PETROLEUM AUTHORITY OF THAILAND (PTT) COMPUTER DISASTER RECOVERY SYSTEM

by

Mr. Samatcha Kowitwongsa

A Final Report of the Three - Credit Course
CE 6998 Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer and Engineering Management
Assumption University

July, 2000

PETROLEUM AUTHORITY OF THAILAND (PTT) COMPUTER DISASTER RECOVERY SYSTEM

by
Mr. Samatcha Kowitwongsa

A Final Report of the Three-Credit Course
CE 6998 Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer and Engineering Management
Assumption University

July 2000

Project Title	Petroleum Authority of Thailand (PTT) Computer Disaster Recovery System
Name	Mr. Samatcha Kowitwongsa
Project Advisor	Asst.Prof.Dr. Vichit Avatchanakorn
Academic Year	July 2000

The Graduate School of Assumption University has approved this final report of the three-credit course, CE 6998 PROJECT, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer and Engineering Management.

Approval Committee:

(Asst.Prof.Dr. Vichit Avatchanakorn)
Advisor

(Prof.Dr. Srisakdi Charmonman)
Chairman

(Dr. Chamnong Jungthirapanich)
Dean and Co-advisor

(Asst.Prof.Dr. Boonmark Sirinaovakul)
Member

(Dr. Prapon Phasukyud)
Member

(Assoc.Prof. Somchai Thayarnyong)
MUA Representative

July 2000

ABSTRACT

This project examines the design and implementation of computer disaster recovery system (CDR) for Petroleum Authority of Thailand (PTT) for business continuity purpose. It focuses on computer systems related to core business of PTT.

The gathering of information is the first process in this project. The information gathered in this project includes current system infrastructure, constraint and specification of PTT system that is related to the goal of this project. After that, the CDR system is designed to match to the requirement by concerning about technical limitations that may happen. The CDR system helps the PTT to continue their business in event of disaster. In addition, the CDR system can be use for other profit such as training center, DR service provider as PTT's wish.

The project also includes system implementation and will discuss about cost analysis and feasibility study.

This project is the first complete CDR solution project in Thailand. It has been implemented since October 1999 and PTT has been using the system for disaster recovery until now. There is periodic testing every 6 months to ensure that the CDR system still works correctly.

I hope that this project will help anybody who wants to implement CDR at least for getting an idea. Thank you in advance.

ACKNOWLEDGEMENTS

I am indebted to the following people and organizations. Without them, this thesis would nct have been possible.

Especially, I would like to express and enlarge my deepest gratitude to Asst.Prof.Dr. Vichit Avatchanakorn whose valuable comments and recommendations eliminated many ambiguous points in this project. I would like to thank all my friends at the Computer and Engineering Management for their interest and support during the course of my work, especially Surapong Somrang and Norachet Saetang for the inspiring discussions which have many times clarified my ideas. I would like to thank Mr. Sutat, Mr. Worradej, Ms. Ajcharee and the staffs at IT Department, Petroleum Authority of Thailand for their help in establishing and finishing of this project and also Mr. Narongrit, Mr. David and Mr. Floyd for their supporting and knowledge.

Sp:cial appreciation is due to my family for their fervent and continuous encouragement. Above all, I am forever grateful to my parents whose willingness to invest in my future has enabled me to achieve my educational goal.

TABLE OF CONTENTS

Chapter Page

ABSTRACT i

ACKNOWLEDGEMENTS ii

LIST OF FIGURES vi

LIST OF TABLES vii

I. INTRODUCTION 1

1.1 Overview 1

1.2 Business Continuity Concept 3

1.3 Objectives of This Project 8

1.4 Scope of Work 8

1.5 Project Schedule 9

1.6 Solution Outline 9

II. SYSTEM ANALYSIS 12

2.1 Problem Statement 12

2.2 Terms of Reference 12

2.3 Current Systems 14

2.4 User Requirements Summary 21

III. CDR SYSTEM DESIGN FOR UNIX SYSTEMS 24

3.1 Overview 24

3.2 Disaster Recovery Systems 25

3.3 Standby Applications 27

3.4 CDR Design Assumptions 28

3.5 Standby Methodology Described 29

Chapter Page

3.6 Standby Database Design 34

IV. DETAIL DESIGN FOR NT CDR SYSTEMS 38

4.1 Overview 38

4.2 Working Hour Data Synchronization 38

4.3 End Day Data Synchronization 39

4.4 ARCserveIT Replication Architecture 39

4.5 Application Specific Detail Design 45

4.6 Application Programs Consideration 51

4.7 Day-to-day Backup Detail Design 52

4.8 Restore Design 54

4.9 Recover Design 56

V. CDR SYSTEM IMPLEMENTATION 58

5.1 Disaster Recovery Site Build 58

5.2 Disaster Recovery Site Services Build 61

VI. COST ANALYSIS 66

6.1 Overview 66

6.2 Costs 66

6.3 Revenues 73

6.4 Return on Investment (ROI) 74

6.5 Feasibility Analysis Study 74

VII. PROJECT SUMMARY 77

7.1 Benefits 77

7.2 Opportunities & Recommendations 77

7.3 Conclusions 79

Chapter Page

APPENDIX A ORACLE DATABASE CONCEPT 81

APPENDIX B SERVER INFORMATION 89

APPENDIX C APPLICATION INFORMATION 97

APPENDIX D UNIX SHELL SCRIPTS 102

APPENDIX E ARCSERVE IT SOFTWARE INSTALLATION 136

APPENDIX F DISASTER RECOVERY DIAGRAM 140

APPENDIX G LIST OF DELIVERABLE DOCUMENT 145

APPENDIX H DETAIL DESIGN FOR EACH NT APPLICATION 148

BIBLIOGRAPHY 179

LIST OF FIGURES

Figure Page

1.1 Business Continuity Life Cycle 4

1.2 CDR Project Gantt Chart 9

3.1 Archive Log Creation 35

3.2 Standby Database Architecture 36

4.1 NT CDR Replication 42

4.2 NT Critical Failure 44

4.3 Control Returned to Primary Server 45

6.1 Breakeven Point 76

F.1 CDR System Diagram 141

F.2 NT CDR System Diagram (1) 142

F.3 NT CDR System Diagram (2) 143

F.4 UNIX CDR System Diagram 144

LIST OF TABLES

Table | Page

2.1 Accessibility Requirements 14

2.2 Oracle Financials Data Centre Server 16

2.3 Oracle Financials DR Service Level Required 16

2.4 SAP/HR Data Centre Server (Production) 17

2.5 SAP/HR Data Centre Server (Development) 17

2.6 SAP/HR DR Service Level Required (Production) 17

2.7 SAP/HR DR Service Level Required (Development) 17

2.8 Data Warehouse Data Centre Server 18

2.9 Data Warehouse DR Service Level Required 18

2.10 MS Exchange Data Centre Server 21

2.11 MS Exchange DR Service Level Required 21

3.1 Standby Applications and their Clients 27

3.2 TOR Requirements 28

3.3 User Requirements 29

3.4 Standby Applications 30

4.1 CDR Servers 52

4.2 Restoration Methods 55

6.1 Probability of Occurrence of Security Event 69

6.2 Probable Economic Loss 70

6.3 Feasibility Analysis 75

B.1 Server Information 90

C.1 Application Information 98

Table Page

D.1 Lis. of UNIX Shell Scripts 103

E.1 NT Software Component Installation Table 137

G.1 List of Deliverable Documents 146

I. INTRODUCTION

1.1 Overview

1.1.1 Disaster Overview

A disaster is any unplanned event that causes business/MIS services to be interrupted to a point (hours to days) that users require alternate services to prevent a business loss. All the operational and planning aspects of a data processing facility, then, become an integral part of disaster recovery (business continuity planning) when an unplanned event shuts down the business, disrupts operations, or causes physical damage to the facility.

The various types of disaster to consider include:

Natural disasters:

(1) Floods

(2) Winter storms

(3) Forest fires

(4) Earthquakes

(5) Hurricanes

(6) Tornadoes

Man-made disasters:

(1) Building fires

(2) Transportation accidents

(3) Chemical accidents

(4) Sabotage or willful destruction

(5) Bomb threats

(6) Burst pipes

(7) Building collapse

Political disasters:

(1) Riots, strikes, and civil disturbances

(2) War and nuclear attack

The term disaster usually refers to a large-scale event or natural disaster, but each event must be considered within the context of the impact it has on the data center and the company's business. What may be considered a nuisance to a large processing facility could be deemed a disaster to a small data center. This is why it is the responsibility of the company to define what is considered a disaster within context of their business practices and what is an acceptable business loss.

1.1.2 PTT CDR Project Overview

The Computer Disaster Recovery (CDR) project is intended to give Petroleum Authority of Thailand (PTT) a disaster recovery capability for certain business applications. Disaster recovery capability means having the ability to transact against mission critical applications on an alternate server located at a remote site when there is a loss of the primary server. Each application included in the scope of the CDR project has recovery parameters that PTT has given to the development team. This document identifies the mission critical applications, their recovery parameters, and a general design guideline for each server.

There are two CDR sites. The primary CDR site is located on 13th floor PTT Headquarter Building, Viphawadee-Rangsit, Bangkok and the backup site is located at PTT research and training center, Wang-Noi, Ayuttaya. The distance between the two sites are around 60 Kilometers.

1.2 Business Continuity Concept

1.2.1 Introduction of Business Continuity

One reason for this increased vulnerability is the growth of distributed systems and PC Servers. In the past, when all computer systems were in glass houses, managers could locate those facilities in geographically safe places. They could put in raised floors, install computer-safe-fire-suppression systems, and add backup power to run mission-critical systems for days or weeks if necessary.

1.2.2 Business Continuity Life Cycle

The first steps in serious DR planning are a risk assessment based on an inventory of IT resources and analysis of how disasters could affect those resources. Ideally, the assessment should include a business impact analysis and a business resumption plan, in which the company determines how a disaster could impact the business (not just the IT infrastructure) and how to get back to work again.

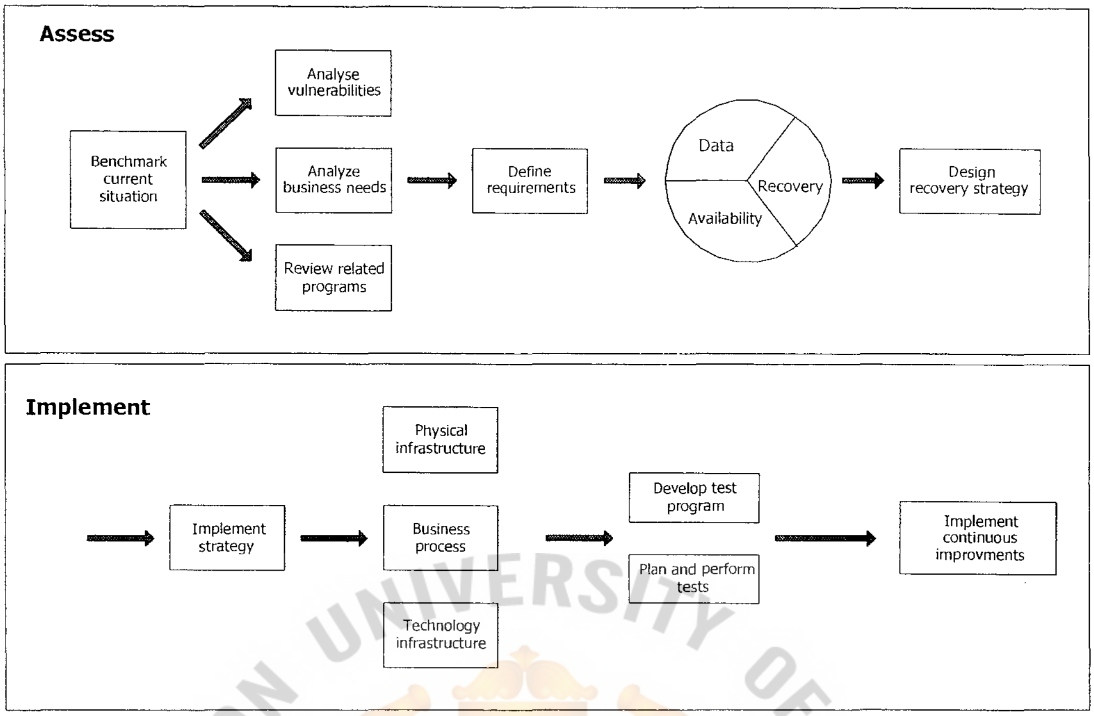


Figure 1.1. Business Continuity Life Cycle.

Plans should also cover general crisis management, including employee safety and how to cooperate with police, medics, fire fighters, and other emergency workers. The next step is to determine an acceptable recovery window for each network or system. Can a system be down for 48 hours, or is one hour too much? When data is involved, a recovery point also has to be determined; that is, can the business afford to lose a day's transactions?

Using recovery time and recovery point objectives, companies must then decide on a technical approach to disaster recovery, ranging from tape backup to electronic vaulting to mirroring.

1.2.3 Backup by Tape

The traditional DR plan calls for a full daily backup to tape. After backup, the tapes are shipped to a safe site. The theory is that if a disaster occurs, the tapes will be shipped to an alternate site, loaded onto a computer, and the business will be up and running again.

Such a plan yields a recovery point of 24 hours maximum (if the disaster hits just before you were going to ship that night's tapes). It might yield a recovery window of 48 hours: one day to ship the tapes to the alternate site and get them loaded, and another day to fiddle with networks, troubleshoot, and actually get things working. The 48-hour estimate assumes a "hot" alternate site, where the entire computer and network infrastructure is already installed and only needs to be configured and fired up.

The advantages of tape backup include relatively low costs and tried-and-true procedures. However, for many businesses, 24 hours represents too much data loss and 24 hours to 48 hours is too much downtime.

To cut the downtime associated with tape backup, one possible enhancement is a journaling file system or database that records all changes. Journal files can be sent offsite electronically or even written directly to a remote site. This yields a more recent recovery point because after the tape is loaded it can be updated with all the journaled transactions; only transactions that were processing when the computer went down may be lost.

Even with journaling, the recovery window remains the same because tapes still have to be shipped and loaded. To reduce the recovery window, companies may look to electronic vaulting for some system.

1.2.4 Electronic Vaulting

With electronic vaulting, data is backed up over a network to a remote site. Some companies use vaulting as a more automated and convenient (and perhaps more reliable) way to do nightly backups. Others use it to back up more frequently. Typical recovery intervals range between two hours and 12 hours.

The recovery window may also be reduced by having a computer at the vaulting site to step in and fulfill some or all of the functions of a downed machine. This method also covers situations in which the disaster disrupts critical transportation systems.

The replacement computer may be the "catcher" that receives the vaulting data from several other machines. To fulfill the role of temporary substitute, the catcher also has to have the necessary application software loaded to substitute for a machine that goes down.

Performance may suffer if the catcher has less horsepower or bandwidth than the downed machine, and some functions may not be supported if the catcher isn't equipped with all the same software and hardware as the computer that was lost. However, service won't be totally interrupted for any extended period of time. This approach is economical because it minimizes the number of computers sitting around waiting for something to go wrong-yet it may be able to provide partial recovery within minutes.

Another approach is to have the DR service provider bring a mobile data center or work space to your location, with machines already loaded with your data, or to drop ship those machines to you so you can set them up. With a mobile data center, you can often achieve a recovery window of 48 hours. Drop shipping usually takes longer (perhaps 96 hours), since you have to install the machines.

Companies can reduce the backup time for databases by using techniques that back up only updates and additions, which are usually a tiny fraction of the entire database. This method also reduces the required network bandwidth, which is usually the largest cost of a vaulting solution.

However, electronic vaulting is usually at least twice as expensive as tape backup and often proves to be three or four times as expensive.

1.2.5 Mirror Method

For many companies, a 12 hour to 24 hour recovery window is unacceptable. Such short recovery windows are typically impossible with vaulting alone; although the backup storage device is often dedicated to a single protected system, there's not necessarily a standby system dedicated to the protected system.

To achieve a recovery window of less than an hour, it is necessary to mirror data to an identical, dedicated system. A probe can then be installed on the protected machine that continuously sends "I'm OK" messages to the mirroring machine. If the probe sends out a distress call or simply stops sending messages, the mirroring machine kicks in.

Theoretically, this provides instantaneous, transparent recovery. However, in case of a large-scale emergency, there's always a good chance that something will prevent instant recovery. For instance, telecommunications or Internet services may be affected, client software may have trouble with the changeover, or key personnel may be running for the hills. Nevertheless, with proper planning, it is often possible to get the mirroring system up and running within an hour.

Companies can achieve an even greater level of protection by continuously operating both servers in parallel. In this setup, clients are transparently directed to which ever server happens to be most available at the moment. This multi-path solution also offers higher performance during normal operations. On the downside, it consumes expensive WAN bandwidth, as well as management resources and power at the backup site. Even with all this expense, a few transactions may still be lost.

1.3 Objectives of This Project

(1) The objective of this project is to design and implement a disaster recovery solution for PTT's information systems that meets the following requirements:

(2) Manageability---PTT needs to be able to simply manage the routine operations required in the disaster recovery facility.

(3) Scalability---Systems must scale easily both in performance and capacity in order to accommodate fast-changing conditions and growth rates.

(4) Accessibility---Users must be able to access their authorized data and applications quickly and reliably as defined by PTT's disaster recovery goals.

(5) Measurability---PTT must be able to measure the effectiveness of the disaster recovery infrastructure.

1.4 Scope of Work

There are many issues for disaster recovery but for this project, we will focus on disaster for computer system only.

This approach has three phased, comprehensive solution based on PTT's stated requirements. The overall project will consist of these three phases:

(1) System Analysis---to acquire the system infrastructure, constraint and specification of PTT system related to the goal of CDR project.

(2) System Design---design the DR system including the operation procedure when having the disaster at both primary site and backup site.

(3) System Implementation---consists of the following three parts:

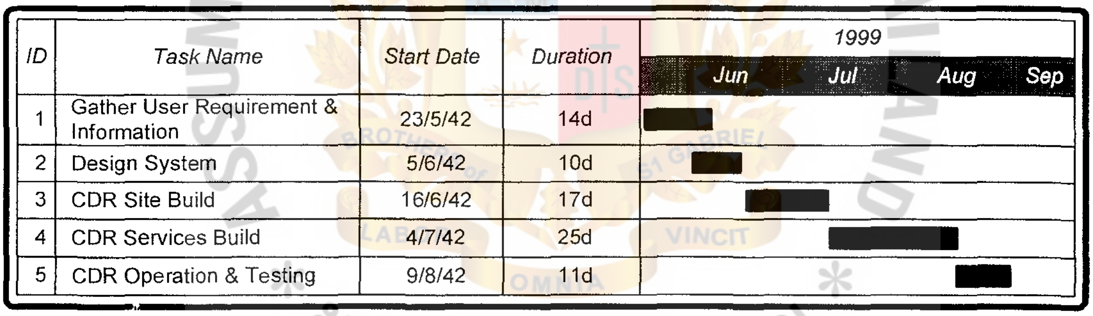
(a) Disaster recovery site build: This part is to prepare the DR site by installing the hardware, system configuration and network configuration.

(b) Disaster recovery operation services build: After the first part, the next operation is to copy all of primary site to the backup site including configuration the DR services and Fail-over procedure

(c) Disaster recovery operations testing and acceptance: To ensure that the CDR is worked to meet the PTT's requirement, testing is required for this step.

1.5 Project Schedule

Project Schedule details are identified in the Gantt Chart as the following:



ID	Task Name	Start Date	Duration	1999
				Jun Jul Aug Sep
1	Gather User Requirement & Information	23/5/42	14d	
2	Design System	5/6/42	10d	
3	CDR Site Build	16/6/42	17d	
4	CDR Services Build	4/7/42	25d	
5	CDR Operation & Testing	9/8/42	11d	

Figure 1.2. CDR Project Gantt Chart.

1.6 Solution Outline

The project will be implemented in a phase approached based on PTT's stated requirements and the industry skills and experience of the consortium involved in this project. The overall project will consist of the following phases, they are:

Phase 1: Project Initiation.

During this phase of the project, the project plan, Document deliverable templates are identified written and submitted to the customer for review, comment, and signoff.

Phase 2: Disaster recovery site build.

This phase of the project consists of unpacking and installing the hardware at the development site. The operating systems will be pre-installed by LOGIC prior to installation at PTT head office.

Phase 3: Disaster recovery operation services build.

This phase of the project consists of two parts. First is the implementation of Oracle stand by server. This necessitates the build (Installation) of each replicated Oracle database and accompanying software on each server, the writing of scripts for the transfer of data (Archive Logs) and monitoring of both the production and stand by systems. Second is backup on NT server. This part will use ArcserveIT products from Computer Associates to develop end-day backup and restore/recovery procedure. Beside the end-day backup, we will provide on-line periodical backup for non-critical server by using ArcserveIT open-file technology and on-line data replication by using ArcserveIT Data Replication technology.

Phase 4: Disaster recovery operations testing and acceptance (Phase III). There are 3 sub-phases.

(1) We will develop test plans for both UNIX application system and Windows NT application system. These test plans will ensure that the system is thoroughly Quality Assurance (QA). PTT will be invited to participate in the final QA testing. Upon completion of the testing, the project will be submitted to PTT for final acceptance and project signed off.

(2) This phase of the project consists of the implementation of Oracle stand by server for Unix part and ArcserveIT product for NT part. The Unix part necessitates the build (Installation) of each replicated Oracle database and accompanying software on each server, the writing of scripts for the transfer of data (Archive Logs) and monitoring of both the production and stand by systems. The NT part will create the scenario for simulate cases which cover all possible causes of failure and demonstrate the recovery plan for each failure.

(3) During this phase of the project, testing procedures will be formulated and test plans written for each system. These test plans will ensure that the system is thoroughly QA. PTT will be invited to participate in the final QA testing. Upon completion of the testing, the project will be submitted to PTT for final acceptance and project signed off.

II. SYSTEM ANALYSIS

2.1 Problem Statement

The Petroleum Authority of Thailand (PTT) wishes to develop a backup site for critical and non-critical information so that in case of a disaster, PTT has the option to do the following:

(1) Fail one or more applications over to the disaster recovery system and;

(a) Point existing clients at the head office to the disaster recovery application and continue transacting business in a limited fashion.

(b) Transport a subset of employees to the disaster recovery site and continue to do business in a limited fashion.

(2) PTT also has the option of not failing over to the Disaster Recovery site if it is judged that the production failure does not constitute a disaster in the normal sense.

So in this section, we will talk about the requirement of the application system that has been related in this project.

The application system means the business procedure that primarily uses the computer system to operate. It can be separated into 2 categories:

(1) Critical Application: The application system that cannot stop the operation more than 24 hrs.

(2) Non-Critical Application: The all other application, other than critical application, that cannot stop the operation more than 48 hrs.

2.2 Terms of Reference

All information in this section is relevant for this project only.

2.2.1 Purpose

The purpose of this project is to ensure that PTT has the ability to continue with business, in a limited fashion, on a replicated system(s) until the production system(s) are rebuilt. The clients will again be pointed to the production systems and normal business operations will be re-established.

A disaster is defined as a total failure of equipment through a fire, flood, earthquake, or some other catastrophic event. A short-term failure of equipment is not considered a disaster within the context of this document.

2.2.2 Objectives

The objective of this project is to develop a disaster recovery solution for the Petroleum Authority of Thailand (PTT) and to propose a disaster recovery solution that meets the following requirements:

2.2.3 Manageability

PTT staff are able to administer the day to day operations necessary to ensure that the disaster recovery facility is always in operational readiness.

2.2.4 Scalability

Systems must be scaleable. The Disaster Recovery systems must be able to operate at peak capacity in order to accommodate changing conditions and growth rates.

2.2.5 Accessibility

A subset of production users must be able to access their authorized data and applications quickly and reliably as defined in PTT's disaster recovery TOR.

Table 2.1. Accessibility Requirements.

Application	Normal Users	DR Users
AP/GL	50	35
Data Warehouse (PBMS)	84	12
SAP/HR	118	6
MS-Exchange	900	X
Oracle on NT	121	9
SQL-Server	82	80
Btrive	17	11
Others(NT)	172	32

2.2.6 Measurability

PTT must be able to measure the effectiveness of the disaster recovery infrastructure through testing of the system as implemented. It is also expected that PTT test the implementation on a quarterly or semi-annually basis to ensure the continued viability of the system.

