Microsoft

Covers Service Pack 1

Foreword by David Espinoza

Foreword and Technical Review by Tony Redmond

Microsoft Exchange Server 2010

Siegfried Jagott and Joel Stidley with the Microsoft Exchange Server Team

Best Practices

Microsoft®

Microsoft® Exchange Server 2010

Best Practices

Siegfried Jagott and Joel Stidley
with the Microsoft Exchange Server Team

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2010929323

Printed and bound in the United States of America.

3 4 5 6 7 8 9 10 11 12 M 6 5 4 3 2 1

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to mspinput@microsoft.com.

Acquisitions Editor: Martin DelRe
Developmental Editor: Karen Szall
Project Editor: Carol Vu
Editorial Production: Christian Holdener, S4Carlisle Publishing Services
Technical Reviewers: Tony Redmond and Scott Schnoll; Technical Review services provided by Content Master, a member of CM Group, Ltd.
Cover: Tom Draper Design

Body Part No. X17-00144

I dedicate this book to my mum, Johanna, for all the support and love she gave to me throughout my whole life. Without her effort I would not be where I am today.

—Siegfried Jagott

To my wife, Andrea. Without her patience, love, and support I would not be able to take on new and exciting challenges.

—Joel Stidley

Contents at a Glance

Contents

PART III UPGRADING TO EXCHANGE SERVER 2010

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

About the Sidebars

This book includes sidebars that provide you with real-world experience and insights from Microsoft Exchange product group members as well as well known Exchange subject matter experts. Each sidebar covers a specific topic of expertise and reflects the opinion of the sidebar contributor, not necessarily the opinion of Microsoft or the authors of this book.

Sidebars in this book are categorized into the following distinguishing sidebar elements:

- **Notes from the Field** Insights and experiences from Microsoft consultants, technical support professionals, partners, and early adopter customers.
- Inside Track Insider information or tips from Microsoft program managers, technical product managers, developers, and testers.
- Lessons Learned Examples of things that did not go well or what not to do. Learn from others so that you don't repeat their mistakes.
- Trade-Offs Best practices are rarely absolute. We point out key decisions that you should be weighing.

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9

Chapter 10

Chapter 11

Chapter 12

Chapter 13

Chapter 14

Chapter 15

Chapter 16

Chapter 17

Foreword

Every day we rely more and more on electronic mail to handle our most basic communication needs. Our reliance leads us to require dependability. To ensure an efficient transition from an older system to Exchange 2010, you must determine how to integrate a myriad of systems. Your users will demand compatibility and high levels of uptime, and managers will demand lower costs in terms of servers and storage. I have spent 15 years at Microsoft working with teams to enhance the end-user experience. I've never been as excited about the work we've done as I am now with the release of Exchange 2010. With Exchange 2010, our development team was dedicated to building a brand-new release that effectively took a deliberate approach to building new features, refining existing features, and making sure at every step that we stayed true to our goals of delivering an awesome release of Exchange. The breadth and depth of the technologies Microsoft Exchange 2010 finally delivers is astounding. Exchange 2010 provides new features such as Exchange Control Panel (ECP), Directory on the Middle Tier (DoMT), High Availability (HA), and Role-Based Access Control (RBAC). Federated sharing, archiving, and lower storage cost options are knocking down barriers that have traditionally stopped customers from deploying or meeting user needs. Any one of the features I just mentioned would be interesting on its own, but the combination is truly compelling.

Exchange is easy to install, but to get the most out of it you need to explore the many features and capabilities that more than 20 million lines of code bring to it. You want to understand the software in detail, and the authors of this book have the experience to show you all of the features and components. The authors have done an awesome job getting the details right and have taken great care in bringing you what I think is the best book on the subject. Recently there has been talk about books like this being out of date as soon as they go to press, or that getting information from the Internet is the new way to learn. To this I say, "Nonsense!" With this book, you will gain from the authors' vast experience with a topic that is vast in scope. How did the authors get such in-depth, detailed experience with a product released in November of 2009? That level of detail—including best practices for deployment—requires time and teamwork, and that is where the Technology Adoption Program (TAP) comes into play.

Microsoft's Technology Adoption Program is designed to validate new versions of Exchange by having customers test and run production deployments of pre-release builds of the next version of Exchange. This gives participants the opportunity to provide real-time design feedback to the Exchange product

development team. Microsoft deployed the first production Exchange 2010 server on April 16, 2007, and in January of 2008 released bits to TAP customers and partners for review. Shortly thereafter, the authors and other customers were running Exchange 2010 in their production deployments. When Microsoft officially shipped Exchange 2010 on November 9, 2009, TAP partners had already deployed more than 200,000 mailboxes into production! Through this preliminary process, the authors participated in every step of the final design, gaining valuable experience with each TAP release for deployment. During this TAP deployment phase, all TAPs work together with Microsoft to find the best product and best ways to deploy. Here is what one TAP had to say about this process:

> *"We have learned a lot through this process, and not only about Exchange 2010. By interacting with other TAP members and the product group on a daily basis we have been able to remove the blinders we sometimes wear from administering the same system day in and day out. This has allowed us to consider alternate approaches we could take to improve our system overall and to identify where some of our own shortcomings are. I've seen things posted I've never even thought of before and hope that our contributions have done the same . . ."*

Individually and collectively the authors who wrote this book have been working with Exchange 2010 for as long as many senior developers at Microsoft. They have done an awesome job of providing readers with the ins and outs of the full range of features of Exchange 2010, which will help you get the most out of the product. Exchange administrators will find the experienced, hands-on approach of this book invaluable in designing and deploying Exchange 2010. You wouldn't want a book that only skimmed and introduced new features. Fortunately for you, this book is based on the experience of years of successful deployments in complex environments and a teamwork approach to the final design process. Microsoft and TAPs have built a product that we are truly proud of, and this book brings you the right way to walk through it. This book definitely belongs on the shelf of every serious Exchange administrator and IT manager.

David Espinoza

Senior Program Manager, Exchange Ship Team

Microsoft Corporation

May 2010

I love the idea of a best practice book. The initial challenge is to capture the knowledge of real-life designs and deployments that underpin best practice. The next challenge is to validate that the claimed best practice is actually valuable. The final challenge is to focus on a best practice that has enduring value rather than the tenets that flame into existence sparked by a notion of someone at a conference or other event and expire just as quickly when everyone realizes that the proposition being advanced isn't such a good idea after all. Active Directory designs for Exchange are an example of best practice that has changed since 1999. The initial designs for large corporations all seemed to favor the "minimal root domain and geographic sub-domains" design at a time when we assumed that a domain was a security boundary and that it was good to segment administration across sub-domains. Of course, at that time we were influenced by PC LAN networks and couldn't quite comprehend how Active Directory would evolve to accommodate the range of design options that are available and in use today. Of course, saying what best practice is for Active Directory is another question. The answer is that there is no best practice, but there are solid guiding principles that any designer needs to understand and respect before deployment.

