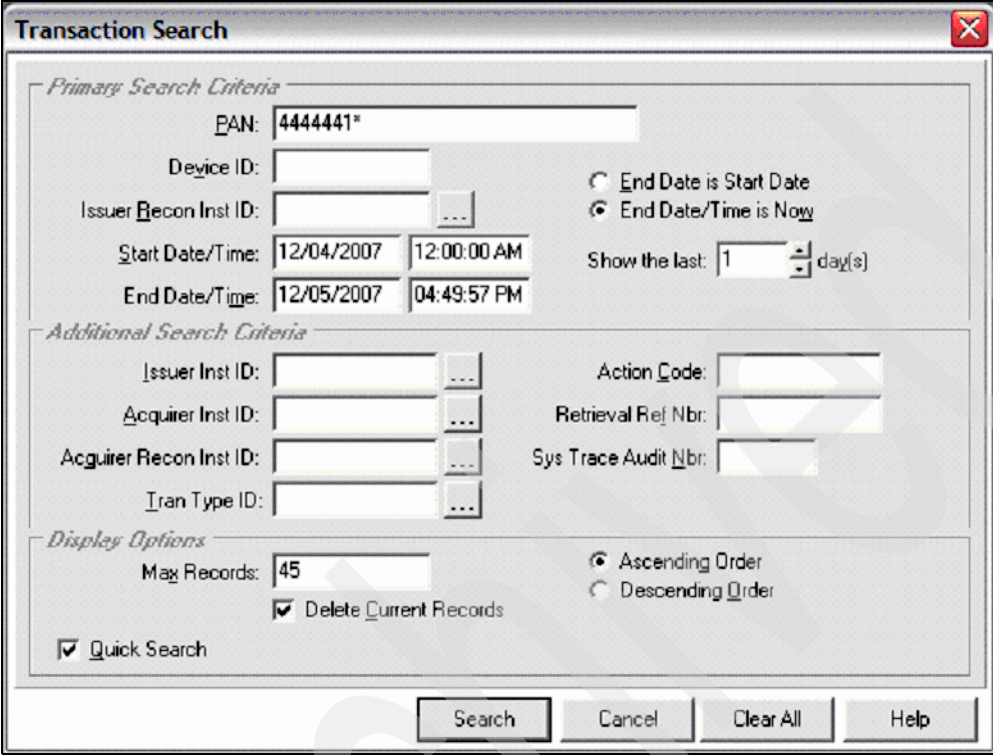


Figure 5-4 Delphi client



Selecting **Transaction Search** on the screen points you to the DataNavigator Delphi client research transaction criteria. Its screen is shown in Figure 5-5.



The **Transaction Search** dialog box is divided into three main sections: *Primary Search Criteria*, *Additional Search Criteria*, and *Display Options*.

**Primary Search Criteria:**

- PAN:** 4444441\*
- Device ID:** [Empty]
- Issuer Recon Inst ID:** [Empty]
- Start Date/Time:** 12/04/2007 12:00:00 AM
- End Date/Time:** 12/05/2007 04:49:57 PM
- End Date is Start Date:** ☐
- End Date/Time is Now:** ☒
- Show the last:** 1 day(s)

**Additional Search Criteria:**

- Issuer Inst ID:** [Empty]
- Acquirer Inst ID:** [Empty]
- Acquirer Recon Inst ID:** [Empty]
- Iran Type ID:** [Empty]
- Action Code:** [Empty]
- Retrieval Ref Nbr:** [Empty]
- Sys Trace Audit Nbr:** [Empty]

**Display Options:**

- Max Records:** 45
- Ascending Order:** ☒
- Descending Order:** ☐
- ☒ **Delete Current Records**
- ☒ **Quick Search**

Buttons at the bottom: **Search**, **Cancel**, **Clear All**, **Help**.

Figure 5-5 Entering search criteria

After filling in the search criteria fields, click **Search**. This shows the results based on the previous search criteria. See Figure 5-6.

**DataNavigator® Application [Redbook DN] - [Transaction List]**

File Edit Options Functions Window Help

**Selected Criteria**  
 Start: 12/04/2007 12:00:00 AM  
 End: 12/05/2007 04:51:43 PM

**Disposition**  
☒ Approvals ☒ Advices  
☒ Tran w/ Case ☒ Fraud Advices  
☒ Denials ☒ Still Except  
☒ Reversals ☒ Archived  
☒ Suspects

**Changed Selection Criteria**  
 PAN: 4444441000000109  
 Device ID: IA1003  
 Issuer Recon Inst ID: 000000013  
 Acq Recon Inst ID: 000000013  
☐ 1 ☐ 4  
☐ 2  
☐ 3  
 < Add these to search criteria >

| Type                                | New                                 | Hide                                | PAN               | Device ID | Tran Date  | Tran Time  | Net Recon Amt w/Fee | Action Cd |
|-------------------------------------|-------------------------------------|-------------------------------------|-------------------|-----------|------------|------------|---------------------|-----------|
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 4444441000000109  | IA1003    | 12/04/2007 | 02:48:53PM | \$ 25.00            | 000       |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 4444441000000104  | IA1002    | 12/04/2007 | 02:48:53PM | \$ 25.00            | 000       |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 44444410000001316 | IA1003    | 12/04/2007 | 02:48:54PM | \$ 25.00            | 000       |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 44444410000001316 | IA1003    | 12/04/2007 | 02:48:54PM | \$ 25.00            | 000       |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 44444410000001297 | IA1000    | 12/04/2007 | 02:48:54PM | \$ 25.00            | 000       |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 44444410000001310 | IA1002    | 12/04/2007 | 02:48:54PM | \$ 25.00            | 000       |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 44444410000003734 | IA1002    | 12/04/2007 | 02:48:55PM | \$ 25.00            | 000       |

45 Records Retrieved  
☒ Show Hidden Rows  
☐ Show All Late Response Reversals

Export Grid Print Grid Customize Grid

Search Again Retrieve More Close Details Help

View all transactions or select one for details

Figure 5-6 Search results

## 5.5.2 DataNavigator Web interface

DataNavigator has a Web interface, also, with the same functions of the Delphi client. The Web interface is becoming increasingly popular. The DataNavigator Web Interface logon screen is shown in Figure 5-7.

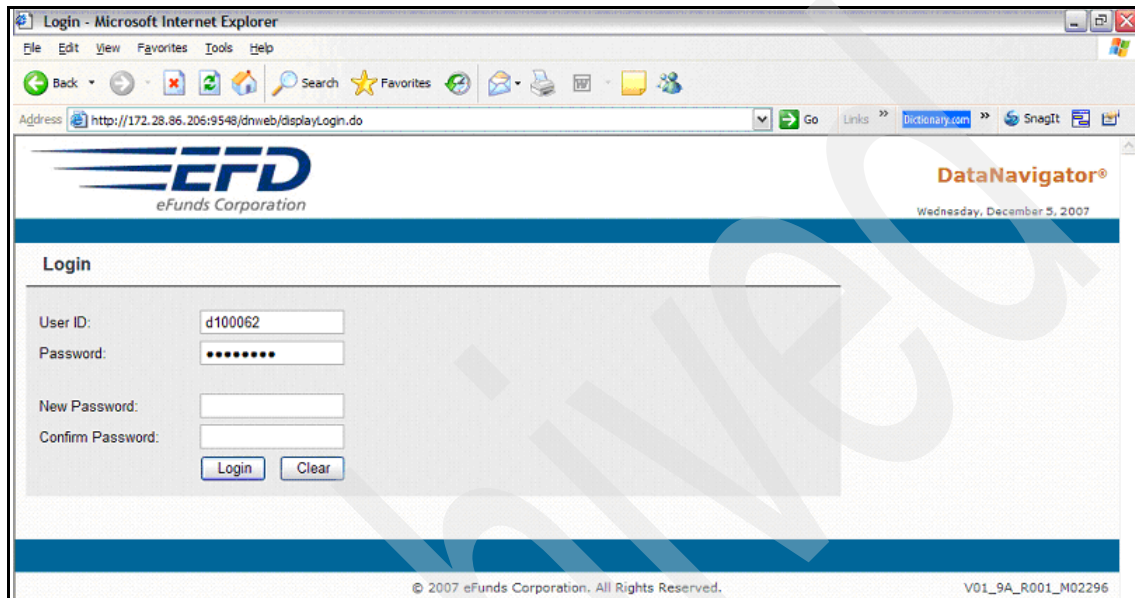


Figure 5-7 Login screen

After logging into the system, we are on the client main page. See Figure 5-8.

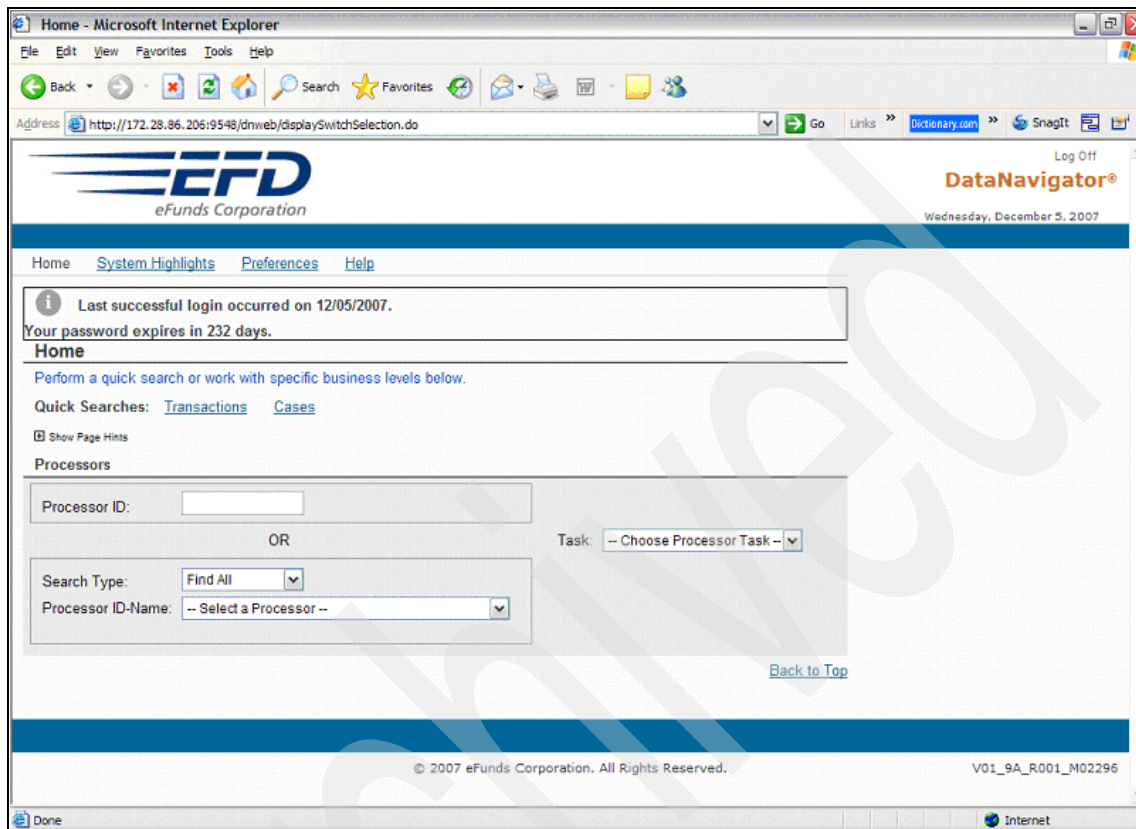


Figure 5-8 Client home page

Next, click **Transaction link** and you will see Figure 5-9.

Transaction Search - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://172.28.86.206:9548/dnweb/transaction/displayTransactionSearch.do> Go Links Dictionary.com Snagit

**EFD**  
eFunds Corporation

Log Off  
**DataNavigator®**  
Wednesday, December 5, 2007

[Home](#) [System Highlights](#) [Help](#)

### Transaction Search

**Specify Switch Date Range**

Start Date/Time: 12/04/2007 00:00:00  
End Date/Time: 12/05/2007 23:59:59  
☐ Show newest first ☒ Show oldest first

**Choose search data**

At least one of the following fields is required to search.  
Add an asterisk (\*) after partial PAN entries:

PAN: 4444441\* (5 digit min) [Add >>](#)

Device ID:  [Find](#)

Acq Recon Inst ID:  [Find](#)

Iss Recon Inst ID:  [Find](#)

Merch Rptng Level:  [Find](#)

Acq Network ID:  [List](#)

Iss Network ID:  [List](#)

[Show additional search data](#)

**Search on this**

[Search](#) [Remove](#) [Clear All](#)

[Search](#) [Remove](#) [Clear All](#)

Internet

Figure 5-9 Searching transactions



After filling in the search criteria fields, click **Search**. This shows the results based on the previous search criteria. See Figure 5-10.

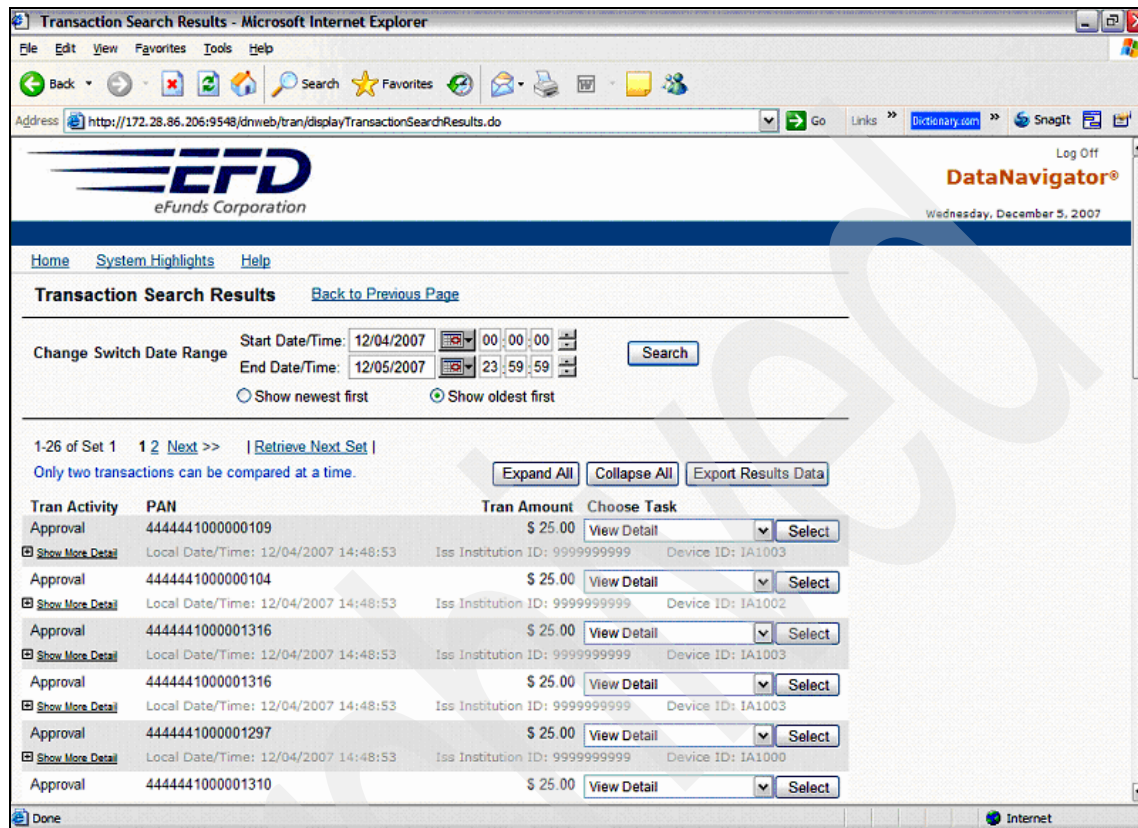


Figure 5-10 Search results

## FIS Fraud Navigator

At this point we have FIS Fraud Navigator processing the transactions generated during Connex on IBM online processing (currently these are in a logfile). The next step is to launch the Enterprise Control tool to look inside.

## EnterpriseView

Figure 5-11 shows the EnterpriseView main screen, which contains useful information such as ATM status, acquirer/issuer activity, and settlement position. See 2.4.4, “FIS EnterpriseView” on page 30, for more details on benefits and functions.



Figure 5-11 EnterpriseView main screen

## 5.6 Onsite customization of all products

Here are some tips when customizing each product.

### 5.6.1 Connex on IBM

Sample operations JCL is included in the FIS-supplied tape. This JCL includes jobs to operate your test and production Connex on IBM systems. Certain values

in the operations JCL are different for the test and production Connex on IBM Systems. Verify that all the operations JCL is in the supplied user.cntl data set, as follows:

- ▶ **CONNEX:** This job runs the primary address space for the online Connex on IBM System in a single image. The image is specified in the PARM option on the EXEC statement as I=I01 for image one (the default), I=I02, I=I03, and so on, for subsequent images. For PARM='I=I01' and subsequent images, copy the JCL but give a different PARM to indicate which image it is. The job name must be unique. You cannot submit the same job name with a different image.
- ▶ **CONNEXS:** This job runs the secondary address spaces for online Connex on IBM Systems.
- ▶ **LOGUPRI:** This job unloads the primary Connex on IBM System logs into a disk data set named LOGUNLDP. The Connex on IBM System automatically submits this job when a primary log becomes full. However, it also submits this job when a user enters the CUTLOG command from the Connex on IBM console.

**Note:** In a production Connex on IBM System, remove PARM=GETALL from the EXEC statement of step CXBPLG10. This ensures that the active log is not included in the unload process when the switch is down.

- ▶ **LOGUSEC:** This job unloads the secondary Connex on IBM System logs into a disk data set named LOGUNLDS. The Connex on IBM System automatically submits this job when a secondary log becomes full. However, it also submits this job when a user enters a CUTLOG command from the Connex on IBM Console.
- ▶ The LOGUSEC job that the Connex on IBM Installation Facility creates unloads all logs, including the active log. This is fine for the installation and test Connex on IBM systems, but not for a production Connex on IBM system. See the note above for LOGUPRI.
- ▶ **SETLPRI:** This job does the same thing as the LOGUPRI job. However, it also submits the SETTLE job for the settlement processing. The Connex on IBM system automatically submits the SETLPRI job when an end-of-day (EOD) event occurs. It also submits this job when a user enters an EOD command from the Connex on IBM console. See the note for LOGUPRI.
- ▶ **SETLSEC:** This job only creates a secondary log of transactions for use if the primary log is destroyed. The Connex on IBM system automatically submits this job when an end-of-day (EOD) event occurs. It also submits this job when a user enters an EOD command from the Connex on IBM console. See the note for LOGUPRI.



- ▶ **SETTLE:** The SETLPRI job submits this job to begin end-of-day settlement processing. The SETTLE job can use FIS settlement software or other settlement software. If you are using FIS settlement software, the SETTLE job does the following:
  - It creates the standard input file for the FIS Connex on IBM Management System.
  - It produces an audit trail report of the contents of the Connex on IBM log.
  - It creates a Connex on IBM error report.
  - It creates a Connex on IBM electronic mail report.
  - It creates a NETSTAT audit trail report.
  - It creates the NETSTAT ATM service report.
  - It creates IBM ELARG error reports, if they apply to your environment.
  - It updates the online history logs, if you are using the Connex on IBM Online History optional component *Netstat Online History* or *Financial Transaction Inquiry*.
  - It creates all settlement reports.
- ▶ **TRACE:** The Connex on IBM system automatically submits this job each time a Connex on IBM trace file fills up. It also submits this job when a user enters a RESET TRACE command on the Connex on IBM console. This job unloads the current trace file and prints its content. The online TRACE task also begins writing information to a new trace file. This protects you from reusing a trace file before you review the captured trace data. TRACE is standard name of FIS Corporation for the associated Connex on IBM system task.
- ▶ **MULTIPLE IMAGES:** The supplied JCL is set up to run on a single image. Before changing it for multiple images, you must initialize all the image-specific files, such as log and traces, using the Data Set Manager option in the configuration repository (CR) Batch Facilities. To change the JCL for multiple images, copy the JCL and change the EXEC parm from I=I01 to I=I02, to I=I03, and so on, for the number of images you are running.

## **Certify system generation**

You can now run the online Connex on IBM system. You may test it using the FIS CrossCheck Integrated Test Facility, or another testing utility, to certify that the system has been generated correctly.

## Review the installation output

You can review the following reports that were produced from running various jobs, as follows:

- ▶ The switch review transaction form of an audit trail that lists C0000 transactions. These are the transactions that cannot be processed by the online system. Each record contains some header information and the external messages format that is received. In production, this report should be reviewed daily to detect any developing problems in your network.
- ▶ The reports from SETLPRI, SETLSEC (UNLDPRI, UNLDSEC) provide the I/O tuning opportunity. They provide the statistics on busy counts for each of the logging data areas and for each of the logs that were unloaded. The log allocation can be adjusted through data placement on a faster disk drive, for example, and through definition of additional log data areas to produce acceptable busy and wait counts.
- ▶ The settlement reports for your system are described in the *Data Preparation and Daily Reporting Manual*.

See *Connex on IBM System Installation Guide* for more information.

## 5.6.2 DataNavigator

The install and customization of the DataNavigator system requires the participation of various experts. The roles and responsibilities are as follows:

### Database Administrator (DBA)

The DBA:

- ▶ Determines the size of objects in the DataNavigator database
- ▶ Creates databases and corresponding storage groups
- ▶ Compresses tablespaces
- ▶ Loads data into tables, and bind plans
- ▶ Sets up security for IBM DB2 tables
- ▶ Reviews DDLs to be used to establish new tables

### Technical support

For technical support:

1. Create the LOGON procedure for the configuration repository.
2. Establish APF authorization for DataNavigator code.
3. Set up application program nodes in VTAM.
4. Add DataNavigator server and ViewDirect procedures to the system PROCLIB.

5. Install WebSphere MQ and create a queue for the data on both the DataNavigator platform and EFT switches.

**Note:** DataNavigator and ViewDirect tasks must run in the same LPAR.

## Applications

For the applications:

- ▶ Restore installation software and create libraries.
- ▶ Perform customization of pre-generated JCL.
- ▶ Configure DataNavigator components, such as ViewDirect and configuration repository.
- ▶ Generate and install extract files for the DataNavigator Server.
- ▶ Set up startup and shutdown scripts for the DataNavigator Server.
- ▶ Start DataNavigator.

## PC support

For PC support:

- ▶ Install CR/editor on designated workstations.
- ▶ Install Web server.
- ▶ Install the standard client on designated workstations.

## Security

For security:

- ▶ Set up Access Security for the DataNavigator Console.
- ▶ Add entity IDs and user IDs to the configuration repository.

See *DataNavigator Server Installation Guide* for more information.

### 5.6.3 FIS Fraud Navigator

The installation and customization of FIS Fraud Navigator is described in the installation and customization manual. In general, your steps are as follows:

1. Before installation, be sure that your system meets all the system prerequisites.
2. If this is not your first install (for example, distribution of an update or release), do not assume that everything is unchanged. You need to carefully follow each step in the process.