2.3 Current Systems

There are two categories for application systems and we call them as:

(1) Critical Application systems

(2) Non-Critical Application systems

The Term of Reference that defines the meaning of "critical applications" means application system at CDR site must be used within 24 hour after deciding to use CDR site and data from head office must apply to and up to date and not loss over 30 minutes. While “non-critical application” differ from critical application only the fail-over time must not exceed 48 hours after deciding to use CDR site.

The complete listing of Hardware and Application including defined service level are shown in Appendix B and Appendix C.

In this chapter, all applications will be categorized into two major platform types that are UNIX platform and Window NT platform and we will discuss in term of these platform types.

2.3.1 UNIX Platform

The systems that are within the scope of this project and using UNIX platform are identified below. They are:

(1) Oracle Financials (AP/GL/FA)

The Oracle Financial Application consists of accounts payable, general ledger and fixed assets. The database and version is Oracle 7.3.4. The user(s) interact with the application via the DL200W Graphical User Interface (GUI) installed on a PC workstation.

Table 2.2. Oracle Financials Data Centre Server.

Server Type	Processors	Memory	Disk Capacity	OS	Application SW
E3000	2x167 MHz	512 MB	63 GB RAID 0+1	Solaris 2.5.1	Oracle: 7.1.6.2.5 Payable, GL, Asset 10.5

Table 2.3. Oracle Financials DR Service Level Required.

Recovery Time	Data Loss	Users / DR Users	Performance
24 Hours	Preceding 24 Hours, Max	50 / 35	Equivalent

(2) MAXIMO

The Maximo application is the asset control application servicing the PTT Head Office operation. The Maximo database and version is Oracle 7.3.4. The user(s) interact with the application via the MAXIMO Graphical User Interface (GUI) installed on a PC workstation.

(3) SAP/HR

The SAP HR application is the human resources application servicing the PTT Head Office operation. This application is based on SAP R3. The database and version is Oracle 7.3.4. The user(s) interact with the application via the SAP/GUI Graphical User Interface (GUI) installed on a PC workstation.

Table 2.4. SAP/HR Data Centre Server (Production).

Server Type	Processors	Memory	Disk Capacity	OS	Application SW
E3000	4	1024 MB	76 GB RAID 0+1	Solaris 2.5.1	Oracle 7.3.3.5.0 SAP 3.0D w/HR

Table 2.5. SAP/HR Data Centre Server (Development).

Server Type	Processors	Memory	Disk Capacity	OS	Application SW
E3000	2	1024 MB	80 GB RAID 0+1	Solaris 2.5.1	Oracle 7.3.3.5.0 SAP 3.0D w/HR

Table 2.6. SAP/HR DR Service Level Required (Production).

Recovery Time	Data Loss	Users / DR Users	Performance
24 Hours	Preceding 24 Hours, Max	118 / 16	Equivalent

Table 2.7. SAP/HR DR Service Level Required (Development).

Recovery Time	Data Loss	Users / DR Users	Performance
24 Hours	Preceding 24 Hours, Max	58 / 5	Equivalent

(4) Data Warehouse

The Data Warehouse application is the decision support and reporting application servicing the PTT Head Office operation. This application is based on Oracle Express. The database and version is Oracle 8.0.5. The user(s) interact with the application via several Clients or Graphical User Interface (GUI) installed on a PC workstations.

Table 2.8. Data Warehouse Data Centre Server.

Server Type	Processors	Memory	Disk Capacity	OS	Application SW
Digital Alpha 2100	2	512 MB	23.1 GB RAID 5	Digital Unix 3.2G	Oracle 7.3.2.2.0 Oracle Express Object, Analyser, Personal Express 2.0.1 + Express Object 2.0.5

Table 2.9. Data Warehouse DR Service Level Required.

Recovery Time	Data Loss	Users / DR Users	Performance
24 Hours	Preceding 24 Hours, Max	84 / 12	N/A

2.3.2 MS Windows NT Platform

The systems that are within the scope of this project and using UNIX platform are identified below by system name. They are:

Critical application on NT platform

(1) Loan Administration System

(2) Financial Planning System

(3) Payroll System

(4) HO Mail System

(5) Inter Mail System

(6) OT Approval System

(7) IT Corporate Filing System

Non-Critical application on NT platform

(1) Staff Recruitment System

(2) Time Access Control Management System

(3) Personal Information System for Top Executives

(4) Performance Based Management System

(5) News Clipping System

(6) Document Circulation System

(7) PTT Board Meeting System

(8) Library Web System

(9) PTT-Course Catalog Service System

(10) PTT-Postcard System

(11) Conference Room Reservation System

(12) Car Booking System

(13) PTT-Telephone Directory System

(14) Display Phone Expense System

(15) Help Desk System

(16) PTT-Internet Homepage

(17) IT Corporate Inventory System

(18) IT Corporate PM System

(19) Document Management System

(20) ASCOPE Filing System

(21) Y2K Filing System

(22) Empower Filing System

(23) PTT Reforest System

(24) Library Management System

(25) PTT Clinical System

(26) Legal Document filing System

(27) Leasing Car Profile System

(28) Document Warehouse System

(29) Car Profile System

(30) Internal Telephone Billing System (CDR)

(31) Health & Environment Data Repository System

(32) Telephone Billing System

(33) Business Development System

(34) Budget Data on Web

(35) Time Attendance

(36) Car Management System

(37) Guest Registration

(38) PTT Print Check

But we can summarize all of application and categorize them by application data store into 5 categories.

(1) Oracle workgroup server application

(2) MS-SQL Server application

(3) MS-Exchange Mail server application

(4) Magic Pervasive SQL application

(5) File base application (MS-Access, Internet, Excel, and others)

All applications are running on around 22 NT systems. The following is the specification of one server of the total NT systems that is MS-Exchange Mail server.

Table 2.10. MS Exchange Data Centre Server.

Server Type	Processors	Memory	Disk Capacity	OS	Application SW
Compaq Proliant 5000	4	256 MB	20 GB RAID 5	MS NT 4.0	MS Exchange

Table 2.11. MS Exchange DR Service Level Required.

Recovery Time	Data Loss	Users / DR Users	Performance
24 Hours	Preceding 24 Hours, Max	800 / 1	N/A

2.4 User Requirements Summary

A system requirement definition is an abstract description of the services which the system must operate. It should only specify the external behavior of the system and should not be concerned with design characteristics. It should be written in such a way that it is understandable without knowledge of specialized notations. Requirements fall into two categories:

(1) Functional system requirements: These are system services which are expected by the user of the system. In general, the user is uninterested in how these services are implemented so we should not introduce implementation concepts when describing these requirements.

(2) Non-functional requirements: These set out the constraints under which the system must operate and the standards which must be met by the delivered system. For example, a non-functional constraint might be a requirement that all information input should be expressible using the ASCII character set. A standard might be a requirement for the maximum system response time for any user command to be less than 2 seconds.

2.4.1 Functional Requirements

(1) The CDR system must be used by Key Users when the 13th Fl. Computer System cannot be used in event of disasters by the following criteria:

(a) The Critical Application must be used within 24 hrs after the authorized person has an announcement for using CDR system.

(b) The Non-Critical Application must be used within 48 hrs after the authorized person has an announcement for using CDR system.

(2) When the PTT's Head Office does not access in any conditions, the CDR system must be used and some of users will be temporary operate at the CDR site by using the computers in training rooms or waiting rooms.

(3) 30-minutes Data Loss on both Critical and Non-Critical Application.

(4) PTT's can use the CDR computer system for application development and training in normal condition.

2.4.2 Non-functional Requirements

(1) PTT CDR Staffs training for using DR system in event of disaster.

(2) Additional ARCserve licenses, PCs, Printer, Solstice Backup and Oracle

(3) Y2K Patching, or OS Upgrade on Production Servers

(4) WAN Bandwidth utilization or improvement

(5) System Assessments

(6) Skill/Staff Assessments

(7) Fail-back Procedures and Manuals

(8) Naming Service (DNS, WINS) Configuration

(9) Front-end Software Installation at Workstations (What, How, When)

(10) CDR Backup Site preparation

(11) All CDR system insurance.

(12) Customer Support must response within 30 minutes

III. CDR SYSTEM DESIGN FOR UNIX SYSTEMS

3.1 Overview

This section describes the design and implementation of a Disaster Recovery solution for Oracle databases. It provides a set of scripts which can be used on most Oracle systems to enable log shipping to a standby database. These enable a warm copy of the production database to be kept available on the Disaster Recovery site. This reduces the impact of data loss which would occur, if recovery had to be made from the previous nights backup tape, the standby database provides a consistent, up to date copy of the production database, with only minor data loss. The extent of this loss will typically be slightly greater than half the log switch time. Other factors, such as wide area network failure could result in a greater loss of data.

Alternative solutions, to the Oracle Standby Database that could reduce the potential of data loss, present problems of their own.

(1) Database replication in synchronous mode has a significant performance impact causing local updates to take longer. It can be configured in asynchronous mode, but in this case, suffers from the same loss of data as log shipping and introduces a large number of changes to the production system which it is desirable to avoid.

(2) Remote disk mirroring is constrained by the requirement for fibers to be available between the two sites. Maximum distances of 20 km for SPARCstorage arrays and 10km for FC-AL disks like the A5000 is possible using single-mode fiber-optic and the appropriate repeaters or GBICs. Remote disk mirroring also imposes a small performance overhead due to the i/o elongation.

(3) Database snapshots enable tables to be copied to remote systems. It is inherently an asynchronous mechanism whereby the remote system has an old copy of the data. Unless the snapshots are kept simple, the entire table will be sent. This method was not considered further for this application.

The log shipping approach selected has the advantage of requiring minimal changes to the production database systems. No changes to the database schema are necessary, only some minor changes to one maintenance script, and the addition of some shell scripts. Given the critical nature of the systems this will be deployed on, this was considered a major advantage.

The scripts have been designed to be easy to configure and to allow multiple instances of them to be running on the same system. Whilst they have been originally written and tested for use in the production CMS environment. They are parameter file driven, allowing them to be readily deployable on other systems.

3.2 Disaster Recovery Systems

The systems that are within the scope of this project are identified below, they are:

(1) Oracle Financials (AP/GL/FA)

The Oracle Financial Application consists of accounts payable, general ledger and fixed assets. The database and version is Oracle 7.3.4. The user(s) interact with the application via several Clients or Graphical User Interfaces (GUI) are installed on a PC workstation, they are shown in Table 3.1: Standby Applications and Their Clients.

There are other "applications" that use the base tables of the Oracle Financials AP/GL database. These applications are specific views on information contained in Oracle Financials, they are:

(a) Budgeting System

(b) Financial System

(c) Accounting System

(d) Warrantee System

(e) Hospital Expense System

The user(s) interact with the applications via several Clients or Graphical User Interfaces (GUI) installed on a PC workstation, they are shown in Table 3.1: Standby Applications and their Clients.

(2) MAXIMO

The Maximo application is the asset control application servicing the PTT Head Office operation. The Maximo database and version is Oracle 7.3.4. The user(s) interact with the application via the MAXIMO client or Graphical User Interface (GUI) installed on a PC workstation, and is shown in Table 3.1: Standby Applications and Their Clients.

(3) SAP/HR

The SAP HR application is the human resources application servicing the PTT Head Office operation. This application is based on SAP R3. The database and version is Oracle 7.3.4. The user(s) interact with the application via the SAP GUI, Graphical User Interface (GUI) installed on a PC workstation, and is shown in Table 3.1: Standby Applications and Their Clients.

(4) Data Warehouse

The Data Warehouse application is the decision support and reporting application servicing the PTT Head Office operation. This application is based on Oracle Express. The database and version is Oracle 8.0.5. The user(s) interact with the application via several Graphical User Interface

(GUI) installed on a PC workstation, they are shown in Table 3.1: Standby Applications and their Clients.

3.3 Standby Applications

The Oracle Financials application residing on the apgl server has many "Applications or Systems" accessing the database. These other "Applications" are essential views into the Oracle Financials database. These systems and their Graphical User Interface (GUI) are:

Table 3.1. Standby Applications and Their Clients.

System	Abbreviation	Server	Client (GUI)
Accounts Payables	AP	apgl	DL200w
General Ledger	GL	apgl	DL200w
Asset Management	FA	apgl	DL200w
Budgeting System	BUDGET	apgl	Developer/2000
Financial System	FINANCE	apgl	Developer/2000
Accounting System	ACCOUNT	apgl	Developer/2000
Warrantee System	WARRANTEE	apgl	Developer/2000
Hospital Expense Tracking System	HOSPITAL	apgl	Developer/2000
MAXIMO	HOMAX	apgl	MAXIMO GUI
Data Warehouse	DW	IW	Oracle Express Analyzer Ver 2.0.5
Data Warehouse	DW	IW	Oracle Express Application 2.2

Table 3.1. Standby Applications and Their Clients. (Continued)

System	Abbreviation	Server	Client (GUI)
Data Warehouse	DW	IW	Oracle Discoverer 3.1
Data Warehouse	DW	IW	Oracle Procedure Builder
Data Warehouse	DW	IW	Oracle Data Mart Suite
Data Warehouse	DW	IW	Platinum Decision Base
SAP/HR	SAP	HR3	SAPGUI

3.4 CDR Design Assumptions

3.4.1 Terms of Reference (TOR)

The Terms of Reference (TOR) for the project define the design constraints. For the most part the requirements are identical for all systems that are within the scope of this project, they are shown in Table 3.2 TOR Requirements.

Table 3.2. TOR Requirements.

Server/Application	Recovery Time	Maximum Data Loss
Apgl (Oracle Financials)	48 Hours from outage	30 Minutes Maximum
IW (Data Warehouse - PBMS)	48 Hours from outage	4 Hours Maximum
HR3 (SAP/HR)	48 Hours from outage	30 Minutes Maximum

3.4.2 Maximum Time to Recover from a Disaster

Recover within the prescribe time defined in Table 3.2 TOR Requirements, after the outage, or after it has been decided that the disaster was of sufficient magnitude and duration that, a disaster is called and the effected system(s) are not available.

3.4.3 Maximum Data Loss

Not more than the prescribe time defined in prior to the outage.

3.4.4 Number of Users in Disaster Mode

In recovery mode, the backup server must support users as shown in Table 3.3. User Requirements.

Table 3.3. User Requirements.

Server/Application	Normal Users	DR Users
Apgl (Oracle Financials)	50	35
IW (Data Warehouse – PBMS)	84	12
HR3 (SAP/HR)	118	6

3.5 Standby Methodology Described

The solution proposed for The Petroleum Authority of Thailand's Disaster Recovery project is to use Oracle Standby Database(s) for each of the three (3) systems defined as being mission critical. The Oracle Standby Database solution will be applied to each of the disaster recovery systems:

(1) SAP/HR

(2) MAXIMO

(3) Oracle Financials (AP/GL/FA)

(4) Oracle Data Warehouse

3.5.1 Standby Operating System

The first step in the system-recovery process is to install the foundation of the operating system on the disaster recovery server. The system should replicate the production system as closely as possible. Files systems, RAM and data should as closely resemble the production system as possible.

3.5.2 Standby Application

The Standby Application exits on the Standby Server. The Standby application for each server is shown in Table 3.4: Standby Applications.

Table 3.4. Standby Applications.

Standby Application	Resident Server
Accounts Payables	apgl
General Ledger	apgl
Asset Management	apgl
Budgeting System	apgl
Financial System	apgl
Accounting System	apgl
Warrantee System	apgl
Hospital Expense Tracking System	apgl
MAXIMO	apgl
Data Warehouse	IW
SAP/HR	HR3

3.5.3 Standby Client

Standby Clients are clients that exist on systems that are dedicated to Disaster Recovery. These clients or Graphical User Interfaces are used to access the Disaster Recovery systems within the disaster recovery framework, they may exist at the disaster recovery datacenter where users would be transported to resume business operations after a disaster has been called.

3.5.4 Standby Database

Oracle Standby Database(s) are a replications of the production database maintained on a separate host, preferably at a location some distance from the production system. Maintaining a Disaster Recovery system at the same location as mission critical production systems serves to nullify the Disaster Recovery effort.

(1) Standby Database Advantages

The Oracle Standby Database solution is well known, used, and documented, this solution has the following features:

(a) Archive Logs from the production database are transported to the standby server and applied to the standby database.

(b) The standby database is maintained in a mounted, but not open mode. This allows the archive logs to be written to the database, and keeps users from transacting with the database.

(c) An exact copy of the production database that lags only by the creation, transfer and application of archived redo logs;

(d) Standby database can serve as a circuit breaker in the event of a serious DBA error by stopping archive log propagation;

(e) Standby database can be activated quickly, much faster than a restore from tape.

(f) Maintain a standby database in a location that is geographically remote from the production database, or maintain several standby databases in several geographically diverse locations.

(2) Standby Database Limitations

The Oracle Standby Database has some limitations to its use. These limitations are not serious, these limitations are:

(a) If the standby database is activated, for any reason, i.e., Disaster Recovery Testing to an actual disaster, it must be used as the production instance or be rebuilt. Once the standby database is activated, the production and standby databases become out of synchronization.

(b) Once the standby database is activated, it cannot be returned to the standby database mode.

(c) An invalidated standby database must be re-initialized from a backup of the production database.

(d) Certain DBA and SQL commands do not write to the online redo log files. If any of these commands are used against the production database, they must also be used against the standby database. Only commands that are written to the online redo logs will be propagated from production to the standby database.

(e) In order to transact against an activated standby database, application GUI's must be directed to the standby database, now the production instance.

3.5.5 Standby Scripts

The standby scripts are UNIX Bourne shell scripts that automate the tedious process of closing and archiving online redo log files, moving the logs to the standby server and applying the archive log to the standby database and finally error checking and e-mail notification to administrators.

The scripts are executed using the UNIX cron, a scheduling utility that will execute scripts on the production and standby servers every 10 minutes. Other cron entries execute scripts that move reports that live on file systems on the production server to the standby server.

(1) Standby Scripts Advantages

Standby Shell Scripts allow the following to occur:

(a) Automatic propagation of archived logs from the production database to the standby database.

(b) The use of compression utilities conserves network bandwidth.

(c) Can "catch up" if the network connection is lost for a time & won't re-transmit successfully transferred logs needlessly.

(d) Automatic application of archive logs on the standby database.

(e) Portable from one Oracle instance to another with small differences between servers, such as path names. These differences are defined in the configuration file for each of the servers.

(f) Low level of effort from DBA's and System Administrators since normal operation is automatic.

(2) Standby Scripts Limitations

(a) The scripts are a compromise between complexity and catering to all eventualities.

(b) The scripts cannot correct for complex problems.

3.6 Standby Database Design

As discussed in earlier sections, the principal behind the standby database is the application of archived online redo logs. The archived logs are generated on the production database and applied to the standby database, thus maintaining the standby database current as defined in Table 3.2: TOR Requirements.

3.6.1 Archive Log Generation

The following narrative is a somewhat simplistic step by step process that shows how transactions are propagated from the users terminal to the database, to the online redo logs and finally the Archive Log File.

(1) Users transact with the Production Server through a client or Graphical User Interface (GUI) these transactions (insert, update, and delete records) occur over the working day. These kind of transactional databases are called OLTP (On Line Transactional Processing).

(2) When users transact against the database, a historical record of each transaction is maintained in the redo buffer in the System Global Area (SGA) in RAM, see

(3) Figure 3.1. Archive Log Creation. The users transactions are buffered in the Redo Buffer Cache, in the SGA.

(4) The transactions are buffered in the Redo Buffer Cache until the transaction(s) are committed (Saved). After the transaction(s) are committed, the redo buffer cache is flushed, transactions are written to the online redo logs, this is done by the LGWR (Log Writer) process.

(5) Transactions are written to the online redo logs until a svrmgrl process tells Oracle that the online redo log is to be archived and closed. The svrmgrl

process is a scheduled process that executes every 10 minutes or when the online redo log is full and automatically closed by Oracle.

(6) This means that online redo logs are archived (written) every 10 minutes. The process that oversees the online redo log writing and closing is the Log Writer (LGWR) process, the process that writes the Archive Log Files is the ARCH process.

(7) The Archive Log File is sequentially numbered, and transported to the standby server through shell scripts also running on a cron (scheduled execution).

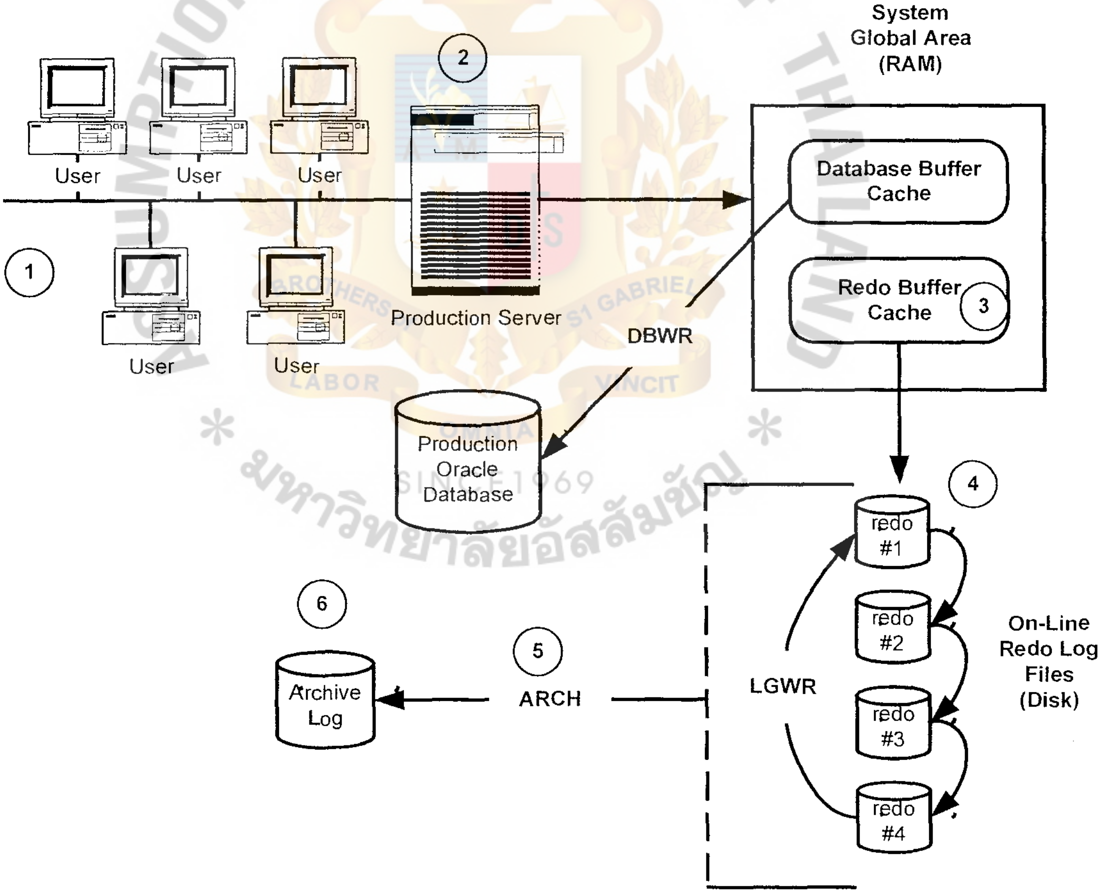


Figure 3.1. Archive Log Creation.

3.6.2 Archive Log Transfer

As the production, redo logs are closed and archived, they are compressed and transferred to the standby server and applied to the standby database. Essentially, these archive log files are used to "recover" the standby database. The Standby Database contents will be consistent with the production database up to the most recent production archive log transferred and applied to the Standby Database.

Changes that are done to the production database are carried to the standby database through the movement and application of the archived redo log files. The standby database is maintained in a "mounted" but not open mode, this allows the archived redo log files to be written to the database, maintaining the integrity between production and standby databases.