I think the same is true for Exchange Server. Best practice is transient and changes from version to version. It also changes over the lifetime of a version as the Exchange community comes to grips with the product and understands the strengths and weaknesses of the software. Microsoft also contributes to the evolution of best practice by publishing a wealth of information through Microsoft TechNet and other sites, including the Exchange development group's blog. Microsoft also changes best practice as they issue roll-up updates and service packs to address product flaws and sometimes even introduce new functionality (and maybe reinforce the old adage that no one should ever deploy a Microsoft server application until the first service pack is available).

Even though I regard best practice as transient, I still think that it is possible to set out solid guiding principles that help system designers and administrators to figure out how to make Exchange work for their organization. Well-organized books like this render a great service to the Exchange community by laying out Exchange 2010 in a practical manner that's based on insight and experience. I guess this could be called best practice, and that's certainly what the title says, but I prefer to think of the knowledge contained here as the guiding principles that every administrator should be acquainted with before deploying Exchange. You won't find a magic bullet here, nor will you find a recipe that you can simply adopt for a deployment. Instead, the chapters unfold to deliver a comprehensive

guide to Exchange 2010 in an informative and easy-to-follow manner. Even better, because this book was written well after Exchange 2010 was released, it doesn't suffer from the "must be first to market" syndrome that afflicts so many technical books and leads to guesses and inaccuracies because the book's content is based on beta code. And as we all know, beta code isn't necessarily what is delivered to customers.

I've enjoyed reading this book and I think it will be valuable to anyone who wants to get to know Exchange 2010. Use it to establish your own foundation but don't forget that best practice evolves over time so be prepared to evolve your own knowledge by keeping up to date with developments.

Tony Redmond

Exchange MVP

May 2010

Acknowledgments

We wanted this book to be something special, something that reflects our passion and dedication to Microsoft Exchange. Our goal was to write a book for Exchange geeks by Exchange geeks. We also didn't want to write something that fell short of our expectations. To accomplish this lofty goal we required input, assistance, and support from a long list of people. This may sound like an award acceptance speech, but it is true. Although only two authors are named on the cover of this book, without a dedicated group of contributors, reviewers, and supporters this book would not exist.

First, we want to thank Stanley Reimer for believing in the project and helping get us the project approved and started. We regret not being able to work with you on this book and we hope to be able to work with you again soon. We also would like to thank Andy Schan and Jeffrey Rosen for being able to fill the void that Stanley left on our project. Without their assistance the project would have never been completed.

Many other people assisted during this project, but a few people in particular from the Exchange product group stand out for their support, patience, and insight—especially as changes were made to the product: Kristian Andaker, Ed Banti, Matthias Leibmann, Alexander Nikolayev, Greg Taylor, Paul Wimmer, Gary Cooper, and Brian Desmond.

In addition to these people, we also want to thank the following teams and companies for their dedicated support and input: everyone on the Microsoft Exchange 2010 TAP List, Siemens Workplace Architecture Team, the Exchange administrators at Axel Springer Media AG, and the supportive people at the Microsoft Enterprise Engineering Center in Redmond.

The three most critical pieces of a successful technical book are its technical accuracy, its grammatical accuracy, and the support of its editing staff. For technical accuracy, we were fortunate to have had two of the most thorough and knowledgeable people in the Exchange server ecosystem to provide technical guidance for the book: Tony Redmond and Scott Schnoll. They provided candid reviews that helped improve the content both technically and logistically. This is a better book thanks to each of them. We also want to thank David Espinoza and Tony Redmond for their kind words and the keen insight they provided in the Foreword for this book.

Although it may be shocking to hear, we as authors do not have perfect grammar, and one of our pet peeves is reading a book with blatant grammatical errors. Thankfully, we had Becka McKay to help ensure that the book's grammatical excellence met the highest standards. She was able to mold our sometimes narrowly focused word choices and improved not only the way the book sounds but also its accuracy and clarity.

The support we received from the editorial staff at Microsoft Press has been unmatched by any of our previous experiences. This book started with Martin DelRe, the acquisitions editor, bootstrapping the project about a year and a half prior to its publication. This happened during the final throes of the Exchange 2010 development process, yet he was still able to wrangle some key players in the Exchange product group to help out. This is a testament both to Martin's ability to get things done as well as to the product group's willingness to assist on this project. Shortly after we got started, Karen Szall, the book's developmental editor, was brought on board. She was critical in helping shape the look and feel of the book, and she also answered our unending barrage of questions and encouraged us to start writing. After Karen provided the momentum, we had the privilege of working with Carol Vu, the book's project editor. Carol was able to keep track of multiple versions of each chapter, deadlines whooshing by, and a variety of other forms of drama all without breaking a sweat. A lesser project editor would have had a panic attack long ago. We'd also like to thank Christian Holdener for managing this seemingly unending project and Maureen Johnson for being able to sift through the pages and pages of technojargon to make an index that is actually useful to our readers.

We want to extend special thanks to the Exchange product group members and Exchange experts who spend long hours of their free time reading our draft chapters to make sure we produced the highest-quality content possible. We gratefully salute the following people who were part of the review process: Alessandro Goncalves, Alexander Nikolayev, Andrew Sullivan, Ankur Kothari, Arno Zwegers, Charlie Chung, Christian Schindler, Colin Lee, Dave Chomas, David Espinoza, Ed Banti, Erik Szewczyk, Evan Dodds, Gary Cooper, Greg Taylor, Henrik Walther, Ilse Van Criekinge, Joe Cirillo, John Glynn, Kamal Janardhan, Korneel Bullens, Kristian Andåker, Kumar Venkateswar, Matthias Leibmann, Nagesh Mahadev, Paul Wimmer, Ross Smith IV, Steve McIntyre, Thierry Demorre, Tim McMichael, Todd Hawkins, Todd Luttinen, and Yesim Koman.

Finally, we would like to thank all of the sidebar contributors; these people really helped add a more comprehensive view of the subject and added depth to many topics. We're proud of the number of practical sidebars in the book, and our thanks go to their creators: Alessandro Goncalves, Alexander Nikolayev, Andreas Bode, Andreas Essing, Andrew Ehrensing, Ankur Kothari, Arno Zwegers, Brian Day,

Brian Desmond, Carsten Allendoerfer, Charlie Chung, Christian Schindler, Colin Lee, Devin L. Ganger, Ed Banti, Ed Wilson, Erick Szewczyk, Gary A. Cooper, Greg Taylor, Henrik Walther, Jeff Mealiffe, Joe Cirillo, John P. Glynn, Jon Webster, Korneel Bullens, Kristian Andaker, Lars Riehn, Manfred Kornagel, Markus Bellmann, Matthias Leibmann, Nicolai Wagner, Paul Wimmer, Robin Thomas, Ross Smith IV, Sascha Schmatz, Steve McIntyre, Thierry Demorre, Todd Hawkins, Todd Luttinen, Tony Redmond, and Ulf Hansen.