3. Ensure that you have allocated sufficient DASD to all database and non-database files.
4. Examine our database DDL. Modify the DDL as required for performance. For example, you should not be changing our table structure or names, but may want to put these tables into your own table spaces. This tuning process is unique by installation.
5. Whenever possible, use system defaults (for example, ports). This will make things easier when you need to upgrade or apply maintenance.
6. Have the proper expertise available during installation. The process goes much smoother if changes can be made quickly (for example, not waiting for overnight turnaround). You need experts in:
  - DBA
  - Network security
  - UNIX (UNIX System Services)
  - WebSphere
  - Security
7. In general, follow these steps:
  - a. Expand the server portion of the system (untar).
  - b. Configure the server:
    - Ports
    - Encryption type
    - Database connections and access
    - Switch type
    - Mode of operation
    - Connections to switch (if applicable)
    - Connections to DataDistributor (if applicable)
    - Connections to HSMs (if applicable)
    - Connections to e-mail mail server
  - c. Web server deployment/configuration of .war files
    - Connections to other Web server components
    - Connections to server components
    - Connections to user authentication components
  - d. Ensure that the FIS Fraud Navigator files have the necessary access rights.
  - e. Ensure that the user ID being used to launch FIS Fraud Navigator has the necessary access rights.
  - f. Ensure that your network security implementation does not block traffic between components.
  - g. Set up default user IDs in your authentication system.

## 5.6.4 EnterpriseView

The installation and customization of the EnterpriseView requires the participation of the various expertise. The major steps for the installation are:

1. Check the system's pre-request and verify the system status.
2. Finish the software test and certification.
3. Finish the client compatibility test.
4. Install the application in the customer system.
5. Configure application setup options, such as configuring database properties and DataNavigator properties.
6. Configure DataNavigator, such as configuring CNDG, CDNP, and security setup.

## 5.7 Maintenance

Here we provide some tips on adding maintenance or feature upgrades.

### 5.7.1 Connex on IBM

Normally, all software changes should be installed while the Connex on IBM System is shut down. This ensures that all code changes are properly applied. However, when using multiple imaging, it is not necessary for all images to be shut down in order to install software changes. The changes can be installed one image at a time with the others still running. In emergency situations, small changes can be installed while the Connex on IBM application is running. This is done by refreshing all applications that use the programs being installed.

A typical installation procedure can be described as follows:

1. Identify all applications of a particular type that use the programs being installed.
2. Install the changed programs on the appropriate system load library.
3. Refresh all of the applications that use the installed programs. This must be done with a "SHUTDOWN type=xxx" command on the Connex on IBM Console, where xxx is the type of application being corrected. Once all tasks have been shut down, an "ATTACH type=xxx" command should be entered on the Connex on IBM Console. This restarts the processes.

For more information about the Connex on IBM Console commands, refer to the *Connex on IBM Console User's Guide*.

**Note:** Do not use the REFRESH command for this function since it causes unpredictable results.

Although this method works in most cases, there are some limitations when using a single image:

- ▶ It cannot be used to install changes to the base system (Process Monitor or any of the low-level Connex on IBM software functions). These programs do not change very often, so this restriction does not usually apply.
- ▶ All applications that use the same programs must be down at the same time for the new programs to be reloaded from disk. This is a restriction imposed by MVS, not by the FIS product.

**Note:** When using multiple imaging, it is not necessary for all images to be shut down in order to install maintenance on one image. This can be done one image at a time with the others still running.

## 5.7.2 DataNavigator

Normally, all software changes should be installed while the DataNavigator System is shut down. This ensures that all code changes are properly applied. In emergency situations, small changes can be installed while the DataNavigator application is running. This is done by refreshing all applications that use the programs being installed.

A typical installation procedure can be described as follows:

1. Identify all applications of a particular type that use the programs being installed.
2. Install the changed programs on the appropriate system load library.
3. Refresh all of the applications that use the installed programs. This can be done using “SHUTDOWN processname” and “START processname” or “REFRESH processname,FULL”.

For the database maintenance procedures refer to *DataNavigator Server Installation Guide*.

### 5.7.3 FIS Fraud Navigator

If you are running a single node FIS Fraud Navigator system, software changes should be installed while the system is down. This ensures that all code changes are properly applied.

- ▶ If new .war files are to be deployed, you only need to be concerned with the Web server. There is no impact on FIS Fraud Navigator's ability to process transactions.
- ▶ If new server code is being deployed, the server is all that you need to be concerned with.
  - If running FIS Fraud Navigator in online mode, the switch will continue to authorize transactions without the benefit of a fraud check. When the interface is restored, the application will update its statistics (for any missed transactions) and immediately begin to validate new transactions.
  - If running in any other mode (for example, post-auth, batch), there is no impact to transaction processing.

Here is a typical installation procedure summarized:

1. Stop the component (for example, server code, UI application).
2. Apply the maintenance.
3. Restart the component.

If you are running a multiple node implementation you can often install the changes one node at a time. This would allow you to stop only one node. While the changes are being applied, the other nodes continue to process normally. Workload is shifted as necessary. When the changes have been applied, you can restart the node. This process is repeated for each node in the system.

### 5.7.4 EnterpriseView

For any FIS product installation or upgrade, the FIS company will provide a detailed guide. For EnterpriseView maintenance, we follow these four steps:

1. Back up all related software and configuration according to our configuration.
2. Deploy the war file into a special directory according to the FIS upgrade guide.
3. Log on to EnterpriseView to test our installation.
4. Recover all data to previous versions (including programs and the configuration) when necessary.

Archived





## Monitoring

There are two ways to monitor the status of an FIS application system. The first one is to use FIS production internal tools, such as System Health Monitor (SHM). The second one is to monitor the system using IBM monitoring tools, such as Resource Measurement Facility (RMF).

# 6.1 FIS monitoring tools

FIS application monitoring tools are designed for monitoring FIS internal data processes, transactions, and statistics. These tools can detect the system health status, such as ATM, POS, and other application node connection failures, authorization rejects, and build failure in the system. The operator can manage the system status and use monitored information to find problems.

Here are some monitoring tools for the FIS products:

- ▶ System Health Monitor (SHM) on Connex on IBM
- ▶ SHM on DataNavigator
- ▶ Enterprise Control (EC) In FIS Fraud Navigator
- ▶ EnterpriseView (EV) on DataNavigator

Figure 6-1 shows a sample system status from SHM.

11:29:53 System Health Monitor - Monitor All Domains  
Command ==>

| Member Name . . . . .   |                           | (optional generic key) |         |         |          |
|---|---------------------------|------------------------|---------|---------|----------|
| Row   | Description               | APPROVED               | DENIED  | BAD PIN | REVERSAL |
| 000001  | ACQUIRING PI OR TH        | 12,770                 | 123,708 | 0       | 3,170    |
| 000002  | ACQUIRER                  | 0                      | 0       | 0       | 0        |
| 000003  | AP AND AS TASKS           | 12,770                 | 123,708 | 0       | 3,170    |
| 000004  | AP TRANSACTION OVER \$500 | 0                      | 0       | 0       | 0        |
| 000005  | ATM DEVICES               | 12,770                 | 123,708 | 0       | 3,170    |
| 000006  | ISSUER                    | 12,770                 | 123,708 | 0       | 3,170    |
| 000007  | POS DEVICES               | 0                      | 0       | 0       | 0        |
| 000008  | SYSTEM AS A WHOLE         | 12,770                 | 123,853 | 0       | 3,170    |
| F1 =Help      F3 =End      F7 =Up      F8 =Down      F9 =Totals |                           |                        |         |         |          |

Figure 6-1 Status screen from System Health Monitor

## 6.1.1 System Health Monitor in Connex on IBM

The System Health Monitor (SHM) in Connex on IBM is designed to help maintain the health status of the Connex on IBM system. SHM can provide automatic monitoring, reporting, and action for problems.

The input to SHM is the messages logged by Connex on IBM. When we understand these messages, we can better operate the application. SHM processes message according to rules that we store in a database and customize according to our requirements. SHM processes messages and provides valuable information to operators. FIS provides a sample database that has rules predefined by experts.

SHM monitors real-time application system information based on:

- ▶ A set of configurable selection criteria
- ▶ Information from the Connex on IBM system log
- ▶ Information from other Connex on IBM system components

Additional benefits from using SHM are:

- ▶ SHM can provide automatic operation for the Connex on IBM system and network. This can increase Connex on IBM system and network system availability.
- ▶ SHM can automatically take action based on defined criteria, which can decrease manual intervention.
- ▶ SHM can generate transaction statistics information, which can help manage the IT system and help make business decisions.
- ▶ SHM keeps default selection criteria in its database. These can be customized in the database and dynamically activated.
- ▶ SHM provides a consolidated online interface to maintain its database, to monitor system statistics reports, and to review action.
- ▶ SHM has adapted the Connex on IBM access security facility to protect resources.

Figure 6-2 shows the main menu for the System Health Monitor.

```
11:27:35 System Health Monitor - Main Menu
Command ==>

    D - DOMAIN      - Show Domains
    B - BUCKET      - Show Statistical Buckets
    X - TRANSACTION - Show Transaction Traces
    S - SELECTION   - Show Selections
    F - FIELDS      - Show Message Fields
    C - CALENDAR    - Show Calendars

    T - TRIGGER     - Show Threshold Triggers
    AL - ACTION LIST - Show Action Lists
    A - ACTION      - Show Actions
    M - MODULE      - Show Modules

    E - EXCEPTION   - Show Exceptions
    W - WATCH       - Set Watch Options
    PP - PULSE PROF - Maintain Pulse Profiles

    H - HEALTH      - Real Time Monitor
    MP - PULSE      - Real Time Monitor Pulse
    P - PROBLEMS    - List Active Triggers

F1 =Help    F3 =End
```

Figure 6-2 SHM main menu

### 6.1.2 System Health Monitor in DataNavigator

This monitor is for long-term business analysis that results in system capability planning, including capacity planning and performance tuning.

Some features are different between SHM in Connex on IBM and SHM in DataNavigator, but most functions are similar. The SHM in Connex on IBM focuses mainly on real-time monitoring for transactions. It shows what is happening in the system. The System Health Monitor in DataNavigator can likewise monitor real-time transaction processing, but the tools mainly focus on statistical reports for business analysis.

### 6.1.3 Enterprise Control

Enterprise Control (EC) in FIS Fraud Navigator has a Web interface to monitor the FIS Fraud Navigator system activity. It provides statistical reports to manage the system and find potential problems. Figure 6-3 shows the Enterprise Control interface.

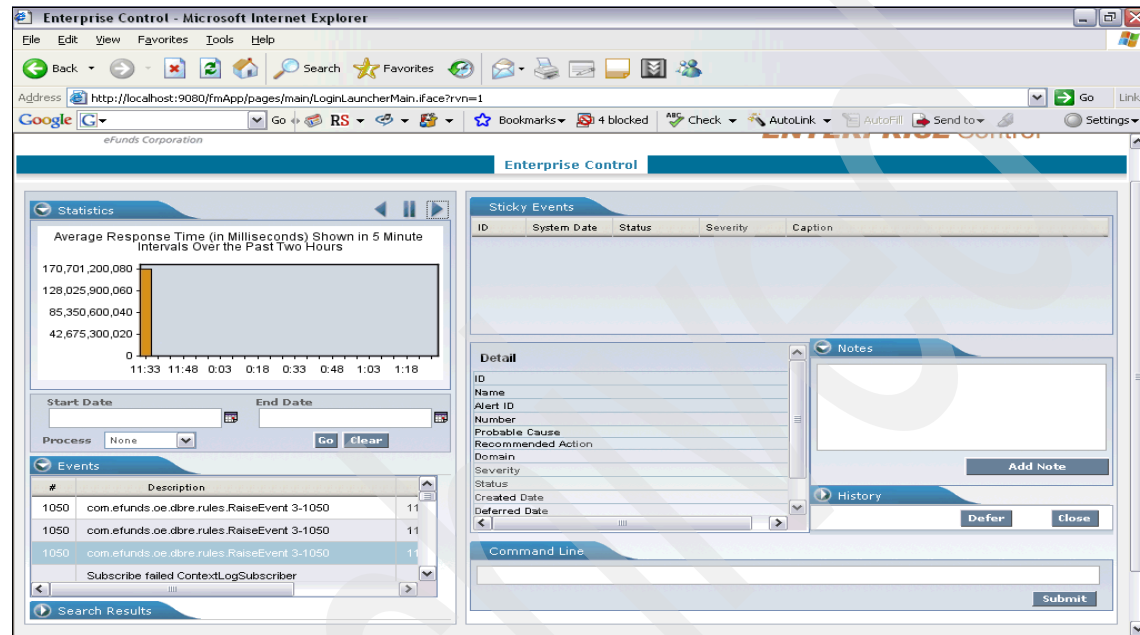


Figure 6-3 Enterprise Control Web interface

Enterprise Control:

- ▶ Can monitor transaction response times. It can isolate system bottlenecks and provide information to help with system tuning and capacity planning.
- ▶ Can check system events and ensure that a problem has been solved.
- ▶ Provides a search engine to check whether the same problem has happened in the past and help find the solution.
- ▶ Can issue system commands to stop and start various functions in the system.

### 6.1.4 EnterpriseView

There are two tools for monitoring the status of DataNavigator. One is SHM and the other is EnterpriseView. EV is a Web interface monitoring tool that is easy to

understand without training. Although it can provide real-time transaction processing, EV focuses mainly on statistical reporting for business analysis and enterprise management. It can connect to multiple platforms through the same user interface.

EV can summarize the enterprise status for management. It can collect information from different systems and display the information in one or more views for different management objectives.

- ▶ EnterpriseView can provide the health status of DataNavigator and find potential problems.
- ▶ EnterpriseView can filter valuable information for ease of access and can decrease the back-office management costs and resources.
- ▶ EnterpriseView has the option of different logon profiles for user access control.

## 6.2 Resource Measurement Facility

Resource Measurement Facility (RMF) is an IBM product for z/OS performance monitoring, measurement, and management. RMF is the base product to collect system and subsystem performance data under z/OS and Parallel Sysplex environments. It allows us to tune and configure a system according to our business requirements. RMF-II and RMF-III are add-on products.

For FIS product monitoring, we can configure IBM tools such as RMF to monitor the address space status. Thus, we can obtain information for application system tuning.

We can obtain the following reports from RMF-II:

- ▶ Address spaces
- ▶ I/O queuing, devices, channels, and HFS
- ▶ Enqueuing, storage, SRM, and program libraries

We can obtain the following reports from RMF-III:

- ▶ Sysplex
- ▶ WFEX, SYSINFO, and detail
- ▶ Information about job delays
- ▶ Processor, device, enqueue, and storage
- ▶ Subsystem information for HSM, JES, and XCF

For more information about RMF monitoring tools, see *z/OS V1R9.0 RMF User's Guide*, SC33-7990-13.

Figure 6-4 shows an RMF example of DASD tuning.

| RMF - DEV Device Activity |     |                        |      |      |      |       |      |             |        |      |      |      | Line 1 of 1     |    |
|---------------------------|-----|------------------------|------|------|------|-------|------|-------------|--------|------|------|------|-----------------|----|
| CPU= 3/ 3 UIC= 65K PR= 0  |     |                        |      |      |      |       |      |             |        |      |      |      | System= PRD1 To |    |
| 5:39:03 I=90% DEV         |     | ACTV RESP IOSQ -DELAY- |      | PEND | DISC | CONN  | %D   | %D          |        |      |      |      |                 |    |
| FG                        | GRP | VOLSER                 | NUM  | PAV  | LCU  | RATE  | TIME | TIME        | CMR DB | TIME | TIME | TIME | UT              | RV |
|                           |     | CNX300                 | 8400 |      | 0029 | 0.001 | 10.3 | 0.0 0.1 0.0 | 0.1    | 0.0  | 10.2 | 0    | 0               |    |
|                           |     | DNDB10                 | 81A0 |      | 0026 | 0.001 | 9.9  | 0.0 0.1 0.0 | 0.1    | 0.0  | 9.9  | 0    | 0               |    |
|                           |     | SYSDAA                 | 871E |      | 002C | 0.022 | 7.2  | 0.0 0.1 0.0 | 0.2    | 0.0  | 7.1  | 0    | 0               |    |
|                           |     | DNDB0C                 | 80AC |      | 0025 | 12.50 | 3.0  | 1.3 0.2 0.0 | 0.3    | 0.2  | 1.5  | 2    | 0               |    |
|                           |     | WASP00                 | 86A0 |      | 002B | 0.060 | 2.6  | 0.0 0.1 0.0 | 0.2    | 0.0  | 2.4  | 0    | 0               |    |
|                           |     | S8RES3                 | 8003 |      | 0025 | 0.454 | 2.5  | 0.0 0.1 0.0 | 0.2    | 0.0  | 2.4  | 0    | 0               |    |
|                           |     | DNDB08                 | 80A8 |      | 0025 | 0.411 | 1.9  | 0.0 0.2 0.0 | 0.3    | 0.1  | 1.6  | 0    | 0               |    |
|                           |     | DNDB07                 | 80A7 |      | 0025 | 0.713 | 1.8  | 0.0 0.1 0.0 | 0.3    | 0.0  | 1.5  | 0    | 0               |    |
|                           |     | DNDB01                 | 80A1 |      | 0025 | 0.275 | 1.7  | 0.0 0.2 0.0 | 0.3    | 0.1  | 1.3  | 0    | 0               |    |
|                           |     | DNDB03                 | 80A3 |      | 0025 | 0.779 | 1.4  | 0.0 0.1 0.0 | 0.3    | 0.0  | 1.2  | 0    | 0               |    |
|                           |     | DCXCF2                 | 8100 |      | 0026 | 2.536 | 1.4  | 0.0 0.1 0.0 | 0.2    | 0.0  | 1.2  | 0    | 0               |    |
|                           |     | DCXCF3                 | 8101 |      | 0026 | 0.110 | 1.4  | 0.0 0.1 0.0 | 0.2    | 0.0  | 1.2  | 0    | 0               |    |
|                           |     | S8RES1                 | 8001 |      | 0025 | 8.273 | 1.2  | 0.0 0.1 0.0 | 0.2    | 0.0  | 1.0  | 1    | 0               |    |
|                           |     | DCXCF1                 | 8013 |      | 0025 | 1.349 | 1.1  | 0.0 0.1 0.0 | 0.2    | 0.0  | 0.9  | 0    | 0               |    |
|                           |     | DNDB0F                 | 80AF |      | 0025 | 0.001 | 1.1  | 0.0 0.4 0.0 | 0.6    | 0.0  | 0.5  | 0    | 0               |    |
|                           |     | CNX100                 | 8200 |      | 0027 | 0.908 | 1.0  | 0.0 0.1 0.0 | 0.2    | 0.0  | 0.8  | 0    | 0               |    |
|                           |     | DNDB00                 | 80A0 |      | 0025 | 0.018 | 1.0  | 0.0 0.2 0.0 | 0.3    | 0.1  | 0.7  | 0    | 0               |    |
|                           |     | CNX203                 | 8306 |      | 0028 | 1.468 | 0.8  | 0.0 0.1 0.0 | 0.2    | 0.0  | 0.6  | 0    | 0               |    |
|                           |     | S8SYS2                 | 830F |      | 0028 | 0.139 | 0.8  | 0.0 0.1 0.0 | 0.2    | 0.2  | 0.4  | 0    | 0               |    |
|                           |     | CNX202                 | 8305 |      | 0028 | 1.110 | 0.8  | 0.0 0.1 0.0 | 0.2    | 0.0  | 0.6  | 0    | 0               |    |
|                           |     | S8RES2                 | 8002 |      | 0025 | 6.286 | 0.8  | 0.0 0.1 0.0 | 0.2    | 0.0  | 0.6  | 0    | 0               |    |
|                           |     | USS102                 | 8312 |      | 0028 | 0.119 | 0.8  | 0.0 0.1 0.0 | 0.2    | 0.0  | 0.6  | 0    | 0               |    |
| ommand ===>               |     |                        |      |      |      |       |      |             |        |      |      |      | Scroll ===> PA  |    |

Figure 6-4 RMF report on DASD status

We see that a DASD volume named DB2100 has a long response time. We need to list the data sets in the volume to find the root cause. These are listed in Example 6-1.

#### Example 6-1 Data sets on volume DB2100

```
CSQ600.CSQA.BSDS01.DATA
CSQ600.CSQA.BSDS01.INDEX
CSQ600.CSQA.LOGCOPY1.DS02.DATA
CSQ600.CSQA.LOGCOPY1.DS04.DATA
CSQ600.CSQA.LOGCOPY2.DS02.DATA
CSQ600.CSQA.LOGCOPY2.DS04.DATA
CSQ600.CSQB.LOGCOPY1.DS02.DATA
CSQ600.CSQB.LOGCOPY1.DS04.DATA
CSQ600.CSQB.LOGCOPY2.DS02.DATA
CSQ600.CSQB.LOGCOPY2.DS04.DATA
DSN810.DB8A.BSDS01.DATA
DSN810.DB8A.BSDS01.INDEX
DSN810.DB8A.LOGCOPY2.DS01.DATA
```

```
DSN810.DB8A.LOGCOPY2.DS02.DATA
DSN810.DB8A.LOGCOPY2.DS03.DATA
DSN810.DB8B.BSDS01.DATA
DSN810.DB8B.BSDS01.INDEX
DSN810.DB8B.LOGCOPY2.DS01.DATA
DSN810.DB8B.LOGCOPY2.DS02.DATA
DSN810.DB8B.LOGCOPY2.DS03.DATA
DSN810.DSNDBD.DB8AWRK.DSN32K01.I0001.A001
DSN810.DSNDBD.DB8AWRK.DSN32K02.I0001.A001
DSN810.DSNDBD.DB8AWRK.DSN32K03.I0001.A001
DSN810.DSNDBD.DB8AWRK.DSN4K01.I0001.A001
DSN810.DSNDBD.DB8AWRK.DSN4K02.I0001.A001
```

---

We can see that bootstrap data sets, active log, work data sets, and MQ bootstrap data sets and log reside on the same volume. These data sets could be heavily used (thus, the root cause of the contention). We should allocate these data sets onto other volumes or different control units to avoid the resource contention.



# Optimization and best practices

The objective of this chapter is to provide tips and techniques for availability and performance.

## 7.1 Throughput

For tips on capacity planning, see Appendix A, “Capacity planning tips” on page 165.

## 7.2 zAAP and FIS Fraud Navigator

System z Application Assist Processor (zAAP) is a special use feature on System z. For Java users, it provides strategic, performance, security, and economic reasons to exploit its benefits. It is designed to process code asynchronously with general CPs. A zAAP reduces the overall cost of code processing, since IBM does not charge for the workload sent to a zAAP. Only Java workloads can be processed on a zAAP. See Figure 7-1 on page 123.

**Note:** There are some hardware and software requirements to implement a zAAP. See *zSeries Application Assist Processor (zAAP) Implementation*, SG24-6386, for more detailed information.