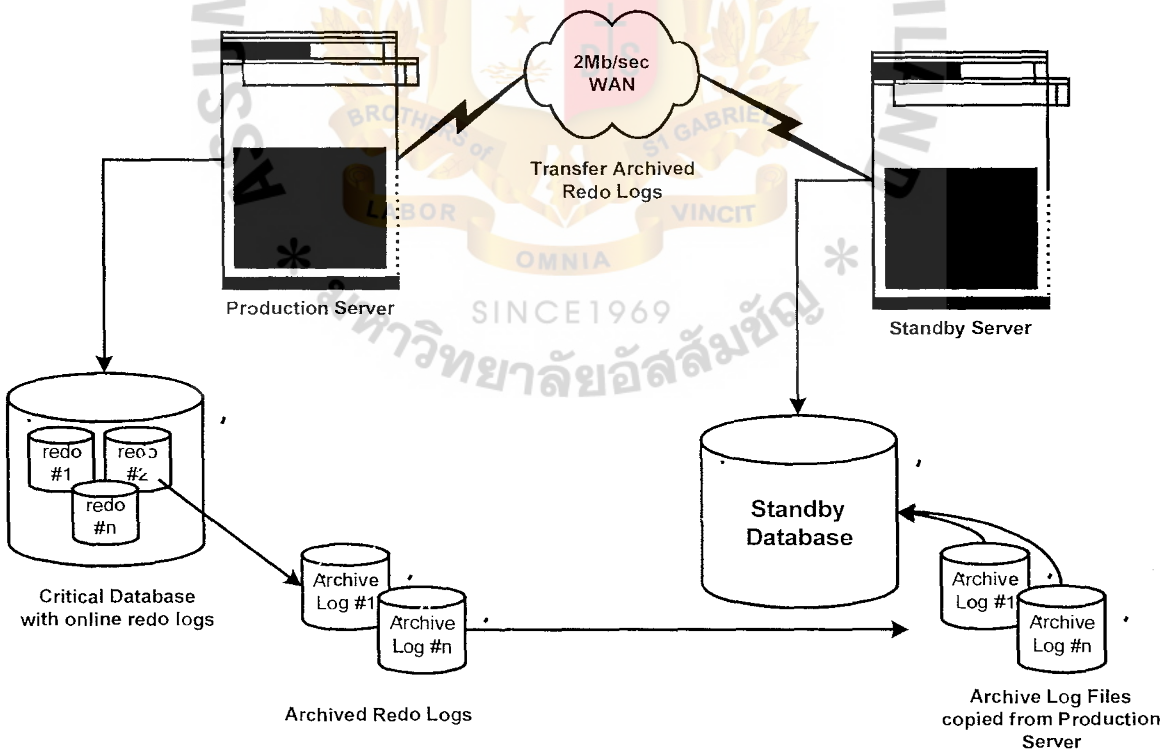


Figure 3.2. Standby Database Architecture.

3.6.3 Archive Log Application

As the archived logs arrive on the standby server, a UNIX Bourne shell script will automatically apply them to the standby database. The script is called every 10 minutes by the oracle user's crontab file. Once the archive log file has been applied successfully to the standby database, it is will be deleted from the standby server.

All of the critical systems archive (write) their redo logs every 10 minutes. Once archived, these logs remain on the production system until they are written to tape and then deleted. The archive log files should remain on the production system for at least 24 hours to ensure that the files have been successfully applied to the standby database.

IV. DETAIL DESIGN FOR NT CDR SYSTEMS

4.1 Overview

The detail design for NT CDR system will consist of two methods for optimize the PTT 's current resources and new hire ones. Those methods are:

(1) Working hour period Data Synchronization

(2) End Day period Data Synchronization

4.2 Working Hour Data Synchronization

This method will take care data of PTT 's applications that use many kinds of DBMS such as Oracle, MS-SQL Server, MS-Access, Pervasive SQL, and Exchange Mail Server. The PTT design will use the replication technology from Computer Associates International (CAI) to maintain data synchronization between PTT home office servers and CDR servers at the CDR site in Wang-Noi. It on-line replicates selected data from primary site to CDR site at the working hour period. Any change will be propagated automatically to CDR site without effect the users response 's time or the availability of the application system. Some non-critical applications will not directly use this method. We will copy data from the selected application in the working hour period to disk and tape. We send them directly to CDR site via the network infrastructure. This will reduce the human effort to carry cartridge tape to CDR site at the normal time. However, he has to carry the cartridge tape to CDR site in the case of disaster.

The addition description for non-critical application data synchronization is that will copy data from original location to backup server 's disk and backup to tape every 20 minutes. If that day doesn't have any problems, the backup server will copy data from backup disk via wide area network to CDR site otherwise, PTT 's staff has to carry tapes to CDR site by himself.

4.3 End Day Data Synchronization

This is the method for backing up routine for all data and application systems. This method will backup all changed data every day, all unconditional data every weekend and full data and system every end-month.

4.4 ARCserveIT Replication Architecture

In order to understand the detail design that will be used at PTT, we need to get an idea of software, which control the data synchronization between servers, it is ARCserveIT replication. The ARCserveIT Replication Options provides server-level data protection for your network.

For each primary server (any server with data that you want to protect), we set up a replication task. This requires us to nominate a secondary server to hold a replicated set of this data.

Replication Option then synchronizes the files on the primary and secondary servers to make them identical. As soon as synchronization is complete, our data is protected. Subsequent data changes are replicated to the secondary server as they happen. The Replication Option uses proprietary open file technology to replicate files even when they are open and in use.

There can only be one primary replication task for each server (but a single computer can have multiple tasks running if it is both a primary and secondary server). We can start, stop, and pause this task at any time. We can also monitor the status of the Replication task. If our primary server fails, the secondary server can stand in for the failed primary. The Replication Option transfers users to the secondary server with minimum delay and loss of data. This process is called fail-over, and is generally transparent to users. (Note that fail-over are optional: we define the conditions that can trigger them.) When the primary server is repaired, we reinstate it. The Replication Option re-synchronizes data on the primary and secondary servers then transfers users back to the primary server. As before, this process is generally transparent to users. We recommend that you set up the Alert notification system to warn you (or anybody you choose) whenever events such as fail-over occur.

4.4.1 What Are Its Advantages?

The Replication Option allows uninterrupted access to data while a failed server is repaired. Therefore, we can protect any files on your network against data loss, while avoiding the lengthy delays typically associated with archiving and disaster recovery solutions.

The Replication Option enables us to back up mission critical data while still allowing users to access the files they need. For example, if we need to upgrade the hardware for our primary server, we can manually fail-over the primary server, transferring control to a secondary server until the primary upgrade is complete.

The Replication Option requires no additional hardware. It runs on our existing servers, providing there is sufficient disk space for our replicated data.

4.4.2 How the Replication Works?

Here are the steps to configure the replication work for any system.

Step 1: Configuring your servers

First, we must configure any computers that we want to use as primary or secondary servers. These settings control default levels of:

Free drive space we can specify critical levels of free drive space on a server. When free space falls below this level, the Replication Option stops replicating files. For primary servers, we can also specify a shortage of free disk space as a condition that triggers fail-over.

Memory To optimize performance, we can set the amount of system memory available to the Replication Option.

Step 2: Setting up a replication task

We then set up a replication task to protect the files. To do this, we run the replication task wizard. The wizard steps us through:

(1) Selecting the servers First, we identify the primary server holding the files we want to protect. Then we select a secondary server to hold replicated versions of these files and, if necessary, stand in for the primary server. Note that primary and secondary servers do not need to be identical.

(2) Setting the speed of our network connection The Replication Option uses this information to assess how quickly it can detect communication failure.

(3) Select the level of protection we select whether the secondary server stands in for the primary server if it detects a fail-over condition, for example a drive failure on the primary server, or whether the replication task simply stops.

(4) Select the files we want to protect. We select the folders containing the files we want to protect and, if the task has "Full Protection", any shares we want to maintain.

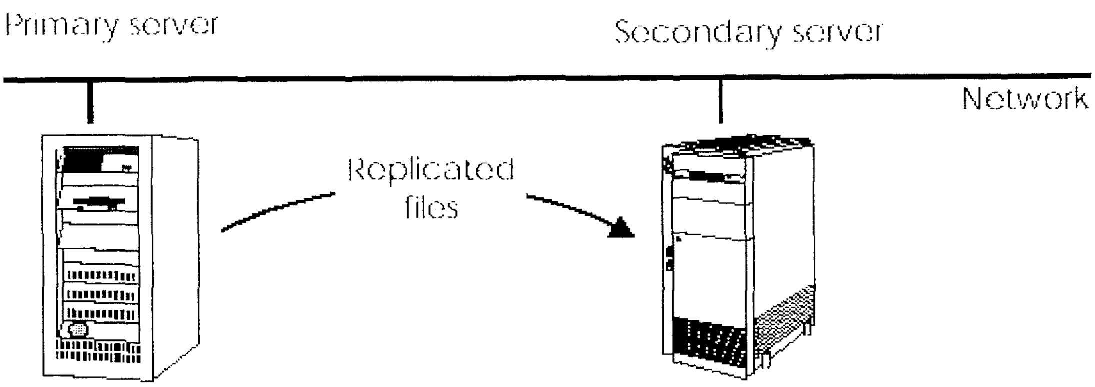


Figure 4.1. NT CDR Replication.

Step 3: Editing a replication task

We can also edit the advanced task settings before we run a replication task. In fact, it strongly recommended that PTT does not edit the advanced settings, particularly if we want to protect our server against communication failure.

At any time when a replication task is running, we can stop the task to edit or review the task settings. (If we do not want to stop the task, we can still view the settings in read-only mode.) We edit replication tasks in the Task Editor. Here, we can edit the settings that the Replication Option uses to detect and respond to failure. In particular, we can:

(1) Edit settings that determine how the Replication Option identifies and responds to communication or drive failure.

(2) Specify the Replication Option's actions when a fail-over occurs or when we reinstate a server.

(3) Specify the alternate name or IP address that the primary server uses when the secondary server is standing in.

(4) Edit settings that control the transactional integrity of replicated files.

(5) Modify the Replication Option's handling of replicated files.

For example, we can change the destination for replicated files.

Note: If we selected 'Full Protection' in step 2, the Replication Option displays a warning when the replication task wizard completes. The warning advises we to edit the advanced task settings. (By default, if the Replication Option detects a drive failure on the primary server, it causes the primary to fail-over. However, if it detects a communication failure, the replication task stops, but the primary does not fail-over.)

In case of PTT 's CDR project, we use only the data protection only because of the less reliability of MS-WINDOWS NT operating system.

Step 4: Running a replication task

We can then run a replication task. When we start a replication task, the Replication Option first synchronizes the files on the primary and secondary servers to make them identical. The Replication Option uses proprietary open file technology to replicate files to the secondary server even when they are open and in use on the primary server. When synchronization is complete, the secondary server is ready to stand in for the primary server, allowing access to the replicated files. Replication now begins.

Replication is the continuous process by which files on the secondary server are constantly updated with changes to the original files on the primary server as they occur. Changes are replicated even if the files are open and in use on the primary server. At any time, we can easily view the status of the replication task. During replication, the

Replication Option sends packets of replicated data to the secondary server as soon as 64 kilobytes of data accumulate in the primary server buffers. If less than 64 kilobytes of data is buffered on the primary server, the Replication Option waits to send this data with the next "heartbeat" signal.

Step 5: Failover

At any time subsequently, if the Replication Option detects a critical failure on the primary server, it can transfer control to the secondary server. This process is called "fail-over".

Fail-over causes the secondary server to stand in for primary server so that external clients and server-based applications can access the replicated files as though they were still available on the primary server. In this situation, the secondary server is known as the stand-in server.

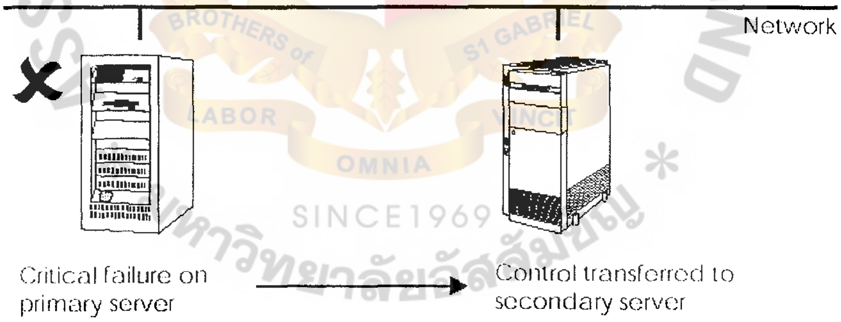

Network
Critical failure on primary server
Control transferred to secondary server

Figure 4.2. NT Critical Failure.

Step 6: Reinstating a server

When the failed primary server has been repaired, we run the reinstate wizard. This reinstates the original roles of the primary and secondary servers. The wizard also enables us to schedule the reinstatement to start at a specific time and broadcast warnings to users logged onto the stand-in server.

The wizard first re-synchronizes the primary server so its files match those currently on the secondary server. When this is complete, the primary server resumes control and users are transferred back to the primary server. We control this transition which is generally transparent to users.

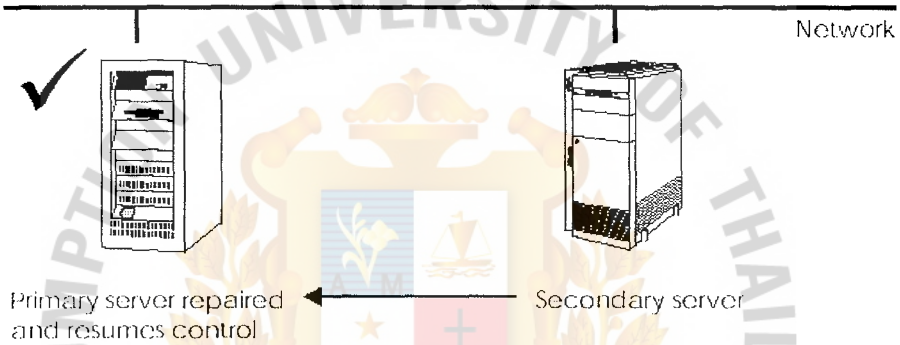


Figure 4.3. Control Returned to Primary Server.

4.5 Application Specific Detail Design

After we understand the concept of Replication. Next, we will discuss about the detail design in each application systems.

Critical application on NT platform

(1) Loan Administration System

(2) Financial Planning System

(3) Payroll System

(4) HO Mail System

(5) Inter Mail System

(6) OT Approval System

(7) IT Corporate Filing System

Non-Critical application on NT platform

(1) Staff Recruitment System

(2) Time Access Control Management System

(3) Personal Information System for Top Executives

(4) Performance Based Management System

(5) News Clipping System

(6) Document Circulation System

(7) PTT Board Meeting System

(8) Library Web System

(9) PTT-Course Catalog Service System

(10) PTT-Postcard System

(11) Conference Room Reservation System

(12) Car Booking System

(13) PTT-Telephone Directory System

(14) Display Phone Expense System

(15) Help Desk System

(16) PTT-Internet Homepage

(17) IT Corporate Inventory System

(18) IT Corporate PM System

(19) Document Management System

(20) ASCOPE Filing System

(21) Y2K Filing System

(22) Empower Filing System

(23) PTT Reforest System

(24) Library Management System

(25) PTT Clinical System

(26) Legal Document filing System

(27) Leasing Car Profile System

(28) Document Warehouse System

(29) Car Profile System

(30) Internal Telephone Billing System (CDR)

(31) Health & Environment Data Repository System

(32) Telephone Billing System

(33) Business Development System

(34) Budget Data on Web

(35) Time Attendance

(36) Car Management System

(37) Guest Registration

(38) PTT Print Check

However, ARCserveIT replication not the application aware replication system. We have to create the failover procedure to handle each application type that is divided by application data store. We can categorize them into 5 categories.

(1) Oracle workgroup server application

(2) MS-SQL Server application

(3) MS-Exchange Mail server application

(4) Magic Pervasive SQL application

(5) File base application (MS-Access, Internet, Excel, and others)

4.5.1 Oracle Workgroup Server Application

Many applications use the Oracle Workgroup Server as their RDBMS. All these applications share same Oracle instance on S1-HO. There are no special procedures for this application set. All replication data place directly to original path in CDR server. We nominate one server at CDR site for these applications. It is named CDR-HO-NT01 (Because we have limit CDR server, we share this server with Head Office mail server application). We install Oracle workgroup server in this CDR server and place all necessary application program related to Oracle Application both NT and UNIX operating system but not activate them. It standby until the CDR site has been used. We just stop the replication service and start the Oracle instance without any special procedure. One thing we have to remind is the user connection service. We have to create the network connection service for all users who will use this type of application at CDR site.

4.5.2 MS-SQL Server Application

There are many applications use MS-SQL Server as a DBMS and distribute in many servers. At the time of implement of this project, they consist of four (4) servers, they are:

(1) S13-HO

(2) S14-HO

(3) S15-HO

(4) S20-HO

These four servers replicate to the same CDR server. It is CDR-HO-NT02. We cannot place all databases file directly to the original path of its original database folder because when we start the MS-SQL database instance, it will start all database in that particular server. All database files have been locked for DBMS. We can't open file for DBMS service and replication service together at the same time. Sometimes we want to use CDR server for only one or two primary servers. We have to stop the replication service related to those primary server. Replication service on other primary server still running for replicating updated information. Therefore, we separate replication path and original path. When we want to use, which database from primary server in case or disaster, just stop the replication service of that server and copy the replication files to the original database path. This procedure will prepare and deliver to PTT staff at the phase of skill transfer.

4.5.3 MS-Exchange Mail Server Application

There are two servers, which use MS-Exchange Server as a mail server, S3-HO-BKK-MSG and S1-INTER-BKK-MSG. We can't share the CDR server for both of them. So we dedicate one small server to serve the request for 2 PTT international unit when the primary server crashed and need to use the CDR site. It is named CDR-HO-NT04. For Head Office Mail server, it needs big hardware configuration to serve every Head Office Staffs. We nominate the big server that is big enough for serving Mail service and other application service like Oracle Workgroup applications. So CDR-HO-NT01 will take care of both HO Mail application and HO Oracle Workgroup applications.

Unlike other enterprise-wide client-server applications, Microsoft Exchange Server is particularly sensitive to the name of the server on which it is running. MS Exchange Server will not run on a system where the name of the computer is different to that of the MS Exchange Server. In setting up the secondary server, it is necessary to have dedicated access to the primary and the ability to down the Exchange Server for a short period. This is to allow the installation of a second Exchange Server with the same name as the primary. In addition, neither the primary nor secondary server may act as the Primary Domain Controller, which needs to be constantly accessible on the network. It is like we must duplicate the primary server with the secondary server when we want to start the MS-Exchange server. Fortunately Oracle RDBMS doesn't care about the server name, we can't start the ORACLE event changing the server name. So we can implement two different application types within the same server.

Before starting the replication service for Ms-Exchange, we have to change the name of CDR server with the same as primary server and install MS-Exchange. After that, restart the machine, change server name back to old name, and mark all exchange services starting behavior as manual then restarts CDR server again. Now we can start the replication service to replicate data from primary site. We can place all replicated data from primary server into the original folder without temporary folder like SQL-Server database because this CDR server take care only one by one primary server. When we want to use Exchange server at CDR site, just change the CDR server name to the same as primary server name, restart server, running fail-over procedure and start exchange service. After that we can allow user connect to this CDR server instead of the crashed primary server.

To simulate environment from head office to CDR site for serving mail server, we dedicate one server for mail service account, it is CDR-HO-NT06. It replicates user information from S6-HO server. Beside that, we will configure CDR-HO-NT03 to serve the Internet mail also.

4.5.4 Magic Pervasive SQL Application

It is like Oracle workgroup server application, its database is stored only one server, S5-HO. We dedicate a small CDR server for this application. It is CDR-HO-NT05. We install the magic application and Pervasive SQL database on this CDR server and then create replication task from S5-HO for replicating data to CDR-HO-

NT05 and place them in the same folder with primary server. Like Exchange applications, when we want to use Magic application, we have to change the server name to be the same as primary server name. This way is easy to maintain in the future because the applications think they running on the original server.

4.5.5 File-Based Application (MS-Access, Internet, Excel, and others)

These applications are placed on CDR-HO-NT02. Most of these applications are non-critical ones and distribute to many servers. We will not use the replication technology to reduce the network traffic. We copy all data files to backup both on disk and tape at backup server and copy to CDR site at night. Tape backup will be useful when the backup server crashed or can't send data to CDR site we assign PTT staff who respond this duty to carry tape to CDR site and restore to disk. This way will reduce the human loading and utilize the network infrastructure. Because these applications are file base, they don't need special configuration. We can use them when we want.

4.6 Application Programs Consideration

All applications will use the same procedure for this consideration. We install application program in the same folder as much as possible and copy all modified programs from primary server to CDR servers depending on its functionality. We target the CDR server functionality as below:

Table 4.1. CDR Servers.

CDR-HO-NT01	Head Office Mail and Oracle Applications
CDR-HO-NT02	SQL Server DBMS, Cold Fusion, Win Forma, Web, MS-Access and other file base applications
CDR-HO-NT03	Internet Mail
CDR-HO-NT04	PTT Inter Mail Server
CDR-HO-NT05	Magic Applications
CDR-HO-NT06	Mail Service Account

4.6.1 Detail Description for CDR System in Each Application

Please see the detail design for each NT application in Appendix H.

4.7 Day-to-day Backup Detail Design

4.7.1 Overview

Enterprise Information system is very valuable. However, we can't backup every part to secondary site by using online data synchronization methodology. Keeping data in the secondary storage within computer center is another choice we choose. This method ensured all data would not be lost when unexpected events take place. This section dedicates for the day-to-day backup detail design for PTT 's HO.

4.7.2 GFS Backup Methodology

The backup methodology we use for PTT is Grandfather Father and Son (GFS) methodology. They will cover the backup procedure on daily, weekly and monthly basis. However PTT doesn't have robotic tape drive which it can automatically mount target tape into tape header or Un-mount the tape when it finishes the operation without human participation. So this design will let PTT data operation administrators take care for tape management. They have to change tape from tape drive every working day. Keep data save set in a safe place, scratch tape when they expired for reusable, print report for keeping activity logs and checking log for any exceptional events.

4.7.3 Backup Servers

Three backup servers backup the data in servers at PTT's HO, they are:

(1) S2-HO

(2) S3-HO-BKK-MSG

(3) S9-HO

The S2-HO backup server back's up S4-HO, S13-HO, S14-HO, S20-HO servers, as well as itself on tape.

The S3-HO-BKK-MSG server takes care of S1-INTER-BKK, S22-HO servers, and itself on tape.

The S9-HO backup server takes care of S1-HO, S5-HO, 15-HO servers, and itself on tape.

Not only, the end-of-the-day backup this server also backup data in working hour period in 20 minutes period fashion. All servers which have non-critical data that not backup by on-line data synchronization will be imbibed by this server before online transport to store at CDR server after mid-night.

4.7.4 End Day Backup

Information has been added, change within that particular day will be backup into tape at back server at the end of the day. It starts every day on 8.00 p.m. It operates on Monday – Thursday. This information will keep in the tape for three weeks before the data administrator brings those tapes back to reuse.

4.7.5 End Week Backup

It backup all data unconditionally. Directory and file are the same with End Day backup method plus system files and registry table file. After it finished backup operation, Data save set will keep in save place for three weeks and return to reuse after that. Like End Day backup, it starts at 8.00 p.m. of Friday. It may take longer than End Day backup so the operation might be finished on Saturday.

4.7.6 End Month Backup

This method will backup every data file and directory like End Week Backup. Besides that the disaster recovery image will be created for disaster case and keep in safe place for one year. The disaster image set consists of data tape and system image diskette.

We can apply this image back to the server after the hardware problem has been corrected without installing original operating system and application system individually. This image will keep all information for building the normal stage before disaster case happens. However, the normal stage depends on the up to date of data tape. This method will work together with on-line data synchronization. That means after we applied all system and data from last disaster image and full backup tape, the End Week and End Day tape will be brought to restore sequentially and the fail-back data from CDR site will be the last if it needs.

4.8 Restore Design

We use the restore methodology from ARCserveIT software. They consist of many methods to restore from backup tape like below.

Table 4.2. Restoration Methods.