We thank you for taking the time to read our book; we hope that everyone's effort comes across and that you find the book both interesting and beneficial.

Introduction

Welcome to *Microsoft Exchange Server 2010 Best Practices*, a book that was developed together with the Microsoft Exchange product group to provide in-depth information about Exchange and best practices based on real-life experiences with the product in use in different environments. Numerous sidebars are also included that detail experiences from skilled industry professionals such as Certified Exchange Masters and Exchange Most Valuable Professionals (MVPs).

> ***NOTE*** ***The book is largely based on the original version of Exchange Server 2010 released in October 2009 together with information about the changes that you can expect in Service Pack 1. Because Service Pack 1 was not yet released when the book was finished, we based our experience in the book on information available from the Microsoft Exchange product group and on a pre-release build of Service Pack 1. To make sure we only cover features that will be in the release of Service Pack 1, we addressed only the most notable changes.***

In November of 2008 Joel was updating an Exchange 2007 book when the two of us began chatting about writing a book on Exchange 2010. Having worked on several books already, we did not want to write the usual "click-here-and-do-this" type of Exchange book. We wanted to do something special, something that reflected our passion for and dedication to Exchange. The idea of working together along with the Microsoft Exchange 2010 product group to produce a book that could document years of experience from so many knowledgeable people thrilled all of us.

From beginning to end, this book took about 17 months to complete, and took a great deal of effort by a lot of hard-working and intelligent people. We hope that this effort comes across to you and that you find this book a worthwhile part of your Exchange library.

Who Is This Book For?

Microsoft Exchange Server 2010 Best Practices is for experienced Messaging architects, Exchange administrators, support professionals, and engineers, especially those who are working in medium to large enterprise organizations and also have at least one year of experience in administering, deploying, managing, monitoring, upgrading, migrating, and designing Exchange Server.

IT professionals who work in smaller companies also will benefit from the recommendations and sidebars presented in this book as well as many of the tips and tricks.

To get the most benefit from this book, prior to reading it you should at least be able to do the following:

- Design and deploy an Exchange messaging enterprise according to business requirements.
- Understand Active Directory concepts, especially how sites and services provide its essential structure.
- Understand the Windows permission model.
- Have good experience with the networking protocol TCP/IP v4 and the messaging protocol SMTP.
- Understand Windows PKI infrastructures and digital certificates.

You should also understand the basics of Exchange Server 2010, including the differences between each of the Exchange server roles (experience gained with Exchange 2007 is valuable here), and you should have experience with using the Exchange Management Console (EMC) and the Exchange Management Shell (EMS). The book does not focus on the "how to" and thus does not include step-by-step guides for each and every setting. This book builds on the knowledge and experience needed to successfully pass the Microsoft 70-663 exam, Pro: Designing and Deploying Messaging Solutions with Microsoft Exchange Server 2010.

The target audience for *Microsoft Exchange Server 2010 Best Practices* is interested in insights and in looking beyond the common administrative tasks performed in Exchange 2010 as well as those who want to unveil the full functionality of the product.

This book is a 300-level technical book; however, the planning and managing chapter will also be very useful to IT managers seeking guidance on understanding technical concepts for managing Exchange projects.

How Is This Book Organized?

This book is organized into four parts:

- Part I: Preparing for Exchange Server 2010
- Part II: Designing Exchange Server 2010
- Part III: Upgrading to Exchange 2010
- Part IV: Deploying and Managing Exchange Server 2010

The first part of this book consists of three chapters that focus on preparing your organization for Exchange Server 2010. Chapter 1, "Introducing Exchange Server 2010," provides an introduction to Exchange Server 2010, including high-level information about Exchange and Windows PowerShell. Chapter 2, "Exchange Deployment Projects," provides a project-oriented approach to Exchange Server implementation as well as information about the imaginary company scenarios that are used throughout the book. Chapter 3, "Exchange Environmental Considerations," then provides information about other areas, such as Active Directory, that you need to consider to have a successful Exchange implementation.

The second part of this book considers areas that are required for designing an Exchange Server 2010 implementation. In Chapter 4, "Client Access in Exchange 2010," you learn about the Client Access Server role of Exchange 2010. Chapter 5, "Routing and Transport," explains how message routing works and how you plan for the Hub Transport server role. Chapter 6, "Mailbox Services," considers the Mailbox server role and explains the database changes introduced in Exchange 2010. Chapter 7, "Edge Transport and Messaging Security," considers the details of the Edge Transport server role and, in addition to discussing messaging security, also covers antivirus and anti-spam functionality. Chapter 8, "Automated Message Processing, Compliance, and Archiving," covers the Exchange compliance and archiving features and also explains how you can perform automated message processing. Chapter 9, "Unified Messaging," explains Exchange Unified Messaging or how to access your mailbox using voice as well as OCS 2007 R2 interoperability with Exchange. Chapter 10, "Federated Sharing," describes how to connect two Exchange Organizations using Federated Sharing. Chapter 11, "Designing High Availability," introduces you to the concept of Database Availability Groups (DAGs) and how DAGs can be implemented to provide high availability for your messaging service as well discussing other availability aspects such as network load balancing. Chapter 12, "Backup, Restore, and Disaster Recovery," takes you through backing up and restoring your Exchange servers, databases, and features to mitigate the need for restores. Chapter 13, "Hardware Planning for Exchange Server 2010," concludes the design part of this book by providing guidance about hardware planning for your Exchange servers.

The third part of this book consists of Chapter 14, "Transitioning from Exchange 2003 and Exchange 2007," which considers how you can approach the upgrade of your existing Exchange 2003 or Exchange 2007 installation to Exchange Server 2010 and what important factors you need to consider beforehand.

The fourth part of this book considers deploying and managing Exchange Server 2010. Chapter 15, "Preparing for and Deploying Exchange Server 2010,"

describes how to prepare Active Directory and the servers for Exchange 2010, how you check your environment to make sure all Exchange requirements are covered, and how you install Exchange 2010 both manually and automatically. Chapter 16, "Managing Exchange," discusses how to manage Exchange Server 2010. Finally, Chapter 17, "Operating and Troubleshooting Exchange Server 2010," provides information about operating and troubleshooting your Exchange 2010 server environment.

How to Read This Book

This book is written as a reference, and each chapter was written to stand on its own, so you do not need to read the chapters in order—you can jump between the chapters that interest you. However, we'd like to point out some chapters that provide an excellent start and are used for other areas in the book as well.

Almost every chapter in the book uses sample scenarios that are introduced in detail in Chapter 2. These fictional scenarios are used as real-world examples and to provide illustrations of how the ideas presented in a chapter could be implemented in practice. Chapter 3 provides the basis for reading about Exchange environmental areas such as networks, operating systems, and certificates. We strongly recommend reading these chapters—they also provide an excellent overview and best practices around the topic you might want to investigate.