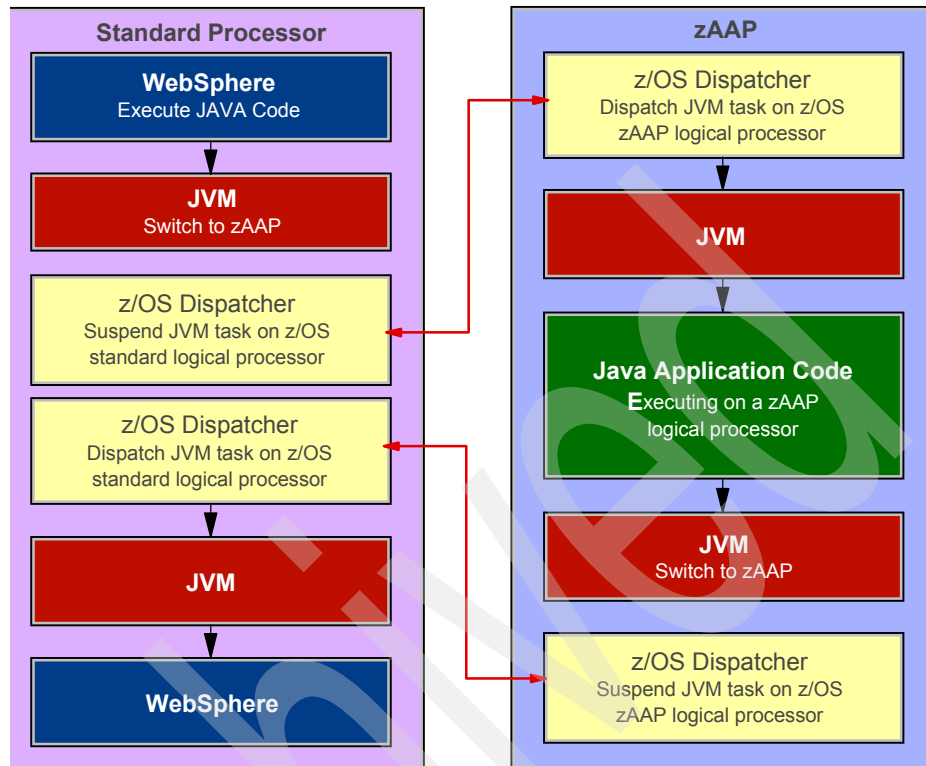


Figure 7-1 zAAP workloads

FIS Fraud Navigator is built on Java technology, so it can take advantage of using a zAAP to process its workloads.

For more information about IBM zAAP technology, refer to:

<http://www.ibm.com/systems/z/zaap/>

## 7.3 Connex on IBM best practices

Here are some recommendations. See Connex on IBM installation documentation for more information.

- A Connex on IBM user should have three system environments available. The test system is used for the installation of Connex on IBM. This environment can be retained or deleted as needed. The second system is called user acceptance testing (UAT). This can be used for client-determined

non-production purposes. Thirdly, the *prod* system is the production environment.

- ▶ Additional system environments can be defined as required. For example, you may require a Certification Connex on IBM System or another test system.
- ▶ Follow the data set naming conventions indicated in the Connex on IBM installation document. This will make the processes of software installation and problem determination easier.
- ▶ Allocate dedicated DASD volumes for the Connex on IBM logs, as well as for the checkpoint and store and forward data sets. These allocations impact the performance of the Connex on IBM system if they are shared.
- ▶ For a high volume and high throughput production environment, configure more than two sets of logs in the configuration repository for Connex on IBM.
- ▶ Define more than seven days of history log data sets, depending on how far back you will need to see the old processed transactions.
- ▶ Use the started task procedure (instead of batch JCL) to bring up the production Connex on IBM environment, since it is not dependent on initiators.
- ▶ The Coupling Facility should be configured to allow a high throughput of messages between Connex on IBM images in separate LPARs on a Parallel Sysplex.

## 7.4 Tuning tips and techniques

Performance problems are complex. The administrator needs to fully understand how these components interact: network, Parallel Sysplex, z/OS system, subsystem, and application system. There are tools to monitor each component in order to understand their activity and find the resource contention.

### 7.4.1 DASD

Often performance problems are related to DASD read/write times. In order to decrease I/O time and increase the transaction rate, monitor DASD response time using RMF. If there are long response times that need tuning, here are some tips:

- ▶ For ISQ waits, add more PAVs to the system or z/OS images.
- ▶ For channel busy and channel utility > 70%, add more channels to the LPAR or use faster channels.

- ▶ If there is high activity on a single DASD volume, separate the files onto different volumes.
- ▶ For a low cache hit ratio, move the database to a different logical control unit (LCU).
- ▶ For heavily used VSAM files, stripe the files to different DASD volumes.

## 7.4.2 DB2

For tuning a database, many tools are available, including OmegaMon. The DB2 Performance Expert provides the following information:

- ▶ Separate buffer pools for different tablespaces and indexspaces for a higher DB2 buffer hit ratio.
- ▶ Buffer pool tuning: Increase the buffer hit ratio as much as possible with limited storage.
- ▶ Lock tuning: For parallel processing, decrease the DB2 lock level and commit as early as possible.
- ▶ Create a suitable index to avoid a table scan or an index scan.
- ▶ Tune SQL statements to fit more indexes. This can decrease data search time.
- ▶ The DB2 utilities LOAD and REORG are used to build the dictionary from sample data. However, building a new dictionary frequently is a load on the system. Instead, use the KEEPDICTIONARY keyword to reuse the old dictionary.

## 7.4.3 VSAM

For VSAM:

- ▶ For better catalog performance in a Parallel Sysplex environment, enable catalog sharing.
- ▶ For FIS production, VSAM will cache the record into memory and batch writes to DASD asynchronously, so there is no need to decrease the CI size for online transactions.
- ▶ Monitor CA/CI splits for VSAM files. Define more free space for high frequency splits, or **repro** the VSAM files more frequently to reorganize them.
- ▶ Define large CI and CA sizes in VSAM for batch processing performance.

## 7.4.4 Workload Manager

For address space and I/O tuning, monitor the system status using WLM reports. Since there are many FIS address spaces in z/OS, our recommendations are:

- ▶ FIS products can take advantage of multiple processors in the system, allowing the system to run more transactions as needed. Define enough CPs for the LPARs in which the FIS application system is running to sustain the desired response time.
- ▶ For FIS production applications, define a separate service class for each of the address spaces and assign an appropriate velocity, as discussed below.
- ▶ Define a different goal for each address space, according to the importance of each application.
- ▶ Generate an RMF WLM report for FIS applications, and tune the FIS application velocity using the WLM report.
- ▶ For decisions on velocity and importance, also use the SHM reports.
- ▶ For the Connex on IBM application, define a higher service class because it needs fast online response. Other applications, such as DataNavigator, can be defined in a lower service class and tuned accordingly.

Figure 7-2 shows a different service class defined for FIS applications. In the RMF report, we can see that the current WLM policy will meet our target.

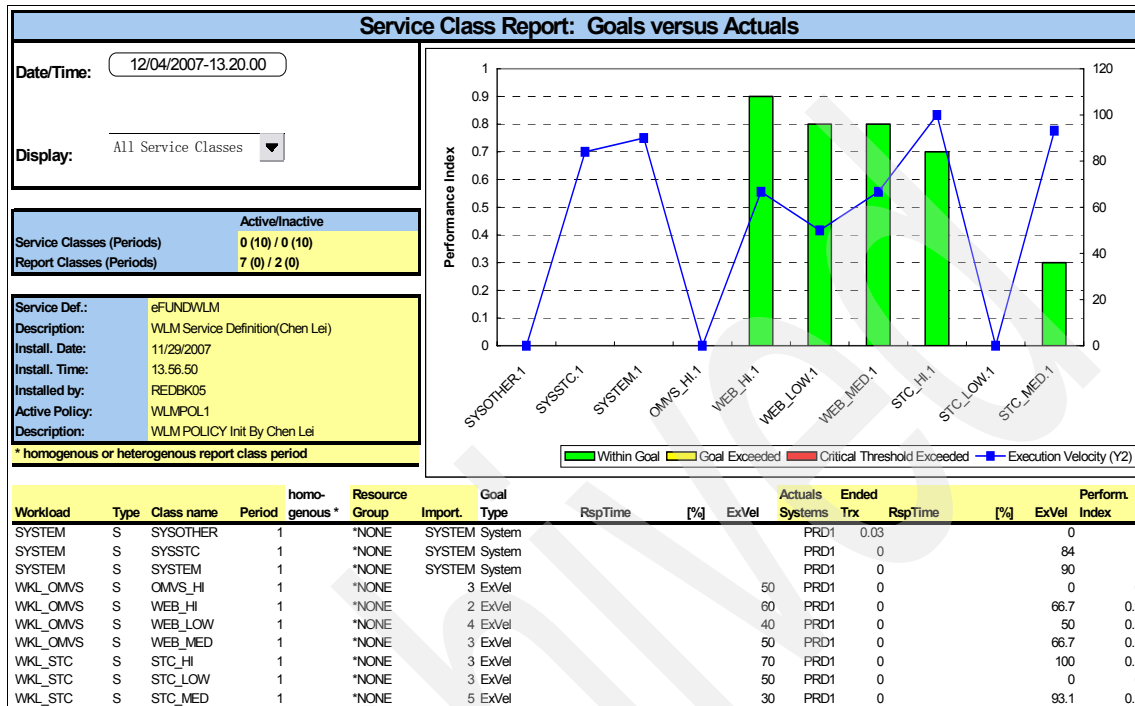


Figure 7-2 Sample RMF report of FIS applications

Archived





## Problem determination on z/OS

The objective of this chapter is to provide tips and tools to help in determining the cause of problems.

## 8.1 Performance troubleshooting tips

In order to troubleshoot performance issues, system resources must be constantly monitored and controlled. These include:

- ▶ Processors (CP capacity and utilization)
- ▶ I/O (delays and utilization)
- ▶ Network traffic
- ▶ Storage (particularly paging rates and amounts)
- ▶ DB2
- ▶ Coupling Facility
- ▶ VSAM

Keep these system components in mind as we proceed through this chapter.

## 8.2 FIS component troubleshooting

As we review the FIS components in this chapter, we consider symptoms, analyze possible causes and consequences, and provide tips about actions that may be taken.

### 8.2.1 Connex on IBM: processes to be monitored

In this section we discuss processes to be monitored.

#### **Logger**

For Logger:

- ▶ Queue depth increases.
  - User response time increases (it can be cash withdrawal transactions).
  - Time-outs implying transaction cancellation.
  - Automatic shutdown.

- Possible causes

A system performance problem or Connex on IBM tuning issue is increasing transaction count. The logger needs tuning to support the new workload.

- Actions

Application tuning:

- Clone the logger process.

- Add logging files and new volumes.
- ▶ Disk failure
  - No consequences
  - Actions
    - Possibly feed the secondary logger into settlement.

## **Communication handlers**

Nothing specific can be described from a performance point of view.

## **Authorization processor**

This handles I/O errors and hardware failures.

Queue depth increases or AP response time increases on certain transactions

- ▶ Consequences
  - User response time increases.
  - Customer transaction deny due to time-outs.
- ▶ Possible causes
  - DB2 access
  - General system performance
- ▶ Actions
  - DB2 tuning
  - Application tuning:
    - Clone the AP process.
    - Database restructuring (for example, split table)

## **Terminal handler and processor interface**

Queue depth increases or TH response time increases on certain transactions.

- ▶ Consequences
  - User response time increases.
  - Customer transaction deny due to time-outs.
- ▶ Possible causes
  - Checkpoint file too busy
  - General system performance

- ▶ Actions
  - Application tuning
  - Cloning the processes

## 8.2.2 DataNavigator

For DataNavigator:

- ▶ WebSphere MQ queue depth increases.
  - Consequences
    - Delay before the transaction is viewable.
    - EnterpriseView has stale data.
  - Possible causes:
    - DB2 access or MQ performance issue
    - General system performance
  - Actions
    - DB2 and MQ tuning.
    - Application tuning.
    - Add another load queue in MQ and a queue reader in DataNavigator.
    - Database restructuring (splitting tables).
- ▶ User interface response time
  - Causes
    - Slow DB2 response time.
    - DataNavigator Query Engine or CI task might be busy.
    - WAS is overloaded.
  - Actions
    - DB2 or WAS tuning
    - Clone either the query engine or the CI task.

### 8.2.3 EnterpriseView

User interface response time:

- ▶ Causes
  - Slow DB2 or DataNavigator response time
  - WAS overloaded
- ▶ Actions
  - Tune DataNavigator, DB2, or WAS

### 8.2.4 FIS Fraud Navigator

For FIS Fraud Navigator:

- ▶ Slow response time to Connex on IBM
  - Consequences
    - Slow user response time
  - Possible causes
    - Out of threads
  - Actions:
    - Add threads or processes spread over several partitions.
- ▶ User interface response time
  - Causes
    - Slow DB2 response time
    - WAS is overloaded
    - CPU dispatching priority
  - Actions
    - DB2 or WAS tuning
    - Adding UI processes or move it to another machine (or both)

## 8.2.5 Causes for application delays

Possible causes are:

- ▶ CPU: dispatching priority, multitasking, capacity
- ▶ I/O: channels (capacity, distance), CU, PAVs, priority, device contention (at any level), VSAM, CF overload
- ▶ Memory: high paging rate

System actions:

- ▶ Raise WLM service class priorities or LPAR priorities.
- ▶ Defer discretionary workloads until capacity is available.
- ▶ Add more CPC capacity.
- ▶ DASD tuning (devices, control units, paths, file systems).

## 8.3 FIS debugging features

This section mentions the trace and dump tools. Refer to FIS's *Connex on IBM Console User's Guide* for additional details.

### 8.3.1 Traces

The tracing facility, one of the FIS PD functions, is used to assist in the determination of system problems. The information contains a record of significant events. This trace function is enabled through the a configuration repository Task option or by entering the TRACE command on the Connex on IBM operator console.

- ▶ To turn on tracing for a specific Processor Interface (PI) task, enter:  
TRACE Processname, ON
- ▶ To turn off tracing for a specific Processor Interface task, enter:  
TRACE Processname, OFF

### 8.3.2 Dumps

For debugging purposes, choose the ABEND (SNAP) dump option in the Process Monitor CR extract. This option provides immediate access to the dump in IOF while the test is running. Each SNAP dump taken during a test is

generated to a unique SYSOUT DDNAME (CXDUMP00, CXDUMP01, and so on).

- ▶ To turn on dump for a specific processor interface task, enter:  
DUMP Processname, ON
- ▶ To turn off dump for a specific processor interface task, enter:  
DUMP Processname, OFF

Archived





## Part 3

# Project details

This part describes our tests for high availability. They are tests that the reader can run to verify that the software that is installed.

Archived

## High availability tests

This chapter documents our test scenarios that demonstrate continuous availability of the FIS Enterprise Payment system applications in an IBM System z environment. First we describe the objective, the injection procedures, and the expected behavior for each scenario. Then we discuss the test that we performed against each scenario and detail the actual behavior.

The chapter covers the following topics:

- ▶ Test environment topology
- ▶ Unplanned outages
  - Scenario 1: canceling a non-critical FIS task
  - Scenario 2: canceling a critical task
  - Scenario 3: failure of a Connex on IBM image
  - Scenario 4: z/OS LPAR failover
- ▶ Planned outages
  - Scenario 5: applying IBM maintenance
  - Scenario 6: applying an upgrade to the FIS application system
- ▶ High availability configurations

## 9.1 Introduction

The topic of high availability is of prime importance in today's online transaction processing systems. Extensive improvements in modern hardware reliability have made fault tolerance less of an issue than in the past. The loss of data in the communications networks far exceeds the loss of information in the processing system itself, and end-to-end protocols designed to recover from loss of data in the communications networks also help ensure financial integrity during infrequent hardware failures. However, contractual obligations and user requirements continue to make high availability vitally important.

Our FIS test configuration was designed with the following architectural principles:

- ▶ Software is generally considered less reliable than hardware. The System z hardware contains all redundant components, making its mean time between failures (MTBF) in the range of decades. Because the System z hardware is so reliable, we allow the System z server to be a single point of failure in this architecture.
- ▶ We duplicate all of the software environments (LPAR, z/OS, WebSphere, DB2), so that none of them is a single point of failure.
- ▶ No failure should be noticeable to the user. The current transaction may fail, but subsequent transactions must succeed. After any single failure, transactions continue at the same rate with no degradation in throughput or response time.
- ▶ The architecture must be rapidly scalable to support increases and decreases in business volume.
- ▶ The architecture should leverage the high availability capabilities and features of z/OS Parallel Sysplex.

## 9.2 Test environment topology

This section provides details of our test system configuration and the reference architecture used.

## 9.2.1 Test configuration

Figure 9-1 shows a simple view of our test system with two LPARs and duplicate transaction processing software.

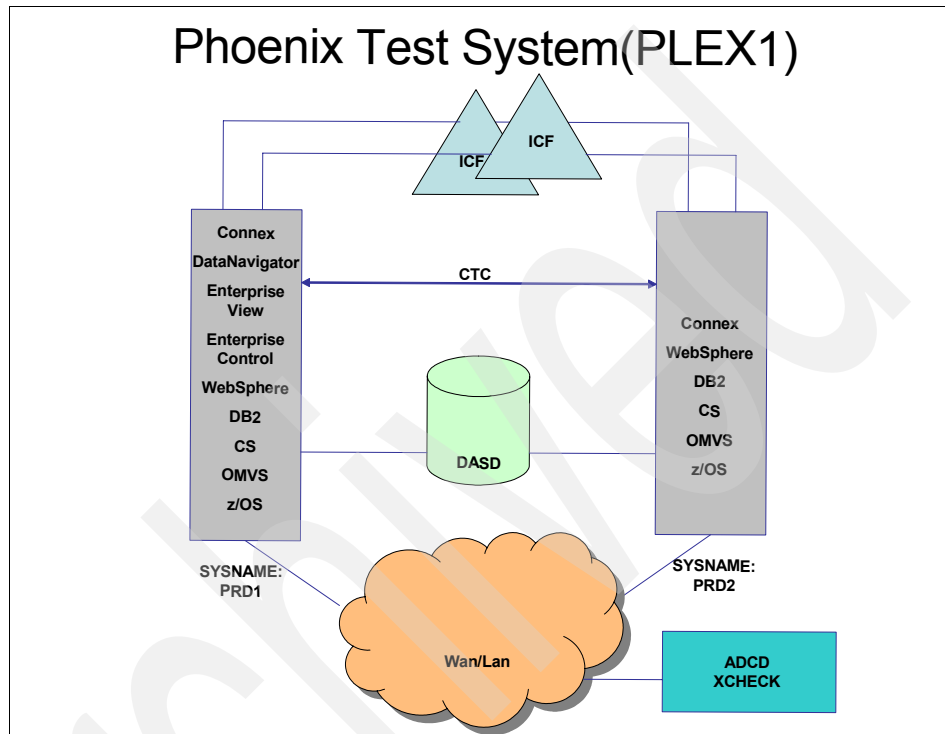


Figure 9-1 FIS test system

## 9.2.2 Environment description

We set up a simple Parallel Sysplex system for our testing environment.

- ▶ We had two z/OS images in our Parallel Sysplex. The two z/OS LPAR names were PRD1 and PRD2 and the SYSPLEX name was PLEX1.
- ▶ We had two DB2 members that comprised our DB2 data sharing environment, named DB8A and DB8B.
- ▶ We had WebSphere MQ sharing queues running on both LPARs.
- ▶ We had a set of WebSphere addresses on each z/OS image.
- ▶ We set up shared HFS files in the Parallel Sysplex.

Figure 9-2 shows our environment.

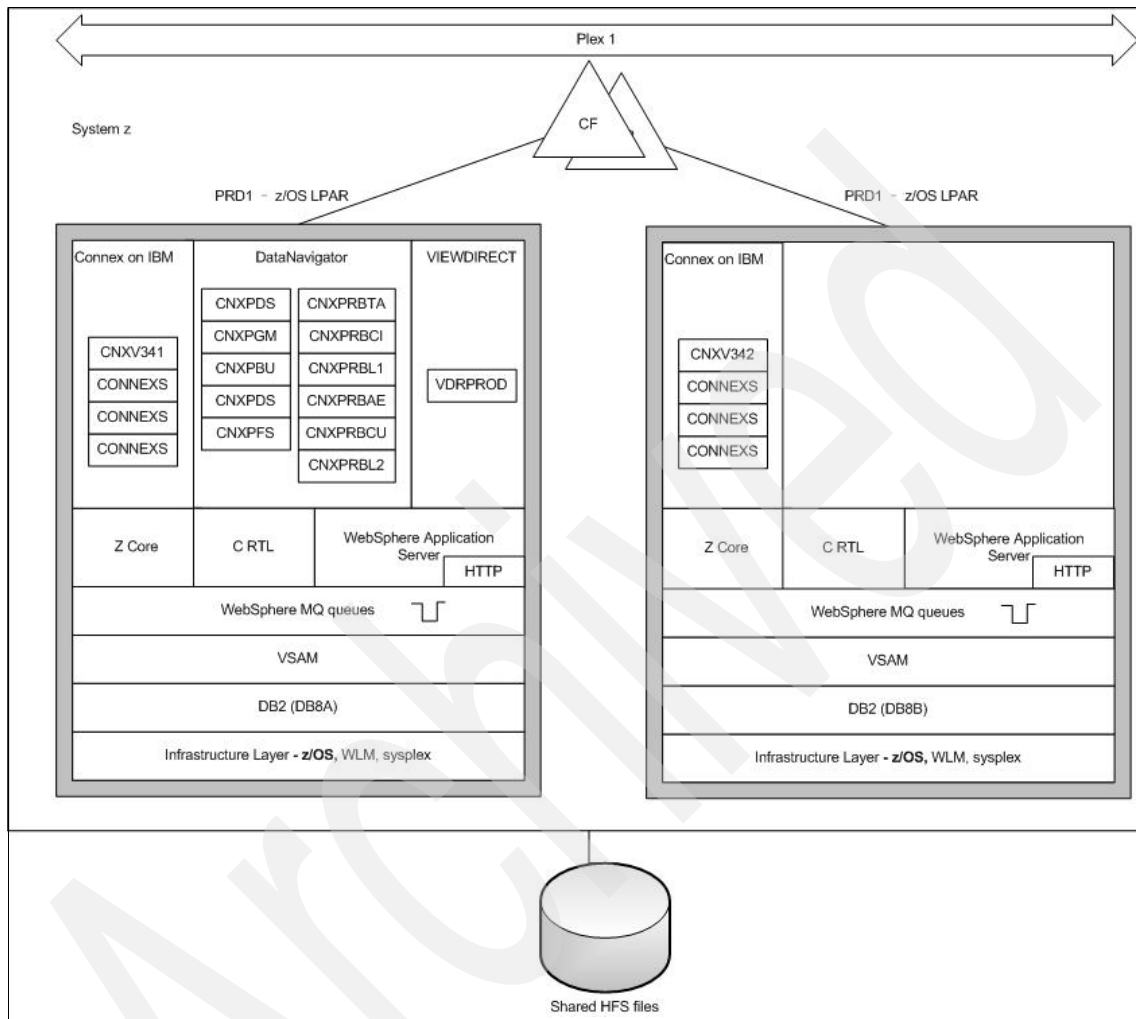


Figure 9-2 System details

For the FIS production applications, we had many address spaces in PLEX1. They are:

- Connex on IBM address spaces
  - One CNXV341 on PRD1
  - One CNXV342 on PRD2
  - Three Connex on IBM on PRD1
  - Three Connex on IBM on PRD2

- ▶ DataNavigator address spaces
  - CNXPDS on PRD1
  - CNXPGM on PRD1
  - CNXPBU on PRD1
  - CNXPDS on PRD1
  - CNXPFS on PRD1
  - CNXPRBTA on PRD1
  - CNXPRBCI on PRD1
  - CNXPRBL1 on PRD1
  - CNXPRBAE on PRD1
  - CNXPRBCU on PRD1
  - CNXPRBL2 e on PRD1
- ▶ VIEWDIRECT
  - VDRPROD on PRD1

## 9.2.3 Reference architecture

In this section we discuss the reference architecture.