Restore by Tree	Restores a specific directory or drive from a display of files or directories that were backup with ARCserveIT. Use this method when we don't know which tape contains data you need, but you know which machine it came from.
Restore by session	Allows you to select the session, and the files and directories you want to restore. Use method when you know the tape name, but are not certain about the session you want to restore.
Restore by Query	Restores files based search pattern given to locate the name of files and directories. Use this method when you know the name of the file or directory you want to restore, but do not know the machine it was backed up from or media(s) it was backed up to. This view uses ARCserveIT 's database.
Restore by backup media	Restores a complete backup session from a specified media in a storage device. All files in the session are restored to the destination, unless filters are added to restore job. Use this method when a tape was created by different version of ARCserveIT or if the database did not recognize it.

The above restore methodology will be adapted to meet the requirement in any restore environment. Normally we choose the restore by session and restore from the oldest to the newest version. This method will apply to all data from tape to the original destination with the most up to date information.

4.9 Recover Design

We have designed the recovery method when servers have disaster. There are some steps to follow when the server falls into a "Disaster Mode".

4.9.1 Recover without Disaster Recovery Option

ARCserveIT Server

(1) Re-install the MS-Windows NT operating System.

(2) Re-install ARCserveIT Enterprise Edition.

(3) Apply license key.

(4) Configure tape management until we can use tape drive.

(5) Select full backup, incremental backup for that server from tape library

(6) Restore the most current ARCserveIT database (optional).

(7) Use the "Restore by Tree" source view in the Restore Manager to select what data we want to recover.

(8) Select machine, drive, directory, or files to restore. Choose the destination tab and select target. Then choose schedule time to restore. It should be now.

(9) Check the data correctness by testing some application in that particular server.

ARCserveIT client

(1) Re-install the MS-Windows NT operating System.

(2) Re-install ARCserveIT NT client.

(3) Select full backup, incremental backup for that server from tape library

(4) At ARCserveIT server, use the "Restore by Session" source view in the Restore Manager to select what data we want to recover.

(5) Select machine, drive, directory or files to restore. Choose the destination tab and select target. Then choose schedule time to restore. It should be now.

(6) Check the data correctness by testing some application in that particular server.

4.9.2 Recover with Disaster Recovery Option

We will follow the recovery steps from ARCserveIT Disaster Recovery option guide both ARCserveIT server and ARCserveIT client.

V. CDR SYSTEM IMPLEMENTATION

5.1 Disaster Recovery Site Build

5.1.1 Overview of DR Site Build

The main focus in this phase of the project is to install, configure and document the hardware, software and network components required to ensure that PTT's disaster recovery efforts actually meet the stated requirements.

We do the following activities:

(1) Installation, configuration and documentation of all required computer systems and peripherals.

(2) Installation, configuration and documentation of computer the OS and applications software required for disaster recovery of the covered services.

(3) Installation, configuration and documentation of network equipment and software required for disaster recovery of the covered services.

(4) Installation, configuration and documentation of disaster recovery workstations.

5.1.2 Installation of Computer Systems and Peripherals

This task involves the initial set-up of the computer systems required for support of PTT's disaster recovery goals.

(1) Description

These work steps are required for the initial set-up of the disaster recovery computer systems and their peripherals. They are detailed below:

(a) Receipt of materials at PTT's DR site

(b) Unpacking and inventory of components

(c) Data entry into asset management systems as required

(d) Proper placement of systems according to PTT standards

(e) Cabling of peripherals and network connections

(f) Initial power up testing of all components

(g) Initial configuration as required

(h) Labelling of components

(i) Create documentation

(1) Delivery report

(2) Systems test report

(3) Installation and configuration guide

(2) Objectives

The main objective of this activity is to install and prepare the disaster recovery site computer systems for the software build phase.

5.1.3 Installation of OS and Applications Software

This task involves the load of the software components required to support PTT's disaster recovery plan.

(1) Description

The following tasks performed on each disaster recovery server:

(a) Operating System software load

(b) OS patching as required

(c) Operating System configuration required for the specific Applications software to be loaded

(d) Applications software load

(e) Applications software configuration and licensing

(f) Configuration conformance check between the Data Center and Disaster Recovery site

(2) Objectives

The objective of this set of tasks is to match the disaster recovery systems to their Data Center counterparts and prepare them for data loading of customer's applications.

5.1.4 Installation of Network Equipment and Software

This task involves the load of the network equipment and software components required to support PTT's disaster recovery plan.

(1) Description

The following tasks will be performed in the disaster recovery site:

(a) IP addressing of servers

(b) DNS configuration

(c) Network Management configuration as required

(2) Objectives

The objective of this set of tasks is to enable the data path to PTT's main data center.

5.1.5 Installation of the Disaster Recovery Workstations

This task involves the initial set-up of the data entry and reporting systems required for support of PTT's disaster recovery goals. These systems will serve as the replacement workstations at the disaster recovery site.

(1) Description

These work steps required for the initial set-up of the disaster recovery computer systems and their peripherals. They are detailed below:

(a) Receipt of materials at PTT's DR site

(b) Unpacking and inventory of components

(c) Data entry into asset management systems as required

(d) Proper placement of systems according to PTT standards

(e) Cabling of peripherals and network connections

(f) Initial power up testing of all components

(g) Front end software load and test

(h) Labelling of components

(i) Create documentation

(1) Delivery report

(2) Systems test report

(3) Installation and configuration guide

(2) Objectives

The main objective of this activity is to install and prepare the disaster recovery site workstations.

5.1.6 Project Management

The project management approach ensures that clear lines of communication, responsibility and authority are defined up-front to allow for schedule controls along with meaningful status reporting.

The scope of this activity is to ensure that this phase of the project is delivered on time and according to the scope of the project and to enable a successful services build phase.

5.2 Disaster Recovery Site Services Build

5.2.1 Overview of Disaster Recovery Site Services Build

In this phase of the project we will design and implement a set of processes and procedures that are designed to meet the requirements as defined in the SLA's completed by PTT. Processes and procedures will consist of the following:

(1) Data Center / DR site data copy

(2) Data Center / DR Site change management

(3) DR Fail-over procedures

(4) DR functionality test routines

(5) DR roles, responsibilities and notification procedures

5.2.2 Data Centre / DR Site Data Copy

This task involves the creation of the daily copy routines required to ensure that the DR site servers data sets are up to date, so that in the event of disaster fail over the required recovery parameters are met.

(1) Description

We evaluate and test various strategies for maintaining the data sets for each covered service and select the most appropriate methods for use on a routine basis. The methods chosen may vary depending on the volume of data requiring updating per day for each covered service. As the methods are selected and implemented, suitable documentation sets will be created to enable PTT personnel to operate and maintain the data copy operations.

(2) Objective

The objective of these tasks is to select and implement the most efficient and reliable methods to maintain the disaster recovery servers.

5.2.3 Data Centre / DR Site Change Management

This task involves the creation of change control processes required to ensure that the DR site server configurations remain synchronized with their Data Center counterparts, so that in the event of disaster fail over the required recovery parameters are met.

(1) Description

We evaluate the PTT change control system and effect changes to the system that will ensure that a parallel change process is enabled between Data Center servers and DR site servers. This parallel change control process will ease fail-over operations and ensure user operations continuity in the event of disaster. As the process changes are implemented, suitable documentation sets will be created to enable PTT personnel to operate and maintain the change control process.

(2) Objective

The objective of this task is to implement the change control process required to maintain the disaster recovery servers in parallel with Data Center servers.

5.2.4 DR Site Data Entry / Reporting Work Stations Configurations

This task involves the configuration of the DR site workstations to enable the transactional capabilities required in the stated service levels.

(1) Description

We prepare the DR site workstations with the required front-end software and configurations. The configuration of these PC's may be required to remain in place if it is determined that pre-staging is needed to meet the recovery parameters. If it is possible to package a rapid configuration procedure that meets the recovery parameters, we also create and document that package.

(2) Objective

The objective of this task is to ensure that proper workstation facilities are available in the event of disaster.

5.2.5 DR Fail-over Procedures

This task involves the creation of the procedures required to perform a fail-over operation, so that in the event of disaster fail over the required recovery parameters are met.

(1) Description

We create, test and optimize a procedure for each covered service that would be performed in the event of disaster. These may include IP re-hosting, license manipulation, hostname manipulation, staff reassignment and any other required processes.

(2) Objective

The objective of this task is to create the most efficient and reliable methods to perform disaster recovery fail-over.

5.2.6 DR Functionality Test Routines

This task involves the creation of a set of test routines whose results will measure the effectiveness of the disaster recovery services. The test routines will be used by PTT staff in the normal course of operations to periodically check the effectiveness of and identify any areas for improvement of, the DR systems and services as provided by this project.

(1) Description

We create a set of test criteria designed to measure the ability of the disaster recovery services to meet the stated service levels required. Various aspects of performance will be measured to include, recovery time, data completeness, transaction capability, staff performance.

(2) Objective

The objective of these tasks is to provide a measurement method to PTT to raise the level of confidence in the disaster recovery services.

5.2.7 DR Roles, Responsibilities and Notification Procedures

This task involves the creation of a disaster recovery organization chart and roles and responsibility definitions, so that in the event of disaster fail over the required recovery parameters are met.

(1) Description

We discuss with PTT personnel in the creation of a disaster recovery response team. The roles and responsibilities of the team members will be defined. Notification and escalation procedures must be defined.

(2) Objective

The objective of these tasks is to provide PTT staff with a framework for disaster recovery response.

VI. COST ANALYSIS

6.1 Overview

Because the CDR is new occurring in Thailand and is very hard to define about profit that is gained from the project, PTT focused on the other opportunities in term of business (presented in chapter 6) more than investment cost and try to make more profit with CDR site in the near future. However, cost is still a big issue for project implementing, so we will discuss about it in terms of concept.

The following is a simple approach to cost analysis, which probably will be needed at least for budget purpose.

First, the possible service disruption incidents must be considered and analyzed as their probability of occurrence. Second, each specific disaster recovery strategy designed to meet these incidents should be studied independently, considering its priority level and need. These disaster costs, and the recovery strategies to reduce them, can be presented for comparison in either tabular form or simply as a memo description. All assumptions of probability of occurrence should be stated.

6.2 Costs

There are many cost issues related in CDR solution, some are the investment cost, some are saving cost that gain from the project. So this section will discuss about them.

6.2.1 Fixed & Variable Costs

A key to understanding cost behavior is distinguishing variable costs from fixed costs. A variable cost is a cost that changes in direct proportion to changes in the cost driver. In contrast, a fixed cost is not immediately affected by changes in the cost driver.

Fixed Costs in Data Center are such as:

(1) Monthly rent fees, or fixed loan interests

(2) Depreciation

(3) Etc.

Variable Costs in Data Center are such as:

(1) Charges for electric consumption

(2) Charges for water consumption

(3) Additional charges from unexpected downtime

(4) Manpower

(5) Etc.

6.2.2 Cost of Possible Lost

In discussing the probability of disruptions and the probable costs with user management, a serious problem will arise. The probable costs given by different users will not be directly comparable, because they will be based on different assumption. Some users will plead importance, but will have no wish to pay for security. Other users will underestimate the catastrophe to their operations in the event of a major data processing breakdown. Still other users will ask for high reliability, and will back up their requests with sufficient available funds.

We proposed two worksheet to help users for estimating the cost of possible lost that effect to their business.

Table 6.1 Probability of Occurrence of Security Event provides an opportunity to state the most likely problem areas and intended to define, by type of asset, the probability of occurrence of service disruption incidents for each of several threats. Determining the potential cost of disasters for each of the categories of assets is the purpose of Table 6.2 Probability Economic Lost.

The objective of these two worksheets is to facilitate determining the areas of greatest exposure. In areas where probability of occurrence and economic loss combine to produce great potential business exposure, prime consideration must be given to disaster recovery strategies.

Table 6.1. Probability of Occurrence of Security Event.

	DESTRUCTION						Fraud, Theft, Etc.	Employee Error
	Fire	Flood	Earthquake	Accident	Sabotage	Other		
EDP Equipment								
Installation Facilities								
Data								
Programs & OSs								
Documentation								

Probability of Occurrence Codes:

(1) High Probability

(2) Medium Probability

(3) Low Probability

Table 6.2. Probable Economic Loss.

	Replacement or Reconstruction Cost (B)		Performance Failure Loss								
	Without Backup	With Backup	Without Backup			Current Backup			Desired Backup		
			ETR	EE	BIL	ETR	EE	BIL	ETR	EE	BIL
EDP Equipment											
Installation Facility											
Data											
Programs & OSs											
Documentation											
Total											

Definition of Terms

(1) Destruction---The loss from accidental and natural causes (e.g., flood, fire, earthquake), malicious mischief, and sabotage. Such events as riot, explosion, and erasure of magnetic files are included.

(2) Fraud, Theft, and Embezzlement---The deliberate alteration of data and programs (e.g., modification of tape, disk pack, card files, etc.) plus the removal of physical objects (e.g., tape reels, check forms, printouts, etc.) Occasionally, disastrous events are planned by the perpetrator to cover up such actions.

(3) Employee Error---Losses resulting from inadequate procedures or systems design, as well as carelessness or indifference by employees. The items particularly susceptible to this problem are data and programs, which can be completely destroyed.

(4) EDP Equipment---Computer mainframes and peripherals, plus data entry and related equipment.

(5) Installation Facilities---The total computer center other than EDP equipment. It includes the computer and key entry room and library, lighting, air conditioning, wiring, furniture, fixtures, bursters and related ancillary equipment, supplies, forms tapes, disk packs, punched card (but not replacement cost of the data or programs contained in these media) and related support facilities.

(6) Without Backup---Of or referring to a backup facility for each of the categories of assets (e.g. I/S Equipment) that either foes not exist, does not function, or for nay reason fails to provide the recovery capability that is was intended to furnish. For example, assume that the data in the secondary

storage facility was totally destroyed along with the data in the regular library. The data must then be reconstructed from source documents because there is no backup file.

(7) Current Backup---The current backup facility provides only the recovery capability that is was designed to furnish. For example, if an agreement with another installation had been made to provide computer time equivalent to 50 percent of your current work load, assume that the agreement will be honored.

(8) Desired Backup---The reasonable, cost-effective level of recovery capability you would like to archive. For example, if you currently do not have a secondary storage facility for data, programs and operating system, and documentation, assume this facility has been created and that the aforementioned materials are fully protected in this storage facility for backup purposes.

(9) ETR (Estimated Time to Recovery)---The elapsed time (in days) it is estimated it will take to fully replace the asset (e.g. data for a given level of backup.)

(10) EE (Extra Expense)---The necessary additional cost required to continue the normal operations of the business immediately following the destruction of an asset.

(11) BIL (Business Interruption Loss)---The financial loss resulting from the inability to conduct the company's normal business operations as the result of the destruction of an asset. For example, if a customer order processing system is inoperative because of loss of the programs, this event would have and impact on the profitability of the company.

6.2.3 Cost of Downtime

Cost of downtime is the cost that we will save by having the backup strategy to reduce the downtime that makes us lose the revenue and business opportunities. We can estimate the cost of downtime by the following equation:

(1) (downtime hrs.) x (average revenue per hr.) = cost of downtime

(2) (downtime hrs.) x (average # of transactions per hr.) x (average transaction value) = cost of downtime

(3) (downtime hrs.) x (# of users, i.e., employees or customers) x (cost of user hr.) = cost of downtime

6.3 Revenues

Revenues for data center mainly come from services, disaster recovery service, by charging back to other business units, or companies. However, it should be clearly defined about charging policy.

Two ideas for providing services:

(1) Provide total solutions, including hardware, software, facilities

(2) Provide only facilities, customers will be responsible for hardware, software themselves

If PTT provides total CDR solutions, charging policies are such as:

(1) Resource usage

PTT can select to charge for monthly computing time (CPU usage), or disk space occupation.

Pros: real life -- most reliable

Cons:rather difficult (need special tools)

(2) Number of key users in each application (by licenses)

Pros: easier

Cons:rather fixed -- not so good with customers

(3) Server-based

Pros: easiest

Cons:most expensive -- not suitable, or applicable

If customers acquire for hardware, software themselves, and PTT provides only facilities, charging policy is:

(4) Monthly charge (floor space, air conditioner, operations)

Pros: most applicable

Cons:- dependent to details -

6.4 Return on Investment (ROI)

A better test of profitability than a profit is the rate of return of investment (ROI), which is a percentage the return you have made relative to the amount you have invested:

ROI = [(benefits – investment) x 100] / investment

This rate can be compared with rates inside and outside the organization and with opportunities in other projects and industries.

6.5 Feasibility Analysis Study

After having idea about cost analysis, the next step is the feasibility study that are the following:

Table 6.3. Feasibility Analysis.

Cost - Benefit Anslysis Table						
Petroleum Authority of Thailand Computer Disaster Recovery System : Cost-Benefit Analysis						
Input Section						
Estimated Costs				Esimated Benefits		
Hardware	฿7,000,000.00			Reduced Time Possible Lost		฿3,000,000.00
Software	฿500,000.00			Assumptions		
Consultant	฿5,500,000.00			Discount Rate:		10%
Training	฿500,000.00			Sensitivity Factor (Cost)		1.1
Maintenance	฿450,000.00			Sensitivity Factor (Benefits)		0.9
				Annual Change in Prod. Costs		7%
				Annual Change in Benefits		5%
Calculations/Output Sections						
Costs	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Development Costs						
Hardware	฿7,000,000.00					
Software	฿500,000.00					
Consultant	฿5,500,000.00					
Training	฿500,000.00					
Total Development Costs	฿13,500,000.00					
Production Costs						
Network Leasing & Support		฿500,000.00	฿535,000.00	฿572,450.00	฿612,521.50	฿655,398.01
Maintenance/Upgrades		฿450,000.00	฿481,500.00	฿515,205.00	฿551,269.35	฿589,858.20
Annual Production Costs		฿1,000,000.00	฿1,016,500.00	฿1,087,655.00	฿1,163,790.85	฿1,245,256.21
(Present Value)		฿1,100,000.00	฿1,229,965.00	฿1,447,668.81	฿1,703,906.18	฿2,005,497.58
Accumulated Cost (Development and Production Present		฿14,600,000.00	฿15,829,965.00	฿17,277,633.81	฿18,981,539.99	฿20,987,037.57
Benefits						
Reduced Time Consuming Process		฿3,000,000.00	฿3,150,000.00	฿3,307,500.00	฿3,472,875.00	฿3,646,518.75
(Present Value)		฿3,300,000.00	฿3,811,500.00	฿4,402,282.50	฿5,084,636.29	฿5,872,754.91
Other Profit		฿500,000.00	฿525,000.00	฿551,250.00	฿578,812.50	฿607,753.13
(Present Value)		฿550,000.00	฿635,250.00	฿733,713.75	฿847,439.38	฿978,792.49
Accumulated Benefits		฿3,850,000.00	฿8,296,750.00	฿13,432,746.25	฿19,364,821.92	฿26,216,369.32
(Present Value)						
Present Value of Accumulated Gain or (Loss)		-฿10,750,000.00	-฿7,533,215.00	-฿3,844,887.56	฿383,281.93	฿5,229,331.75
Benefit/Cost Ratio						1.2491696

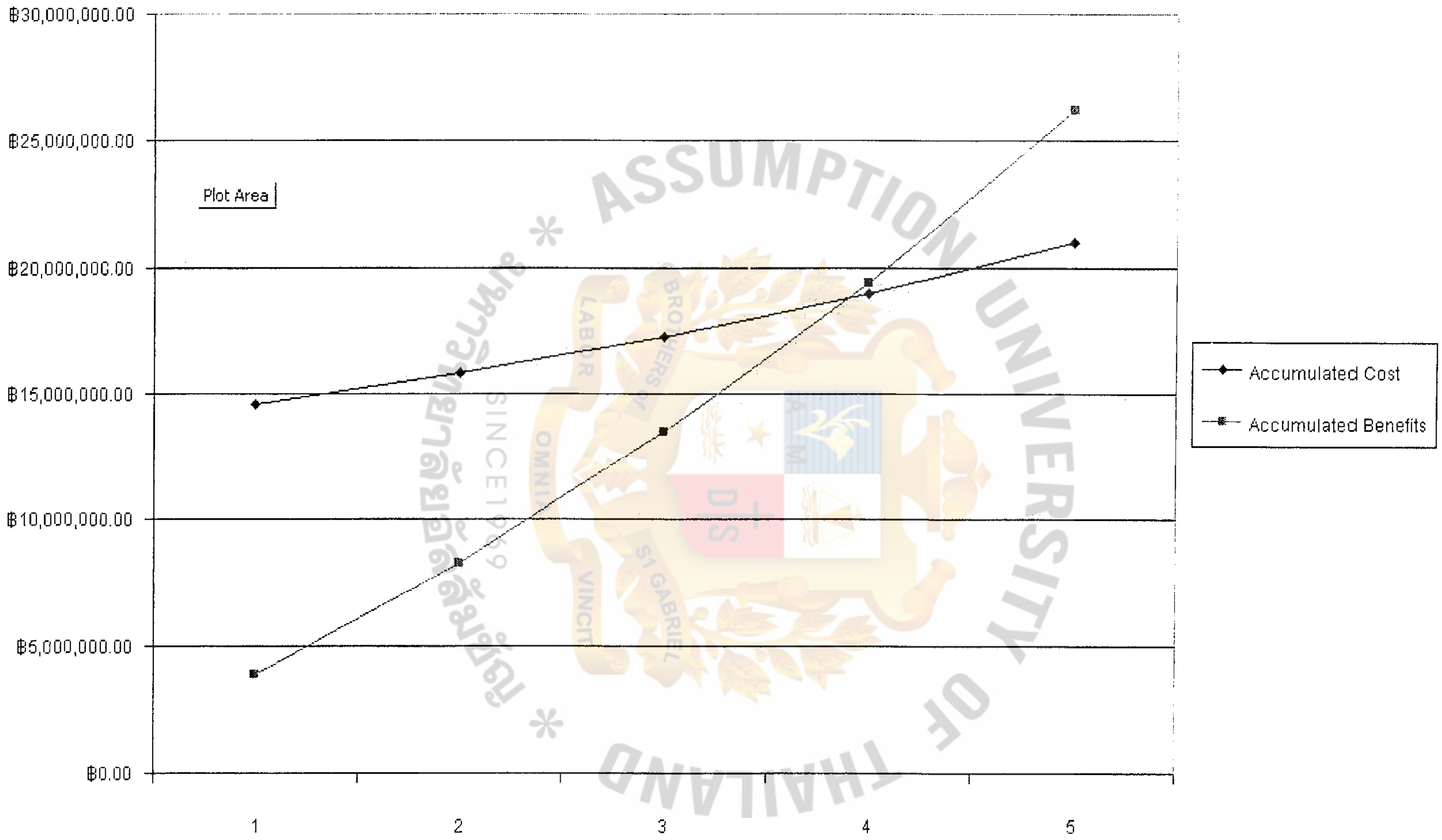


Figure 6.1. Breakeven Point.

VII. PROJECT SUMMARY

7.1 Benefits

The disaster recovery solution offers PTT the following benefits and new capabilities:

(1) Business continuity and security for PTT's IT: The CDR system helps the PTT does not lose its business opportunity which cause the losing of huge revenues in this competitive market and make all PTT's customers and partners have more confidence.

(2) Enable PTT's Information Technology unit to offer Disaster Recovery of distributed systems services to its clients.

(3) We will use this project as an opportunity to transfer knowledge to PTT's IT staffs.

(4) Simplicity in event of disaster.

(5) PTT can use the DR site for expandability and other purpose such as Training and Seminar Center , or Testing Center

(6) Increasing PTT's reputation as leader of Thailand IT organization : PTT is the first organization that implemented complete CDR solution. This shows the vision of PTT's executives that are concerned about business opportunities for the global organization

7.2 Opportunities & Recommendations

Opportunities after CDR implementing are such as:

7.2.1 Disaster Recovery

As mentioned before in section 6.1. and PTT can leverage the people skill of CDR team and facility in order to expand supporting outside (not limited to PTT group only). In this case, PTT IT may be transferred to be profit center.

7.2.2 Application Hosting

Trend technology of application deployment was effected from internet boom. It's also applied to application as well, which began with email and calendar application, which we can see today. Next step would be business application. For example, Enterprise Resource Planning, major vendors in this area start to migrate to this idea such as SAP (MySAP) and Oracle. Instead of buying everything and be owner for hardware, software package, licensing and maintenance, End-users can deploy by renting right to use application based on amount of user, transaction, etc. So they does not need to have infrastructure to support the application. It would turn user to concentrate to business related more.