What This Book Is Not

In *Microsoft Exchange Server 2010 Best Practices*, we assume that you have a good understanding of Exchange Server 2010 and Windows PowerShell 2.0. For this reason, this book does not teach the basics of every feature nor does it include a how-to section for common administrative tasks.

This book is also not a preparation guide for Exam 70-662: TS: Microsoft Exchange Server 2010, Configuring, or Exam 70-663: Pro: Designing and Deploying Messaging Solutions with Microsoft Exchange Server 2010, even though when you apply the knowledge and experience covered in this book, it will help you to pass these exams.

In general, the book does not include detailed steps for every configuration setting but tries to provide a foundation so that you can make your own decisions for what would be optimal in your environment. It does not dictate one specific way to configure Exchange 2010; instead, it provides the options available and the factors that should influence your decisions. Thus this book is not a guide for how to configure your Exchange servers; it is meant to improve your already configured environment or help you add new features such as Unified Messaging.

System Requirements

This book is designed to be used with the following Exchange 2010 software requirements:

- Windows Server 2008 or Windows Server 2008 R2
- 1 GB of RAM
- x64 architecture-based computer with Intel or AMD processor that supports 64 bit
- 1.2 GB of available disk space
- Display monitor capable of 800 × 600 resolution

The following list details the minimum system requirements needed to run the content in the book's companion Web site:

- Windows XP with the latest service pack installed and the latest updates from Microsoft Update Service
- Display monitor capable of 1024 × 768 resolution
- CD-ROM drive
- Microsoft Mouse or compatible pointing device

The Companion Web Site

This book features a companion Web site that makes available additional information to you such as job aids, quick reference guides, and additional Exchange 2010 resources. We have included these elements to help you plan and manage your Exchange 2010 organization and apply the book's recommended best practices. The companion Web site includes the following:

- **Job Aids** Additional documents on most of the chapters that help you to collect and structure your work through the book.
- Quick Reference Guides Such as the Exchange 2010 Best Practices Quick Reference Guide, which is an overview of all best practice recommendations in the book, and the Exchange 2010 Additional Reference Guide, a collection of all Internet links referenced in the book.
- TechNet Exchange 2010 Resources Additional links that might be useful when reading the book.

You can download these files from the companion Web site, located at *http://wwww.microsoftpressstore.com/title/parentISBN13*.

Full documentation of the contents and structure of the companion Web site can be found in the Readme.txt file in the download.

Support for This Book

Every effort has been made to ensure the accuracy of this book. As corrections or changes are collected, they will be added to a Microsoft Knowledge Base article accessible via the Microsoft Help and Support site. Microsoft Press provides support for books, including instructions for finding Knowledge Base articles, at the following Web site: *http://www.microsoft.com/learning/support/books/*.

If you have questions regarding the book that are not answered by visiting the site above or viewing a Knowledge Base article, send them to Microsoft Press via e-mail to *mspinput@microsoft.com*. Please note that Microsoft software product support is not offered through these addresses.

We Want to Hear from You

We welcome your feedback about this book. Please share your comments and ideas via the following short survey: *http://www.microsoft.com/learning/booksurvey.* Your participation will help Microsoft Press create books that better meet your needs and your standards.

> ***NOTE*** ***We hope that you will give us detailed feedback via our survey. If you have questions about our publishing program, upcoming titles, or Microsoft Press in general, we encourage you to interact with us via Twitter at http://twitter.com/MicrosoftPress. For support issues, use only the e-mail address shown above.***

CHAPTER 3

Exchange Environmental Considerations

This chapter describes all the basic components surrounding Exchange Server 2010 that need to be considered to plan a solid Exchange implementation. These components provide the basis to build Exchange on a solid foundation and to identify potential issues.

It provides a basis for other chapters in this book by describing some of the technologies that will be discussed later. For example, this chapter includes a discussion on namespace design as well as a review of certificate requirements, which are then taken to the next level in Chapter 4, "Client Access in Exchange 2010." Of particular importance when using this book is the "Planning Naming Conventions" section, which explains the names that are used throughout the entire book.

Evaluating Network Topology

Evaluating the network topology through which Exchange Server 2010 will communicate is crucial during the Delivery Phase, Step 2: Assess, as described in Chapter 2, "Exchange Deployment Projects." Often, making changes in the network infrastructure can take a considerable amount of time because the Exchange team isn't necessarily responsible for making changes to the network, and communication and negotiation are often required before network changes can be made, especially in large organizations that support heterogeneous operating systems.

Identifying any required changes and making sure that the execution of the change can occur without any difficulties early in the design process can save time later when you are implementing Exchange Server 2010.

This section provides an overview of the network-related requirements for Exchange 2010.

Reviewing Current and Planned Network Topology

The first step is to collect all information about your internal network, the perimeter network, and its external collections as thoroughly as possible from a variety of sources. These sources include the following:

- **Physical network topology** Verify that TCP/IP is used everywhere, which Internet Protocol is used (IPv4 and/or IPv6), how IP addresses are allocated for servers, and that IP subnets are used according to location.
- **Internal physical network connections or links** This includes LAN and WAN links, router, and so on.
- **External physical network connections** This includes the Internet, partner companies, and so on.
- **Interconnection of physical network connections** This includes hub-and-spoke, ring or star, and point-to-point.
- **Physical network speed** Divide between guaranteed bandwidth, available bandwidth, and latency for each identified network link.
- **Network protection that might interfere** This includes firewalls that protect physical links or network link encryption devices that reduce the link speed.
- **Firewall port availability to both external and internal systems.**
- **Server name resolution used in locations or between locations (DNS/WINS name resolution).**
- **Defined namespaces in DNS** This is described in the "Planning Namespace" section later in this chapter.
- **Perimeter network servers** Including any servers that are located in a perimeter network, especially any server that provides SMTP-relay functionality.

Be sure to identify any known changes that will occur to the network configuration during the interim between the planning phase and the deployment phase so that the impact of the change can be assessed just prior to deployment and the proper adjustments made.

> ***NOTE*** *In large organizations, gathering this information might be quite a time-consuming effort—you may have to meet with many disparate network teams to get a thorough understanding of the network specific details. If you want to evaluate a global network infrastructure that includes many sites or locations, make sure you understand the company structure, the businesses that Exchange will serve, and how these businesses are supplied with IT currently. Having these discussions will provide you with much insight into the current network topology and help identify any problems and potential issues that you should consider when planning the messaging design.*

Domain Name System (DNS)

This section is about the technical foundation on domain name system (DNS). It does not include any discussion about namespace planning. The aspects of namespace planning and disjoint namespace or single label domains are described in the "Planning Namespace" section later in this chapter.

DNS and Active Directory

Microsoft Windows uses the DNS standard as the primary name registration and resolution service for Active Directory. For that reason it is a basic requirement that all clients and servers must be able to reliably resolve DNS queries for a given resource in the appropriate namespace.

DNS provides a hierarchically distributed and scalable database where hosts can automatically update their records. These dynamic records can be fully integrated into Active Directory when using Active Directory–integrated DNS zones.