### ARM

Restarting applications if they fail is important. This capability contributes to the high availability of the environment and minimizes the impact on the user. The z/OS Automatic Restart Manager (ARM) helps ensure that this restart process is carried out quickly and efficiently, according to a user-defined policy.

### WebSphere

Some of the FIS applications were deployed to a WebSphere cluster consisting of two LPARs (nodes). WebSphere manages the deployment of the application onto the nodes of the cluster and can upgrade the application on the fly.

User session data is replicated among the cluster members so that, if a cluster member fails, the transaction can be continued on the other cluster member. We recommend configuring WebSphere to replicate session data and hold it in memory in the cluster members. This option performs well and can scale well, and is simpler to configure than using a database to hold session data.

The following steps are required to accomplish this setup:

1. Make the transaction logs sharable by all members of the cluster. By default these are located in this directory:

```
<WASinstallroot>\profiles\<profilename>\tranlog\<cellname>\<nodename>\<servername>\transaction
```

However, they should be configured for another directory that you will make sharable.

2. After you configure the logs, the only other setup is to check the “Enable high availability for persistent services” box in the ServerCluster.

**Note:** For details on how to set up the HA manager, refer to the section “Transaction Manager High Availability” in the *WebSphere Application Server V6 Scalability and Performance Handbook*, SG24-6392.

### **DB2 data sharing**

The test system made use of a DB2 data sharing group as the data store. DB2 subsystems that share data must belong to a DB2 data sharing group that runs in a Parallel Sysplex cluster. The data sharing function of DB2 for z/OS enables applications that run on different DB2 subsystems to read and write the same data concurrently.

A data sharing group is a collection of one or more DB2 subsystems (or members) that access shared DB2 data. All members of a data sharing group use the same shared DB2 catalog and directory, share user data, and behave as a single logical server with the benefit of higher scalability and availability.

### **MQ shared queue**

The function of a WebSphere MQ shared queue is that a message on the queue can be accessed by one or more queue managers that are running on the Parallel Sysplex. This is called a queue-sharing group and it provides a rapid mechanism for communication. It does not require channels to be active between queue managers. To implement a shared queue, a structure is defined in the CFRM. Messages on a shared queue are stored in the Parallel Sysplex coupling facility. An application can connect to any of the queue managers within the group. All of the queue managers can access any of the queues, providing high availability.

### **Disk multipathing**

Our I/O subsystem was set up so that each disk data set had multiple paths to the application program. Also, disks were shared so that each LPAR had access to as much of the data on the other LPAR as possible.

## **9.3 Unplanned outages**

Our first group of four tests simulates unplanned outages.



### 9.3.1 Scenario 1: losing a non-critical task in Connex on IBM

The FIS product Connex on IBM has the capability to restart any of its tasks if they go down. This test shows the automatic recovery sequence after a Connex on IBM task is lost.

#### Objective

The objective is to cancel an address space that is running non-critical tasks in a Connex on IBM system and to see whether the tasks will be automatically restarted by the Connex on IBM application system.

#### Test procedure

We cancelled a Connex on IBM address space (CNXV3412). We then checked the system log to see whether the Connex on IBM tasks were closed and the Connex on IBM address space was restarted.

#### Expected results

When the Connex on IBM address space (CNXV3412) is cancelled, the Connex on IBM tasks will be closed and the Connex on IBM address space will be restarted, including all tasks.

#### Actual results

We can see that the Connex on IBM address space (CNXV3412) was cancelled and restarted successfully from the system log, as shown in Example 9-1.

*Example 9-1 A task is automatically restarted*

---

#### C Connex on IBMS.CNXV3412

```
IEE301I Connex on IBMS.CNXV3412 CANCEL COMMAND ACCEPTED
I01 LOGTSKS CF000 12/18 12:12:46 LOGSC002 CLOSE SUCCESSFUL
I01 LOGTSKS CF000 12/18 12:12:46 LOGSC001 CLOSE SUCCESSFUL
I01 LOGTSKS CF000 12/18 12:12:46 LOGSC    CLOSE SUCCESSFUL
I01 LOGTSKS CF000 12/18 12:12:46 LOGSB002 CLOSE SUCCESSFUL
I01 LOGTSKS CF000 12/18 12:12:46 LOGSB001 CLOSE SUCCESSFUL
I01 LOGTSKS CF000 12/18 12:12:46 LOGSB    CLOSE SUCCESSFUL
I01 LOGTSKS CF000 12/18 12:12:46 LOGSA002 CLOSE SUCCESSFUL
I01 LOGTSKS CF000 12/18 12:12:46 LOGSA001 CLOSE SUCCESSFUL
I01 LOGTSKS CF000 12/18 12:12:46 LOGSA    CLOSE SUCCESSFUL
I01 LOGTSKP CF000 12/18 12:12:46 LOGPC002 CLOSE SUCCESSFUL
I01 LOGTSKP CF000 12/18 12:12:46 LOGPC001 CLOSE SUCCESSFUL
I01 LOGTSKP CF000 12/18 12:12:47 LOGPC    CLOSE SUCCESSFUL
I01 LOGTSKP CF000 12/18 12:12:47 LOGPB002 CLOSE SUCCESSFUL
I01 LOGTSKP CF000 12/18 12:12:47 LOGPB001 CLOSE SUCCESSFUL
I01 LOGTSKP CF000 12/18 12:12:47 LOGPB    CLOSE SUCCESSFUL
```

```

I01 LOGTSKP CF000 12/18 12:12:47 LOGPA002 CLOSE SUCCESSFUL
I01 LOGTSKP CF000 12/18 12:12:47 LOGPA001 CLOSE SUCCESSFUL
I01 LOGTSKP CF000 12/18 12:12:47 LOGPA    CLOSE SUCCESSFUL
I01 TRACE   CF000 12/18 12:12:47 TRACE01  CLOSE SUCCESSFUL
I01 PMTASK  PC145 12/18 12:12:47 A01      HAS TERMINATED
START Connex on IBMS.CNXV3412
I01 PMTASK  PC171 12/18 12:12:47 A01      WILL BE RESTARTED
I01 PMTASK  PC190 12/18 12:12:47 REGION A01  STARTUP IN P
IEF450I Connex on IBMS CNXV3412 - ABEND=S222 U0000 REASON=00000000
$HASP395 Connex on IBMS  ENDED
$HASP100 Connex on IBMS  ON STCINRDR
IEF695I START Connex on IBMS  WITH JOBNAME Connex on IBMS  IS ASSIGNED
TO US
START2 , GROUP SYS1
$HASP373 Connex on IBMS  STARTED
I01 A01     PC115 12/18 12:12:48 TIMER    ATTACHED SUCCESSFUL
I01 PMTASK  PC119 12/18 12:12:49 TIMER    HAS STARTED
I01 A01     PC115 12/18 12:12:49 TRACE    ATTACHED SUCCESSFUL
I01 A01     PC115 12/18 12:12:49 FHTASK   ATTACHED SUCCESSFUL

```

---

### 9.3.2 Scenario 2: Canceling critical tasks in Connex on IBM

In this section we provide a second scenario.

#### Objective

The objective is to cancel critical tasks in a Connex on IBM system and see whether the tasks will be restarted in Connex on IBM.

#### Test procedure

We cancelled a Connex on IBM address space (CNXV3413) and checked the log.

#### Expected results

When the Connex on IBM address space (CNXV3413) is cancelled, the Connex on IBM tasks will be closed and the Connex on IBM address space will be restarted.

## Actual results

The system log in Example 9-2 shows that the Connex on IBM address space (CNXV3413) was cancelled and restarted successfully.

*Example 9-2 A Connex on IBM address space is automatically restarted*

---

```
0000000 2007317 10:24:57.63 FISDP2 00000290 C Connex on IBMS.CNXV3413
0000000 2007317 10:24:57.66 TSU07454 00000090 IEE301I Connex on IBMS.CNXV3413
CANCEL COMMAND ACCEPTED
0020000 2007317 10:24:57.68 STC07459 00000290 BPXP018I THREAD 121DD7C000000000, IN
PROCESS 83951788, ENDED 362 D
362 00000290 WITHOUT BEING UNDUBBED WITH COMPLETION CODE 40222000, AND REASON CODE E
362 00000290 00000000.
0000000 2007317 10:24:57.68 STC07456 00000290 START Connex on IBMS.CNXV3413
4000000 2007317 10:24:57.70 STC07459 00000281 IEF450I Connex on IBMS CNXV3413 -
ABEND=S222 U0000 REASON=00000000
4000000 2007317 10:24:57.71 STC07459 00000281 $HASP395 Connex on IBMS ENDED
0200000 2007317 10:24:57.74 STC07464 00000281 $HASP100 Connex on IBMS ONSTCINRDR
0020000 2007317 10:24:57.77 STC07464 00000290 IEF695I START Connex on IBMS WITH
JOBNAME Connex on IBMS IS ASSIGNED TO USERS START2 ,
GROUP SYS1
4000000 2007317 10:24:57.77 STC07464 00000090 $HASP373 Connex on IBMS STARTED
0020000 2007317 10:24:59.23 STC07464 00000290 I01 PMTASK PC145 11/13 10:24:57 A03
HAS TERMINATED
0020000 2007317 10:24:59.24 STC07464 00000290 I01 PMTASK PC171 11/13 10:24:57 A03
WILL BE RESTARTED
0020000 2007317 10:24:59.24 STC07464 00000290 I01 PMTASK PC190 11/13 10:24:57 REGION
A03 STARTUP IN PROGRESS
0020000 2007317 10:24:59.24 STC07464 00000290 I01 A03 PC115 11/13 10:24:59 NSMTASK
ATTACHED SUCCESSFULLY
0020000 2007317 10:24:59.24 STC07464 00000290 I01 NSMTASK OF000 11/13 10:24:59
MSGTEXT OPEN SUCCESSFUL
0020000 2007317 10:24:59.24 STC07464 00000290 I01 NSMTASK NS500 11/13 10:24:59
MESSAGE TEXT FILE MSGTEXT IS ACTIVE
0020000 2007317 10:25:01.03 STC07464 00000290 I01 PMTASK PC119 11/13 10:25:01
NSMTASK HAS STARTED
0020000 2007317 10:25:01.03 STC07464 00000290 I01 PMTASK NS000 11/13 10:25:01
**READY TO ACTIVATE COMMUNICATIONS
0020000 2007317 10:25:01.03 STC07464 00000290 I01 PMTASK PC191 11/13 10:25:01
REGION A03 ACTIVE
```

---

### 9.3.3 Scenario 3: failure of a Connex on IBM image

In this section we discuss a third scenario.

## Objective

The objective is to simulate the loss of a Connex on IBM image on one LPAR (PRD2) to check that Connex on IBM on PRD1 will start the recovery process and take over the workload that was running on PRD2.

## Test procedure

We used the command **V XCF,PRD2,OFFLINE** to remove PRD2 from Parallel Sysplex. See Example 9-3.

### Example 9-3 Varying an LPAR offline

---

|   |         |             |                                       |
|---|---------|-------------|---------------------------------------|
| PRD1                                    | 2008014 | 09:20:29.76 | -V XCF,PRD2,OFFLINE                   |
| PRD1                                    | 2008014 | 09:20:29.77 | *0074 IXC371D CONFIRM REQUEST TO VARY |
| SYSTEM PRD2 OFFLINE. REPLY SYSNAME=PRD2 |         |             |                                       |
| PRD1                                    | 2008014 | 09:21:18.11 | -R 0074,SYSNAME=PRD2                  |
| PRD1                                    | 2008014 | 09:21:18.11 | IEE600I REPLY TO 0074 IS;SYSNAME=PRD2 |

---

## Expected results

The LPAR PRD2 is shut down and XCF notifies Connex on IBM on the surviving image (PRD1). Upon receiving the notification, Connex on IBM starts the recovery process on PRD1 and takes over the workload that had run on PRD2.

## Actual results

We can see in Example 9-4 that the Connex on IBM image running on PRD2 was shut down. Connex on IBM on the surviving image started the recovery process and took over the workload that had run on PRD2.

### Example 9-4 System log showing Connex on IBM recovery

---

|          |          |             |       |       |          |                                |             |   |   |       |        |
|----------|----------|-------------|-------|-------|----------|--------------------------------|-------------|---|---|-------|--------|
| 09.09.32 | STC02993 | I02 THD910B | NS000 | 01/14 | 09:09:32 | AT0199                         | AT004       | C | L | 00002 | CLOSED |
| 09.09.32 | STC02993 | I02 THD910B | NS000 | 01/14 | 09:09:32 | AT0200                         | AT004       | C | L | 00002 | CLOSED |
| 09.21.39 | STC02993 | I01 FNTASK  | PC128 | 01/14 | 09:21:39 | IMAGE I02 SHUTDOWN             |             |   |   |       |        |
| 09.21.39 | STC02993 | I01 PMTASK  | PC168 | 01/14 | 09:21:39 | IMAGE I01 RECOVERY STARTED     |             |   |   |       |        |
| 09.21.39 | STC02993 | I01 A01     | PC115 | 01/14 | 09:21:39 | TCHTCP1 ATTACHED SUCCESSFULLY  |             |   |   |       |        |
| 09.21.39 | STC02993 | I01 A01     | PC115 | 01/14 | 09:21:39 | TCHTCP3 ATTACHED SUCCESSFULLY. |             |   |   |       |        |
| 09.21.49 | STC02993 | I01 THN560B | NS000 | 01/14 | 09:21:49 | AT1200                         | AT004       | C | L | 00102 | CLOSED |
| 09.21.49 | STC02993 | I01 PMTASK  | PC119 | 01/14 | 09:21:49 | THD910B                        | HAS STARTED |   |   |       |        |
| 09.21.49 | STC02993 | I01 PMTASK  | PC119 | 01/14 | 09:21:49 | THN560B                        | HAS STARTED |   |   |       |        |
| 09.21.49 | STC02993 | I01 PMTASK  | PC169 | 01/14 | 09:21:49 | IMAGE I01 RECOVERY ENDED       |             |   |   |       |        |

---

### 9.3.4 Scenario 4: z/OS system failover

We wanted to test what would happen if an entire LPAR became unavailable. Would the FIS system continue in another LPAR? This is a highly unlikely scenario, but it does test the restartability of each FIS component in another LPAR, which is also one of our objectives.

Before we did this test, we issued a D XCF command to check the system status (Example 9-5).

*Example 9-5 D XCF command*

---

```
-D XCF,S,ALL
IXC335I 16.40.06 DISPLAY XCF 598
SYSTEM  TYPE SERIAL LPAR STATUS TIME          SYSTEM STATUS
PRD1    2094 3E7D   00  12/14/2007 16:40:04 ACTIVE      TM=SIMETR
PRD2    2094 3E7D   02  12/14/2007 16:40:06 ACTIVE      TM=SIMETR
```

---

#### Objective

The objective is to deactivate an entire LPAR running FIS products to see whether DataNavigator, DB2, Connex on IBM, and EnterpriseView will be restarted successfully.

#### Test procedure

To prepare our environment:

1. We enabled our Parallel Sysplex SFM policy.
2. We configured System Automation (SA) to remove any inactive systems from the Parallel Sysplex automatically.
3. We modified our CNXPGM startup procedure to make sure that DataNavigator and ViewDirect can be started in the same z/OS image.
4. We defined a Parallel Sysplex ARM policy for DataNavigator to restart in another available z/OS image.
5. We also defined an ARM policy for our DB2 subsystem to restart for data recovery.

To run the test, we deactivated the PRD1 LPAR from the HMC. We replied "DOWN" to the Parallel Sysplex message from the z/OS console.

#### Expected results

DataNavigator and DB2 will be restarted successfully so that the FIS application system can be accessed.

## Actual results

We saw that the DataNavigator and DB2 systems were restarted successfully. See the following examples. After PRD1 was removed (Example 9-6), we checked that each component was restarted by ensuring that we could successfully log on to EnterpriseView, DataNavigator, and the Connex on IBM system.

### *Example 9-6 After PRD1 is removed*

---

```
RESPONSE=PRD2
IEE112I 16.44.00 PENDING REQUESTS 694
RM=1    IM=0    CEM=0    EM=0    RU=0    IR=0    NOAMRF
ID:R/K   T SYSNAME  MESSAGE TEXT
      0198 R PRD2   *0198 IXC102A XCF IS WAITING FOR SYSTEM PRD1
                        DEACTIVATION. REPLY DOWN WHEN MVS ON PRD1 HAS
                        BEEN SYSTEM RESET

R 198,DOWN
IEE600I REPLY TO 0198 IS;DOWN
IEA257I CONSOLE PARTITION CLEANUP IN PROGRESS FOR SYSTEM PRD1
```

---

In Example 9-7, the PRD1 image was removed from the Parallel Sysplex. ARM restarted the subsystem, as defined in our policy.

### *Example 9-7 PRD1 image removed*

---

```
IXC467I STOPPING PATHOUT STRUCTURE IXCPATH1 LIST 9 724
      USED TO COMMUNICATE WITH SYSTEM PRD1
      RSN: SYSPLEX PARTITIONING OF REMOTE SYSTEM

IWM051I STRUCTURE(SYSZWLM_WORKUNIT), FOR SYSTEM PRD1 CLEANED UP
IXC467I STOPPING PATHOUT STRUCTURE IXCPATH2 LIST 9 725
      USED TO COMMUNICATE WITH SYSTEM PRD1
      RSN: SYSPLEX PARTITIONING OF REMOTE SYSTEM

IXC467I STOPPING PATHIN STRUCTURE IXCPATH2 LIST 8 726
      USED TO COMMUNICATE WITH SYSTEM PRD1
      RSN: SYSPLEX PARTITIONING OF REMOTE SYSTEM

IXC467I STOPPING PATHIN STRUCTURE IXCPATH1 LIST 8 727
      USED TO COMMUNICATE WITH SYSTEM PRD1
      RSN: SYSPLEX PARTITIONING OF REMOTE SYSTEM

S CNXPGM
IXC812I JOBNAME CNXPGM, ELEMENT CDNREDB_I01 944
FAILED DUE TO THE FAILURE OF SYSTEM PRD1.
THE ELEMENT WAS RESTARTED WITH OVERRIDE START TEXT.
IXC813I JOBNAME CNXPGM, ELEMENT CDNREDB_I01 945
```

WAS RESTARTED WITH THE FOLLOWING START TEXT:  
S CNXPGM  
THE RESTART METHOD USED WAS DETERMINED BY THE ACTIVE POLICY.

S DB8AMSTR  
IXC812I JOBNAME DB8AMSTR, ELEMENT DSN810DB8A 911  
FAILED DUE TO THE FAILURE OF SYSTEM PRD1.  
THE ELEMENT WAS RESTARTED WITH OVERRIDE START TEXT.  
IXC813I JOBNAME DB8AMSTR, ELEMENT DSN810DB8A 912  
WAS RESTARTED WITH THE FOLLOWING START TEXT:  
-DB8A STA DB2  
THE RESTART METHOD USED WAS DETERMINED BY THE ACTIVE POLICY.

---

Example 9-7 on page 150 shows only one member (PRD2) in the Parallel Sysplex. DN is not Sysplex enabled. You can see the ARM restart for DN and DB2 in the block of code starting with “S CNXPGM” (CNXPGM is DataNavigator).

Using the SDSF **DA** command we can see the Connex on IBM address spaces, as shown in Example 9-8.

*Example 9-8 Connex on IBM address spaces*

| Display                         | Filter   | View     | Print  | Options | Help      |   |     |             |      |
|---------------------------------|----------|----------|--------|---------|-----------|---|-----|-------------|------|
| -----                           |          |          |        |         |           |   |     |             |      |
| SDSF STATUS DISPLAY ALL CLASSES |          |          |        |         |           |   |     | LINE 1-25 ( |      |
| NP                              | JOBNAME  | JobID    | Owner  | PrtY    | Queue     | C | Pos | SAff        | ASys |
|                                 | CNXV342  | STC02077 | START2 | 15      | EXECUTION |   |     | PRD2        | PRD2 |
|                                 | CNXPGM   | STC02084 | START2 | 15      | EXECUTION |   |     | PRD2        | PRD2 |
|                                 | CNXPBU   | STC02087 | START2 | 15      | EXECUTION |   |     | PRD2        | PRD2 |
|                                 | CNXPDS   | STC02088 | START2 | 15      | EXECUTION |   |     | PRD2        | PRD2 |
|                                 | CNXPRBL1 | STC02090 | START2 | 15      | EXECUTION |   |     | PRD2        | PRD2 |
|                                 | CNXDFS   | STC02089 | START2 | 15      | EXECUTION |   |     | PRD2        | PRD2 |
|                                 | CNXPRBL2 | STC02091 | START2 | 15      | EXECUTION |   |     | PRD2        | PRD2 |
|                                 | CNXPRBTA | STC02092 | START2 | 15      | EXECUTION |   |     | PRD2        | PRD2 |
|                                 | CNXPRBAE | STC02093 | START2 | 15      | EXECUTION |   |     | PRD2        | PRD2 |
|                                 | CNXPRBCI | STC02094 | START2 | 15      | EXECUTION |   |     | PRD2        | PRD2 |
|                                 | CNXPRBCU | STC02095 | START2 | 15      | EXECUTION |   |     | PRD2        | PRD2 |

Issuing **DA**, we can see the DB2 address space, as shown in Example 9-9.