PTT can leverage from existing infrastructure to do application-hosting service easily. With enough network bandwidth to CDR site or via high-speed dial-up, outside user could access application-hosting service from their locations. Advance charge-back methodology is needed to handle this kind of concept. Comparisons between buying solution and renting to use. The renting service overall cost should be cheaper as it based-on sharing concept. In addition, user (PTT CDR customer) will always get new technology or version along their duration agreement.

7.2.3 Supply Chain Management

Nowadays PTT doesn't only compete in product quality, but business process is also an important part that helps push its own business to compete globally.

Supply chain management is a thought of integrating all business processes from bringing raw materials to production process, and delivering finished goods to customers. It will create efficient flow of information, products, and service (cost and time reduction), and also cover internal & external management, such as co-working between suppliers, producers, distributors, retailers, and customers.

PTT, in the status of Producer, has one aim to minimize costs in production, by resource management. Resource management, especially production planning for serving customers' requirements, must be managed efficiently to be more flexible in responding to the market.

7.3 Conclusions

We define a disaster as any unplanned event that causes business/MIS services to be interrupted to a point (hours or days) that users require alternate services to prevent a business loss. All the operational and planning aspects of a data processing facility, then, become an integral part of disaster recovery if an unplanned event shuts down the business, disrupts operations, or causes physical damage to the facility.

The Computer Disaster Recovery (CDR) project is intended to give Petroleum Authority of Thailand (PTT) a disaster recovery capability for certain business applications. Disaster recovery capability means having the ability to transact against mission critical applications on an alternate server located at a remote site when there is a loss of the primary server. There are two CDR sites. The primary CDR site is located on 13th floor PTT Headquarter Building, Viphawadee-Rangsit, Bangkok and the backup site is located at PTT research and training center, Wang-Noi, Ayuttaya. The distance between the two sites are around 60 Kilometers.

This project is the first complete CDR solution project in Thailand. It has been implemented since October 1999 and PTT has been using the system for disaster recovery until now. There is periodic testing every 6 months to ensure that the CDR system still works correctly.

This solution will allow PTT's Information Technology unit to serve its customers in the event of disaster.

APPENDIX A

ORACLE DATABASE CONCEPT

ORACLE DATABASE ARCHITECTURE

In order to understand the design that will be used at PTT, it is necessary to touch on a bit of Oracle Architecture that will be used in the solution. Following are short descriptions of Oracle structures that need to have a cursory understanding to get a better idea of how the whole solution works.

(1) Logical Database Structures

(a) Tablespaces

The basic storage allocation in an Oracle database is a tablespace. Each tablespace is composed of one or more physical (operating system) files. Every database is created with the SYSTEM tablespace. Other tablespaces are created by the DBA.

(b) Schemas

In Oracle, a schema is essentially the same as an account or username. Each object in the database is owned by a schema. Every Oracle database is created with two initial schemas:

(1) SYS, which is used to store the data dictionary

(2) SYSTEM, which often stores some data dictionary extensions as well as critical tables for other tools.

Other schemas are created by the DBA. Each schema can be granted quotas in any tablespace.

(c) Segments

Each object that takes up space is created as one or more segments. Each segment can be in one and only one tablespace.

(d) Extents

Each segment is composed of one or more extents. An extent is a contiguous allocation of space within a data file of a tablespace. At the time a segment is created, you can specify the size of the initial and next extents, as well as the minimum and maximum number of extents.

(e) Roll back segments

Every time a table is updated, Oracle writes the old value into the rollback segment, which allows other users to maintain a consistent, read on the table. It also allows Oracle to restore the contents of the table in case a change is not committed (Saved).

(f) Tables

All data in a database is stored in table(s). Data includes not only the user data, but also the contents of the data dictionary.

(g) Indexes

Indexes are used both to facilitate the quick retrieval of data from the table and to enforce the uniqueness of column values. Indexes are stored in separate segments from the table data.

(2) Files

(a) Data Files

Data files contain exactly what you would expect to find in a relational database management system. A series of records of data arranged in tables. There are a number of types of database objects (tables, indexes, views, and so forth) that you will find in these files.

(b) Control Files

Every Oracle instance must have one or more control files. The control file is a binary file that is critical to Oracle, but is not directly readable by a user, nor is it editable by a text editor. The control file can be thought of as a software "boot-strap" file; it contains information that Oracle requires to start. Information stored in the control files includes:

(1) Name and locations of data files;

(2) Names and locations of redo log files;

(3) Information on the status of archive log files;

(4) The current redo log sequence number;

(5) Redo log information required for recovery;

(6) Timestamp information on the instance creation and startup/shutdown;

(7) Essential parameters specified at database creation (e.g., MAXDATAFILES);

(8) The information stored in the control file is so critical that if the control file is lost or damaged, the only options available for recovery are either to create a new control file (assuming that the DBA has access to all pertinent information required) or to rebuild the database and restore from a backup. Because of the critical nature of the control file, Oracle allows the DBA to maintain multiple mirrored control files, as specified by the CONTROL_FILES parameter in the INIT.ORA file.

(c) INIT.ORA File

It is the primary file that contains configuration and tuning parameters. An INIT.ORA file must exist for each Oracle instance. This file is used by Oracle when starting the database, and therefore must be located in a known location, or Oracle must be told where it is located through the use of a command-line parameter to Server Manager.

(d) On-line Redo Logs

An Oracle database must have at least two redo log files, and most databases have more than two. The online redo logs are static files that are over written many times during the course of a day of transactions against the database. These files are written by the LGWR process in a circular fashion; that is, when the last log file is filled, the first log file is reused. When archive logging is active, all the transactions are written to the Archive Logs, these log files are sequentially numbered.

Redo log files are operating system files used by Oracle to maintain logs of all transactions performed against the database. Redo logs record every transaction made to the database in a file separate from the main data files. These files can be used to recover all changes made to the database in the event that a data file is damaged. Oracle uses several of these files so that when it gets done writing to the last file in the series, it begins overwriting to the first online redo log file. The primary purpose of these log files is to allow Oracle to recover changes made to the database in case of a failure.

(e) Archive Log Files

The contents of the online redo log file and the archive log file is identical. When you finish writing to an online redo log file, a background process makes a copy of that redo log file to a separate file, Archive Log Files, that is given a unique sequential file numbered file name. This file can be placed on a magnetic disk drive or tape. Then if you want to recover a data file that has been destroyed, you get a copy from the backup media that you have been religiously making and apply all transactions in the redo and archive log files that have occurred since that backup was made.

(3) System Global Area

The System Global Area (SGA) is the most important memory structure of Oracle and consists of several different components. The SGA is a shared memory structure and contains all memory-resident data and control information for a single Oracle instance (each instance must have its own SGA). Because the SGA data is shared among all users of the instance, it is sometimes called the Shared Global Area. Oracle Processes can both read from and write to the SGA.

(a) Redo Log Buffer

The redo log buffer is an area of memory within the System Global Area (SGA) that holds information about changes to the database, called redo log entries. These entries are used if database recovery is necessary, and they contain information required to reconstruct changes made by INSERT, UPDATE, DELETE, CREATE, DROP or ALTER statements.

The redo log buffer is circular, that is, when it is full, entries are written to it from the beginning. The LGWR process writes the contents of the redo log buffer to the active redo log file on disk. The size of the redo log buffer is determined by the INIT.ORA parameter LOG_BUFFER, which is expressed in bytes.

(b) Database Buffer Cache

The largest component of the SGA is usually the database buffer cache, which is the part of the SGA that holds copies of blocks of data read from the Oracle datafiles on disk.

(4) Processes

Oracle uses a number of processes to manipulate and access the data contained in an Oracle database. Some of these processes are required in all cases, while others apply only to specific Oracle Options.

(a) Database Writer (DBWR)

The database writer process is responsible for the actual writing of data to Oracle's physical database files at the operating system level. An important part of this responsibility is the management of the database buffer cache.

When an application makes a change to data contained in an Oracle database, that change, including inserts and deletes, is first made to a memory buffer. When data is written to a memory buffer, it is marked as "dirty" as opposed to "clean" or "free". The DBWR process is responsible for writing the contents of dirty buffers to the physical disk files, thereby keeping the buffer cache clean. This action has the effect of maintaining an adequate supply of free buffers for use when a user process needs to read data from disk.

(b) Log Writer (LGWR)

The log writer process is responsible for transferring transactions that have occurred from the redo log buffer in the System Global Area (SGA) to the online redo log files. It is responsible for keeping track of which log file is ready to receive data. It is also responsible for ensuring that the online redo log file has been copied to the archive log file before the online redo log file is overwritten. The Archive Log file is only written when the database is operating under archive log mode.

(c) Archive Log Writer (ARCH)

The ARCH process is used to copy the contents of an online log file to another location, typically a disk file, when that log file becomes full. Oracle uses the location, typically a disk file, when that log file becomes full. Oracle uses the online log files in a “round robin” fashion, that is, when all available online log files become full, the first file is reused. The mode of operation whereby the contents of each file are save prior to reuse is called archivelog mode, and is controlled by the ARCHIVELOG parameter in the ALTER DATABASE statement. The ARCH process runs only when the instance is running in archivelog mode.

APPENDIX B

SERVER INFORMATION

Table B.1. Server Information.

No.	SERVER name	Brand	Model	Serial Number	CPU	MEMORY	DISK CAP.	Disk Free	OS	Application	Critical Level
1	AP/GL	SUN	Enterprise 3000	635F123A	2	512	63	84.3	Solaris 2.5.1	Financial System (AP/GL)	H
2	SAPUX2 (Productions)	SUN	Enterprise 3000	635F123B	4	1024	75.6	34.91	Solaris 2.5.1	Human resources mgnt. System (Production)	H
3	SAPUX1 (Dev. & Con.)	SUN	Enterprise 3000	713F1218	2	1024	79.8	17.97	Solaris 2.5.1	Human resources mgnt. System (Development)	M
4	PBMS	DIGITAL	Alpha 2100	N160302VC7	2	512	23.1	5.23	UNIX v 3.2 G	Performanc Base Management System (KPI)	H
5	S3-HO-BKK-MSG	Compaq	Proliant 5000	8632BHK20023	4	256	20	6.41	NT 4.0	HO. Mail System (exchange & domain server)	H

Table B.1. Server Information. (Continued)

No.	SERVER name	Brand	Model	Serial Number	CPU	MEMORY	DISK CAP.	Disk Free	OS	Application	Critical Level
6	S8-HO	Compaq	Proliant 2500	8747BPV10333	1	128	10	-	NT 4.0	PTT INTER Mail System	H
7	S4-HO	Compaq	Proliant 5000	8720HWR10032	4	256	29.9	5.3	NT 4.0	Top-Ex Information System, Calls Details Reporting System	H
8	NS1	SUN	Netra i2/1200	651F0148	1	128	9.3	7.4	Solaris 2.5.1	Soltice Firewall 1 V.2.1, DNS	H
9	S2-HO	Compaq	Proliant 5000	8649BHK20091	4	256	20'	5.67	NT 4.0	Intranet Web Page (HTML), Intranet Applications	H
10	S9-HO	Compaq	Proliant 2500	8750BPV10070	1	128	10	6.894	NT 4.0	Intranet Web Page (HTML), Intranet Applications	H

Table B.1. Server Information.

No.	SERVER name	Brand	Model	Serial Number	CPU	MEMORY	DISK CAP.	Disk Free	OS	Application	Critical Level
11	S5-HO	Compaq	Proliant 5000	8720HWR10038	4	256	29.9	5.729	NT 4.0	Telephone Information System, Projects Gurantee Deposite System, Leasing Car Profile System, PTT-Car Profile Filing System, OT Approval System, Inventory System (Maximo), Working Hours Reporting System, PTT Docmument Keeping Warehouse System	M

Table B.1. Server Information. (Continued)

No.	SERVER name	Brand	Model	Serial Number	CPU	MEMORY	DISK CAP.	Disk Free	OS	Application	Critical Level
12	S15-HO	Compaq	Workstation 5000	7734BJS20029	1	32	2.1	0.138	NT 4.0	Calls Details Recording System	M
13	S1-HO	Compaq	Proliant 5000	8652BHK20009	4	256	20	7.497	NT 4.0	PTT Forest Project Information System, Legal Document Filing System, PTT Clinical System, PTT Library Information System, Petrochemical Group Information System	M
14	S10-HO	Compaq	Deskpro XL590	115/230	1	32	0.5	0.3	NT 4.0	Data Picture Forest Project	L

Table B.1. Server Information. (Continued)

No.	SERVER name	Brand	Model	Serial Number	CPU	MEMORY	DISK CAP.	Disk Free	OS	Application	Critical Level
15	S13-HO	Compaq	Proliant 2500	8728BPT10454	1	128	20	8	NT 4.0	ASCOPE Filing System (SQL), Docmument Management System (SQL)	M
16	S16-HO	IBM	PC300GL	90-D4LL8	1	32	2.3	2.1	NT 4.0	PTT Internet Web Server, GAS BU Internet Web Server	M
17	NMM	SUN	Sparc 10	600-2933-05	1	228	6		Solaris 2.5.1	Network Management	L
18	BACKUP-HO	SUN	Enterprise 1200	717F10FD	1	256	8.4		Solaris 2.5.1	Backup Server	M
19	S6-HO	IBM	PC350	90-A55BD	1	32	1.2	0.4	NT 4.0	Messaging Domain	H

Table B.1. Server Information. (Continued)

No.	SERVER name	Brand	Model	Serial Number	CPU	MEMORY	DISK CAP.	Disk Free	OS	Application	Critical Level
20	S7-HO	Compaq	Proliant 2500	8750BPV10071	1	128	10	7.177	NT 4.0	Proxy Server	M
21	S12-HO	Compaq	Deskpro XL590	7533HHR40057	1	56	0.5	0.046	NT 4.0	Shiva Access Manager, WinS	H
22	S17-HO	IBM	PC300GL	90-D4LB3	1	32	2.3	0.3	NT 4.0	Print Server for SAP	H
23	PTT-ISFS	SUN	Sparc 5	444F1355	1	32	0.426		Solaris 2.5.1	Router Information	L
24	PTT-0A	Netframe	NF-450	D00208	1	64	4.9		Netware 3.12	Share Files, Share Printing	L
25	ISDMIS	Netframe	NF-450	E00492	1	40	4		Netware 3.12	Share Files, Share Printing	L

Table B.1. Server Information. (Continued)

No.	SERVER name	Brand	Model	Serial Number	CPU	MEMORY	DISK CAP.	Disk Free	OS	Application	Critical Level
26	FSDIC	Netframe	NF-450	E01107	1	40	2.4		Netware 3.12	Share Files, Share Printing	L
27	S11-HO	Gateway	2000	4657969	1	64	4	0.783	NT 4.0	Intranet Development	L
28	BOISIP	DIGITAL	486d2 MTE	TA414K3955	1	32	5.5		Netware 3.12	General Information Sharing	L
29	S14-HO	Compaq	Proliant 2500	8728BPT10128	2	128	20	20	NT 4.0	Backup Maximo	L

APPENDIX C

APPLICATION INFORMATION

Table C.1. Application Information.

Application	Critical	# act	KeyUsr	Bckup	Associate	DISK
	lev.[1]	usr[2]		schm.[3]	Server	USED(M)
Financial System (AP/GL)	H	50	35	D,W,M	AP/GL	13000
Performance Base Management System (KPI)	H	84	12	D,M	PBMS,S4	11971
HO. Mail System (exchange & domain server)	H	800	1	D,M	S3,S6	9530
PTT INTER Mail System	H			-	S8-HO	-
Human resources mgnt. System (Production)	M	118	16	D,M	SAPUX2	26000
Human resources mgnt. System (Development)	M	58	5	D,M	SAPUX1	17000
Top-Ex Information System	L	95		D,M	S4-HO	708.6
Calls Details Reporting System	L	4	4	D,M	S4-HO	792
Telephone Information System	L		3	D,M	S5-HO	7.83
Projects Gurantee Deposite System	L		3	D,M	S5-HO	13.2

Table C.1. Application Information. (Continued)

Application	Critical lev.[1]	# act usr[2]	Key User	Bckup schm.[3]	Associate Server	DISK USED(M)
Legal Document Filing System	L	5	3	D,M	S1-HO	44.2
Leasing Car Profile System	L		3	D,M	S5-HO	383
PTT-Car Profile Filing System	L		2	D,M	S5-HO	8.1
OT Approval System	L	800		D,M	S5-HO	13
Inventory System (Maximo)	L	5	3	M	S5-HO	240
Working Hours Reporting System (e-softwin)	M		3	M	S5-HO	17
Intranet Web Page (HTML), Intranet Applications	M	800	22	M	S2,S9,S13	23234
Calls Details Recording System	L		4	D	S15-HO	1380.6
PTT Forest Project Information System	L		2	D,M	S1-HO,S10-HO	467
PTT Docmument Keeping Warehouse System (SQL)	L		3	D,M	S5-HO	15.6

Table C.1. Application Information. (Continued)

Application	**Critical lev.**[1]	**# act usr**[2]	**Key User**	**Bckup schm.**[3]	**Associate Server**	**DISK USED(M)**
PTT Clinical System	M	5	4	D,M	S1-HO	672
PTT Library Information System	L		3	D,M	S1-HO	192
ASCOPE Filing System (SQL)	L	4	1	D,M	S13-HO	500
Docmument Management System (SQL)	M	21	5	D,M	S13-HO	1400
Backup Server	M		5	-	BACKUP-HO	
Proxy Server	M		2	-	S7-HO	2581
Print Server for SAP	M	20	4	-	S17-HO	2080
PTT Internet Web Server, GAS BU Internet Web Server	L		8	M	S16-HO	280
Soltice Firewall 1 V.2.1, DNS	H		3	-	NS1	
Shiva Access Manager, WinS	H		5	-	S12-HO	459
Network Management	L		3	-	NMM	
Petrochemical Group Information System	M		1	D,M	S1-HO	157.7
Intranet Development	L		3	M	S11-HO	3196

Table C.1. Application Information. (Continued)

Application	Critical lev.[1]	# act usr[2]	KeyUsr	Bckup schm.[3]	Associate Server	DISK USED(M)
Router Information	L		3	-	PTT-ISFS	
Share Files, Share Printing	L		2	-	PTT-OA,ISDMIS,FSDIC	
General Information Sharing	L		1	-	BOISIP	

Notes:

(1) Critical lev[1] : critical level แบ่งเป็น 3 class คือ H : high risk to business, M : medium risk to business , L : Low risk to business

(2) # act usr[2] : number of active user จำนวน user ที่มีการ assign ไว้ในระบบและได้มีการใช้ระบบจริง

(3) Bckup schm.[3]: Backup scheme แบ่งเป็น 4 แบบ Class คือ D : Daily backup, W: Weekly, M : monthly backup, Q : quarterly backup

APPENDIX D

UNIX SHELL SCRIPT

UNIX SHELL SCRIPT

For the PTT Disaster Recovery project, a number of UNIX Bourne shell scripts have been created. These scripts, and configuration files manage the transfer and application of the archive logs. These scripts are started automatically by the UNIX cron utility. The script files are contained in the following section. Appendix A contains flowcharts of the scripts.

Table D.1. List of UNIX Shell Scripts.

File Name	Server	Description
PROD_standby.cfg	Production, Standby	File that sets Environment variables.
standby_procs.sh	Production, Standby	Used on both servers and contains the functions used by the pds & sds scripts.
standby_db.pds.sh	Production	Script that is executed by the UNIX cron on the Production Server.
standby_db.sds.sh	Standby	Script that is executed by the UNIX cron on the Standby Server.

(1) standby_db.procs.sh

This is a library of functions called by the standby_db.*ds.sh scripts. Any new functions that are created for either the production or standby server should be placed in this script.

(2) standby_db.pds.sh

This script, run on the production server, is called by the oracle user's crontab every 10 minutes. The script builds a list of all the archived logs on the production server, it then checks if each log has been transferred or is the most recently archived log. If a given log is not the newest log and it has not been transferred already, the script will make a copy of the log, compress it, send it to the standby server, and record the transfer.

(3) standby_db.sds.sh

This script, run on the standby server, is called by the oracle user's crontab every 10 minutes. The script checks for the presence of compressed archive logs in the receiving directory, if any compressed logs are present, it checks if it is the most recently arrived log. If a given log is not the newest log, the script will uncompress the log, copy it to the archive log directory, and apply it to the standby database.

(4) PROD_standby.cfg

This configuration file defines the location of various files and directories, the hostnames of the production and standby systems and the Oracle instance ID's. Use of this configuration file enables the portability of the scripts from one Oracle instance to another.