> ***NOTE*** *In Exchange Server 2003 or earlier, the Windows Internet Name Service (WINS) was required to support multi-domain environments. This is no longer required for Exchange Server 2010.*

The following list provides best practices for DNS settings when implementing Exchange Server 2010 in your Active Directory:

- Use the DNS Server service that is part of Windows Server. This provides you with features such as Dynamic Update and Active Directory–integrated DNS zones. For example, domain controllers register their network service types in DNS so that other computers in the forest can access them.

- If you cannot use the Windows DNS Server for Active Directory and Exchange, make sure the DNS server supports SRV resource records and allows dynamic updates of Locator DNS resource records (SRV and A records). If your company uses BIND, make sure you use BIND 8.x or later.
- Store all DNS zones as Active Directory–integrated in Active Directory to gain the benefit of having DNS and Active Directory replicated by a single mechanism. This prevents the need to use different tools for troubleshooting.
- Configure Dynamic Updates as Secure, thus only allowing authorized clients to register their host name and IP address.
- Only configure Forward Lookups Zones, which are required by Exchange 2010. You do not need to configure Reverse Lookup Zones because they are not used by Windows 2008 or Exchange 2010.

More information can be found in the whitepaper "DNS Requirements for Installing Active Directory" at *http://technet.microsoft.com/en-us/library/cc739159(WS.10).aspx*.

NOTES FROM THE FIELD

DNS Dynamic Updates

John P Glynn
Principal Consultant, Microsoft Consulting Services, US/Central Region

Active Directory is a key dependency for Exchange; without it Exchange does not and will not properly function. Active Directory is based on the DNS service. Without DNS, many components of Active Directory, Exchange, and client interaction fail to function properly. When a domain controller is installed on a domain, a series of records is created. These DNS records contain service location records for Kerberos, LDAP, GC, site-specific information, and a domain record that is a unique GUID.

Exchange servers utilize these DNS records to locate authentication or other specific services. Exchange will use Active Directory site-specific service location records for services such as: locating the closest Global Catalog servers to utilize for name resolution, locating domain controllers to utilize for Exchange configuration information, and routing messages between remote Exchange servers. Exchange servers as well as workstations that run the Exchange management tools rely heavily on Kerberos for authentication. Therefore, it is equally important that the Exchange server A records are registered within DNS correctly as well.

As a best practice, implement DNS with dynamic updates enabled. I have been in a few environments where transient Exchange and client issues were tracked to missing or invalid SRV records. Some of the specific issues that I have seen include the following:

- *Invalid host record for the Exchange Server—the connection suffix of the server did not match the DNS record causing Kerberos authentication failure.*
- *The domain GUID records for the domain were incorrectly entered under the _msdcs zone, causing improper identification of domain controllers for the Active Directory domain.*
- *Slowness issues resulting from missing site location records, causing Exchange to possibly grab a Global Catalog located at a distant site—thus communication needs to flow across WAN links. This might be because some or all of the following records are missing or incorrect in DNS: _ldap._tcp ._sitename._sites._gc._msdcs.domain.com.*

Most modern DNS implementations in use today support dynamic updates. As a best practice it is advisable to allow only secure updates, which prevents rogue systems from injecting invalid entries into your DNS zones.

A few environments refused to globally enable dynamic updates on their zones. We were able to convince the team to allow only domain controllers to dynamically update their records. Exchange server records were created manually. However, A records are familiar to DNS administrators and less likely to be incorrect. As with any manual process, it can be incorrectly created, so always double-check. If this is not possible, try to convince the DNS team to temporarily enable dynamic updates during the DCPROMO process and the subsequent reboot to allow the domain controllers to dynamically create/update all of the necessary records. Obviously this requires more process overhead, but in the long run it will save on issues, outages, and hours of troubleshooting caused by incorrectly configured DNS records.

Several tools are available to validate records and the functionality of DNS, such as DNSLint, DCDiag, and netdiag. Other standard tools include nslookup, ipconfig, and nltest.

DNS Records Used by Exchange 2010

DNS provides a number of critical functions for Exchange 2010. This section provides an overview of the most important records in DNS.

A RECORDS

A records or *Host records* provide a host name to IP address mapping. Host records are required for each domain controller and other hosts that need to be accessible to Exchange Servers or client computers. Host records use IPv4 (A records).

Here is an example of an A record:

```
berlin-dc01.litware.com. IN A 10.10.0.10.
```

SRV RECORDS

All Exchange 2010 servers use DNS to locate a valid domain controller or global catalog. By default, each time a domain controller starts the Netlogon service, it updates DNS with *service (SRV) records* that describe it as a domain controller and global catalog server, if applicable.

SRV resource records are DNS records. These records identify servers that provide specific services on the network. For example, an SRV resource record can contain information to help clients locate a domain controller in a specific domain or site. For that reason, the SRV records for domain controllers and global catalog servers are registered with several different variations to allow Exchange servers locating a suitable domain controller or global catalog during the Active Directory discovery process.

One option is to register DNS records by site name, which enables computers running Exchange Server to find domain controllers and global catalog servers in the local Active Directory site. Exchange Server always favors the selection of a domain controller and/or global catalog from the same site that Exchange is installed into.

Here is an example of an SRV record:

```
_ldap._tcp.litware.com. IN SRV 0 100 389 berlin-DC01.litware.com.
```

MX RECORDS

A *Mail Exchanger (MX) record* is a resource record that allows servers to locate other servers to deliver Internet e-mail using the Simple Mail Transfer Protocol (SMTP). An MX record identifies the SMTP server that will accept inbound messages for a specific DNS domain. Each MX record contains a host name and a preference value. When you deploy multiple SMTP servers that are accessible from the Internet, you can assign equal preference values to each MX record to enable round-robin between the SMTP servers. You also can specify a lower preference value for one of the MX records. All messages are routed through the SMTP server that has the lower-preference-value MX record, unless that server is not available.

Here is an example of an MX record:

```
litware.com MX 10 fresno-ht01.na.litware.com.
```

NOTE ***You don't need MX records for Hub Transport servers that are involved in internal mail routing. That is only required for external SMTP routing—for example, to the Internet.***

More information about MX records and how they are used for SMTP message routing can be found in Chapter 5, "Routing and Transport."

SPF RECORDS

Exchange Server 2010 uses *Sender Policy Framework (SPF) records* to support Sender ID spam filtering. If you want to use this feature, you need to configure the SPF records in DNS. This is described in more detail in Chapter 7, "Edge Transport and Messaging Security."

Split DNS

Split DNS or split-brain DNS is about setting up separate DNS zones so that DNS requests that come from the Internet will resolve to different IP addresses than requests coming from your internal workstations or servers. In other words, as shown in Figure 3-1, if the Internet client resolves mail.litware.com, it will receive an IP address that is associated with an external firewall solution that is sitting in the perimeter network. The internal client will get an IP address associated with the internal Client Access server array.