*Example 9-9 DB2 address space*

---

|         |        |         |       |         |             |
|---------|--------|---------|-------|---------|-------------|
| Display | Filter | View    | Print | Options | Help        |
| -----   |        |         |       |         |             |
| SDSF    | STATUS | DISPLAY | ALL   | CLASSES |             |
|         |        |         |       |         | LINE 1-8 (8 |

| NP | JOBNAME  | JobID    | Owner  | Prty | Queue     | C | Pos | SAff | ASys |
|----|----------|----------|--------|------|-----------|---|-----|------|------|
|    | DB8BMSTR | STC02021 | START2 | 15   | EXECUTION |   |     | PRD2 | PRD2 |
|    | DB8BIRLM | STC02022 | START2 | 15   | EXECUTION |   |     | PRD2 | PRD2 |
|    | DB8BDBM1 | STC02023 | START2 | 15   | EXECUTION |   |     | PRD2 | PRD2 |
|    | DB8BDIST | STC02024 | START2 | 15   | EXECUTION |   |     | PRD2 | PRD2 |
|    | DB8AMSTR | STC02082 | START2 | 15   | EXECUTION |   |     | PRD2 | PRD2 |
|    | DB8AIRLM | STC02083 | START2 | 15   | EXECUTION |   |     | PRD2 | PRD2 |
|    | DB8ADB1  | STC02086 | START2 | 15   | EXECUTION |   |     | PRD2 | PRD2 |
|    | DB8ADIST | STC02096 | START2 | 15   | EXECUTION |   |     | PRD2 | PRD2 |

After PRD1 was removed from the Parallel Sysplex, we checked to see whether each component was restarted in PRD2 by logging onto EnterpriseView, DataNavigator, and the Connex on IBM system.

## 9.4 Planned outages

From time to time, IBM and FIS systems require upgrades for new functions or maintenance. It is important to be able to install these upgrades without affecting our production system, from an user view. Here we describe two scenarios for taking down an LPAR in a planned outage while maintaining production.

### 9.4.1 Scenario 5: applying IBM PTFs

In this section we provide a scenario of applying IBM PTFs.

#### Objective

Our objective is to install IBM software PTFs on a production system.

#### Procedure

The procedure is:

1. Before performing an IBM software upgrade or applying PTFs, check whether there are FIS PTFs that we need to apply. Decide whether to apply these at the same time as the IBM PTFs or separately.
2. Apply PTFs onto the test system before affecting the production system. Run function tests, stress tests, and recovery tests on our test system. If no problems occur, schedule the PTFs for the production systems.
3. Apply PTFs on our production system, ensuring that all prerequisite and co-requisite maintenance is also applied. During this upgrade, move the FIS application system to another z/OS image to maintain high availability.



4. Shut down the z/OS system and re-IPL.
5. Move the FIS application system back to the original z/OS image.
6. Check the system log and FIS JOBLOG for verification (no errors).
7. If a problem occurs and cannot be solved quickly, collect the information required by IBM and rolled back to the previous version of the system.

**Important:** FIS applications, such as Connex on IBM, DataNavigator, FIS Fraud Navigator, and EnterpriseView, interact with IBM products such as DB2, MQ, WebSphere, z/OS UNIX, VTAM, TCP/IP, and VSAM. We need to understand the relationships between these systems. For detailed information, see the *FIS Application System Reference Guide*.

### Expected and actual results

We can apply the IBM product PTFs on the production system with little or no impact on throughput or response time.

## 9.4.2 Scenario 6: upgrading FIS application system

In this section we discuss a sixth scenario.

### Objective

Our objective is to install FIS product PTFs on a production system.

### Test procedure

When we need to upgrade the FIS application system due to maintenance or new functions, we follow this procedure:

1. Check to see whether there are IBM PTFs that need to be applied. Make a decision whether to apply and test them together with the FIS PTFs or separately.
2. Apply the PTFs on a test system first. Testing should include a function test, a stress test, and a recovery test. If no problems occur, we can put these PTFs into production.
3. Upgrade the FIS application system at non-peak time. This can decrease the impact on business if problems occur.
4. Modify z/OS system parameters as directed by the upgrade guide. Put the modifications into effect on the production system. We do not put any LMODs in the LPA, so we do not need to IPL the system.
5. Define new versions of FIS data sets according your enterprise naming convention. An example would be "HLQ.FIS.&version..pdsname".

6. Follow these tips to minimize the impact on production:
  - a. Connex on IBM can run in multiple LPARs and monitor its status in a Parallel Sysplex. We can shut down the LPAR in which we are upgrading FIS code, and the transactions will automatically re-dispatch to an available Connex on IBM system within the Parallel Sysplex.
  - b. Move DataNavigator and FIS Fraud Navigator to another z/OS image.
  - c. If not using the MQ shared queue function in the Parallel Sysplex environment, move the MQ subsystem to the LPAR in which the Connex on IBM system is running. If there are MQ shared queues, simply shut down the MQ subsystem normally on the LPAR being upgraded.
  - d. If WebSphere is started in all end nodes, then stop WebSphere normally in the maintenance LPAR.
  - e. If DB2 data sharing is enabled, then stop the DB2 member normally. There is no need to restart it in the other z/OS image. However, if a standalone DB2 subsystem is being used, move it to another z/OS image.
  - f. Verify that the current system is normal and can process transactions normally.
7. When the upgrade has been installed, start FIS products using the new version name in the startup procedure.
8. Verify the system log and address space JES2 JOBLOG.
9. Restart DB2 subsystem (if DB2 data sharing is enabled) or move back the DB2 subsystem to the original z/OS image.
10. Restart the WebSphere subsystem.
11. Restart the MQ subsystem (if MQ shared queues are enabled), or move back the MQ subsystem to the original z/OS image.
12. Verify that all that subsystems have started successfully.
13. Run a function test on the new application software versions.
14. If a problem occurs that cannot be solved quickly, collect system information as required by FIS and as described in Chapter 8, "Problem determination on z/OS" on page 129. Stop the FIS application system and restart using the previous code versions.

### **Expected and actual results**

We can apply the FIS product PTFs on the system with little or no impact on the production system.

## 9.5 Test summary

In this section we provide a summary of the tests.

### **Connex on IBM**

In a Parallel Sysplex and DB2 data sharing environment, when one z/OS image is down and all the workload can be dispatched to another available z/OS image, business transactions on Connex on IBM can be processed continually. Also, some Connex on IBM tasks are restarted automatically and quickly, which reduces the impact of an FIS application system outage.

### **FIS Fraud Navigator**

If FIS Fraud Navigator is cancelled on one z/OS image (PRD1), it will automatically restart, just as we described in above.

### **DataNavigator**

The DataNavigator application allows the office to monitor the Connex on IBM system status and generate various reports for management. There is only one DataNavigator system allowed in a Parallel Sysplex.

When a z/OS system containing DataNavigator fails, or a DataNavigator address space fails, the Parallel Sysplex ARM policy will automatically restart these.

We can define an ARM or SA policy to restart DataNavigator on an available z/OS image if a problem occurs.

### **EnterpriseView (EV)**

EV is a WebSphere application, so we can define the same port for it on all systems. This provides high availability for EV, as for all WebSphere applications. Then if a z/OS system is down or one set of WebSphere address spaces is down, another WebSphere can take over the workload.

## 9.6 High availability configurations

There are four possible system configurations. Each has its own level of availability and associated costs. These are a standalone system, Parallel Sysplex, Parallel Sysplex + GDPS/PPRC/HyperSwap, and Parallel Sysplex + GDPS/XRC. Each level further reduces the risk of a having a single point of failure. We list each component availability in Table 9-1.

Table 9-1 Degree of high availability

|                        | Failure   | Connex on IBM | DataNavigator | FIS Fraud Navigator | EnterpriseView |
|------------------------|-----------|---------------|---------------|---------------------|----------------|
| z/OS Standalone System | Hardware  | X             | X             | X                   | X              |
|                        | z/OS      | X             | X             | X                   | X              |
|                        | WebSphere | A             | A (delphi)    | A                   | X              |
|                        | DB2       | X             | X             | X                   | X              |
|                        | MQ        | A             | X             | A                   | A              |
|                        | CS        | X             | X             | X                   | X              |
|                        | DASD      | X             | X             | X                   | X              |
|                        | Site      | X             | X             | X                   | X              |
| Parallel Sysplex       | Hardware  | A             | A             | A                   | A              |
|                        | z/OS      | A             | A             | A                   | A              |
|                        | WebSphere | A             | A             | A                   | A              |
|                        | DB2       | A             | A             | A                   | A              |
|                        | MQ        | A             | A             | A                   | A              |
|                        | CS        | A             | A             | A                   | A              |
|                        | DASD      | X             | X             | X                   | X              |
|                        | Site      | X             | X             | X                   | X              |

|   | Failure   | Connex on IBM | DataNavigator | FIS Fraud Navigator | EnterpriseView |
|---|-----------|---------------|---------------|---------------------|----------------|
| GDPS/PPRC HYPERSWAP*+<br>Parallel Sysplex | Hardware  | A             | A             | A                   | A              |
|   | z/OS      | A             | A             | A                   | A              |
|   | WebSphere | A             | A             | A                   | A              |
|   | DB2       | A             | A             | A                   | A              |
|   | MQ        | A             | A             | A                   | A              |
|   | CS        | A             | A             | A                   | A              |
|   | DASD      | A             | A             | A                   | A              |
|   | Site      | X             | X             | X                   | X              |
| GDPS/XRC*+Parallel Sysplex                | Hardware  | A             | A             | A                   | A              |
|   | z/OS      | A             | A             | A                   | A              |
|   | WebSphere | A             | A             | A                   | A              |
|   | DB2       | A             | A             | A                   | A              |
|   | MQ        | A             | A             | A                   | A              |
|   | CS        | A             | A             | A                   | A              |
|   | DASD      | A             | A             | A                   | A              |
|   | Site      | A             | A             | A                   | A              |

Legend:  
X=Not Available  
A=Available

In the following section we discuss some high availability considerations for each environment.

## 9.6.1 Standalone z/OS system

In this section we discuss the advantages and disadvantages of the standalone z/OS system.

### ► Advantages

- Highly available z/OS operating system and supported subsystems
- High throughput System z hardware
- Low hardware and software cost/per transaction

► Disadvantages

A standalone System z is very reliable (commonly above 99% available) from both a hardware and a software perspective. However, it does not have any redundancy. So in the rare case of a failure it would have these exposures:

- No hardware backup
- No software backup
- Vulnerable to a DASD failure
- Vulnerable to a general site failure

## 9.6.2 Parallel Sysplex

In this section we list the advantages and disadvantages of Parallel Sysplex.

► Advantages

- Utilizes z/OS to increase availability
- High performance System z hardware
- Low hardware and software cost per transaction
- Avoids many single points of failure problem
- Efficient use of all hardware and software available

► Disadvantages

- The complexity requires a more experienced operator for managing the system
- Vulnerable to a DASD failure
- Vulnerable to a general site failure

## 9.6.3 Parallel Sysplex+GDPS/PPRC/HyperSwap

In this section we list the advantages and disadvantages of Parallel Sysplex+GDPS/PPRC/HyperSwap.

► Advantages

- Utilizes z/OS to increase high availability
- High performance System z hardware
- Low hardware and software cost per transaction
- Avoids many single points of failure
- Efficient use of all hardware and software available
- High availability even if there is a DASD failure

- Disadvantages
  - The complexity requires a more experienced operator for managing the system.
  - More hardware is required for an additional cost.
  - Vulnerable to a general site failure.

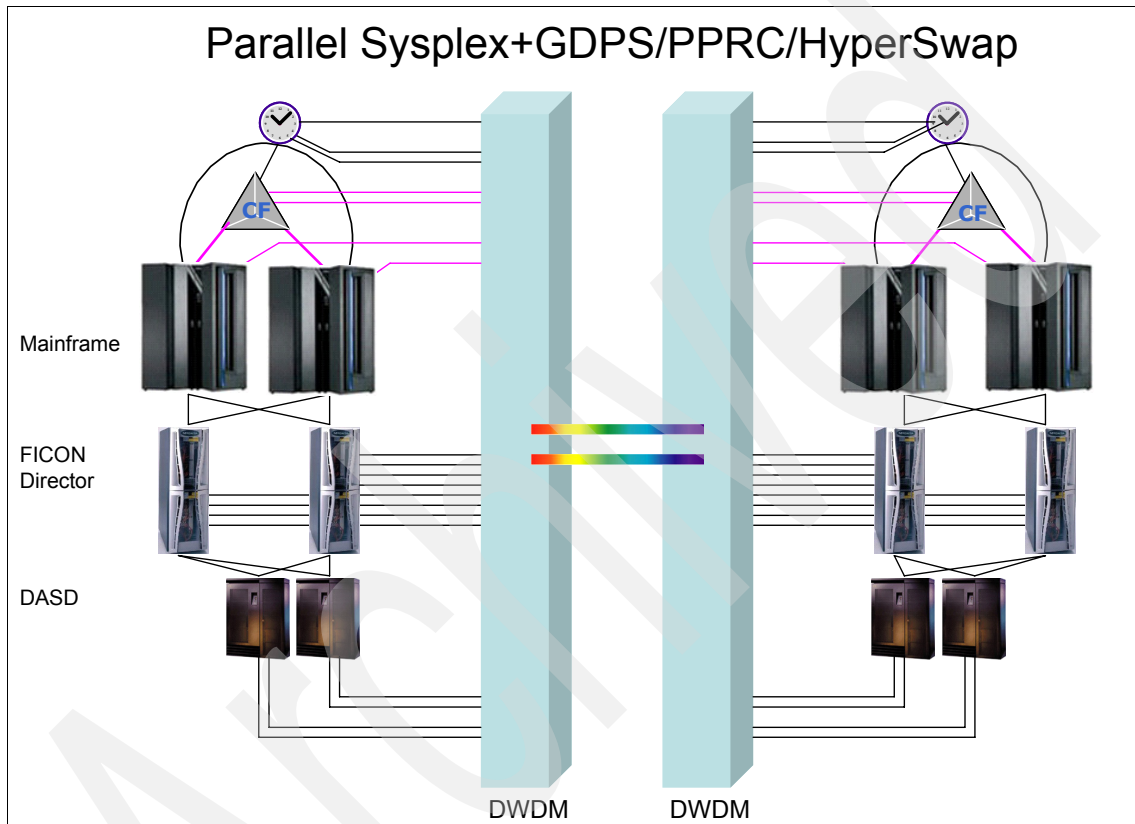


Figure 9-3 Parallel Sysplex+ GDPS/PPRC/HyperSwap Solution

For more information about GDPS/PPRC, see *GDPS Family An Introduction to Concepts and Capabilities*, SG24-6374.

## 9.6.4 Parallel Sysplex+GDPS/XRC

In this section we list the advantages and disadvantages of Parallel Sysplex+GDPS/XRC.

- ▶ Advantages
  - Utilizes z/OS to increase high availability
  - High performance System z hardware
  - Low hardware and software cost per transaction
  - Avoids many single points of failure
  - Efficient use of all hardware and software available
  - High availability even if there is a DASD failure
  - High availability even there is site failure
- ▶ Disadvantages
  - The complexity requires a more experienced operator for managing the system.
  - More hardware is required for an additional cost.



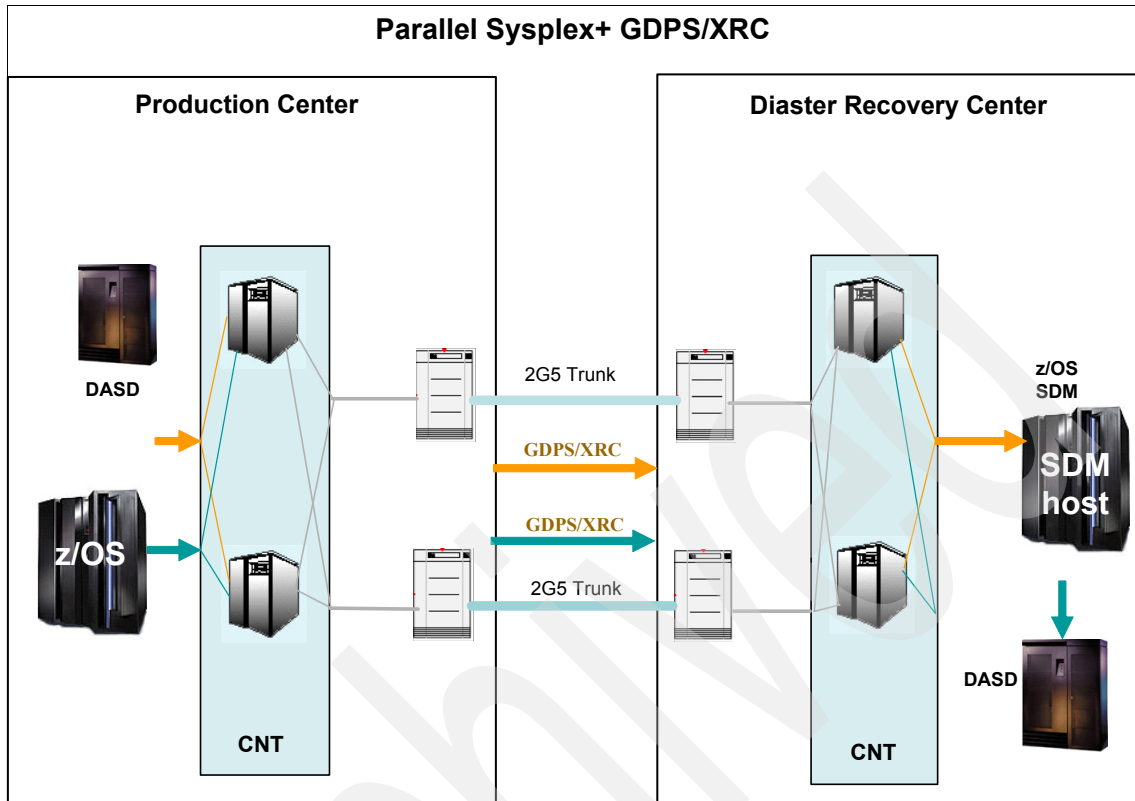


Figure 9-4 Parallel Sysplex plus GDPS/XRC

For more information about GDPS/XRC, see Tier 7 in *IBM System Storage Business Continuity: Part 1 Planning Guide*, SG24-6547.

Archived



# Part 4

## Appendixes

Archived



## Capacity planning tips

Archived

# Introduction

A large-scale commercial computing environment is one that exists to process very large volumes of corporate data. The following sections use the example of the banking industry, where data relating to client accounts may be processed for many reasons, including:

- ▶ Cash withdrawal from an ATM
- ▶ Check processing
- ▶ Online inquiry
- ▶ Telephone inquiry
- ▶ Statement printing
- ▶ Overnight accounting
- ▶ Data mining
- ▶ Money laundering legislation

Much of the data is also subject to regulatory controls that determine how long information must be kept.

The term *capacity* has several definitions, including:

- ▶ The potential or suitability for holding, storing, or accommodating
- ▶ The facility or power to produce, perform, deploy, or simply process

A large bank most likely will have millions of clients. Many of these may hold several accounts, and the data relating to these accounts has to be available to multiple functions, which may be reading or updating it.

In order to meet the first definition, there needs to be sufficient disk storage to allow fast access—and a less expensive and secure media for long-term storage.

To meet the second definition, there needs to be sufficient computing capacity to run the programs that will process the data.

## Balancing resources for capacity planning

Capacity planning resources, in a balanced system, are not independent variables in the determination of system performance. Instead, the interrelationship of processing power, I/O capability, processor storage size, and networks determines system performance. Capacity can then be determined based on performance objectives. In z/OS, the workload manager manages the achievement of user objectives.

A *balanced system* is one in which all resources are dimensioned toward each other, thereby optimizing the system's capacity and achieving the best performance and throughput in accordance with the goals. A balanced system should have no single bottleneck. However, if a single bottleneck is unavoidable, then it should be the most expensive one to eliminate (for example, a CPU-constrained system). All other bottlenecks should be eliminated.

Balanced systems provide these benefits:

- ▶ They optimize your enterprise's IT investments.
- ▶ They increase your IT productivity.
- ▶ They increase communication between users and WLM services.
- ▶ They provide more transparency for management by establishing service level agreements (SLAs).
- ▶ They help to avoid performance problems that can seriously impact your business.

## Managing mixed workloads

If large amounts of data are to be stored in one place, a large-scale commercial computing environment can be expected to provide service to the following:

- ▶ Online clients
- ▶ Online in-house users
- ▶ Batch jobs

All of these have the same basic requirement, which is to have access to processors, main storage, and disk or tape storage. The time frame in which a unit of work must be completed is the distinguishing factor.

Online clients, such as Internet users, have an expectation of fast response time. If that expectation is not met, they could switch to a competitor's site. In-house users, such as application developers, require a responsive system in order to be productive, but their needs will not take precedence over those of the clients.

Batch jobs are generally the least demanding regarding a specific completion time, as long as they are completed at a certain time. They can usually give up resources to the work in the previous two categories.

The relationship between capacity, workload, and response time demands of each of the categories should be defined in a service level agreement.

# Distributed versus central system

In the banking example, the storing of information relating to only one client's accounts usually will not be particularly large and could just as easily be stored and retrieved from a small distributed system as from a large central system.

## Distributed system

If data can be stored and processed on a single server, a large number of small servers could be a more appropriate solution than a centralized data store. If data needs to be accessed and updated from different servers, however, the following problems must be addressed:

- ▶ Data retention may be required by law. When data is changed, does the owning system or the updating system create a copy?
- ▶ Data may need to be processed for both reading and updating, a data locking mechanism has to be implemented, and all servers processing the data need to participate in this locking scheme.
- ▶ Connectivity has to be established between the various servers.

The system management of such a setup can be problematic, and it often impacts many people.

## Large central system - IBM System z approach

To circumvent the problems concerning distributed data and system management, a more centralized approach is used in most mainframe computing centers. The IBM System z philosophy is that the best utilization of the capacity of a server is obtained by running mixed workloads.

IBM System z architecture has evolved over many years. The design has adhered to the following criteria:

- ▶ Large volumes of commercial data are best held in one place.
- ▶ Applications should not need to be rewritten as new technology is introduced.
- ▶ The total system—that is, hardware, system software, and applications—must be extremely robust in the areas of reliability and availability.

The concept of running a mixed workload on a single system derives from the fact that, with modern processor chips, it is very unlikely in a commercial environment that one program could keep a processor fully utilized over a long period of time.



There are several approaches to achieving better processor utilization, such as:

- ▶ Level 1 (internal) and level 2 (external) processor cache
- ▶ Preloading of programs into main storage
- ▶ Preloading of data into main storage
- ▶ Running several copies of an application

### **IBM System z input/output architecture**

If large amounts of data are to be held in one place, then enough bandwidth must be provided to enable that data to be accessed in a timely manner.

The communication bus between IBM System z server and its input/output (I/O) devices uses a channel architecture. This architecture is implemented in dedicated microprocessors that communicate across fiber optic cables (the channel) to the control units that operate the I/O devices without affecting the server.

Note that the I/O devices are *not* directly connected to the channel. One or more of them are connected to a control unit. This design allows new function to be introduced in just the microprocessor and control units. Additionally, the design has implications for scalability.

The performance of an I/O control unit, such as a disk control unit, has an impact on capacity.

## **Managing the system resources**

A service-level agreement (SLA) is an agreement between a service provider and a recipient, generally the server owner and a business unit. There should be several SLAs in place to cover the various aspects of the business that will be run on the server. For capacity management, having correct and precise definitions of SLAs is very important, because these SLAs are the baseline against which the capacity demands are measured and compared.