(1) PROD_standby.cfg

This is the configuration file that defines the location of various files and directories, the hostnames of the production and standby systems and the Oracle instance ID’s. Use of this configuration file enables the portability of the scripts from one Oracle instance to another.

```
#!/bin/sh
################################################################################
#      PROGRAM INFORMATION:
#      NAME:                         standby.conf
#      TITLE:                        Oracle standby database configuration file
#      VERSION:                      1.0
#      USAGE:                        {STANDBY_SID}standby.conf
#      ARGUMENTS:
#
#      DESCRIPTION:
#                               Sets environment variables used by stanby_db
#                               scripts for compression of redo log files and
#                               transmission to remote database system.
#                               Scripts designed to allow multiple instances to
#                               run concurrently.
#
#                               Each Oracle Sid has its own configuration file.
#                               i.e. ${STANDBY_SID}standby.conf
#                               Configuration files reside in $STANDBY_BASE/etc
#                               Script files reside in $STANDBY_BASE/bin
#                               Log files are generated in $STANDBY_BASE/log
#
#      SPECIAL REQUIREMENTS:
```

```
#                                   These have to be customized to match your system
#                                   they will be inherited from the environment if
#                                   set, otherwise they have to be explicitly
#                                   defined here
#
#       CALLED BY:
#
#
#       CALLS:                              Sourced standby_procs.sh
#
#
##################################################################################
#
#
#       Standby Oracle SID to be processed
#
STANDBY_SID=PROD; export STANDBY_SID
echo "STANDBY_SID = $STANDBY_SID"
#
#
#       Identify Oracle Software Location
#
STANDBY_ORACLE_HOME=/oracle734/app/oracle/product/7.3.4; export
STANDBY_ORACLE_HOME
echo "STANDBY_ORACLE_HOME = $STANDBY_ORACLE_HOME"
#
#
#       Base directory location where standby
#       Scripts and configuration files are installed
#
```

```
STANDBY_BASE=/oracle734/app/oracle/standby; export STANDBY_BASE

echo "STANDBY_BASE = $STANDBY_BASE"

#

#

#       Standby control file

#

STANDBY_CONTROLFILE=/oracle734/oradata/PROD/ctlstby.ora

echo "STANDBY_CONTROLFILE = $STANDBY_CONTROLFILE"

#

#

#       Location of redo Logs

#

STANDBY_REDOARCH=/db_archive/PROD; export STANDBY_REDOARCH

echo "STANDBY_REDOARCH = $STANDBY_REDOARCH"

#

#

#       Redo log file name prefix, oracle default is "arch_SID_"

#       this format must also be defined in the init.ora file for the SID

#

STANDBY_ARCH_PREFIX=apgl${STANDBY_SID} l_; export STANDBY_ARCH_PREFIX

echo "STANDBY_ARCH_PREFIX = $STANDBY_ARCH_PREFIX"

#

#

#       Should be defined if a suffix is used. Oracle default is ".dbf"

#

STANDBY_ARCH_SUFFIX=".dbf" export STANDBY_ARCH_SUFFIX

echo "STANDBY_ARCH_SUFFIX = $STANDBY_ARCH_SUFFIX"

#

#       Production Database Server Host name

#
```

```
STANDBY_PDS_SRV=apgl; export STANDBY_PDS_SVR
echo "STANDBY_PDS_SRV = $STANDBY_PDS_SRV"
#
#
#       Standby Database Server Name
#       must have .rhosts enabled to allow remote copy
#
STANDBY_SDS_SRV=cdr-ho-ux01; export STANDBY_SDS_SVR
err_msg "STANDBY_SDS_SRV = $STANDBY_SDS_SRV"
#
#
#       Destination directory to send compressed redo archives
#
STANDBY_DEST=/db_archive/PROD_CMPR; export STANDBY_DEST
echo "STANDBY_DEST = $STANDBY_DEST"
#
#
#       Destination directory for log files
#
STANDBY_FDEST=${STANDBY_BASE}/log; export STANDBY_FDEST
echo "STANDBY_FDEST = $STANDBY_FDEST"
#
#
#       Files to be copied to Standby Database Server
#
STANDBY_UFILES="/var/opt/oracle/oratab
${STANDBY_ORACLE_HOME}/network/admin/listener.ora
${STANDBY_ORACLE_HOME}/network/admin/tnsnames.ora
${STANDBY_ORACLE_HOME}/pfile/init${STANDBY_SID}.ora
```

```
${STANDBY_ORACLE_HOME}/pfile/config${STANDBY_SID}.ora
${STANDBY_ORACLE_HOME}/dbs/sgadef${STANDBY_SID}.dbf"; export STANDBY_UFILES
echo "STANDBY_UFILES = $STANDBY_UFILES"
#
#
#       Set to "true" allow deletion of Archive Logs
#
STANDBY_ARCHIVE_DELETE=false; export STANDBY_ARCHIVE_DELETE
echo "STANDBY_ARCHIVE_DELETE = $STANDBY_ARCHIVE_DELETE"
#
#
#       Number of days before redo file deletion
#       (Valid only if STANDBY_ARCHIVE_DELETE is set to true)
#
STANDBY_ARCHIVE_REMOVAL_DAYS=5; export STANDBY_ARCHIVE_REMOVAL_DAYS
echo "STANDBY_ARCHIVE_REMOVAL_DAYS = $STANDBY_ARCHIVE_REMOVAL_DAYS"
#
#
#       Lock file to indicate that hot backup is
#       running the redo logs should not be touched
#     If no hotbackups, set to impossible file name
#
STANDBY_HOT_BACKUP_LOCK=/tmp/no_SID_backup; export
STANDBY_HOT_BACKUP_LOCK
echo "STANDBY_HOT_BACKUP_LOCK = $STANDBY_HOT_BACKUP_LOCK"
#
#
#       Number of seconds to wait between polls set
#       this to a value, slightly less than the log switch
#       time for the database instance
```

```
#
STANDBY_LOOP_TIME=1400; export STANDBY_LOOP_TIME
echo "STANDBY_LOOP_TIME = $STANDBY_LOOP_TIME"
#
#
#       Identify Standby Log file name and locations
#       These should not require changing
#
STANDBY_LOG=$STANDBY_BASE/log/${STANDBY_SID}log; export STANDBY_LOG
echo "STANDBY_LOG = $STANDBY_LOG"
#
#
#       Temporary list of Archive Logs
#
ARCHIVE_LIST=$STANDBY_BASE/tmp/archive_list; export ARCHIVE_LIST
echo "ARCHIVE_LIST = $ARCHIVE_LIST"
#       Log of compressed files
#
STANDBY_LOG_FILE_Z=$STANDBY_BASE/log/${STANDBY_SID}log_file_z; export
STANDBY_LOG_FILE_Z
echo "STANDBY_LOG = $STANDBY_LOG"
#
#
#       Log of Transferred files
#
STANDBY_LOG_FILE_T=$STANDBY_BASE/log/${STANDBY_SID}log_file_t; export
STANDBY_LOG_FILE_T
echo "STANDBY_LOG_FILE_T = $STANDBY_LOG_FILE_T"
#
#
```

```
#       Program Terminates after next loop if file is removed
#
STANDBY_RUN_FILE=/tmp/${STANDBY_SID}run; export STANDBY_RUN_FILE
echo "STANDBY_RUN_FILE = $STANDBY_RUN_FILE"
#
#
#       Error Log File
#
STANDBY_ERROR_FILE=$STANDBY_BASE/log/${STANDBY_SID}error_file; export
STANDBY_ERROR_FILE
echo "STANDBY_ERROR_FILE = $STANDBY_ERROR_FILE"
#
#
#       Temp files that are written to in many of the shell scripts
#
STANDBY_TMP_F1=$STANDBY_BASE/log/${STANDBY_SID}tmp1 ;export STANDBY_TMP_F1
STANDBY_TMP_F2=$STANDBY_BASE/log/${STANDBY_SID}tmp2 ;export STANDBY_TMP_F2
STANDBY_TMP_F3=$STANDBY_BASE/log/${STANDBY_SID}tmp3 ;export STANDBY_TMP_F3
STANDBY_TMP_F4=$STANDBY_BASE/log/${STANDBY_SID}tmp4 ;export STANDBY_TMP_F4
echo "STANDBY_TMP_F1 = $STANDBY_TMP_F1"
echo "STANDBY_TMP_F2 = $STANDBY_TMP_F2"
echo "STANDBY_TMP_F3 = $STANDBY_TMP_F3"
echo "STANDBY_TMP_F4 = $STANDBY_TMP_F4"
#
#
#
#
STANDBY_UFILE_LOG=$STANDBY_BASE/log/${STANDBY_SID}ufile; export
STANDBY_UFILE
echo "STANDBY_UFILE_LOG = $STANDBY_UFILE_LOG"
```

```
#
#
#        File kept to archive the alert file name
#
STANDBY_ALERT_FILE=${STANDBY_BASE}/log/${STANDBY_SID}alert_filename; export
STANDBY_ALERT_FILE
echo "STANDBY_ALERT_FILE = $STANDBY_ALERT_FILE"
#
#
#        Add to distribution list
#
STANDBY_UFILES="$STANDBY_UFILES $STANDBY_ALERT_FILE"; export
STANDBY_UFILES
echo "STANDBY_UFILES = $STANDBY_UFILES"
#
#
#
#
STANDBY_PID_FILE=$STANDBY_BASE/log/${STANDBY_SID}pid; export STANDBY_PID_FILE
echo "STANDBY_PID_FILES = $STANDBY_PID_FILES"
#
#
#        Directory to send acknowledgement log file
#
STANDBY_ACK_FILEP=${STANDBY_BASE}/log; export STANDBY_ACK_FILEP
echo "STANDBY_ACK_FILEP = $STANDBY_ACK_FILEP"
#
#
#        Acknowledgement file name
#        Contains list of successfully uncompressed
```

```
#	redo files
#
STANDBY_ACK_FILE=ack.file; export STANDBY_ACK_FILE
echo "STANDBY_ACK_FILE = $STANDBY_ACK_FILE"
#
#
#	Maximum number of lines in history log file
#
STANDBY_MAX_HIS_SIZE=5000; export STANDBY_MAX_HIS_SIZE
echo "STANDBY_MAX_HIS_SIZE = $STANDBY_MAX_HIS_SIZE"
#
#
#	Retry time if rcp or compress operation fails
#
STANDBY_RETRY_TIME=120; export STANDBY_RETRY_TIME
echo "STANDBY_RETRY_TIME = $STANDBY_RETRY_TIME"
#
#
#	email id to send alert scan messages
#
STANDBY_ALERT_MAILID=cdr@ptt.or.th; export STANDBY_ALERT_MAILID
echo "STANDBY_ALERT_MAILID = $STANDBY_ALERT_MAILID"
#
#
#	Customer command to be executed in the event
#	of a problem found when scanning logs etc
#	(i.e., open change ticket)
#
STANDBY_CUSTOMER_COMMAND="echo no customer command"; export
STANDBY_CUSTOMER_COMMAND
```

```
echo "STANDBY_CUSTOMER_COMMAND = $STANDBY_CUSTOMER_COMMAND"
#
#
#
#
STANDBY_SCAN_F1=$STANDBY_BASE/log/${STANDBY_SID}scan_alter
STANDBY_SCAN_F2=$STANDBY_BASE/log/${STANDBY_SID}scan_alter_hits
STANDBY_SCAN_F3=$STANDBY_BASE/log/${STANDBY_SID}scan_hits
echo "STANDBY_SCAN_F1 = $STANDBY_SCAN_F1"
echo "STANDBY_SCAN_F2 = $STANDBY_SCAN_F2"
echo "STANDBY_SCAN_F3 = $STANDBY_SCAN_F3"
#
#
#       Set Oracle variables now with STANDBY this
#       prevents having to reset them everywhere
#
ORACLE_HOME=$STANDBY_ORACLE_HOME; export ORACLE_HOME
echo "ORACLE_HOME = $ORACLE_HOME"
#
#
#       Set the Oracle SID to the Standby SID
#
ORACLE_SID=$STANDBY_SID; export ORACLE_SID
echo "ORACLE_SID = $ORACLE_SID"
#
#
#       Define Path
#
PATH=$STANDBY_BASE/bin:$STANDBY_ORACLE_HOME/bin:/sbin:$PATH; export PATH
echo "PATH = $PATH"
```

```
#

#

#        Initialize (Source) standby internal procedures

#

. standby_procs.sh


#get_seqno_svrmgr
```

(2) standby_procs.sh

This is a library of functions called by the standby_db.*ds.sh scripts. Any new functions that are created for either the production or standby servers must be adoed to this script on both servers.

```
#!/bin/sh
####################################################################################
#
#       PROGRAM INFORMATION:
#       NAME:                       standby_procs.sh
#       TITLE:                      Procedures and functions used by Standby scripts
#       VERSION:                    1.0
#       USAGE:                      standby_procs
#       ARGUMENTS:
#
#       DESCRIPTION:
#                               Used to load various procedures used by the
#                               standby scripts into memory for usage.
#
#       SPECIAL REQUIREMENTS:
#                               STANDBY_* environment variables should already
#                               be set by executing the SID.standby.cfg file.
#
####################################################################################
#
####################################################################################
#
# PROGRAM INFORMATION:
#       NAME:                       err_msg()
#       TITLE:                         Append error message to log file
#       VERSION:                       1.0
#       USAGE:                         err_msg message
#       ARGUMENTS:                     message Error message to append to file.
#
#       DESCRIPTION:                   Used to add an error message to the log file
#                                      current
#
#       SPECIAL REQUIREMENTS:
#
#
#####################################################################################

err_msg () {
echo  "`date` $* " >> $STANDBY_ERROR_FILE
}

#####################################################################################
#
# PROGRAM INFORMATION:
#       NAME:                          trim_logs.sh
#       TITLE:                         Trim the standby logs
#       VERSION:                  1.0
#       USAGE:                         trim_logs
#       ARGUMENTS:
#
#       DESCRIPTION:                   Keeps the log files trimed to the value
#                                 $STANDBY_MAX_HIS_SIZE set in
#                                 $STANDBY_BASE/etc/standby.cfg
#
```

```
#       SPECIAL REQUIREMENTS:
#
######################################################################################

trim_logs () {
for f in $STANDBY_LOG_FILE_T $STANDBY_ERROR_FILE $STANDBY_LOG
do
if [ -f $f ];
then
wc -l $f > $STANDBY_TMP_F2
read SIZE FILE <  $STANDBY_TMP_F2
if [ $SIZE -gt $STANDBY_MAX_HIS_SIZE ];
then
tail -$STANDBY_MAX_HIS_SIZE $f > $f.new
rm $f
mv $f.new $f
fi
else
echo "0" > $STANDBY_LOG_FILE_T
fi
done
}

######################################################################################
#
# PROGRAM INFORMATION:
#       NAME:                   newest_log.sh
#       TITLE:                  Get the newest log number
#       VERSION:            1.0
#       USAGE:                  newest_log
#       ARGUMENTS:
#
#       DESCRIPTION:            Finds the newest log available in $STANDBY_REDOARCH
#                       on either STANDBY_PDS_SERVER or STANDBY_SDS_SERVER
#                       depending on where you run it. It then strips out the
#                       sequence number...
#
#       SPECIAL REQUIREMENTS:
#
#
##############################################################################


newest_log () {
NEWEST_LOG=`ls -lt $STANDBY_REDOARCH | grep dbf | head -1 | awk '{print $9}' | cut -c11-20 |
cut -d . -f1
export NEWEST_LOG
}

##############################################################################
#
# PROGRAM INFORMATION:
#       NAME:                   newest_cmpr_log.sh
#       TITLE:                  Get the newest compressed log number
#       VERSION:        1.0
#       USAGE:                  newest_cmpr_log
#       ARGUMENTS:
#
#       DESCRIPTION:            Finds the newest compressed log available in
#                       $STANDBY_DEST on either STANDBY_PDS_SERVER or
#                       STANDBY_SDS_SERVER depending on where you run it.  It
#                                  then strips out the sequence number...
#
#       SPECIAL REQUIREMENTS:
#
################################################################################


newest_cmpr_log () {
NEWEST_CMPR_LOG=`ls -lt $STANDBY_DEST | grep Z | head -1 | awk '{print $9}' | cut -c11-20 | cut
-d . -f1`
export NEWEST_CMPR_LOG
}


################################################################################
#
# PROGRAM INFORMATION:
#       NAME:                    svrmgrl_seq.sh
#       TITLE:                   Determine the sequence number
#       VERSION:            1.0
#       USAGE:                   svrmgrl_seq
#       ARGUMENTS:
#
#       DESCRIPTION:
#
#       SPECIAL REQUIREMENTS:
#
################################################################################


svrmgrl_seq () {
svrmgrl <<EOF
connect internal
```

```
set echo on

select max(sequence#) from v\$log_history;

exit

EOF

}


##########################################################################################

#

# PROGRAM INFORMATION:

#       NAME:                         sds_log_check.sh

#       TITLE:                        Check logs application

#       VERSION:                1.0

#       USAGE:                        sds_log_check

#       ARGUMENTS:

#

#       DESCRIPTION:

#

#       SPECIAL REQUIREMENTS:

#

##########################################################################################


sds_log_check () {

svrmgrl_seq > $STANDBY_BASE/log/log_data

newest_log

REQUESTED_LOG=`cat $STANDBY_BASE/log/log_data | tail -3 | head -1`

if [ `expr $REQUESTED_LOG + 6` -gt $NEWEST_LOG ]

then

rmail cdr@ptt.or.th << EOF

Subject: Maintenance problem on Standby database for $STANDBY_ORG
```

```
Alert log problem on Standby database
server `hostname`
instance $SID
Check Standby database for potential maintenance requirements
EOF
fi
}

#####################################################################################
#
# PROGRAM INFORMATION:
#       NAME:                   send_logs.sh
#       TITLE:                  Log packing & sending PDS ---> SDS
#       VERSION:          1.0
#       USAGE:                  send_logs
#       ARGUMENTS:
#
#       DESCRIPTION:
#
#       SPECIAL REQUIREMENTS:
#
#       HISTORY:
#####################################################################################

send_logs () {
if [ `/usr/bin/hostname` != $STANDBY_PDS_SRV ]
then
        err_msg "$0: Should only be run on the production database server. "
else
        for FILE in `ls $STANDBY_REDOARCH | grep dbf`
```

```
    do
        SEQ_NUM=`echo $FILE | cut -c11-20 | cut -d . -f1`
        TRANSFERED=`fgrep $SEQ_NUM $STANDBY_LOG_FILE_T`
        if [ $SEQ_NUM -lt $NEWEST_LOG ] && [ -z "$TRANSFERED" ]
        then
            cp -pr $STANDBY_REDOARCH/$FILE $STANDBY_DEST
            if [  $? -ne 0 ]
            then
                err_msg "$0: Copy of $FILE from $STANDBY_REDOARCH to $STANDBY_DEST failed"
                exit 1
            else
                echo "`date` Copy of $FILE from $STANDBY_REDOARCH to $STANDBY_DEST complete" >> $STANDBY_LOG
            fi
            compress -f $STANDBY_DEST/$FILE
if [  $? -ne 0 ]
then
err_msg "$0: Compress of $FILE in $STANDBY_DEST failed"
exit 1
else
echo "`date` Compress of $FILE in $STANDBY_DEST complete" >> $STANDBY_LOG
fi
            rcp -pr $STANDBY_DEST/$FILE.Z $STANDBY_SDS_SRV:$STANDBY_DEST
            if [  $? -ne 0 ]
            then
                err_msg "$0: rcp of $FILE.Z to $STANDBY_SDS_SRV failed"
                exit 1
            else
```

```
                              echo "`date` rpc of $FILE.Z to $STANDBY_SDS_SRV complete" >>
$STANDBY_LOG
                              echo "$SEQ_NUM" >> $STANDBY_LOG_FILE_T
                              rm $STANDBY_DEST/$FILE.Z
                        fi
                  fi
      done
fi
rm /tmp/PRODrun
}

#######################################################################################
#
#     PROGRAM INFORMATION:
#     NAME:             apply_logs.sh
#     TITLE:            Apply logs to SDS
#     VERSION:          1.0
#     USAGE:            apply_logs
#     ARGUMENTS:
#
#     DESCRIPTION:
#
#     SPECIAL REQUIREMENTS:
#
#######################################################################################

apply_logs () {
if [ `/usr/bin/hostname` != $STANDBY_SDS_SRV ]
then
err_msg "$0: Should only be run on the standby database server. "
```

```
else
for FILE in `ls $STANDBY_DEST | grep Z`
do
        SEQ_NUM=`echo $FILE | cut -c11-20 | cut -d . -f1`
        if [ $SEQ_NUM -lt $NEWEST_CMPR_LOG ]
        then
            uncompress -f $STANDBY_DEST/$FILE
            if [  $? -ne 0 ]
            then
                err_msg "$0: Uncompress of $FILE in $STANDBY_DEST failed"
                exit 1
            else
                echo "`date` Uncompress of $FILE in $STANDBY_DEST complete" >>
$STANDBY_LOG
            fi
        fi
    done
    for FILE in `ls $STANDBY_DEST | grep -v Z`
    do
        cp -pr $STANDBY_DEST/$FILE $STANDBY_REDOARCH
        if [  $? -ne 0 ]
        then
            err_msg "$0: cp of $FILE to $STANDBY_REDOARCH failed"
            exit 1
        else
            echo "`date` cp of $FILE to $STANDBY_REDOARCH complete" >>
$STANDBY_LOG
            rm $STANDBY_DEST/$FILE
        fi
    done
```

```
fi

svrmgrl <<EOF
connect internal
set echo on
RECOVER AUTOMATIC STANDBY DATABASE UNTIL CANCEL
cancel
exit
EOF

rm /tmp/PRODrun
}

################################################################################
#       End of Script
################################################################################
```

(3) standby_db_pds.sh

This script, run on the production server, is called by the oracle user's crontab every 10 minutes. The script builds a list of all the archived logs on the production server, it then checks if each log has been transferred or is the most recently archived log. If a given log is not the newest log and it has not been transferred already, the script will make a copy of the log, compress it, send it to the standby server, and record the transfer.

```
#!/bin/sh
##################################################################################
#
# PROGRAM INFORMATION:
#       NAME:                   standby_db.pds.sh
#       TITLE:                  Oracle standby database script for PDS
#       VERSION:                1.0
#       USAGE:                  standby_db.pds.sh STANDBY_SID
#       ARGUMENTS:
#                       Argument    Description
#                       -------------------
#                       SID         Instance this process is being run against.
#
#       DESCRIPTION:
#
#       SPECIAL REQUIREMENTS:   STANDBY_BASE environment variable must be defined.
#                       Requires rhost permissions on remote system.
#
##############################################################################
##############################################################################
```

```
#       Check for arguments
################################################################################


if [ $# -ne 1 ]
then
        echo  "usage $0 STANDBY_SID"
        exit 1
fi


################################################################################
#       Assisgn SID to input argument
################################################################################


SID=$1


################################################################################
#       Check to see if user is ora734 & source .profile
################################################################################


ID=`/usr/xpg4/bin/id -un`


case "$ID" in
'ora734')       ;;
*)
        echo "Must be run as user ora734"
        echo "Exiting"
        exit 1
        ;;
esac
. /oracle734/app/oracle/standby/etc/.profile
```

```
################################################################################
#        Find configuration files
################################################################################

if [  -f ${STANDBY_BASE}/etc/${SID}_standby.cfg ]
then
        ENVF=${STANDBY_BASE}/etc/${SID}_standby.cfg
else
        echo "can't locate config file"
        echo "${STANDBY_BASE}/etc/${SID}_standby.cfg"
        echo "Exiting"
        exit 1
fi

################################################################################
#        Initialize (Source) shells
################################################################################

. $ENVF
. ${STANDBY_BASE}/bin/standby_procs.sh

################################################################################
#        Log the start of the loop
################################################################################

err_msg "Started $STANDBY_SID"

################################################################################
#        Check if program already running
```

```
################################################################################


if [ -f $STANDBY_RUN_FILE ]; then

        err_msg "Process $$ already running, remove lockfile  $STANDBY_RUN_FILE to\

                        allow restart"

exit 1

else

        touch $STANDBY_RUN_FILE

fi


################################################################################

#       Packup and send the Oracle archive logs to the SDS server

################################################################################


newest_log


send_logs


####################################################################################

#     House keeping on standby database log files

####################################################################################


trim_logs


################################################################################

#       End of script

################################################################################
```

(4) standby_db.sds.sh

This script, run on the standby server, is called by the oracle user's crontab every 10 minutes. The script checks for the presence of compressed archive logs in the receiving directory, if any compressed logs are present, it checks if it is the most recently arrived log. If a given log is not the newest log, the script will uncompress the log, copy it to the archive log directory, and apply it to the standby server.

```
#!/bin/sh
################################################################################
#
#       PROGRAM INFORMATION:
#       NAME:                   standby_db.sds.sh
#       TITLE:                  Oracle standby database script for SDS
#       VERSION:                1.0
#       USAGE:                  standby_db.sds.sh STANDBY_SID
#       ARGUMENTS:
#                               Argument    Description
#                               --------- -----------
#                               SID         Instance this process is being run against.
#
#       DESCRIPTION:                    Queries database for SEQ number of last redo log file
#                               Maintains a file history of all redo log files which have
#                               been compressed or transferred to the remote system
#                               A script on the remote system is required to uncompress
#                               and apply the redo logs to the Standby database.
#
#                               Each Oracle Sid has its own configuration file.
#                               i.e. ${STANDBY_SID}standby.pds.cfg
#                         Configuration files reside in          $STANDBY_BASE/etc
#                         Script files reside in                 $STANDBY_BASE/bin
#                         Log files are generated in             $STANDBY_BASE/log
#
#                         This program runs continuoulsy.  It can be stopped by
#                         deleting  the  STANDBY_RUN_FILE.    If  STANDBY_RUN_FILE
exists it assumes that
#                         another instance of the program is already running, so it
#                         will not start.
#                         The          default          STANDBY_RUN_FILE          is
/tmp/{$STANDBY_SID}run_standby_db
#                         It sleeps for STANDBY_LOOP_TIME seconds.  This time should be
set to
#                         be slightly less than the log switch interval
#
#      SPECIAL REQUIREMENTS:
#                         STANDBY_BASE environment variable must be defined.
#                         Requires rhost permissions on remote system.
#                         Must be run as User oracle
#
##########################################################################################
##########################################################################################
#      Check for arguments
##########################################################################################

if [ $# -ne 1 ]
then
      echo  "usage $0 STANDBY_SID"
      exit 1
```

```
fi

################################################################################
#       Assisgn SID to input argument
################################################################################

SID=$1

################################################################################
#       check to see if user is Oracle
################################################################################

ID=`/usr/xpg4/bin/id -un`

case "$ID" in
'ora734')       ;;
*)
        echo "Must be run as user ora734"
        echo "Exiting"
        exit 1
        ;;
esac
. /oracle734/app/oracle/standby/etc/.profile

################################################################################
#       Find configuration files
################################################################################

if [  -f ${STANDBY_BASE}/etc/${SID}_standby.cfg ]
then
```

```
        ENVF=${STANDBY_BASE}/etc/${SID}_standby.cfg
else
        echo "can't locate config file"
        echo "${STANDBY_BASE}/etc/${SID}_standby.cfg"
        echo "Exiting"
        exit 1
fi

################################################################################
#       Initialize (Source) shells
################################################################################

. $ENVF
. ${STANDBY_BASE}/bin/standby_procs.sh

################################################################################
#       Echo STANDBY_SID
################################################################################

err_msg "Started $STANDBY_SID"

################################################################################
#       Check if program already running
################################################################################

if [ -f $STANDBY_RUN_FILE ]; then
        err_msg "Process $$ already running, remove lockfile  $STANDBY_RUN_FILE to\
                        allow restart"
exit 1
else
```

```
        touch $STANDBY_RUN_FILE
fi

##########################################################################################
#       Start loop processing
##########################################################################################

newest_cmpr_log

apply_logs

##########################################################################################
#     House keeping on standby database log files
##########################################################################################

trim_logs

##########################################################################################
#       End of script
##########################################################################################
```

APPENDIX E

ARCSERVE IT SOFTWARE INSTALLATION INFORMATION

Table E.1. NT Software Component Installation Table.