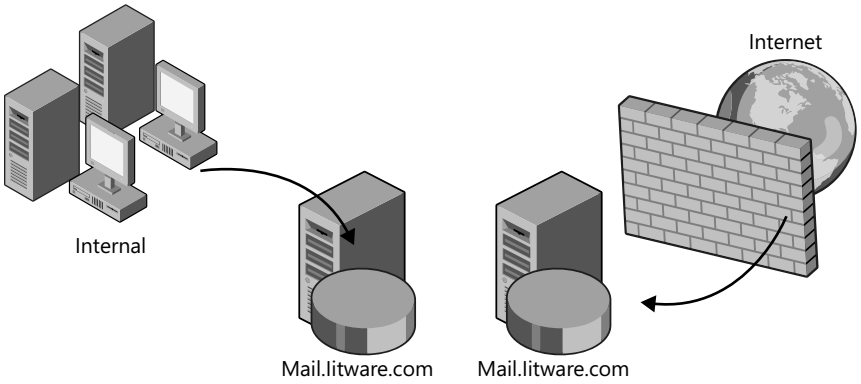

FIGURE 3-1 How split DNS works

The benefit of using split DNS is that it helps control client access. Internal clients use the internal systems instead of the external systems. In other words, internal users' sessions aren't handled by the firewall application and you do not expose internal IP addresses or host names to the Internet.

You can also limit access to specific hosts that are part of the perimeter network or force users to take a specific communication route. For this reason it is a best practice to implement split DNS in every Exchange organization that has server roles exposed to the Internet.

Fixed IP Address vs. Dynamic IP Address

It's important to know whether your company has an Internet provider that provides your company with fixed IP addresses or if you're using dynamic IP addresses to access the Internet. If your servers that have some relationship to external communication, such as Edge Transport servers, have fixed IP addresses and your DNS entries (MX or A records) are registered accordingly, you're working with the best practices approach.

However, a fixed IP address might be a cost issue, especially in small companies. Thus some companies might want to implement Exchange 2010 based on an Internet provider that only provides a dynamic IP address. If you're in this situation, you should consider a Dynamic DNS service that lets you register your dynamic IP address to their DNS service. However, make sure the dynamic DNS service includes the following:

- Your IP addresses should automatically register in DNS when the IP address changes. Your router and/or Dynamic DNS service provider need to support this.
- IP updates should be replicated in DNS real time to make sure the change is known to the Internet immediately.
- For external SMTP servers to know how to send messages to your domain, the DNS record for your domain should include an MX record.
- The Dynamic DNS service should provide you with an SMTP relay host to send messages to the Internet. If you directly send messages, your server is quite likely to be detected as spam because of your changing IP addresses. Many SMTP servers consider dynamic IP addresses as not trustworthy and thus don't accept messages from them.

If you consider these points, you'll have no problem operating Exchange Server 2010 when using a dynamic IP address.

Internet Protocol (IPv4 and IPv6)

Internet Protocol Version 4 (IPv4) is commonly available and the basis for communication between any device on the Internet. The successor of IPv4 is called Internet Protocol Version 6 (IPv6), as defined in RFC 2460 in 1996.

IPv6 was developed to correct many of the shortcomings of IPv4, such as the limited pool of available IP addresses and the lack of extensibility. Because IPv6 addresses are 128 bits long (compared to IPv4 addresses, which are 32 bits long), there are enough IPv6 addresses available for every living insect, animal, and person on earth.

Unfortunately, IPv6 is not an extension of IPv4 but a completely new protocol. Therefore, an IPv4 network can't communicate directly with an IPv6 network and vice versa. Any network device, such as a router, needs to be able to understand IPv6; otherwise, IPv6 causes communication problems.

IPv6 for Windows

The client and server software needs to support IPv6 to use it. The following Microsoft server operating systems support IPv6:

- Windows Server 2003 (IPv4 is installed and enabled; IPv6 is not installed by default.)
- Windows Server 2008 (IPv4 and IPv6 are installed and enabled by default.)
- Windows Server 2008 R2 (IPv4 and IPv6 are installed and enabled by default.)

> ***IMPORTANT*** ***Microsoft also recommends that you do not turn off IPv6 in a clustered environment because Windows Server 2008 R2 Clustering uses IPv6 for internal communication.***

Not only does the server need to support IPv6, but also the client operating system. The following Microsoft client operating systems support IPv6:

- Windows XP Service Pack 1(SP1) or later (IPv4 is installed and enabled; IPv6 is not installed by default.)
- Windows Vista (IPv4 and IPv6 are installed and enabled by default.)
- Windows 7 (IPv4 and IPv6 are installed and enabled by default.)
- For more information about IPv6, see the IPv6 for Microsoft Windows FAQ at *http://go.microsoft.com/fwlink/?LinkId=147465*.