The SLA for a batch job is generally an agreement as to when the output from the job will be available. The earliest start time will be when the data to be processed is available.

The 24x7 applications, such as for Internet processing, should be available at all times, and the SLA will apply to the response to each client enquiry, referred to as a *transaction*.

Both the batch program and the 24x7 transaction server application are run in the same environment. Each has an address space on the same System z

server. Therefore, it is important to manage the system or server resources in the most efficient way to achieve both SLAs.

## CPU management

The z/OS operating system provides for a sophisticated feature called Workload Manager (WLM). WLM supplies the means for managing CPU usage to meet goals or Service Level Objectives (SLOs).

Through Service Class definitions written in a rule-based *policy*, the client selects which workload is important over others. The *importance* of the workload will allow the job request to obtain more or less CPU. The CPU management for a job or workload is incorporated in the policy.

CPU usage is defined within the service class as service units (SUs). Using the policy definition, the WLM administrator can put in place the rules for managing the minimum and maximum CPU consumption for a job or workload. The WLM policy is the overall method used to manage resource usage for a sysplex.

## Disk management

There is a unique element of z/OS called Data Facility Storage Management Subsystem (DFSMS). DFSMS provides all essential disk, storage, and device management functions of the system. In a system-managed storage environment, DFSMS automates and centralizes storage administration based on the policies that an installation defines for availability, performance, space utilization, and security.

Storage management policies reduce the need for users to make multiple detailed decisions that are unrelated to their business objectives. DFSMS provides functions that reduce the occurrence of system outages, and enhance disaster recovery capabilities and system security. DFSMS also improves business efficiency by providing better system performance and throughput, as well as usability enhancements that increase storage administration productivity.

## Storage management

The z/OS software was designed to provide security and integrity, while also allowing communication between functions and access to common services. To help achieve this goal, the memory of the mainframe is divided into *address spaces*. Each program or subsystem of the z/OS operating system is loaded into its own private address space.

A program can be located in the physical memory of the system or on DASD storage. The combination of the two is called *virtual memory* or *virtual storage*.

The physical memory of the system or main storage is referred to as *real storage* and is managed by a Real Storage Manager (RSM™) in elements of 4 kilobytes, known as *frames*. Each frame of real storage holds a page of virtual storage. Real storage is not directly addressed by application programs. RSM uses tables to establish a link between a virtual address and a real address. A hardware mechanism known as Dynamic Address Translation (DAT) converts the virtual addresses seen by a program into real addresses that the CPs can access.

The majority of programs have sections that are frequently executed and have data objects that they refer to on a regular basis. The RSM, in common with other systems, tries to keep as many of the pages of programs and data that have been used in storage, as a performance aid. Eventually, all of the real storage will be full of virtual pages. To accommodate new ones, the pages that have not been used recently are copied to a disk file. This is referred to as a *page-out*, and if the virtual page is referred to at a later time, it will be paged in. Avoiding page-in and page-out activity aids the performance of a program and it is therefore a capacity requirement to have sufficient real storage to achieve this.

## System z architecture

A z/OS system is capable of driving the servers' processors at 100% for sustained periods of time. This is not difficult on any system if there are sufficient resources for workload. The problem is making sure that the high-importance units of work achieve their objectives.

One technique for achieving this is by allowing the high-importance work to run until it must give up control of the CP (for example, while waiting for an I/O operation). The I/O is considered an interrupt and at this point a context switch occurs. A *context switch* is where the executing unit of work status is kept in special hold areas called register save areas (RSAs). After the executing work status is saved, the interrupt processing can begin. The RSA is used to provide a return point back to the executing program to pick up execution where it left off after the interrupt is processed.

In the case of an I/O interrupt, the I/O request may have been to retrieve a record from a data set. The return data from the I/O can then be used by the executing program and continue execution to process the information. The interrupt provides the means to give up control to other executing work so that their instruction requests can be processed. This is known as *multi-tasking*.

z/OS is able to achieve balance between processor utilization and allowing as many units of work as possible to meet their objectives, defined in WLM. z/OS

does so by exploiting the hardware architecture of IBM System z server. The significant architectural elements of the IBM System z server are the Program Status Word (PSW) and six types of interrupts. The combination of these allows control of the work that runs on the CPs.

## **LPAR**

Most online environments need to have sufficient capacity to handle their peak workload volumes and are therefore sized such that they are not, generally, fully utilizing the resources that they are allocated. For reasons such as systems management and security, it is often preferable to have separate environments for system testing, application development, and production systems. In order to meet the requirement for function separation and to optimize processor utilization, the logical partition (LPAR) was introduced.

An LPAR is a subdivision of a machine's resources, which shares use of the central processors (CPs) with other LPARs but has main storage dedicated for its use. From the point of view of the programs running in an LPAR, the other LPARs do not exist (unless you connect them using a Coupling Facility, channel, hiperspace, OSA, or shared DASD). There is no common storage for them to use for communications. A hardware interface allows the definition of the amount of main storage to be allocated to an LPAR and the number of processors that it can use concurrently. We refer to a *processor* as the resources assigned and operating system running in the LPAR.

## **Parallel Sysplex**

While the combination of IBM System z hardware and software provides excellent availability and reliability characteristics, both planned and unplanned outages can occur. If 24x7 availability is required, then more than one LPAR must be configured to share the critical workload. This is known as a Parallel Sysplex.

Parallel Sysplex has aspects that relate to capacity, availability, and scalability, and it is described in more detail under those topics. Briefly, it is a form of clustering that allows up to 32 systems to be linked together and share data. The components required to do this, in addition to the servers and I/O devices, are a time source and a Coupling Facility.

## ***Coupling Facility***

The Coupling Facility (CF) is either a standalone server or an LPAR that runs a specialized operating system and provides facilities to control data sharing. A CF has high-speed links to the LPARs in the Parallel Sysplex, and can store selected data in its main storage. This ability to make shared data available quickly is the major contribution of the Coupling Facility to capacity.

### ***Common time source***

Transaction processing software writes log entries about the actions taken to update data, including timestamps. This is why a common time signal is needed in a clustered eComputing environment. In a Parallel Sysplex, a single high resolution time source is used to ensure that every subsystem of the Parallel Sysplex uses exactly the same time.

## **Capacity planning methodology**

There are several methods and tools that you can use when performing capacity planning. Regardless of the method used, the same basic steps followed. Here we briefly discuss a few of these steps.

### **Select or create a capacity planning method**

Depending on the time, cost, skill, and level of detail and accuracy that is available, capacity planners choose the method that best meets their expectations. Here is a brief description of the five different methods to help you determine which one is most suitable to your situation:

- ▶ *Guidelines:* A guideline falls somewhere between a guess and a rule of thumb (ROT), and is simple to execute. Simple, informal projections are often done without documentation and may be based on experience and knowledge of the system.
- ▶ *Linear projections:* A linear projection has better accuracy than a guideline, but you have to be sure of using this approach because it does not apply for all cases. For example, newer types of workloads, such as WebSphere applications or the unpredictable peaks of transactions, can easily transform a forecast into an unreliable assumption.
- ▶ *Analytic methods:* An analytic method based on mathematical methods such as the queuing theory can provide insight into forecasting CPU utilization, response time evaluation, capture ratios, effect of buffering, effect of queuing, and so forth. Based on measured data from today's environment, various configurations and growth scenarios are studied and documented.
- ▶ *Discrete methods:* A discrete method uses a *discrete event simulation* to employ a next-event technique to control the behavior of the model. Many applications of discrete simulation involve queuing systems of one kind or another. The queuing structure may be obvious, as in a queue of jobs waiting to be processed on a batch computer or in a stack of aircraft waiting for landing space at an airport. Other examples are customers waiting in line for service at a bank or grocery store. In the simplest case, the queue operates with a first-in, first-out (FIFO) discipline.

In capacity planning, a discrete method can be used as an application of the discrete simulation. Unlike analytic simulation, discrete simulation is not based on mathematical formulas. Using discrete capacity planning methods, dissimilar workloads and their effect on each other are modelled.

- ▶ *Benchmarking*: Benchmarking refers to running a real workload in a particular hardware configuration to measure CPU power. It is often more effective than the previous methods, but is also more expensive and time consuming. Benchmarks are divided into two categories:
  - A benchmark from others: With these benchmarks, the evaluations of standard workloads running in a controlled environment result in numbers that are used as a reference. Normally, consulting companies do these studies.
  - Your own benchmark: With these benchmarks, you run your own workload on a specific hardware configuration. Unlike the analytic and discrete methods, the benchmark requires executable code and installed hardware to predict how a certain combination of resources will perform in a specific situation.

Whatever method used, the customer creates and manages a database with SMF registers that will be utilized as input by the capacity planning tool.

### **Measure current workload**

It is always necessary to check the health of the system being measured in order to avoid undesired deviations. To process this collected data, customers can use worksheet tools or tools that are available in the marketplace. However, customers are increasingly relying on the C200 tool from IBM, in which case the processing step is executed by the IBM technical sales force. The inferences, predictions, and interpretations are done by IBM technical support relative to customer expectations.

### **Present the results**

Discussions of capacity planning often emphasize the final step, presenting your results. Keep in mind, though, that high-level executives are generally more interested in *the big picture* than in the planning details. Using clear graphics and showing only relevant statistics can help you communicate your results most effectively.

## **Measurements**

As stated earlier, capacity requires that the CPs, main storage, and data access must be in balance, and it is essential that regular measurements are taken to detect imbalance before performance is impacted.

Both IBM System z hardware and software produce information that may be recorded and processed by batch jobs to produce reports. The reports need to be read in the context of the SLAs, because the SLAs determine whether the capacity is sufficient. If elements of work are missing the SLA targets, then the causative factors must be addressed.

### **Central Processor (CP) usage**

Because the workload can be a mix of high and low importance, a CP utilization of 100% does not necessarily mean that capacity has been exceeded. In fact, it is not unusual for the CPs on IBM System z servers to be utilized at 100% for long periods of time. However, if the reports indicate that the reason for the failure to meet the SLA was due to waiting for CP resources, then a closer look is warranted. Delays due to waiting for CP resources do not necessarily mean that more CPs are required. Perhaps running part of the workload at a different time might be all that is needed.

### **Main storage usage**

Delays can be caused if the ratio of virtual storage to real storage is such that workloads are competing for the real storage. As with the CPs, it is not necessarily required to add more resources—there might be too many address spaces running concurrently, and an alteration to the scheduling will be all that is required.

### **Access to disk storage**

Delays due to elongated response times from disk storage can be the most difficult to identify. This is due to the interrelationship between the components involved. Possible solutions are to add more channels, move data to different volumes, or change the workload mix.

### **Capacity tools**

IBM offers a number of tools, including RMF and CP2000.

#### ***RMF***

The z/OS Resource Measurement Facility (RMF) is an optional element of z/OS. It is a product that supports installations in performance analysis, capacity planning, and problem determination. For these disciplines, different kinds of data collectors are needed:

- ▶ Monitor I is a long-term data collector for all types of resources and workloads. The SMF data collected by Monitor I is mostly used for capacity planning, but is also used performance analysis. Additionally, there are some products like Tivoli Decision Support that use SMF records gathered by Monitor I.

- ▶ Monitor II is a snapshot data collector for address space states and resource usage.
- ▶ Monitor III is a short-term data collector for problem determination, workflow delay monitoring, and goal attainment supervision. This data is used by RMF PM Java Client, Tivoli DM, and Tivoli TBSM.
- ▶ Data collected by all three gatherers can be saved persistently for later reporting.
- ▶ Monitor II and Monitor III are online reporters. Monitor I and Monitor III can store the collected data long term.

To learn more about RMF, visit the RMF home page:

<http://www.ibm.com/servers/eserver/zseries/zos/rmf>

Regardless of the method or tool used, RMF data is a complete source of information for resource capacity planning.

### ***CP2000***

CP2000 is a tool that provides performance analysis and capacity planning information from SMF registers. CP2000 requires an input that is extracted from the SMF files by the CP2KEXTR program.

IBM Technical Support teams and Business Partners use the extractor. A support manual for CP2KEXTR, available from IBM, explains how to customize the batch process of extraction. Generally, this task is executed at the customer site. A Technical Support Analyst from IBM or a Business Partner demonstrates the process to the customer Technical Support Analyst, who then selects the best period to analyze with the CP2000. Normally, the periods chosen are peak periods.

CP2000 analysis helps customers to understand and manage their resources (such as processors, storage, DASD I/O) and features (such as Parallel Sysplex).





## Business continuity concepts

Today's on demand business climate is highly competitive. Customers, employees, suppliers, and business partners expect to be able to use applications and access data at any time from anywhere. Businesses must also be increasingly sensitive to data protection and security. Add regulatory requirements and the inherent demands of participating in the global economy, and the demands of modern IT management become apparent.

# Overview

Business Continuity is the ability to adapt and respond to risks, as well as opportunities, in order to maintain continuous business operations.

A 24x7 enterprise needs a comprehensive business continuity plan that supports high availability, continuous operations, and disaster recovery. Implementing such a plan can help organizations achieve resiliency by:

- ▶ Supporting competitive efforts through more efficient and cost-effective risk management
- ▶ Helping to achieve critical business application and data availability based on business value
- ▶ Facilitating compliance with government rules and regulations
- ▶ Safeguarding against internal and external threats and providing sustained operations even in the event of a disaster.

Most enterprises cannot afford the cost of downtime due to planned or unplanned system outages. Although the indirect, longer term impacts of downtime (lost market share, decreased productivity, noncompliance with regulations, reduced competitiveness, damaged brand reputation, and eroded customer loyalty) are harder to measure, they are equally important. Strengthening the resiliency of your business can help mitigate or avoid them.

The business continuity strategy for key applications and business processes should provide for:

- ▶ High availability

High availability is the ability to provide access to applications. High availability is often provided by clustering solutions that work with operating systems coupled with a hardware infrastructure that has no single points of failure. If a server that is running an application suffers a failure, the application is picked up by another server in the cluster, with minimal or no interruption to the users. Today's servers and storage systems are also built with fault-tolerant architectures to minimize application outages due to hardware failures.

You can think of high availability as a resilient IT infrastructure that masks failures, and thus continues to provide access to applications.

- ▶ Continuous operations

Continuous operations is the ability to keep things running under normal operations, for example, where applications can remain online during scheduled backups or planned maintenance. Continuous operations

technologies provide the ability to perform repetitive, ongoing, and necessary infrastructure actions, while still maintaining high availability.

Normally, all the components providing continuous operations are situated in the same computer room. The building, therefore, becomes the single point of failure. Thus, a continuous operation setup does not usually of itself provide a disaster recovery solution.

You can think of continuous operations as the ability to keep applications running during scheduled backups or planned maintenance.

► Disaster recovery

Finally, disaster recovery is the ability to recover a datacenter at a different site if a disaster destroys the primary site or otherwise renders it inoperable. In a disaster recovery scenario, the processing resumes at a different site, and on different hardware. A non-disaster problem, such as a corruption of a key customer database, may indeed be a catastrophe for a business, but it is not, by our definition, a disaster, unless processing must be resumed at a different location and on different hardware.

You can think of disaster recovery as the ability to recover at a different site from unplanned outage.

The objectives of business continuity are to protect critical business data, to make key applications available, and to enable operations to continue after a disaster. These must happen in such a way that recovery time is both predictable and reliable, with predictable and manageable costs.

## IT recovery timeline

Figure B-1 shows a diagram of an IT recovery. Time proceeds from left to right.

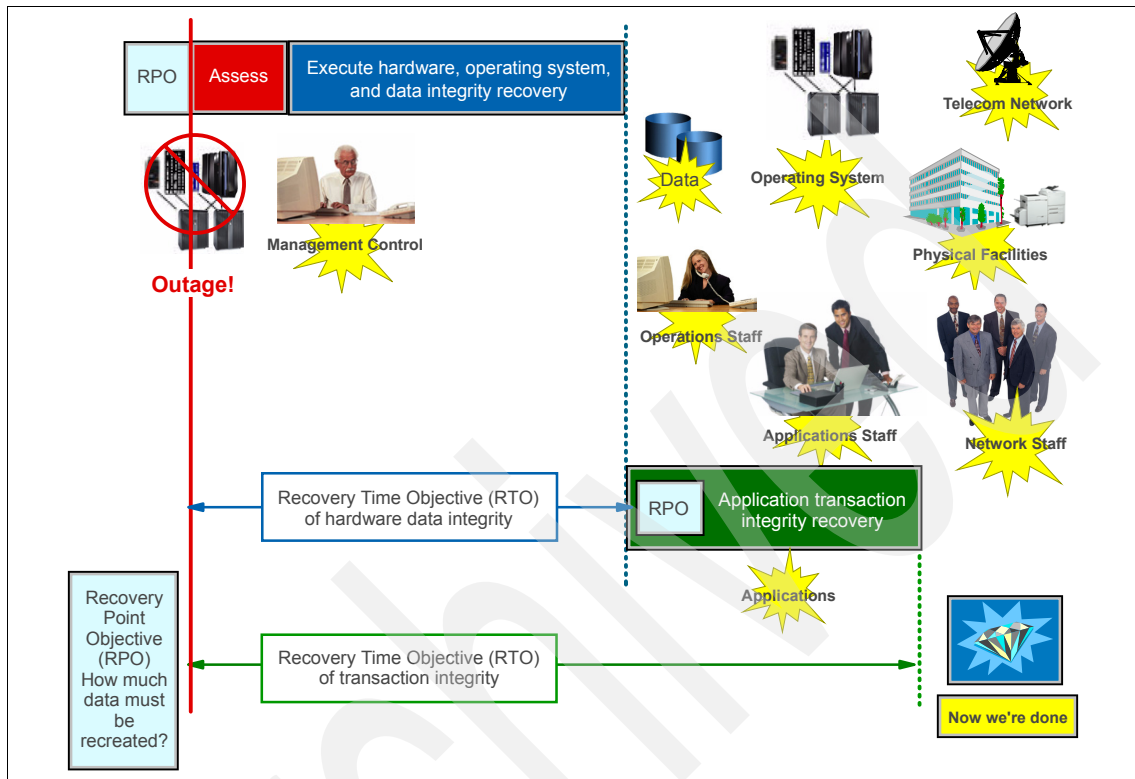


Figure B-1 Timeline of an IT recovery

Three major components are:

- ▶ **Assessment**

After the outage occurs, the first step is that management must assess the outage (this incurs elapsed time). Because there is a significant capital cost to declaring a disaster and executing a recovery, management must be sure that the situation warrants committing their organization to that expense. Once management has decided to declare a disaster, then they initiate the business continuity process.

- ▶ **Hardware and operating system recovery**

The first stage of the business continuity process is to recover the hardware, operating systems, and the data itself. Operations, networking, telecommunications, physical facilities, and associated staff are involved. At the end of this stage, the operating systems and the data are recovered.

Ideally, the data is accurate and consistent to a point-in-time prior to the outage. The time duration to this point is the *recovery time objective of hardware data integrity*.

► Transaction integrity recovery

Hardware data integrity is not the same as database/application integrity. The storage and servers cannot know what the logical database relationship is between multiple data blocks in the database. Therefore, once the first stage is complete, the transaction integrity recovery must next be performed by the applications staff, on the application and database. The applications staff performs transaction integrity recovery. Hopefully, this is a database restart and not a database recovery. This process backs out incomplete logical units of work, and restores the database to logical integrity as of the most recent time possible. When the transaction integrity recovery (rollback or roll forward) is complete, we now have the application and databases ready for user access. This duration is the *recovery time objective of transaction integrity*.

**Note:** There is a difference in elapsed time between RTO of hardware data integrity and RTO of transaction integrity. When discussing the recovery time objective (RTO), it is important to distinguish which of the two is being referred to. Operations and application staff can have differing perceptions of the RTO depending on whether the RTO is assumed to be at the hardware recovery level or at the application recovery level.

► Recovery point objective (RPO)

Finally, observe how the recovery point objective is depicted in Figure B-1 on page 180. RPO (which is how much data must be recreated) is shown as the time offset before the outage occurred.

**Note:** The RPO data recreation happens in the transaction integrity recovery stage. RPO data recreation cannot happen in the hardware and operating system recovery stage, because the server and storage components do not have knowledge of the logical relationships between multiple application and database blocks of data.

## Business continuity solution methodology

From an IT infrastructure standpoint, there is a large variety of valid business continuity products. The fundamental challenge is to select the optimum blend of all these business continuity products and technologies.

A common problem in the past has been a tendency to view the business continuity solution as individual product technologies and piece parts. Instead, business continuity solutions need to be viewed as a whole, integrated multi-product solution. In this section we propose a business continuity solution selection methodology that can be used to sort, summarize, and organize the various business requirements in a methodical way. Then we methodically use those business requirements to efficiently identify a proper and valid subset of business continuity technologies to address the requirements.

### **IT technology components**

To combine and properly choose between multiple products, disciplines, and skills to effect a successful IT business continuity solution, we first observe that we can categorize all valid business continuity IT technologies into five component domains:

- ▶ Servers
- ▶ Storage
- ▶ Software and automation
- ▶ Networking and physical infrastructure
- ▶ Skills and services required to implement and operate the above

All IT infrastructure necessary to support the business continuity solution can be categorized as one of these five components. These five categories provide a framework to organize the various component evaluation skills that are needed. To gather the proper mix of evaluation skills together facilitates an effective comparison, contrast, and blend of all five product component areas to arrive at an optimum solution.

### **Infrastructure simplification**

Complexity can prevent even the best organization from acting nimbly to meet the ever-changing market and client demands. Solutions for infrastructure simplification can help to improve efficiency, lower total cost of ownership, and reduce time-consuming and costly errors.

From a business continuity perspective, infrastructure simplification can do the following:

- ▶ Introduce common architectural recovery platforms (operating systems, applications, databases, servers, storage systems).
- ▶ Reduce the components to recover.
- ▶ Reduce the difficulties in managing dynamic application changes, managing IT environment changes, and managing data and storage allocations.

An excellent idea is to do a good job of infrastructure simplification as a prerequisite to implementing improved IT business continuity. This has the effect

of introducing consolidation (and cost savings) as a foundation for implementing the new business continuity functions.

### The tiers of business continuity

By categorizing business continuity technology into the various tiers, we have the capability to more easily match our desired RTO time with the optimum set of technologies. The reason for multiple tiers is that as the RTO time decreases, the optimum business continuity technologies for RTO must change. For any given RTO, there is always a particular set of optimum price/performance business continuity technologies.

The concept of business continuity tiers (Figure B-2) is a common method used in today's best practices for business continuity solution design. It was originally developed by the IBM US SHARE User Group in 1988. The concept of tiers is powerful, because the tiers concept recognizes that for a given customer recovery time objective, all business continuity products and technologies can be sorted into a RTO solution subset that addresses that particular RTO range.