Servers	SOFTWARE COMPONENTS
S1-HO	ArcserveIT Replication ArcserveIT NT Agent
S2-HO	ArcserveIT SERVER ENTERPRISE EDITION ArcserveIT Open file backup ArcserveIT Disaster Recovery
S3-HO	ArcserveIT SERVER ENTERPRISE EDITION ArcserveIT Replication ArcserveIT Disaster Recovery
S4-HO	ArcserveIT SERVER ENTERPRISE EDITION ArcserveIT Open file backup ArcserveIT Disaster Recovery ArcserveIT NT Agent
S5-HO	ArcserveIT Replication ArcserveIT NT Agent ArcserveIT SERVER ENTERPRISE EDITION ArcserveIT Disaster Recovery
S6-HO	ArcserveIT NT Agent
S1-Inter-BKK	ArcserveIT Replication ArcserveIT NT Agent

Table E.1. NT Software Component Installation Table. (Continued)

Servers	SOFTWARE COMPONENTS
S9-HO	ArcserveIT SERVER ENTERPRISE EDITION ArcserveIT Open file backup ArcserveIT Disaster Recovery
S13-HO	ArcserveIT Replication ArcserveIT NT Agent
S14-HO	ArcserveIT Replication ArcserveIT NT Agent
S15-HO	ArcserveIT Replication ArcserveIT NT Agent
S16-HO	ArcserveIT NT Agent ArcserveIT Open file backup
S18-HO	ArcserveIT NT Agent
S20-HO	ArcserveIT Replication ArcserveIT NT Agent
S22-HO	ArcserveIT NT Agent
CDR-HO-NT01	ArcserveIT Replication ArcserveIT NT Agent
CDR-HO-NT02	ArcserveIT SERVER ENTERPRISE EDITION ArcserveIT Replication ArcserveIT NT Agent

Table E.1. NT Software Component Installation Table. (Continued)

Servers	SOFTWARE COMPONENTS
CDR-HO-NT03	ArcserveIT NT Agent
CDR-HO-NT04	ArcserveIT Replication ArcserveIT NT Agent
CDR-HO-NT05	ArcserveIT Replication ArcserveIT NT Agent
CDR-HO-NT06	ArcserveIT NT Agent

APPENDIX F

DISASTER RECOVERY DIAGRAM

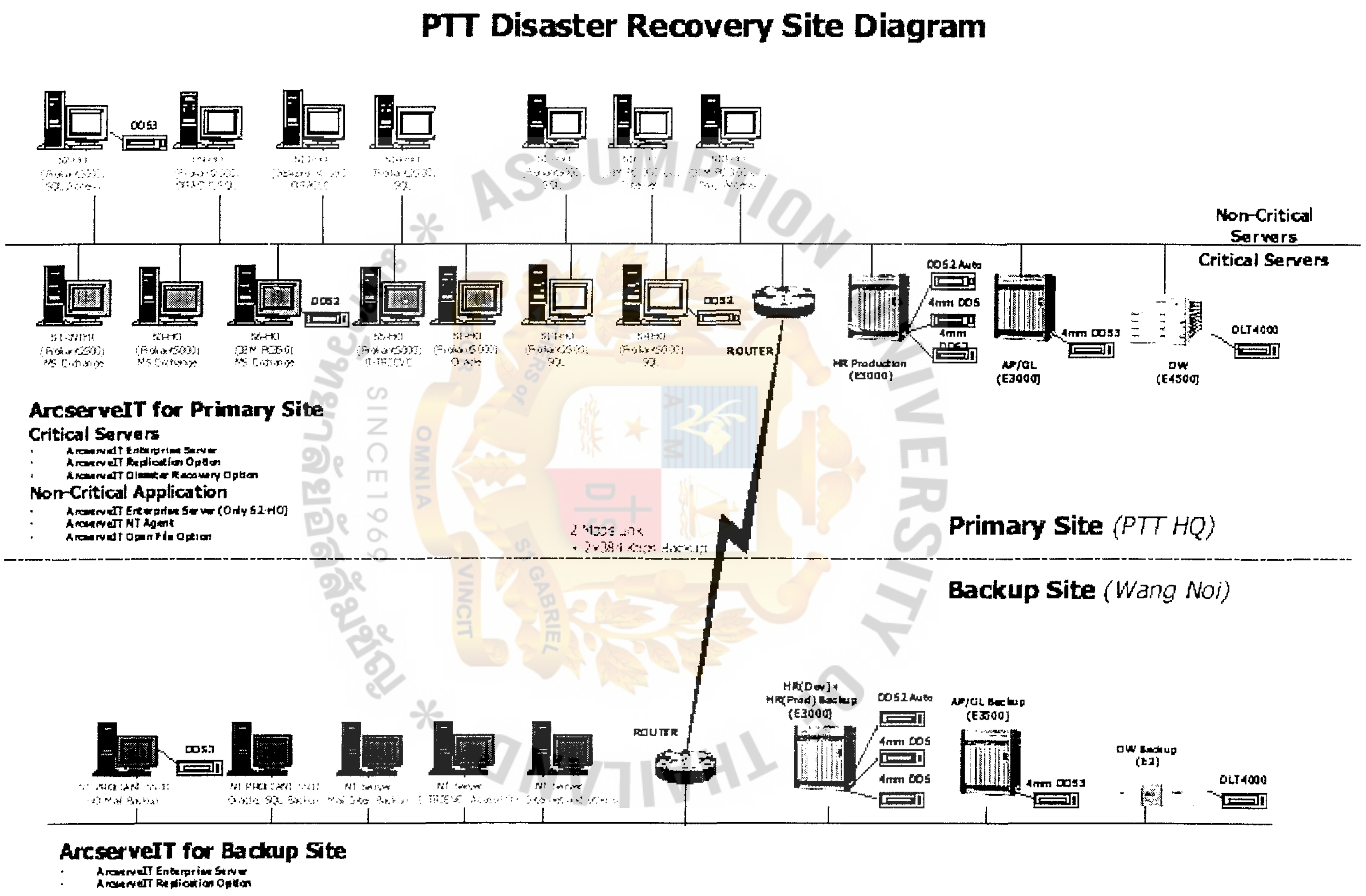


Figure F.1. CDR System Diagram.

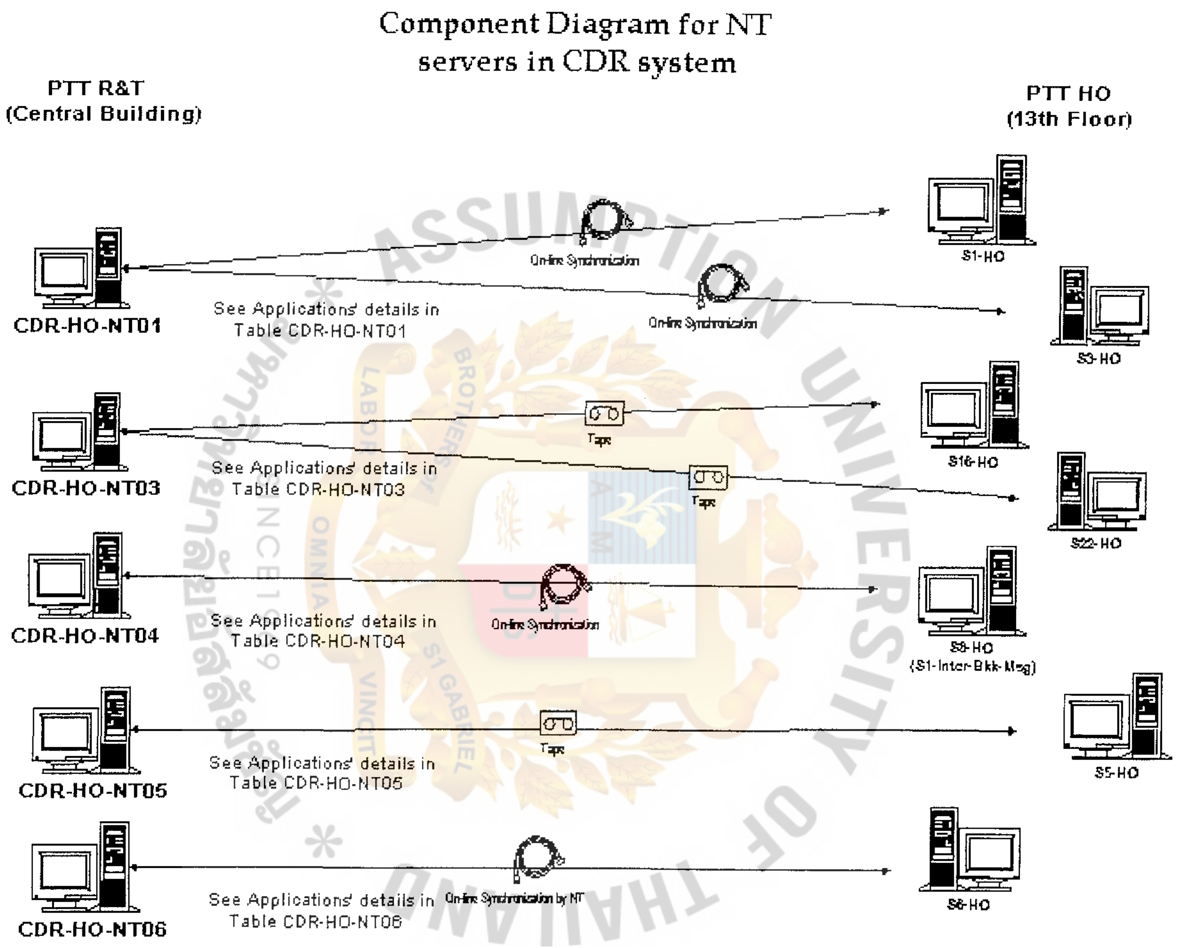

Component Diagram for NT servers in CDR system
PTT R&T (Central Building)
PTT HO (13th Floor)
CDR-HO-NT01
CDR-HO-NT03
CDR-HO-NT04
CDR-HO-NT05
CDR-HO-NT06
S1-HO
S3-HO
S16-HO
S22-HO
S8-HO (S1-Inter-Bkk-Mrg)
S5-HO
S6-HO
On-line Synchronization
On-line Synchronization
Tape
Tape
On-line Synchronization
Tape
On-line Synchronization by NT
See Applications' details in Table CDR-HO-NT01
See Applications' details in Table CDR-HO-NT03
See Applications' details in Table CDR-HO-NT04
See Applications' details in Table CDR-HO-NT05
See Applications' details in Table CDR-HO-NT06

Figure F.2. NT CDR System Diagram (1).

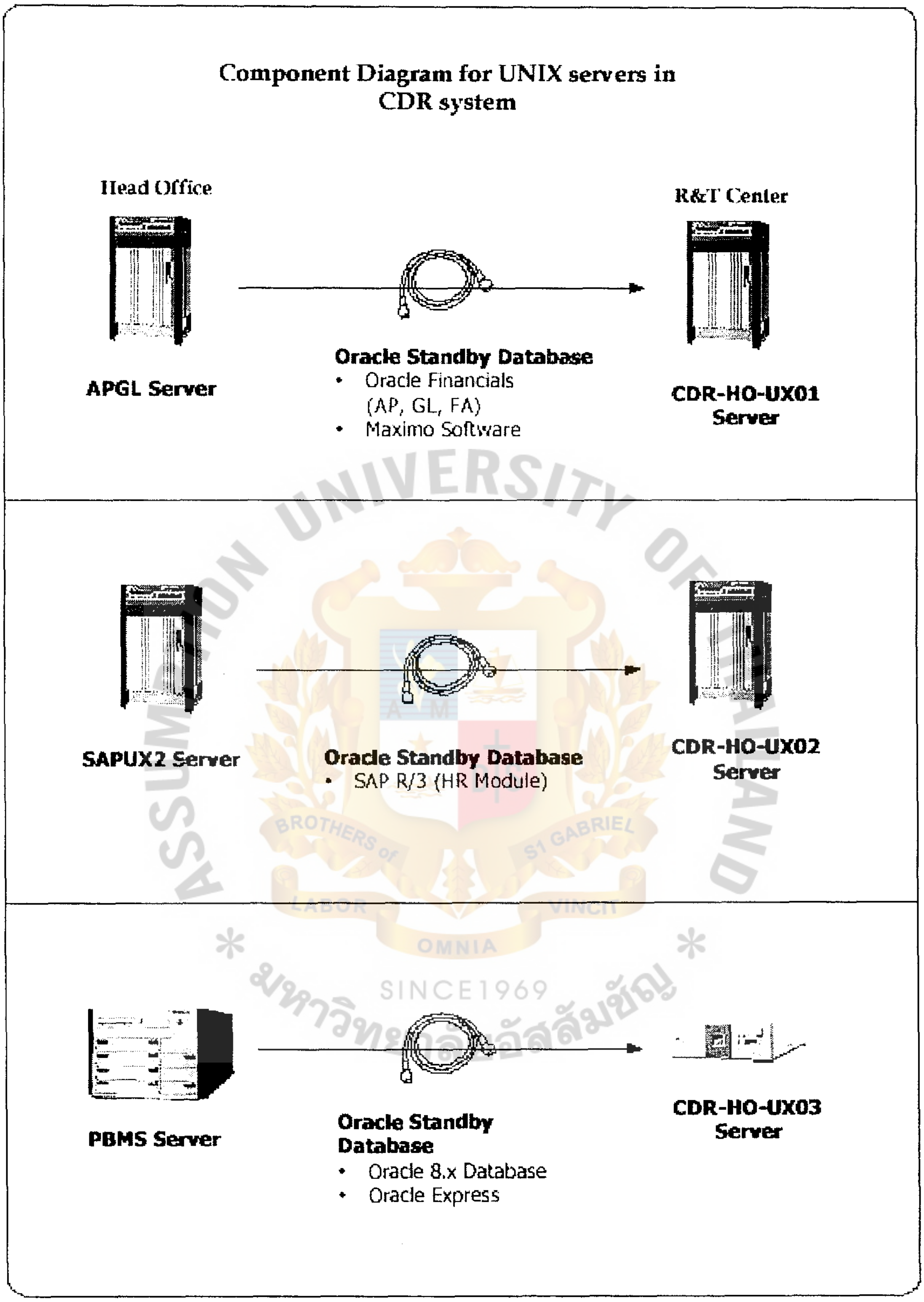


Figure F.4. UNIX CDR System Diagram.

APPENDIX G

LIST OF DELIVERABLE DOCUMENTS

Table G.1. List of Deliverable Documents.

No	Documents (Following TOR)	Deliveried Documents
1	Analysis of Current IT Infrastructure	Analysis of Current IT Infrastructure
2	CDR Design, Methodologies	Oracle Standby Database Design NT Detail Design Document
3	Operation Manuals	
	CDR Roles and Responsibility	UNIX Processes and Procedures
	CDR Hardware Installation	Hardware Installation & Configuration
	CDR Software Installation	Software Installation & Configuration
		NT Site Build Administration
		NT Replication Build
	CDR Workstation Configuration Procedure	Workstation Configuration
		UNIX Processes and Procedures
	CDR Fail-over Procedure	UNIX Processes and Procedures
		NT Fail-over & Fail-back Procedure
	CDR Data Backup/Update Routine	UNIX Processes and Procedures
		NT Backup Procedure
	CDR Data Recovery Routine	UNIX Processes and Procedures
		NT Restore Procedure
	CDR Change Management	Change Management
	CDR Test Routine	
	CDR Functionality Acceptance Test Procedure	

Table G.1. List of Deliverable Documents. (Continued)

No	Documents (Following TOR)	Deliveried Documents
4	Training Evaluation	Training Evaluation
5	CDR Business Opportunities	CDR Business Opportunities
Others	Skill Assessment Document	Skill Assessment Document
	Quality Assurance	Quality Assurance
	Hardware Test Report	Hardware Test Report
	Asset Control Document	Asset Control Document
	Asset Control Document	Asset Control Document
	APGL Deployment Plan	APGL Deployment Plan
	MAXIMO Deployment Plan	MAXIMO Deployment Plan
	Data Warehouse Deployment Plan	Data Warehouse Deployment Plan
	SAP/HR Deployment Plan	SAP/HR Deployment Plan

APPENDIX H

DETAIL DESIGN FOR EACH NT APPLICATIONS

(1) Loan Administration System

This application is a critical one. It uses oracle work group server version 7 as a database server and using Oracle developer/2000 as a client front-end. Database server was configured and running at S1-HO server. We installed and configured ARCserveIT Replication primary server in this server and replicate database to CDR site. The CDR server for this application is CDR-HO-NT01. For the Application programs, we backup them by using off-line Data synchronization method. The backup server for this application is S9-HO. S1-HO installed only ARCserveIT NT agent (in the case of backup) to collect data when the Backup server request for. S9-HO will collect all application files into tape at the end of the day and apply them to CDR backup site in case of disaster. Here are the detailed file locations.

Primary Server:	
Database Folder	\\S1-HO\E$\CRANT\DATABASE
Application Folder	\\S1-HO\E$\APPL\LOAN
CDR Server:	
Database Folder	\\CDR-HO-NT01\E$\ORANT\DATABASE
Application Folder	\\CDR-HO-NT01\E$\APPL\LOAN

Because its application files are not quite frequently change, we will inform the application developer about the change management procedure for the application program update at CDR site. This procedure will apply for all application systems. The detail procedure will declare later in the separate document. The idea of this procedure is that every time developers modify the application program, they have to copy the program files to the CDR site follow the application table.

(2) Financial Planning System

This application is EXCEL spread sheet. It has kept in the S4-HO server at primary site.

(3) Payroll System

This application is a critical application. It uses Pervasive SQL as a database server and using MAGIC as a client front-end. Database server was configured and running at S5-HO server. We installed and configured ARCserveIT Replication primary server in this server and replicate database to CDR site. The CDR server for this application is CDR-HO-NT05. For the Application programs, we backup them by using off-line Data synchronization method. The backup server for this application is S9-HO. S9-HO will collect all application files into tape at the end of the day and apply them to CDR backup site in case of disaster.

Here is the detail of location file.

Primary Server:

Database Folder	\\S5-HO\\\S1-HO\E$\ORANT\DATABASEE$\APPL\MAGIC\LEASING
Application Folder	\\S5-HO\\\S1-HO\E$\ORANT\DATABASEE$\APPL\MAGIC\ LEASING

CDR Server:

Database Folder	\\CDR-HO-NT05\E$\APPL\MAGIC\LEASING
Application Folder	\\CDR-HO-NT05\E$\APPL\MAGIC\LEASING

(4) Staff Recruitment System

This application is a non-critical application. It uses the MS-SQL Server as a database server and a Web application, which was developed using the cold fusion development tool. Users use a web browser to maintain this application. The web server, IIS in this case will handle the data from the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S9-HO is a web server. We install and setup ARCserveIT Replication primary server in S13-HO for on-line Data synchronization. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Detailed file locations are shown below:

Primary Server:

Database Folder	\\S13-HO\\\S1-HO\E$\ORANT\DATABASEF$\ RECRUIT
Application Folder	\\S9-HO\\\S1-HO\E$\ORANT\DATABASE\F$\ INETPUB\WWW\ ROOT\ INTRANET\RECRUIT

CDR Server:

Database Folder	\\CDR-HO-NT02\F$\RECRUIT
Application Folder	\\CDR-HO-NT02\F$\INETPUB\WWWROOT\ INTRANET\RECRUIT

In the case of SQL database, we can't put the database file that replicate from primary site to the same location directly because SQL server will start all database when we start the SQL instance, we can't stop other server 's replication tasks for any particular server. So we store replication data in other place and will copy those data to the database location when we want to use them. Here is the replication folder for this application.

Replication Folder \\CDR-HO-NT02\F$\S13\RECRUIT

When we want to use this application at the CDR site, we will stop the replication task from the replication console and copies replicated data to the database folder and will start the SQL Server instance. After that, we will be able to use web browser to access data from this application properly.

(5) Time Access Control Management System

This application is a non-critical application. It has no database on NT server. This application input by E-SOFTWIN tool at a client machine and sends data to store at S5-HO before sending to SAP/3 UNIX server. We will use the End Day period Data Synchronization for this application and send data to backup at CDR site after mid-night. The CDR server for this application is CDR-HO-NT02. Detailed file locations are shown below:

Primary Server:	
Application Folder	\\S5-HO\O$
CDR Server:	
Application Folder	\\CDR-HO-NT02\E$\TIME

(6) Personal Information System for Top Executives

This application is a non-critical application. It uses Oracle as a database server and using MAGIC as development software to create web base application and place on the S9-HO. Database server was configured and running at S1-HO server. We installed and configured ARCserveIT Replication primary server in this server and replicate database to CDR site. The CDR server for this application is CDR-HO-NT01. For the Application programs, we backup them by using End Day period Data synchronization method. The backup server for this application is S9-HO. It will collect all application files into tape at the end of the day and apply them to CDR backup site in case of disaster. Detailed file locations are shown below:

Primary Server:

Database Folder \\S1-HO\E$\ORANT\DATABASE

Application Folder \\S9-HO\F$\INETPUB\WWW\ROOT\INTRANET\PIS

CDR Server:

Database Folder \\CDR-HO-NT01\E$\ORANT\DATABASE

Application Folder \\CDR-HO-NT02\F$\INETPUB\WWW\ROOT\INTRANET\PIS

(7) Performance Based Management System

This application is a non-critical application. It uses Oracle for Unix as a database server and keeps application file and some data on S4-HO by using MS-Access. We will use End Day period Data synchronization method and copy data to tape every 20 minutes for backing up the application program in S4-HO and the database will take care by UNIX database administrator. Detailed file locations are shown below:

Primary Server:

Database Folder IW UNIX server

Application Folder \\S4-HO\E$\APPL\KPI

CDR Server:

Database Folder IW Backup Server

Application Folder \\CDR-HO-NT02\E$\APPL\KPI

(8) HO Mail System

This application is a critical application. It uses MS-Exchange as a mail server for Home Office. It runs on S3-HO-BKK-MSG and use S6-messaging server as a mail service account server. We installed and configured ARCserveIT Replication primary server S3-HO-BKK-MSG server and replicate database to CDR site. The CDR server for this application is CDR-HO-NT01. We install additional NT operating system at the CDR site as a backup mail service account. When the primary has been disaster, we still use Mail service at the CDR site. Detailed file locations are shown below:

Primary Server:

Exchange Folder	\\S3-HO-BKK-MSG\C$\EXCHSRVR;\\S3-HO-BKK-MSG\F$\EXCHSRVR

CDR Server:

Exchange Folder	\\CDR-HO-NT04\C$EXCHSRVR;\\CDR-HO-NT04\F$\EXCHSRVR

(9) Inter Mail System

This application is a critical application. It uses MS-Exchange as a mail server for Home Office. It runs on S1-INTER-BKK and use S6-messaging server as a mail service account server. We installed and configured ARCserveIT Replication primary server S1-INTER-BKK server and replicate database to CDR site. The CDR server for this application is CDR-HO-NT04. We install additional NT operating system at the CDR site as a backup mail service account. When the primary has been disaster, we still use Mail service at the CDR site. Detailed file locations are shown below:

Primary Server:

Exchange Folder \\S1-INTER-BKK\C$\EXCHSRVR;\\S1-INTER-BKK\D$\EXCHSRVR

CDR Server:

Exchange Folder \\CDR-HO-NT04\C$\EXCHSRVR;\\CDR-HO-NT04\D$\EXCHSRVR

(10) Job Description System

This application is a non-critical application. It uses Fox Pro develop application and create a database as well. The application programs locate on S4-HO and database file location on S18-HO. We will copy the database file to tape every 20 minutes by ARCserveIT server.