NOTES FROM THE FIELD

Hardware Provider Recommended to Disable IPv6 for Windows Server 2008

For a recent project I consulted my server hardware provider, who recommended turning off IPv6 because their network interface card (NIC) drivers caused problems especially when using NIC Teaming. The operating system was Windows Server 2008; thus we used the following registry key to disable IPv6 from all LAN interfaces, connections, and tunnel interfaces:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip6\Parameters\
DisabledComponents set to 0xFFFFFFFF (DWORD type)
```

After a reboot, even IPCONFIG.exe no longer showed IPv6 addresses. You can find more information on how to disable IPv6 at http://support.microsoft.com/kb/929852/.

Windows Failover Clustering on Windows Server 2008 R2 requires IPv6 and the Exchange 2010 Database Availability Group uses some elements of Windows Failover Clustering. However, Exchange 2010 does not use IPv6 within a DAG. This does not mean that you can disable IPv6. Even though the DAG neither depends on nor uses IPv6, disabling IPv6 completely for Windows Server 2008 R2 is not a tested scenario and could therefore result in unpredictable consequences for a DAG.

IPv6 for Exchange Server 2010

Because Exchange Server 2010 runs on Windows Server 2008 or R2, you might think that it automatically supports IPv6. However, you should consider a few things before planning for IPv6 and Exchange 2010. The following Exchange Server roles can cause issues when using IPv6 addresses:

- **Hub or Edge Transport** Features such as IP Allow List Providers or Sender reputation do not support IPv6 because they require static IP addresses.
- **Unified Messaging** All features do not support IPv6 but need IPv4 to work correctly.
- **Client Access Server** Autodiscover and EWS Web services endpoints because you cannot configure an IIS binding for an IPv6 address—WCF throws a Watson exception if you try to configure it.
- **Database Availability Group (DAG)** Even though you cannot define an IPv6 DAG IP address, IPv6 is supported. When static IPv4 addresses are specified for a DAG, it only uses IPv4. When no static IPv4 address is specified, Exchange use DHCP for the IPv4 addresses and also creates IPv6 address resources.

Exchange Server uses the Windows network stack to process any request. Each request depends upon two things:

- Name resolution (when initiating the request)
- Packet type (when receiving a request)

First, the name resolution will determine how to initiate the request to another computer based on which address (IPv4 or IPv6) is resolved first. When the name resolution comes back with an IPv6 address, this address is used to initiate the request.

Second, packet types do not mix IP versions midstream. If the request comes in as IPv4, the response will also use IPv4, and the same is true for IPv6. As for unfulfilled requests, they should be handled before name resolution is completed and would otherwise fail after name resolution in the same way other transient network failures occur. For features that don't support IPv6, Exchange causes the request to fail so that the client initiates another request using IPv4. Thus an IPv6 request, even to the Unified Messaging role, does not cause a problem but is just ignored.

NOTE *The official Microsoft support statement for IPv6 is that Exchange Server 2010 running on Windows Server 2008 or R2 requires an IPv4 address. Exchange Server 2010 is not supported in a pure IPv6 environment where you disable the IPv4 protocol.*

Understanding Client Load Patterns

Another important aspect that should be understood when planning for Exchange 2010 is the current client load patterns—namely the traffic between Outlook clients (or other mail clients) and the Exchange server.

The scope of this task depends on what your current mail clients are. If most of your clients use POP3 or IMAP4 clients, the load on an Exchange server is significantly lower and you can plan for many more users on a single server.

If you're using a MAPI-based client, such as Microsoft Outlook 2003 or Outlook 2007, you need to analyze which profile your average users fall into to understand the impact of the traffic to the Exchange server. You can use the information available from your monitoring system, such as Microsoft System Center Operations Manager if available. Alternatively you can use Windows Performance Monitor to collect the performance information of your clients. Consider using the following performance counters:

- Messages sent/received per day
- Average message size
- Messages read per day
- Messages deleted per day
- Outlook Web Access logon and logoff per day

Consider collecting the client data from each Exchange server or mail server (when coming from a non–Exchange System) for at least a couple of days (at peak times, not on weekends) to have a representative aggregation of performance data.

NOTES FROM THE FIELD

Identifying Current Client Load

Andy Schan
Senior Consultant, Schan Consulting Inc., Canada

To work out the current client load, my last couple of projects had Quest MessageStats in place, so I sat down with the MessageStats data and Microsoft Excel and crunched the numbers for the last couple of quarters to come up with a profile of their typical users.

To come up with a useful client load picture for scaling the new environment, I typically make assumptions similar to the following:

- *10-hour workdays*
- *5-day workweeks, Monday–Friday*
- *Messages/day/user derived from the data are all sent during normal working hours*

I also use the messaging data for the previous calendar year quarter and derive message/day data from that, to ensure that I'm seeing a reasonably accurate picture of the average client activity in the environment and to minimize any effects from brief periods of increased activity. I may also go back two quarters, if the previous quarter includes quiet periods such as summer vacation or Christmas holidays.

I focused on figuring out messaging activity (number of messages/day sent and received, calendaring activity, and so on) rather than actual bits over the wire, which changes once you get them to cached mode and a newer version of Outlook. Together with the data from the client, I used the following Microsoft whitepaper to assess the load: "Outlook Anywhere Scalability with Outlook 2007, Outlook 2003, and Exchange 2007" at http://technet.microsoft.com/en-us/library/cc540453(EXCHG.80).aspx.

After you collect the results, you can compare the data with Table 3-1 to identify how to classify your client profile according to Microsoft's most common client profiles.

TABLE 3-1 Common Client Profiles

TASK/PROFILE	LIGHT	MEDIUM	HEAVY	VERY HEAVY
Sent per day	5	10	20	30
Received per day	20	40	80	120
Average message size	50k	50k	50k	50k
Messages read per day	20	40	80	120
Messages deleted per day	10	20	40	60
Outlook Web Access logon and logoff per day	2	2	2	2

When you have identified your typical client load pattern, you should plan to implement a load-generating tool such as Exchange Load Generator 2010 to verify your Exchange server hardware performance. Exchange Load Generator 2010 (64 bit) is available

at *http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=cf464be7-7e52-48cd-b852-ccfc915b29ef*.

You'll find more details on how to plan your Exchange hardware and how to use the Exchange Load Generator in Chapter 13, "Hardware Planning for Exchange Server 2010."

Perimeter Network

Communication to the Internet or external network is also important. You'll find information about the required firewall ports or protocols that need to be configured later in this chapter in the "Planning Network Port Requirements" section. This discussion does not cover other security options, such as IPsec or VPN, that allow clients on the Internet to directly connect to their internal network.

The recommended deployment for Exchange Server 2010 Internet access includes two firewalls or routers in a back-to-back firewall scenario, which enables you to implement a perimeter network between the two. An external firewall faces the Internet and protects the perimeter network. You then deploy an internal firewall between the perimeter and your internal corporate network.

In the perimeter network you place any Internet-facing server, such as the Edge Transport role of Exchange Server 2010. Microsoft does not support any topologies that put firewalls between a Client Access, Hub Transport, Unified Messaging, and a Mailbox (MBX) server. Putting a firewall between these roles could cause issues because they use dynamic ports that could be blocked unintentionally by the firewall. The only Exchange 2010 role supported for deployment in a perimeter network—and with a firewall server separating it from other Exchange servers it talks to—is the Edge Transport role.

IMPORTANT ***The Edge Transport server role should never be a member of your internal domain, but should be a stand-alone server or member of an available perimeter Active Directory forest.***

The most common servers that are placed in the perimeter to support Internet access are:

- A smart host to route SMTP messages between the internal and external network, such as Edge Transport server role or any other smart host.
- A reverse proxy or application-layer firewall that supports client-related traffic such as Autodiscover, Outlook Web App (OWA, previously known as Outlook Web Access), Outlook Anywhere, ActiveSync, POP3, IMAP4, SMTP, and so on to the internal network. Microsoft Forefront TMG and Microsoft ISA Server 2006 are example application-layer firewalls. However, don't underestimate the scalability challenges for software-based reverse proxy servers. Any implementation that needs to handle more than 100,000 concurrent connections on an ongoing basis should focus on a hardware solution.