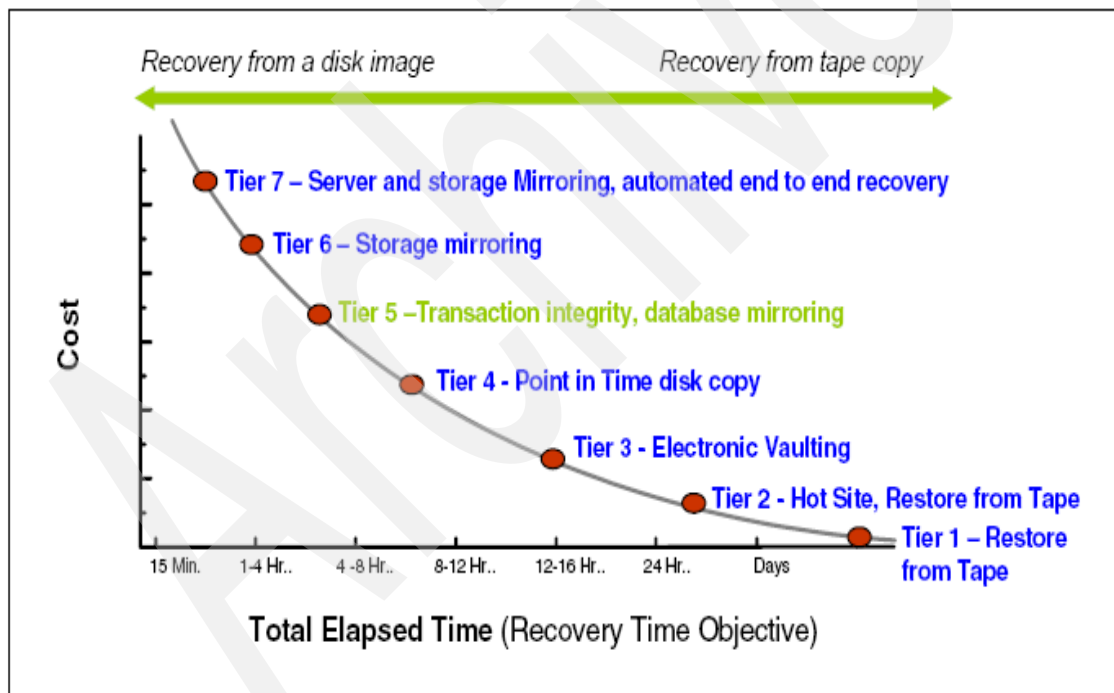


Figure B-2 Business continuity tiers

The tiers concept is flexible. As products and functions change and improve over time, the tier chart only needs to be updated by the addition of that new technology into the appropriate tier and RTO.

The tiers chart below gives a generalized view of some of today's IBM business continuity technologies by tier. As the recovery time becomes shorter, more aggressive business continuity technologies must be applied to achieve that RTO (carrying with them their associated increase in value and capital cost).

The tiers also reflect the way an IT organization can incrementally grow and improve their IT BC over time. Each preceding tier provides a foundation for the subsequent higher tier. Notice that implementing a higher tier does not remove the need for the lower tier. In fact, the higher tier can exist because it is based upon the foundation of the tiers below it.

These tiers are generally accepted examples of today's IT BC tiers. We recommend that you refine these tiers to create your own specific version of a business continuity tiers chart, specific to your organization, installation, and recovery times.

► Tier 0: no off-site data

Businesses with a tier 0 business continuity solution have no business continuity plan. There is no saved information, no documentation, no backup hardware, and no contingency plan. The length of recovery time in this instance is unpredictable. In fact, it may not be possible to recover at all.

► Tier 1: data backup with no hotsite

Businesses that use tier 1 business continuity solutions back up their data at an off-site facility. Depending on how often backups are made, they are prepared to accept several days to weeks of data loss, but their backups are secure off-site. However, this tier lacks the systems on which to restore data.

Sample solutions include:

- Pickup truck access method (PTAM)
- Disk subsystem or tape-based mirroring to locations without processors

► Tier 2: data backup with a hotsite

Businesses using tier 2 business continuity solutions make regular backups on tape. This is combined with an off-site facility and infrastructure (known as a hotsite) in which to restore systems from those tapes in the event of a disaster. This tier of solution will still result in the need to recreate several hours to days worth of data, but it is more predictable in recovery time.

Sample solutions include PTAM with hotsite available.



► Tier 3: electronic vaulting

Tier 3 solutions utilize components of tier 2. Additionally, some mission-critical data is electronically vaulted. This electronically vaulted data is typically more current than that which is shipped via the PTAM. As a result, there is less data recreation or loss after a disaster occurs.

Sample solutions include electronic vaulting of data.

► Tier 4: point-in-time copies

Tier 4 solutions are used by businesses that require both greater data currency and faster recovery than users of lower tiers. Rather than relying largely on shipping tape, as is common on the lower tiers, tier 4 solutions begin to incorporate advanced core technologies, such as point-in-time disk copy, basic disk mirroring without data consistency, forwarding of database log files and journals, and so on. Several hours of data loss is typical, but these tiers of solutions use methodologies such as disk point-in-time (PIT) copies, managed with software such as Tivoli Storage Manager, to effect a solution with better currency of data than can typically be achieved through tape-based solutions.

Sample solutions include:

- Peer-to-Peer Virtual Tape Server
- Global Copy

► Tier 5: transaction integrity

Tier 5 has traditionally been reserved for database-specific solutions in which transaction integrity-based database replication solutions are employed. These solutions are used by businesses with a requirement for database transaction and synchronization of data between one production and one or more remote data centers. These solutions can offer little or no data loss, and they can provide methodologies that employ deferred application of data at the remote site to provide protection against propagation of logical data corruption. The implementation of these solutions is entirely dependent on the software application in use.

Sample solutions include:

- Two-phase commit, such as DB2 remote replication
- WebSphere Information Integrator Q replication

► Tier 6: storage mirroring with little to zero data loss

Tier 6 business continuity solutions are defined as storage mirroring solutions. In other words, the storage subsystem (whether it is disk or tape) maintains a very high level of data currency, with built-in data integrity. This tier is used by businesses with little or no tolerance for data loss that need to restore data to applications rapidly. These solutions provide power-outage data consistency, and have no dependence on the applications.

Sample solutions include:

- Metro Mirror
- Global Mirror
- z/OS Global Mirror
- GDPS HyperSwap Manager

► Tier 7: highly automated business integrated solution

Tier 7 solutions include all the major components being used for a tier 6 solution with the additional integration of automation of all server, storage, software, and networking components. This allows a tier 7 solution to provide near continuous availability, with automated recovery of the applications.

Sample solutions include Geographically Dispersed Parallel Sysplex (GDPS) for System z.

## Blending tiers into an optimized solution

A best practice today in designing a business continuity solution is to further use the tiers concept to derive a blended business continuity solution for the entire enterprise. The most common result, from an enterprise standpoint, is a strategic architecture of three bands in a blended business continuity solution. Three bands generally appear as an optimum number, because at the enterprise level, two bands generally are insufficiently optimized (in other words, overkill at some point and underkill at others), and four bands are more complex but generally do not provide enough additional strategic benefit.

To use the tiers to derive a blended, optimized enterprise business continuity architecture, we suggest these steps:

1. Categorize the business' entire set of applications into three bands:
  - Low tolerance to outage
  - Somewhat tolerant to outage
  - Very tolerant to outage

Some applications that are not in and of themselves critical actually feed the critical applications. Therefore, those applications would need to be included in the higher band.

Within each band, there are tiers. The individual tiers represent the major business continuity technology choices for that band. It is not necessary to use all the tiers, and it is not necessary to use all the technologies.

2. Once we have segmented the applications (as best we can) into the three bands, we usually select one best strategic business continuity methodology for that band. The contents of the tiers are the *candidate technologies* from which the strategic methodology is chosen.

Business impact analysis, risk assessments, and program assessments are excellent methodologies and essential tools to assist in defining your application segmentation. These three bands are strategic objectives, which the organization by necessity will implement over time.

Note that the IT business continuity technologies chosen to service each band become the strategic, consolidated technology platform standards upon which this band's IT business continuity is based.

IBM business continuity solutions in the System Storage™ Resiliency Portfolio have been segmented into the continuous availability band, the Rapid Data Recovery band, and backup/restore band, shown in Figure B-3.

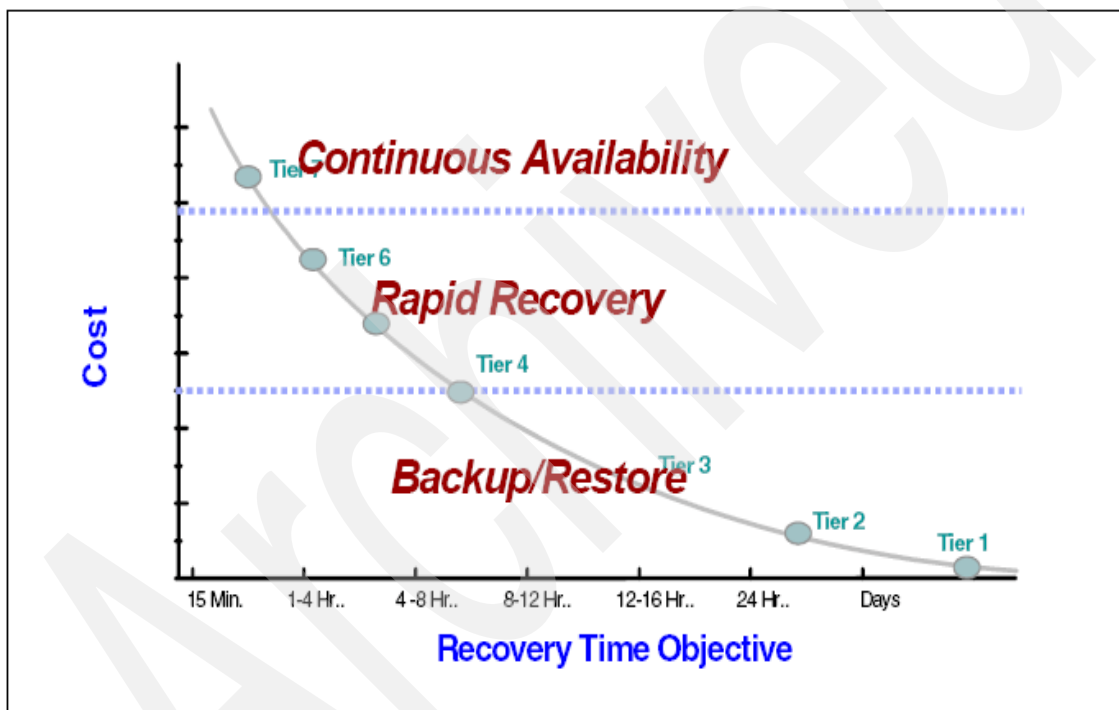


Figure B-3 Business continuity solutions

The implementation of an end-to-end IT business continuity solution does not need to be done all at once. You should plan for a step-by-step incremental IT business continuity project, and start with the current environment. From there you can incrementally build towards the final objective. Each stage provides the necessary foundation for the next step.

Now that we have introduced the three solution segment bands (backup/restore, rapid data recovery, and continuous availability) we further describe each segment band.

### **Backup and restore solutions (tiers 1–4)**

Backup and restore is the most simple and basic solution to protect and recover your data from failure by creating another copy of data from the production system. The second copy of data allows you to restore data to the time of the data backup. Backup and restore spans tier 1 through tier 4.

A sample solution is IBM Tape storage systems. IBM tape drives, libraries, and virtualization products can be used to build solutions for data archiving, backup, and disaster recovery. They can provide an important component of a comprehensive business continuity strategy that supports high availability, near continuous operations, and disaster recovery.

### **Rapid data recovery solutions (tiers 4–6)**

Rapid data recovery is based on maintaining a second copy of data that is consistent at a point-in-time as close to the time of a failure as possible. This consistent set of data allows for the restart of systems and applications without having to restore data and reapply updates that have occurred since the time of the data backup. It is possible that there may be a loss of a minimal number of in-flight transactions.

Rapid data recovery spans tier 4 through tier 6 and is different from continuous availability, because it does not have the end-to-end server, storage, software, and networking automation that is required to be a tier 7 solution.

Rapid data recovery solutions based on replication technology can be implemented on three different levels:

- ▶ Application/database replication
  - Requires less bandwidth.
  - Span of consistency is the application or database only.
  - More complex implementation. Must implement for each application.
- ▶ Server replication
  - Can be less complex than application implementation. Application independent.
  - Uses server cycles. Span limited to that operating system.
- ▶ Storage replication
  - Requires more bandwidth.

- Implementation is largely platform and application independent. Mirrors logical disks. Supports multiple heterogeneous systems.

Application/database and server replication is implemented in software, whereas storage replication exploits the hardware directly.

Here we focus on rapid data recovery solutions based on storage replication. We describe the following environments:

- ▶ Rapid data recovery for System z (GDPS/PPRC HyperSwap Manager)
- ▶ IBM System Storage SAN Volume Controller

Before we discuss the particular solutions, we first need to explain some aspects of data consistency when using disk-based mirroring techniques in database environments. We introduce important terms, including the consistency group.

### ***Data consistency for database environments***

In a disaster recovery solution using disk remote mirroring, we want to restart a database application following an outage without having to restore and recover the database. This process has to be consistent, repeatable, and fast (measurable in minutes). Restoring the database means restoring the last set of image copy tapes and then applying the log changes to bring the database up to the point of failure. This can take many hours, which would not constitute a tier 4 or later solution.

Moreover, actual disasters (fire, explosion, earthquake) are messy. You cannot expect your entire complex to fail simultaneously. Failures will be intermittent and gradual, and the disaster will occur over many seconds or even minutes. This is known as the *rolling disaster*. A viable disk mirroring disaster recovery solution must be designed to avoid data corruption that is caused during a rolling disaster.

In any operating system, the sequence in which updates are being made is what maintains the integrity of the data. If that sequence is changed, data corruption occurs. The correct sequence must be maintained within a volume, across volumes, and across multiple storage devices. For example, in Figure B-4, we show the relationship between a database and its log, which demonstrates the requirement for maintaining I/O integrity. Data consistency across the storage enterprise must be maintained to ensure data integrity.

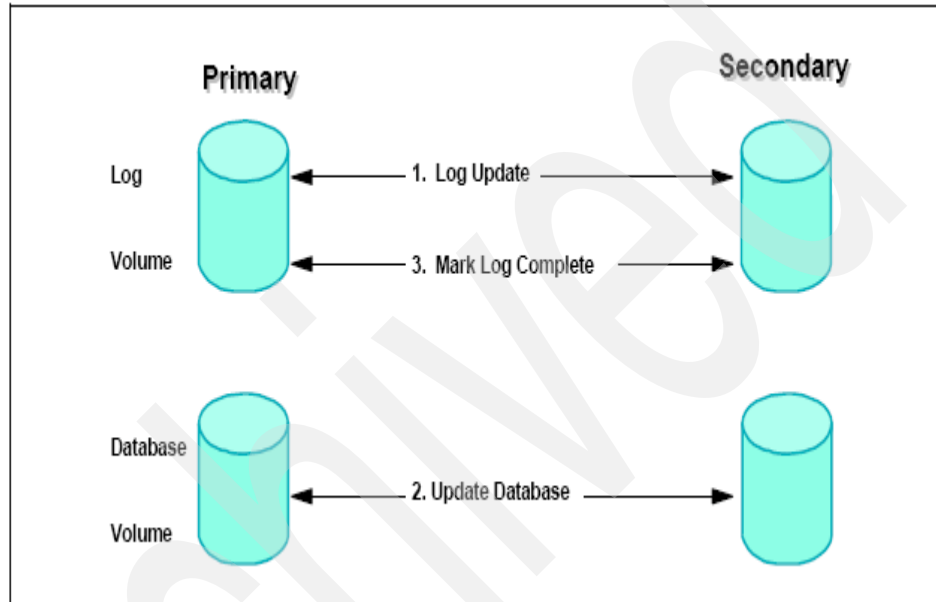


Figure B-4 Database logging

The order of *dependent writes* across volumes must be maintained at remote locations. Failure to do so results in data corruption and introduction of data inconsistency.

In Figure B-5 on page 191 we illustrate this concept with an example. The intention to update the database is logged in the database log files at both the primary and secondary volume (step 1). The database data file is updated at the primary volume, but the update does not reach the remote volume that contains the mirrored data file. The primary location is not aware of the write failure to the secondary volume (step 2). The database update is marked complete in the log files at both the primary and remote locations (step 3). The result is that the secondary site log files say that the update was done, but the updated data is not in the database at the secondary location. There is no way to know that the data was corrupted.

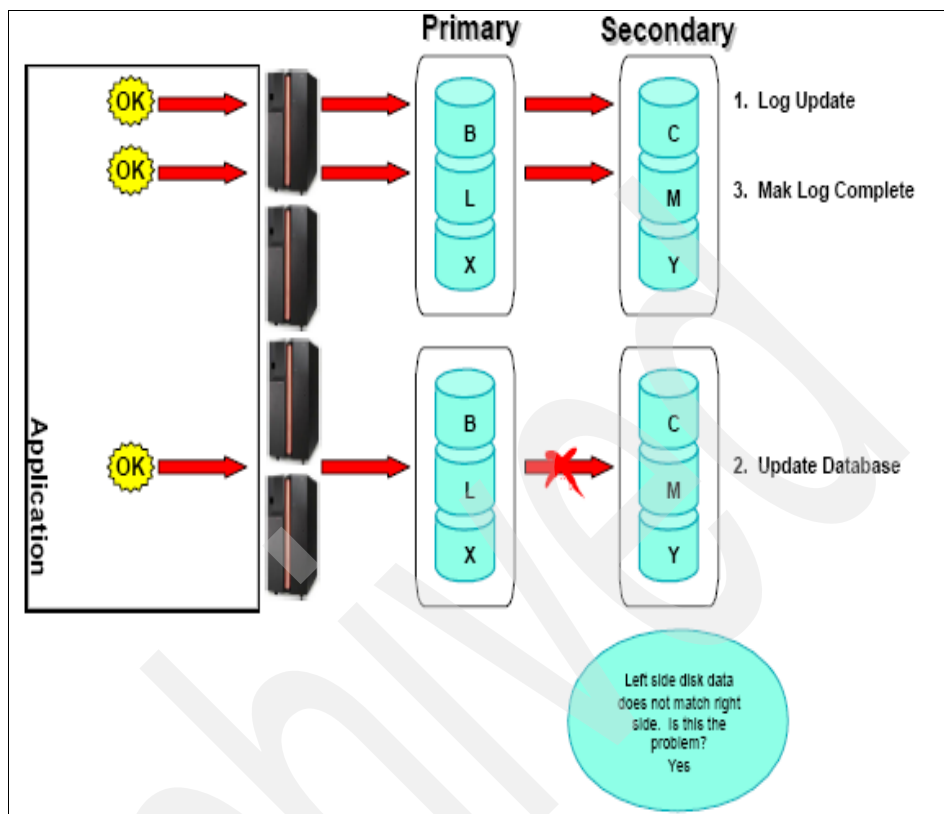


Figure B-5 Dependent writes

So, the issue is that the disk subsystem cannot by itself perform the steps required to avoid the rolling disaster system problem.

For this reason, we strongly recommend, at a minimum implementing *consistency groups* in any mirroring solution. Consistency groups are an implementation of technology that assists with the consistency of application data capable of dependent writes. To guarantee a fully consistent remote copy, multiple volumes require a consistency group functionality. ESS, DS6000™, and DS8000™ Metro/Global Mirror microcode already have the consistency group function for both System z and open systems.

If any volume within a consistency group cannot complete a write to its counterpart in the remote mirror relationship, an *extended long busy* (ELB) will be issued, preventing further writes to any of the volumes within the consistency group. This wait period is the perfect time to issue a freeze to all volumes involved to maintain consistency. If a write cannot complete, the storage system will not back out incomplete transactions on its own. Instead, the application will

need to recognize that the transaction was incomplete and take the appropriate actions. Once the storage system pauses the application I/O to the affected primary volumes, the write dependent mechanism of the application prevents the Metro/Global Mirror secondary volumes from becoming inconsistent.

### ***Rapid data recovery for System z***

When used in a two-site implementation, the GDPS/PPRC HyperSwap Manager can be a tier 6 rapid data recovery solution for disaster recovery situations. It is not a tier 7 solution because it lacks the recovery automation provided by a full GDPS/PPRC implementation.

Rapid data recovery for System z is provided by an IBM Global Services service offering, Geographically Dispersed Parallel Sysplex (GDPS) HyperSwap Manager, in the GDPS suite of offerings. It uses IBM System Storage Metro Mirror (previously known as Synchronous PPRC) to mirror the data between disk subsystems. Metro Mirror is a hardware-based mirroring and remote copying solution for the IBM System Storage DS6000, DS8000, and SVC systems.

Further information can be found on the GDPS Web site:

<http://www.ibm.com/systems/z/gdps/>

### ***IBM System Storage SAN Volume Controller***

IBM System Storage SAN Volume Controller (SVC) creates a virtual pool of storage so that it appears as one logical device to centrally manage and to allocate capacity as needed. It also provides one solution to help achieve the most effective on demand use of your key storage resources. The SVC addresses the increasing costs and complexity in data storage management by shifting the storage management intelligence from individual SAN controllers into the network by using virtualization.

SVC falls under tier 4 for FlashCopy® and tier 6 for Metro Mirror.

### ***Continuous availability solutions (tier 7)***

Tier 7 solutions are distinguished by their built-in automation capabilities

Here we briefly describe a continuous availability solution for IBM System z (GDPS).

### ***Geographically Dispersed Parallel Sysplex (GDPS)***

GDPS is a family of IBM Global Services offerings for a single or multi-site environment, which provides an integrated, end-to-end solution for enterprise IT Business Continuity, integrating software automation, servers, storage, and networking.



GDPS automation provides the capability to manage the remote copy configuration and storage subsystems, automate System z operational tasks, manage and automate planned reconfigurations, and do failure recovery from a single point of control. GDPS offerings include tier 7 and tier 6 recovery capability.

The GDPS family of System z Business Continuity solutions consists of two major offering categories, with sub-offerings in each category. They are:

- ▶ GDPS/PPRC solutions, based on IBM System Storage Metro Mirror (formerly PPRC), including:
  - GDPS/PPRC, a tier 7 solution
  - GDPS/PPRC HyperSwap Manager, a tier 6 solution
  - RCMF/PPRC, a remote copy management solution for PPRC
- ▶ GDPS/XRC solutions, based on System Storage z/OS Global Mirror (formerly XRC), including:
  - GDPS/XRC, a tier 7 solution
  - RCMF/XRC, a remote copy management solution for XRC

### **GDPS/PPRC overview**

GDPS/PPRC is designed as a *continuous availability and disaster recovery solution*. Metro Mirror (PPRC) hardware disk mirroring synchronously mirrors data that reside on a set of disk volumes, called the primary volumes, to secondary disk volumes in a second system.

The physical topology of GDPS/PPRC, shown in Figure B-6, consists of a System z base or Parallel Sysplex cluster spread across two sites that are separated by up to 100 km/62 miles of fiber, with one or more z/OS systems at each site.