(11) News Clipping System

This application is a non-critical application. It uses SQL Server as a database server and using Web application, which develop by Win Forma tool. Users use web browser to maintain this application. The web server, IIS for in this case will handle the data from the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S9-HO is a web server. The application development server is S4-HO. We install and setup ARCserveIT Replication primary server in S13-HO for on-line Data synchronization. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Because PTT did not have the Win Forma License at CDR site, So we will only backup the source program into tape and copy runtime web page to CDR site and setup until it can connect to web server and use data in the MS-SQL database. Detailed file locations are shown below:

Primary Server:	
Database Folder	\\13-HO\E$\APPL\DATABASE\PTTNEWS
Application Folder	\\S4-HO\E$\APPL\WINFORMA
CDR Server:	
Database Folder	\\CDR-HO-NT02\E$\APPL\DATABASE\PTTNEWS
Application Folder	\\CDR-HO-NT02\E$\APPL\WINFORMA

In the case of SQL database, we can't put the database file that replicate from primary site to the same location directly because SQL server will start all database when we start the SQL instance, we can't stop other server's replication tasks for any particular server. So we store replication data in other place and will copy those data to the database location when we want to use them. Here is the replication folder for this application.

Replication Folder	\\CDR-HO-NT02\E$\S13-HO\APPL\DATABASE\PTTNEWS

When we want to use this application at the CDR site, we will stop the replication task from the replication console and copies replicated data to the database folder and will start the SQL Server instance. After that, we will be able to use web browser to access data from this application properly.

(12) Document Circulation System

This application is a non-critical application. It uses MS-Access and FoxPro as a database and application. We will backup data file every 20 minutes with tape at S9-HO backup server and send data to CDR site at night to the temporary folder. When we want to use this application, the fail-over procedure for this application has to initialize before use. The detailed locations are shown below:

Primary Server:

Database Folder	\\S4-HO\E$\APPL\SOFTWARE\DOC_CIR

CDR Server:

Database Folder	\\CDR-HO-NT02\ E$\APPL\SOFTWARE\DOC_CIR
Temporary Folder	\\CDR-HO-NT02\E$\OF_BACKUP\S4-HO

(13) PTT Board Meeting System

This application is a non-critical application. It uses SQL Server as a database server and using Web application, which develop by Cold Fusion tool. Users use web browser to maintain this application. The web server, IIS for in this case will handle the data from the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup in to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S2-HO is a web server. The application development server is S2-HO. We install and setup ARCserveIT Replication primary server in S13-HO for on-line Data synchronization. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. The detailed locations are shown below:

Primary Server:

Database Folder	\\S13-HO\F$\AGENDA
Application Folder	\\S2-HO\E$\InetPub\wwwroot\Intranet\Agenda

CDR Server:

Database Folder	\\CDR-HO-NT02\F$\AGENDA
Application Folder	\\CDR-HO-NT02\E$\InetPub\wwwroot\Intranet\Agenda

It is like other SQL database application. We have to put the data at the other folder until we want to use them.

Replication Folder	\\CDR-HO-NT02\F$\\S13-HO\AGENDA

(14) Library Web System

This application is a non-critical application. It uses MS-Access and FoxPro as a database and application. We will backup data file every 20 minutes with tape at S9-HO backup server and send data to CDR site at night to the temporary folder. When we want to use this application, the fail-over procedure for this application has to initialize before using it. The detailed locations are shown below:

Primary Server:

Database Folder	\\S4-HO\E\APPL
Application Folder	\\S4-HO\E\APPL

CDR Server:

Database Folder	\\CDR-HO-NT02\ E$\APPL
Application Folder	\\CDR-HO-NT02\ E$\APPL

Temporary Folder	\\CDR-HO-NT02\E$\OF_BACKUP\S4-HO

(15) PTT-Course Catalog Service System

This application is a non-critical application. It uses MS-Access as a database and some web application (Cold Fusion). We will backup data file every 20 minutes with tape at S9-HO backup server and send data to CDR site at night to the temporary folder. When we want to use this application, the fail-over procedure for this application has to initialize before using it. The detailed locations are shown below:

Primary Server:

Database Folder	\\S1-HO\c$\Netscape\server\docs\ptt\course
Application Folder	\\S1-HO\c$\Netscape\server\docs\ptt\course

CDR Server:

Database Folder	\\CDR-HO-NT02\C$\Netscape\server\docs\ptt\course
Application Folder	\\CDR-HO-NT02\C$\Netscape\server\docs\ptt\course
Temporary Folder	\\CDR-HO-NT02\E$\OF_BACKUP\S2-HO

(16) PTT-Postcard System

This application is a non-critical application. It uses SQL Server as a database server and using Web application, which is developede by Cold Fusion tool. Users use web browser to maintain this application. The web server, IIS for in this case will handle the data from the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that is explained before. The database engine for this application locates at S13-HO and S9-HO is a web server. We install and setup ARCserveIT Replication primary server in S13-HO for online Data synchronization. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. The detailed locations are shown below:

Primary Server:

Database Folder	\\S13-HO\F$\POSTCARD
Application Folder	\\S9-HO\F$\InetPub\wwwroot\Intranet\Ptt

CDR Server:

Database Folder	\\CDR-HO-NT02\F$\POSTCARD
Application Folder	\\CDR-HO-NT02\E$\InetPub\wwwroot\Intranet\Ptt

It is like other SQL database application. We have to put the data at the other folder until we want to use them.

Replication Folder	\\CDR-HO-NT02\F$\S13-HO\POSTCARD

(17) Conference Room Reservation System

This application is a non-critical application. It uses SQL Server as a database server and using Web application, which develop by Cold Fusion tool. Users use web browser to maintain this application. The web server, IIS for in this case will handle the data from the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S9-HO is a web server. We install and setup ARCserveIT Replication primary server in S13-HO for on-line Data synchronization. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. The detailed locations are shown below:

Primary Server:

Database Folder \\S13-HO\F$

Application Folder \\S9-HO\F$\InetPub\wwwroot\Intranet\asp

CDR Server:

Database Folder \\CDR-HO-NT02\F$

Application Folder \\CDR-HO-NT02\E$\InetPub\wwwroot\Intranet\asp

(18) Over Time (OT) Approval System

This application is a critical application. It uses SQL Server as a database server and using Web application, which develop by Cold Fusion tool. Users use web browser to maintain this application. The web server, IIS for in this case will handle the data from the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S20-HO and S2-HO is a web server. We install and setup ARCserveIT Replication primary server in S20-HO for on-line Data synchronization. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. The detailed locations are shown below:

Primary Server:

Database Folder \\S20-HO\E$\Database\overtime

Application Folder \\S2-HO\E$\InetPub\wwwroot\Intranet\Overtime

CDR Server:

Database Folder \\CDR-HO-NT02\E$\Database\overtime

Application Folder	\\CDR-HO-NT02\E$\InetPub\wwwroot\Intranet\Overtime

It is like other SQL database application. We have to put the data at the other folder until we want to use them.

Replication Folder	\\CDR-HO-NT02\E$\S20-HO\ADATABASE\OVERTIME

(19) Car Booking System

This application is a non-critical application. It uses SQL Server as a database server and using Web application, which develop by Cold Fusion tool. Users use web browser to maintain this application. The web server, IIS for in this case will handle the data from the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S9-HO is a web server. We install and setup ARCserveIT Replication primary server in S13-HO for on-line Data synchronization. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. The detailed of locations are shown below:

Primary Server:

Database Folder	\\S13-HO\F$\CARPTT
Application Folder	\\S9-HO\F$\InetPub\wwwroot\Intranet\Car_Ptt

CDR Server:

Database Folder	\\CDR-HO-NT02\F$\CARPTT

Application Folder \\CDR-HO-NT02\E$\InetPub\wwwroot\Intranet\Car_Ptt

It is like other SQL database applications. We have to put the data at the other folder until we want to use them.

Replication Folder \\CDR-HO-NT02\F$\S13-HO\CARPTT

(20) PTT-Telephone Directory System

This application is a non-critical application. It uses SQL Server as a database server and using Web application, which develop by Cold Fusion tool. Users use web browser to maintain this application. The web server, IIS for in this case will handle the data from the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S2-HO is a web server. We install and setup ARCserveIT Replication primary server in S13-HO for on-line Data synchronization. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Here is the detail of location file.

Primary Server:

Database Folder \\S13-HO\F$\DIRECTORY

Application Folder \\S2-HO\E$\InetPub\wwwroot\Intranet\Directory

CDR Server:

Database Folder \\CDR-HO-NT02\F$\DIRECTORY

Application Folder \\CDR-HO-T02\E$\InetPub\wwwroot\Intranet\Directory

It is like other SQL database applications. We have to put the data at the other folder until we want to use them.

Replication Folder \\CDR-HO-NT02\F$\S13-HO\DIRECTORY

(21) Display Phone Expense System

This application is a non-critical application. It uses SQL Server as a database server and using Web application, which develop by Cold Fusion tool. Users use web browser to maintain this application. The web server, IIS for in this case will handle the data from the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S9-HO is a web server. We install and setup ARCserveIT Replication primary server in S13-HO for on-line Data synchronization. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Here is the detail of location file.

Primary Server:

Database Folder \\S13-HO\F$\BILL

Application Folder \\S9-HO\F$\InetPub\wwwroot\Intranet\TB

CDR Server:

Database Folder \\S9-HO\F$\InetPub\wwwroot\Intranet\TB

Application Folder \\CDR-HO-NT02\E$\InetPub\wwwroot\Intranet\TB

It is like other SQL database applications. We have to put the data at the other folder until we want to use them.

Replication Folder \\CDR-HO-NT02\F$\S13-HO\BILL

(22) Help Desk System

This application is a non-critical application. It uses SQL Server as a database server and using Web application, which develop by Cold Fusion tool. Users use web browser to maintain this application. The web server, IIS for in this case will handle the data from the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S9-HO is a web server. We install and setup ARCserveIT Replication primary server in S13-HO for on-line Data synchronization. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Here is the detail of location file.

Primary Server:

Database Folder	\\S13-HO\F$\HELPDESK
Application Folder	\\S9-HO\F$\InetPub\wwwroot\Intranet\HELPDESK

CDR Server:

Database Folder	\\CDR-HO-NT02\F$\HELPDESK
Application Folder	\\CDR-HO-NT02\E$\InetPub\wwwroot\Intranet\HELPDESK

It is like other SQL database applications. We have to put the data at the other folder until we want to use them.

Replication Folder	\\CDR-HO-NT02\F$\S13-HO\HELPDESK

(23) IT Corporate Inventory System and IT Corporate PM System

These two applications use Oracle on UNIX as a database and MAXIMO as a application program which stored on S5-HO. In this document will mention only the application program maintenance. They will backup by the normal backup procedure at S9-HO backup server and the applied to CDR-HO-NT02 at CDR site.

(24) Document Management System

This application is a non-critical application. It uses SQL Server as a database server and using Web application, which develop by Win Forma tool. Users use web browser to maintain this application. The web server, IIS for in this case will handle the data from the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S4-HO is a web server. The application development server is S4-HO. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Here is the detail of location file.

Primary Server:	
Database Folder	\\13-HO\E$\WFM
Application Folder	\\S4-HO\E$\APPL\WINFORMA
CDR Server:	
Database Folder	\\CDR-HO-NT02\E$\WFM
Application Folder	\\CDR-HO-NT02\E$\APPL\WINFORMA

In the case of SQL database, we can't put the database file that replicate from primary site to the same location directly because SQL server will start all database when we start the SQL instance, we can't stop other server's replication tasks for any particular server. So we store replication data in other place and will copy those data to the database location when we want to use them. Here is the replication folder for this application.

Replication Folder \\CDR-HO-NT02\E$\S13-HO\WFM

When we want to use this application at the CDR site, we will stop the replication task from the replication console, copy the replicated data to the database folder, and start the SQL Server instance. After that, we will be able to use web browser to access data from this application properly.

(25) ASCOPE Filing System

This application is a non-critical application. It uses SQL Server as a database server and Win Forma as an application development tools. Users use web browser to maintain this application. The web server, IIS for in this case will maintain the data via ODBC though the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S4-HO is a web server. The application development server is S4-HO. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Here is the detail of location file.

Primary Server:

Database Folder	\\13-HO\E$\APPL\DATABASE
Application Folder	\\S4-HO\e:\appl\winforma\ascope

CDR Server:

Database Folder	\\CDR-HO-NT02\E$\APPL\DATABASE
Application Folder	\\CDR-HO-NT02\E$\appl\winforma\ascope

It is like other SQL database applications. We have to put the data at the other folder until we want to use them.

Replication Folder	\\CDR-HO-NT02\E$\S13-HO\APPL\DATABASE

(26) IT Corporate Filing System

This application is a critical application. It uses SQL Server as a database server and Win Forma as an application development tools. Users use web browser to maintain this application. The web server, IIS for in this case will maintain the data via ODBC though the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S4-HO is a web server. The application development server is S4-HO. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Here is the detail of location file.

Primary Server:

Database Folder	\\13-HO\E$\APPL\DATABASE
Application Folder	\\S4-HO\E$\appl\winforma

CDR Server:

Database Folder \\CDR-HO-NT02\E$\APPL\DATABASE

Application Folder \\CDR-HO-NT02\E$\appl\winforma

It is like other SQL database applications. We have to put the data at the other folder until we want to use them.

Replication Folder \\CDR-HO-NT02\E$\S13-HO\APPL\DATABASE

(27) Y2K Filing System

This application is a non-critical application. It uses SQL Server as a database server and Win Forma as an application development tools. Users use web browser to maintain this application. The web server, IIS for in this case will maintain the data via ODBC though the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S4-HO is a web server. The application development server is S4-HO. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Here is the detail of location file.

Primary Server:

Database Folder \\13-HO\E$\APPL\DATABASE

Application Folder \\S4-HO\e$\appl\winforma\y2k

CDR Server:

Database Folder \\CDR-HO-NT02\E$\APPL\DATABASE

Application Folder \\CDR-HO-NT02\E$\appl\winforma\y2k

It is like other SQL database applications. We have to put the data at the other folder until we want to use them.

Replication Folder \\CDR-HO-NT02\E$\S13-HO\APPL\DATABASE

(28) Empower Filing System

This application is a non-critical application. It uses SQL Server as a database server and Win Forma as an application development tools. Users use web browser to maintain this application. The web server, IIS for in this case will maintain the data via ODBC though the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S14-HO and S4-HO is a web server. The application development server is S4-HO. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Here is the detail of location file.

Primary Server:

Database Folder \\14-HO\E$\APPL\DATABASE\EMPOWER

Application Folder \\S4-HO\e$\appl\winforma\empower

CDR Server:

Database Folder \\CDR-HO-NT02\E$\APPL\DATABASE\EMPOWER

Application Folder \\CDR-HO-NT02\E$\appl\winforma\empower

It is like other SQL database applications. We have to put the data at the other folder until we want to use them.

Replication Folder	\\CDR-HO-NT02\E$\S14-HO\APPL\DATABASE\EMPOWER

(29) PTT Reforest System

This application is a non-critical one. It uses oracle work group server version 7 as a database server and using Oracle developer/2000 as a client front-end. Database server was configured and running at S1-HO server. We installed and configured ARCserveIT Replication primary server in this server and replicate database to CDR site. The CDR server for this application is CDR-HO-NT01. For the Application programs, we backup them by using off-line Data synchronization method. The backup server for this application is S9-HO. S1-HO installed only ARCserveIT NT agent (in the case of backup) for collect data when the Backup server request for. S9-HO will collect all application files into tape at the end of the day and apply them to CDR backup site in case of disaster. Here is the detail of location file.

Primary Server:	
Database Folder	\\S1-HO\E$\ORANT\DATABASE
Application Folder	\\S1-HO\E$\APPL
CDR Server:	
Database Folder	\\CDR-HO-NT01\E$\ORANT\DATABASE
Application Folder	\\CDR-HO-NT01\E$\APPL

(30) Library Management System

This application is a non-critical application. It uses MS-Access as a database and application. We will backup data file every 20 minutes with tape at S9 HO backup server and send data to CDR site at night to the temporary folder.

When we want to use this application, the fail-over procedure for this application has to initialize before using it. Here is the detail of location file.

Primary Server:

Database Folder	\\S1-HO\E$\PROLIB\DATABASE
Application Folder	\\S1-HO\E$\PROLIB\DATABASE

CDR Server:

Database Folder	\\CDR-HO-NT02\ E$\PROLIB\DATABASE
Application Folder	\\CDR-HO-NT02\ E$\PROLIB\DATABASE
Temporary Folder	\\CDR-HO-NT02\E$\OF_BACKUP\S1-HO

(31) PTT Clinical System

This application is a non-critical one. It uses oracle work group server version 7 as a database server and using Oracle developer/2000 as a client front-end. Database server was configured and running at S1-HO server. We installed and configured ARCserveIT Replication primary server in this server and replicate database to CDR site. The CDR server for this application is CDR-HO-NT01. The data synchronization method is the same as other ORACLE Workgroup Applications. Here is the detail of location file.

Primary Server:

Database Folder	\\S1-HO\E$\ORANT\DATABASE
Application Folder	\\S1-HO\E$\APPL

CDR Server:

Database Folder	\\CDR-HO-NT01\E$\ORANT\DATABASE
Application Folder	\\CDR-HO-NT01\E$\APPL

(32) Legal Document Filing System

This application is a non-critical application. It uses MS-Access as a database Delphi application tool. We will backup data file every 20 minutes with tape at S9-HO backup server and send data to CDR site at night to the temporary folder. When we want to use this application, the fail-over procedure for this application has to initialize before using it. Here is the detail of location file.

Primary Server:

Database Folder	\\S1-HO\E$\PVNET
Application Folder	\\S1-HO\E$\PVNET

CDR Server:

Database Folder	\\CDR-HO-NT02\E$\PvNet
Application Folder	\\CDR-HO-NT02\E$\PvNet
Temporary Folder	\\CDR-HO-NT02\E$\OF_BACKUP\S1-HO

(33) Leasing Car Profile System

This application is a non-critical application. It uses Pervasive SQL database from Magic as a database server and Magic tools as an application development tools. There are three version of magic software installed at home office. Magic version 5, 7, and 8. This application uses magic version 5. The database server and application server for this application locates on S5-HO. The CDR server for this application is CDR-HO-NT05. This server is delicate for the magic application. When the S5-HO has been down, we can promote this server to be S5-HO and use all magic application from this server until the exact S5-HO will come back. Here is the detail of location file.

Primary Server:

Database Folder	\\S5-HO\E$\APPL\MAGIC\LEASING
Application Folder	\\S5-HO\E$\APPL\MAGIC\LEASING

CDR Server:

Database Folder	\\CDR-HO-NT05\E$\APPL\MAGIC\LEASING
Application Folder	\\CDR-HO-NT05\E$\APPL\MAGIC\LEASING

(34) Document Warehouse System

This application is a non-critical application. It uses Pervasive SQL database from Magic as a database server and Magic tools as an application development tools. It used Magic version 7. Like other magic applications, it locates on S5-HO and its CDR server is CDR-HO-NT05. The fail-over and fail-backup procedure is the same one. We will explain those procedures later in this document. Here is the detail of location file.

Primary Server:

Database Folder	\\S5-HO\E$\APPL\MAGIC7\DOCUMENT
Application Folder	\\S5-HO\E$\APPL\MAGIC7\DOCUMENT

CDR Server:

Database Folder	\\CDR-HO-NT05\E$\APPL\MAGIC7\DOCUMENT
Application Folder	\\CDR-HO-NT05\E$\APPL\MAGIC7\DOCUMENT

(35) Car Profile System

This application is a non-critical application. It uses Pervasive SQL database from Magic as a database server and Magic tools as an application development tools. It used Magic version 7. Like other magic applications, it locates on S5-HO and its CDR server is CDR-HO-NT05. The fail-over and fail-backup procedure is the same one. We will explain those procedures later in this document. Here is the detail of location file.

Primary Server:	
Database Folder	\\S5-HO\E$\APPL\MAGIC7\CAR
Application Folder	\\S5-HO\E$\APPL\MAGIC7\CAR
CDR Server:	
Database Folder	\\CDR-HO-NT05\E$\APPL\MAGIC7\CAR
Application Folder	\\CDR-HO-NT05\E$\APPL\MAGIC7\CAR

(36) Internal Telephone Billing System

This application is a non-critical application. It uses SQL Server as a database server and cooperates with other application program that collect data from the PABX telephone system. The database engine for this application locates at S15-HO. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Here is the detail of location file.

Primary Server:	
Database Folder	\\S15-HO\C$\MSSQL\DATA
Application Folder	\\S15-HO\C$\BILLING
CDR Server:	
Database Folder	\\CDR-HO-NT02\E$\MSSQL\DATA
Application Folder	\\CDR-HO-NT02\E$\BILLIING

It is like other SQL database applications. We have to put the data at the other folder until we want to use them.

Replication Folder	\\CDR-HO-NT02\E$\S15-HO\MSSQL\DATA

(37) Telephone Billing System

This application is a non-critical application. It uses SQL Server as a database server and MS-Visual Basic as an application development tools. The database engine for this application locates at S13-HO and S9-HO is an application server. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Here is the detail of location file.

Primary Server:

Database Folder	\\13-HO\F$\Bill
Application Folder	\\S9-HO\F$\InetPub\wwwroot\Intranet\tb

CDR Server:

Database Folder	\\CDR-HO-NT02\F$\BILL
Application Folder	\\CDR-HO-NT02\F$\InetPub\wwwroot\Intranet\tb

It is like other SQL database applications. We have to put the data at the other folder until we want to use them.

Replication Folder	\\CDR-HO-NT02\F$\S13-HO\BILL

(38) Business Development System

This application is a non-critical application. It uses SQL Server as a database server and Win Forma as an application development tools. Users use web browser to maintain this application. The web server, IIS for in this case will maintain the data via ODBC though the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S14-HO and S4-HO is a web server.

The application development server is S4-HO. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server. Here is the detail of location file.

Primary Server:

Database Folder	\\14-HO\E$\APPL\DATABASE\BusDev
Application Folder	\\S4-HO\E$\appl\winforma\BusDev

CDR Server:

Database Folder	\\CDR-HO-NT02\E$\APPL\DATABASE\BusDev
Application Folder	\\CDR-HO-NT02\E$\appl\winforma\BusDev

It is like other SQL database applications. We have to put the data at the other folder until we want to use them.

Replication Folder	\\CDR-HO-NT02\E$\S14-HO\APPL\DATABASE\ BusDev

(39) Budget Data on the Web

This application is a non-critical application. It uses SQL Server as a database server and Win Forma as a application development tools. Users use web browser to maintain this application. The web server, IIS for in this case will maintain the data via ODBC though the SQL database. Therefore, we will treat web page as an application program that has to maintain and backup to save place and propagate to CDR site. We still apply the same change management procedure with other application program systems that explain before. The database engine for this application locates at S13-HO and S9-HO is a web server. The CDR server for this application is CDR-HO-NT02, both backup database and application program files will locate on this server.

Here is the detail of location file.

Primary Server:	
Database Folder	\\13-HO\F$\BUDGET
Application Folder	\\S9-HO\F$\InetPub\wwwroot\Intranet\budget
CDR Server:	
Database Folder	\\CDR-HO-NT02\F$\BUDGET
Application Folder	\\CDR-HO-NT02\F$\InetPub\wwwroot\Intranet\budget

It is like other SQL database applications. We have to put the data at the other folder until we want to use them.

Replication Folder	\\CDR-HO-NT02\E$\S13-HO\BUDGET

(40) ID Request

This application is implemented by MS-EXCEL spreadsheet placed on \\S4-ho\E$\Appl\ID Request.

We will check the modification of this file and copy it to S9-HO before backup to tape if it was modified in that period. Besides that, we will copy to CDR site after mid-night to reduce the human workload. The CDR server for this application is CDR-HO-NT02.

(41) Payroll Data Entry System

The Payroll Data entry System is being developed and will be deployed and in the future will not be addressed in this design document. However, PTT can implement the same CDR methodology on these applications if PTT desires this functionality after production deployment.

BIBLIOGRAPHY

1. "ARCserveIT Administration Guide." Computer Associates, 1999.

2. Baylus, Eileen P. Disaster Recovery Handbook. PA: McGraw Hill, 1991.

3. "Oracle Corporation. Oracle 8i Automated Standby Database." Oracle Corporation, 1999.

4. Sommerville, Ian. Software Engineering, 4th Edition. CA: Addison-Wesley, 1992.

5. Taigo, Jon W. Disaster Recovery Planning. London: Prentice-Hall, 1989.