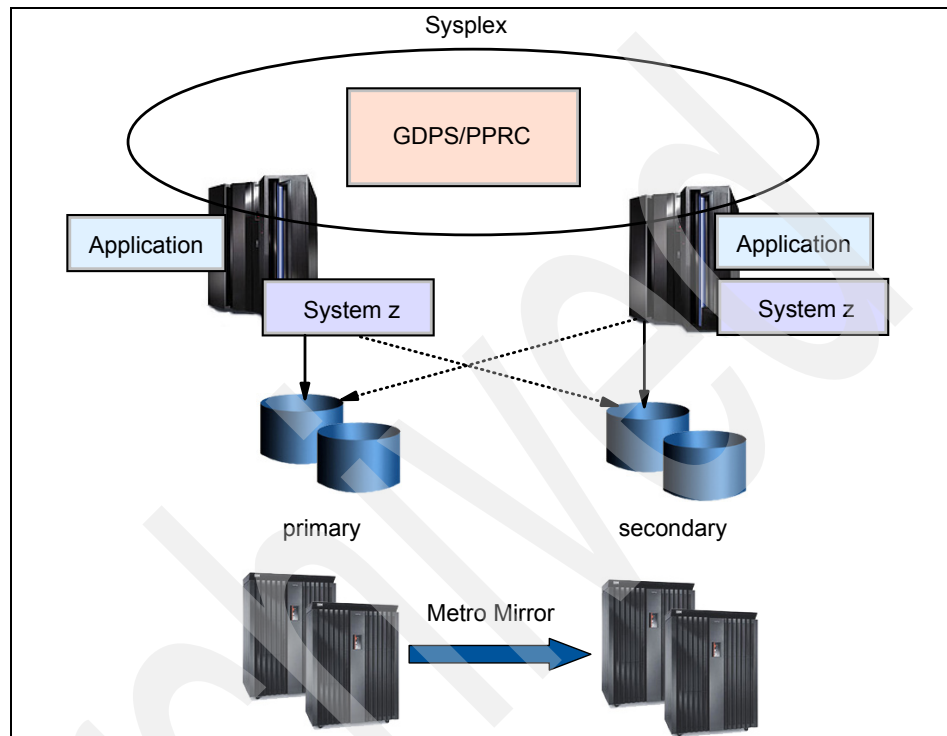


Figure B-6 GDPS/PPRC topology

With GDPS/PPRC, you can perform a controlled site switch for both planned and unplanned site outages, with no or minimal data loss, maintaining full data integrity across multiple volumes and storage subsystems and the ability to perform a normal DBMS restart (not DBMS recovery) in the second site. GDPS/PPRC is application independent, and therefore can cover the complete application environment.

### ***Near continuous availability of data with HyperSwap***

GDPS in the Metro Mirror environment provides the HyperSwap functionality. The HyperSwap function can help significantly reduce the time needed to switch to the secondary set of disks while keeping the z/OS systems active, together with their applications. In this case, HyperSwap broadens the near continuous availability attributes of GDPS/PPRC by extending the Parallel Sysplex redundancy to disk subsystems. GDPS/PPRC has a major sub-offering named

GDPS/PPRC HyperSwap Manager, described in “GDPS/PPRC HyperSwap Manager overview” on page 196. This product extends Parallel Sysplex availability to disk subsystems, even if multiple sites are not available and the Parallel Sysplex is configured in only one site.

### ***Management of System z operating systems***

In addition to managing images within the base or Parallel Sysplex cluster, GDPS can manage a client's other System z production operating systems, including z/OS, Linux for System z, z/VM, and VSE/ESA™. For example, if the volumes associated with the Linux for System z images are mirrored using Metro Mirror, GDPS can restart these images as part of a planned or unplanned site reconfiguration. The Linux for System z images can either run as a logical partition (LPAR) or as a guest under z/VM.

### ***Management for open systems Logical Unit Numbers (LUNs)***

GDPS/PPRC technology has been extended to manage a heterogeneous environment of z/OS and Open Systems data. If installations share their disk subsystems between the z/OS and Open Systems platforms, GDPS/PPRC can manage a common Metro Mirror Consistency Group (described in “Data consistency for database environments” on page 189) for both System z and open systems storage, thus providing data consistency across both z/OS and Open Systems data. This allows GDPS to be a single point of control to manage business resiliency across multiple tiers in the infrastructure, improving cross-platform system management and business processes.

### ***Multi-Platform Resiliency for System z***

GDPS/PPRC has been enhanced to provide a new function called GDPS/PPRC Multi-Platform Resiliency for System z. This function is especially valuable for clients who share data and storage subsystems between z/OS and z/VM Linux guests on System z, for example, an application server running on Linux on System z and a database server running on z/OS. GDPS/PPRC provides the reconfiguration capabilities for the Linux on System z servers and data in the same manner as for z/OS systems and data.

In summary, GDPS/PPRC is capable of the following attributes:

- ▶ Near continuous availability solution for z/OS
- ▶ Near transparent business continuity solution for z/OS
- ▶ Common point of control for recovery of a mixed System z and Open Systems business continuity environment
- ▶ Recovery time objective (RTO) less than an hour
- ▶ Recovery point objective (RPO) of zero (optional)

- Protects against localized area disasters (distance between sites limited to 100 km fiber)

### ***GDPS/PPRC HyperSwap Manager overview***

The GDPS/PPRC HyperSwap Manager solution is a subset of the full GDPS/PPRC solution, designed to provide a affordable entry point to the full family of GDPS/PPRC offerings by providing a rapid data recovery solution for enterprise disk-resident data.

GDPS/PPRC HyperSwap Manager (GDPS/PPRC HM) does this by offering a subset of the full GDPS/PPRC, specifically, the HyperSwap management and Metro Mirror management capabilities.

GDPS/PPRC HyperSwap Manager can provide either of the following two configurations:

- Near continuous availability of data within a single site  
GDPS/PPRC HyperSwap Manager is designed to provide continuous availability of data by masking disk outages that are caused by disk maintenance or failures. For example, if normal processing is suddenly interrupted when one of the disk subsystems experiences a hard failure, thanks to GDPS, the applications are masked from this error because GDPS detects the failure and autonomically invokes HyperSwap. The production systems continue to use data from the mirrored secondary volumes. Disk maintenance can also be similarly performed without application impact by executing the HyperSwap command.
- Near continuous availability of data and disaster recovery solution at metro distances  
In addition to the single site capabilities, in a two-site configuration, GDPS/PPRC HyperSwap Manager provides an entry-level disaster recovery capability at the recovery site, including the ability to provide a consistent copy of data at the recovery site from which production applications can be restarted. The ability to simply restart applications helps eliminate the need for lengthy database recovery actions.

The GDPS HyperSwap Manager offering features specially priced, limited function Tivoli System Automation and NetView® software pricing, thus enabling a more affordable entry point into the full GDPS automation software.

GDPS/PPRC HyperSwap Manager can be upgraded to full GDPS/PPRC at a later time, preserving the implementation investment and GDPS skills and procedures that have already been developed.

### ***GDPS/XRC overview***

GDPS/XRC has the attributes of a disaster recovery solution. z/OS Global Mirror (XRC) is a combined hardware-asynchronous and software-asynchronous remote copy solution. The application I/O is signaled completed when the data update to the primary storage is completed. Subsequently, a DFSMSdfp™ component called System Data Mover (SDM) that is typically running in the recovery site, asynchronously offloads data from the primary storage subsystem's cache and updates the secondary disk volumes.

GDPS/XRC has the following attributes:

- ▶ Disaster recovery solution
- ▶ RTO between an hour and two hours
- ▶ RPO less than two minutes, typically 3–5 seconds
- ▶ Protects against localized as well as regional disasters (distance between sites is unlimited)
- ▶ Minimal remote copy performance impact

### **GDPS/XRC topology**

The physical topology of a GDPS/XRC configuration, as shown in Figure B-7, consists of production systems in one site. The production systems could be a single system, multiple systems that are sharing a disk, or a base or Parallel Sysplex cluster. The recovery site can be located at a virtually unlimited distance from the production site.

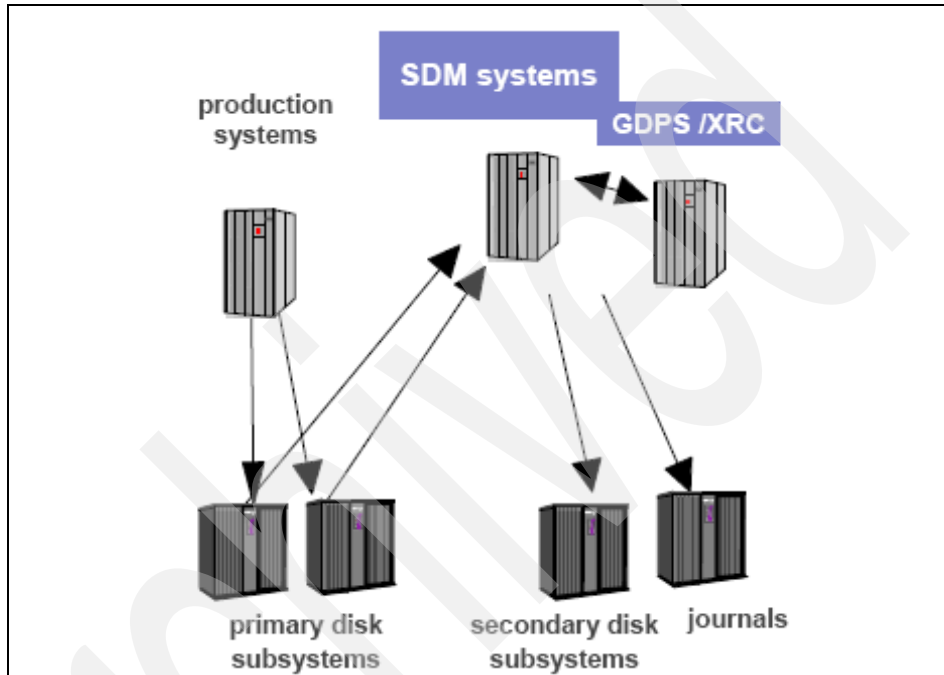


Figure B-7 GDPS/XRC topology

In GDPS/XRC, the production systems located in the production site can be a single system, multiple systems that are sharing disks, or a base or Parallel Sysplex cluster. GDPS/XRC provides a single, automated solution designed to dynamically manage storage subsystem mirroring (disk and tape) that allows a business to attain *near transparent* disaster recovery with minimal data loss. With GDPS/XRC, you can perform a controlled site switch for an unplanned site outage, and maintain data integrity across multiple volumes and storage subsystems. You then only need to perform a normal DBMS restart in the recovery site, rather than a DBMS recovery. GDPS/XRC is application independent and therefore can cover the client's complete application environment.

### ***Additional GDPS information***

For additional information about GDPS solutions or GDPS solution components, refer to these Web sites:

- ▶ GDPS home page  
<http://www.ibm.com/servers/eserver/zseries/gdps/>
- ▶ zSeries® Business Resiliency Web site  
<http://www.ibm.com/systems/z/resiliency/>
- ▶ Also, refer to these IBM manuals and IBM Redbooks:
  - *IBM eServer™ zSeries: Device Support Facilities User's Guide and Reference R17*, GC35-0033
  - *IBM eServer zSeries: z/OS V1R6.0 DFSMS Advanced Copy Services*, SC35-0428
  - *GDPS Family - An Introduction to Concepts and Capabilities*, SG24-6374

Archived



# Related publications

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

## IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 202. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *IBM System z Strengths and Values*, SG24-7333
- ▶ *IBM System z Connectivity Handbook*, SG24-5444
- ▶ *IBM eServer zSeries 990 Technical Guide*, SG24-6947
- ▶ *GDPS Family An Introduction to Concepts and Capabilities*, SG24-6374
- ▶ *Achieving the Highest Levels of Parallel Sysplex Availability*, SG24-6061
- ▶ *Architecting High Availability Using WebSphere V6 on z/OS*, SG24-6850

## Other publications

These FIS publications are relevant as further information sources:

- ▶ *FIS Fraud Navigator Installation and Configuration Guide*
- ▶ *DataNavigator Server Installation Guide*
- ▶ *EnterpriseView Installation Guide*
- ▶ *Connex System Installation Guide*

These IBM publications are also relevant:

- ▶ *Setting up a Sysplex*, SA22-7625
- ▶ *DB2 Version 9.1 for z/OS Installation Guide*, GC18-9846
- ▶ *DB2 UDB for z/OS: Design Guidelines for High Performance and Availability*, SG24-7134

## Online resources

These Web sites are relevant as further information sources:

- ▶ FIS financial services (Search on the product or term of interest.)  
<http://www.fidelityinfoservices.com/fnfis/>
- ▶ FIS System z announcement  
<http://www.fidelityinfoservices.com/fnfis/newsroom/20070301.htm>
- ▶ z/OS UNIX ported tools  
<http://www.ibm.com/servers/eServer/zseries/zos/unix/bpxalty1.html>
- ▶ IBM financial industry offerings  
<http://www-03.ibm.com/industries/financialservices/doc/content/partner/931731103.html/>

## How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)

# Index

## A

- Account lifecycle management services 16
- ACH 29
- acquirer/issuer activity 32
- acquisition 6
- agent bank 9
- alteration 48
- APF authorization 77
- architecture 14
- ATM
  - management 30
  - status 32
- ATM and POS device driving 23
- ATM self-service banking 23
- authentication 6
- authorization 5, 23
  - processor 131
- auto balance 30
- auto reconciliation 29
- Automated Banking Machine (ABM) 5
- Automated Teller Machine (ATM) 5, 9
- automatic recovery sequence 145
- Automatic Restart Manager (ARM) 47, 62, 143
- availability 12, 48, 57, 121
  - continuous 48

## B

- back end 6, 56
- back-office 21, 28, 30, 47
- backup and recovery 47
- best practices 121
- bill payment services 9
- bridging payments 16
- buffer pools 84
- business challenges 3
- business insight 16

## C

- card authorization, activation, and production 9
- card disablement 24
- cardholder
  - notification 24

- services 16
- cash management 30
- central processor complex (CPC) 59
- centralized data storage 46
- certification (CERT) testing 57
- checks, paper 3
- clearing 6
- communication handler 34, 131
- compatibility 45
- competitive advantage 19
- configuration repository (CR) 84
- configurations, customer 66
- CONNEX 104
- Connex on IBM 22
  - auto restart 155
  - best practices 123
  - customization 103
  - high availability 59
  - installation 74
  - maintenance 109
  - processes to be monitored 130
- CONNEXS 104
- consolidation 10
- continuous availability 48
- continuous operations 49
- continuous reliable operation (CRO) 50
- Coupling Facility (CF) 59
- CrossCheck 95
- customer
  - configurations 66
  - cross-sell 17
  - interaction layer 35
  - relationship 10
  - service 28
  - services 15
  - up-sell 17

## D

- DASD 83
  - multipathing 144
  - requirements 75
  - tuning 124
- dashboard 31

- data
  - capture 23
  - integrity 12
- database administrator (DBA) 56
- DataNavigator 27
  - auto restart 155
  - customization 106
  - high availability 63
  - installation 82
  - maintenance 110
  - troubleshooting 132
- DB2 35, 44
  - data sharing 60, 144
  - Performance Expert 125
  - tuning 125
- debugging 134
- decision-making 28
- Delphi 95
  - client 85
- deposit sharing program 9
- deposit verification 30
- design decisions 55
- device inquiry 30
- device problem management 23
- DFSMS storage groups 61
- disk multipathing 144
- downtime, cost of 59
- DUMP 134
- dynamic workload balancing 11

## E

- eavesdropping 48
- EBA Step 2 11
- eCommerce 3, 23
- EFT systems 28
- e-mail notification 29
- encryption 48
- endpoints 34
- Enterprise Control 117
- enterprise naming conventions 75, 83
- enterprise payments 19
- Enterprise Payments framework 21
- enterprise payments system 3
  - high availability 59
  - operations 15
- EnterpriseView 30, 117
  - auto restart 155
  - customization 109

- high availability 64
- installation 92
- maintenance 111
- troubleshooting 133
- ESCON directors 60
- exception management 29
- Extended Remote Copy (XRC) 61

## F

- Faster Payments initiative 3
- FedWire 11
- FICON switches 60
- Fidelity National Information Systems (FNIS) 19
- finance industry
  - requirements 11
  - use of System z 46
- flexibility 17
- FNIS Connex on IBM 22
  - installation 74
- FNIS credentials 20
- FNIS DataNavigator 27
  - exception management 29
- FNIS Enterprise Payments Framework 21
- FNIS Enterprise Payments system overview 33
- FNIS EnterpriseView 30
- FNIS Fraud Navigator 23
  - auto restart 155
  - customization 107
  - high availability 62
  - installation 87
  - maintenance 111
  - troubleshooting 133
- FNIS monitoring tools 114
- FNIS product installation 73
- FNIS value proposition 20
- fraud 10
  - alert generation 24
  - alerts 26
  - detection 25
  - management 10, 14, 16
  - patterns 27
  - prevention 9
- fraud analyst productivity 24
- Fraud Navigator, FNIS 23
- front end 6, 37

## G

- gateway connections 9

GDPS 57  
Geographically Dispersed Parallel Sysplex (GDPS)  
51  
global banking industry 3

## H

hand-off 8  
hardware sizing 75  
Hierarchical Storage Manager (HSM) 75  
high availability 48, 140  
high availability configuration 156  
    Parallel Sysplex 158  
    Parallel Sysplex+GDPS/PPRC/HyperSwap  
    158  
    Parallel Sysplex+GDPS/XRC 160  
    standalone z/OS system 157  
high availability tests 139  
holistic approach 24

## I

I/O symmetry 60  
IBM monitoring tools 118  
IBM WebSphere MQ 40  
insights 21  
installation 73  
    steps 84, 88, 93  
installation verification 79  
integration planning 56  
Internet-based theft 4  
ISO 20022 11  
ISO 8583 37

## L

Linux for System z 52  
Logger 130  
logon procedure 77  
logs 41, 76, 83  
LOGUPRI 104  
LOGUSEC 104

## M

MACLIB 77  
mainframe solution  
    benefits 43  
maintenance 109  
merchant 6  
merchant acquirer 9

Message Delivery System (MDS) 35  
mission-critical applications 46  
Mobius 85  
monitoring 113  
    performance 118  
monitoring and management 23  
monitoring tools  
    FNIS 114  
    IBM 118  
MQ 40

## N

National Bridge Transactions 7  
network 76, 84  
    access 23  
    high availability 62  
Network On Us 7

## O

On Us 6  
operational view 36  
operations 3, 28  
optimization 121  
optimizing 4  
outages  
    limiting 23  
    none 25  
    planned 139  
    reducing 62  
    unplanned 139, 144

## P

packaged rules 24  
paper checks 3  
Parallel Sysplex 44, 50, 140  
payment czars 30  
payments landscape 10  
PDS vs. PDSE 77  
Peer-to-peer Remote Copy (PPRC) 61  
performance 59, 121  
    management 47  
    monitoring 118  
Person to Person 3  
PIN-secured debit 9  
planned outages  
    avoiding 49  
Point of Sale (POS) 5

problem determination 129  
product installation 73

## **R**

RACF 75  
RAID technologies 51  
Reciprocal Transaction 7  
reconciliation 6  
recoverability 58  
recovery 47  
recovery point objective (RPO) 58  
recovery time objective (RTO) 58  
Redbooks Web site 202  
    Contact us xiii  
reference architecture for enterprise payments 14  
regulatory 10  
regulatory pressure 43  
reliability 11, 45  
reliability, availability, and serviceability (RAS) 50  
Research & Decisioning 26  
Resource Measurement Facility (RMF) 118  
Resource Recovery Services (RRS) 47  
responsibilities 56  
responsiveness 28  
retail payments system 13  
risk management 9, 16  
RMF 76  
roles 56  
round the clock settlement 28  
rules  
    configuration 24  
    creation and management 24  
    packaged 24

## **S**

SAN 75  
scalability 59  
Secure Sockets Layer (SSL) 48  
security 76, 83  
serviceability 50  
SETLPRI 104  
SETLSEC 104  
SETTLE 105  
settlement 6  
settlement position 16, 32  
signature debit 9  
simplicity 17  
single point of failure (SPOF) 59

single points of failure (SPOF) 64  
skills 56, 82, 88, 93  
SNA

    HCH 40  
software developer 56  
stability 45  
stand-in authorization 9  
stored value card processing 9  
strategic insight 19  
SWIFT 11  
switching systems 8  
System Health Monitor 114  
system managed storage (SMS) 74  
System z 11, 17, 21  
    Application Assist Processor (zAAP) 122  
    architecture 45  
    benefits 43  
    hardware 59  
    high availability 140  
    value 44  
system-managed storage (SMS) 57

## **T**

TCP/IP  
    HCH 40  
technology 3  
Terminal handler and processor interface 131  
threats 4  
three-layered services oriented architecture 15  
Tivoli 44  
TRACE 105, 134  
transaction  
    national bridge 7  
    network on us 7  
    on us 7  
    reciprocal 7  
transaction analysis 23  
transaction management layer 35  
transaction research 28  
transaction services 15  
transaction switching 23  
troubleshooting 130  
    application delays 134  
TSO logon procedure 77

## **U**

user acceptance testing (UAT) 123

## **V**

value proposition 20  
velocity goals 78  
verification of overall solution 95  
view packs 32  
virtual delivery channel 4  
VSAM 35  
    performance 125  
VTAM 78

## **W**

WebSphere 44, 143  
WebSphere Application Server (WAS) 63  
WebSphere MQ 40, 62  
    shared queue 144  
wholesale 3  
wire transfer 29  
WLM 76  
workload management 17, 46  
Workload Manager (WLM)  
    tuning 126

## **X**

X66496 8  
XCF 77

## **Z**

z/OS performance monitoring 118  
z/OS UNIX 62  
zAAP 122  
zPARMS 84





Archived







# Fidelity National Information Systems Payments Reference Architecture for IBM System z



## Introduction and architecture

## Installation and configuration

## High availability tests and techniques

The number of electronic transactions conducted by financial institutions around the world every day is growing astronomically. Today payments can account for a significant portion of a bank's total cost. The ability to move money quickly and safely and to manage the associated data management flows are necessary components of banking. Banks must also gain critical insights from their data to stay steps ahead of Internet-based theft of bank and customer information. These thefts can dangerously undermine the efficiencies of more cost-effective virtual delivery channels. On the positive side, this data can provide key insights into the way their customers behave and the products and services that they are likely to buy.

The FIS approach to enterprise payments puts the consumer account at the core, surrounded by rich functionality that provides access via multiple channels, extensive options for managing customer service, as well as a range of integrated fraud management tools. A complete payments solution for issuing and acquiring, clearing and settlement, FIS Enterprise Payments also contributes unique insight into the customer relationship.

This IBM Redbooks publication describes the FIS Enterprise Payments offerings and how to implement them. The book is written for decision makers and financial solutions architects, and assumes a basic knowledge of payments systems.

## INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

## BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**  
[ibm.com/redbooks](http://ibm.com/redbooks)