NOTE ***You should not deploy the Client Access Server role in a perimeter network to reduce the attack surface of your internal forest. Because the Exchange computer account of the Client Access Server role has elevated privileges, it can be used by an attacker to destroy your Active Directory. Instead, use an application-layer firewall such as Microsoft Forefront Threat Management Gateway (TMG) to publish the Client Access server services to the Internet.***

If you do not use an application-layer firewall from Microsoft, consider the following key areas for choosing a firewall application of highest security standard:

- **Pre-authenticate traffic** To prevent unauthenticated traffic from entering the corporate network.
- **Packet inspection** Application-layer firewalls allow for identification of known protocol attacks prior to entering the corporate network.
- **Intrusion Detection System (IDS)** Simplifies identification of attacks on the system. If attacks occur internally, chances are they will be successful and difficult to detect because they may look like typical traffic, whereas if the proxy starts requesting RPC to other servers, or tries to get through the firewall, it is blocked and logged.
- **Fixed Ports/IP Addresses** Only specific ports and IP addresses are allowed into the corporate environment.
- **Group Membership allowance** Provides the capability to allow only specific groups to access specific applications; for instance, my current customer does not allow hourly workers to access mail externally.
- **Load balancing** Arrays of reverse proxy servers can distribute network traffic for a single URL.

As a best practice, always implement a reverse proxy or application-layer firewall if you want to provide Internet access to your Exchange servers. However, some companies, especially in the small to medium sector, do not implement any kind of security between their servers and the Internet. If you do not implement an application-layer firewall, consider the following recommendations:

- Deploy a firewall between the internal and external network and open only the ports or protocols you need.
- Implement a server certificate for all your Exchange servers. (This can be a single certificate that includes the required domain names, as described in the "Planning Certificates" section of this chapter.)
- Require SSL to encrypt client communication (for Outlook client traffic).
- Require TLS for SMTP and SSL if you enable POP3 or IMAP4.
- Make sure that any operation requires authentication.
- Implement Forms-based authentication for Outlook Web App.

This provides you with at least minimum security but still might expose some of your user data to the Internet. However, it's better than nothing.

Avoiding Pitfalls by Providing Technical Recommendations

The following list provides ways to avoid potential pitfalls on the network topology side. Any problems must be rectified before Exchange Server 2010 can be installed at the location.

- Make sure that the physical network speed of sites that will host Exchange Server 2010 has at least 64 Kb per second of bandwidth available.
- Exchange Server 2010 does not support a pure TCP/IP v6 (IPv6) environment. If you've already implemented pure IPv6 addresses anywhere in your company, make sure that they also support IPv4 addresses; otherwise, the clients might encounter errors in communicating with Exchange Server 2010.
- IP subnets should map to the locations of the company and should be non-overlapping between locations. However, sometimes single locations have multiple IP subnets, which is fine. If IP subnets are spanned between multiple physical locations, make sure the WAN link between them matches LAN link speed—10 megabits per second (Mbps) or more.
- Make sure your Active Directory sites match IP subnets for each location.
- DNS must be used for network name resolution.
- Active Directory uses service (SRV) resource records in DNS to register a list of domain controllers for client use. If you do not use Windows Server 2008 DNS Service for Active Directory, make sure that your DNS server software supports the resource records!
- To receive messages from the Internet, an appropriate mail exchanger (MX) resource record in DNS is required for the company's domain name.

NOTES FROM THE FIELD

Additional Beneficial Server Settings

Joe Cirillo
Senior Engineer and Architect, Horizons Consulting, US/Central Region

Microsoft has always provided some invaluable guidance regarding the installation and automation of Exchange. Over the years, I have found some additional settings I like to confirm or set to help ease administration or to further ensure that the installation will occur without error.

Network Interface Card Naming Standard

I always try to reduce ambiguity for any objects that I interface with. Because I often refer to the settings on the NIC when troubleshooting, I like to provide an easy-to-follow naming standard for the NICs. This is particularly helpful when multiple NICs are configured, as required when using the Exchange Server 2010 Database Availability Group.

Confirm that the adapter binding order is correct

Because the server host name in the registry will bind to the first interface in the adapter/bindings list, the binding order must be set properly.

I have seen connectivity issues caused by improperly configured bindings. Anytime you make changes to the network configuration (such as adding protocols, adapters, modems, or services), the binding order can potentially change. Be sure to check the binding order on server install and anytime you make changes to the network configuration.

Confirm that Windows Remote Registry Service is running

Exchange Server requires access to the local registry to retrieve various settings. One example is the Exchange Setup program. The Exchange Setup program uses DNS to obtain the fully qualified domain name (FQDN) of the local computer to access the registry. If the Windows Remote Registry Service is not running, setup will be unable to access the registry, causing the installation of Exchange to fail.

Confirm that Date, Time, and Time Zone are set properly

Exchange gets its date and time information from the operating system. Windows Directory servers need to have their time synchronized for Kerberos authentication to work correctly. Kerberos works by exchanging time-stamped authenticator identification tokens. By default, directory servers have a maximum tolerance for computer clock synchronization of five minutes. This is known as clock skew. Clock skew is the range of time allowed for a server to accept Kerberos tickets from a client. If the clock skew is greater than five minutes, Kerberos authentication fails, which results in cascading authentication failures for Exchange Server.

If an Exchange server's time is out of sync with the Domain Controller time, issues will occur such as Exchange services failing to start or client connections being rejected. To prevent issues caused by time skew, make certain of the following:

- *The Exchange server has network connectivity with the Domain Controller.*
- *The Exchange server time is synchronized with either the Domain Controller time or the time server that is used by the Domain Controller.*

Desktop Background

Setting the Desktop Background to display useful information is also useful for eliminating ambiguity when logging onto a server. It is very inefficient to have to click through several diagnostic windows just to find that you logged on to the wrong server, or to find information such as the IP address or operating system version. If you manage multiple computers you will benefit greatly by displaying relevant information about the Windows server on the desktop's background. You can find the BgInfo tool at http://technet.microsoft.com/en-us/sysinternals/bb897557.aspx.

Import any Trusted Root Authorities onto the local computer

Depending on how you are managing certificates (such as by using a stand-alone Certificate Authority, enterprise Certificate Authority, or third-party Certificate Authority), you may need to add the certificate for a foreign Certificate Authority to the Trusted Root Certification Authorities container on the local computer. If this is necessary, be sure to add the certificate for the foreign Certificate Authority to every server you install to ensure consistency in your build and to avoid communication errors between servers.

Any e-mail client connecting to the Exchange Server's secure sites must trust the Exchange Server's site certificates. Before it can successfully negotiate a secure SSL/TLS link with the Exchange Server, the e-mail client must trust the Certificate Authority (CA) issuing the Web site certificate to the Exchange Server's services. If the certificates issued by the foreign CA will be used by client-facing Exchange services (such as Outlook Web App or Outlook Anywhere), be sure to also add the certificate of the foreign CA to the Trusted Root Certification Authorities container on any user workstation or mobile device.

Evaluating and Planning for Active Directory

Active Directory is the integrated, distributed directory service included with Windows Server operating systems. Many applications, such as Exchange Server 2010, integrate with Active Directory. This creates a link between user accounts and applications, which enables single sign-on for applications. Additionally, the Active Directory replication capabilities enable distributed applications to replicate application-configuration data.

How Exchange 2010 Uses Active Directory

The Active Directory database is divided into logical partitions—namely the schema partition, the configuration partition, and a domain partition for every domain.

Windows Server 2008 and R2 includes a tool called *Repadmin* that can be used to list all Active Directory partitions available. Figure 3-2 shows the result from the Litware Scenario using the command *Repadmin /showrepl*.

As shown in the figure, Active Directory is made out of the configuration, schema, application, and domain partitions.

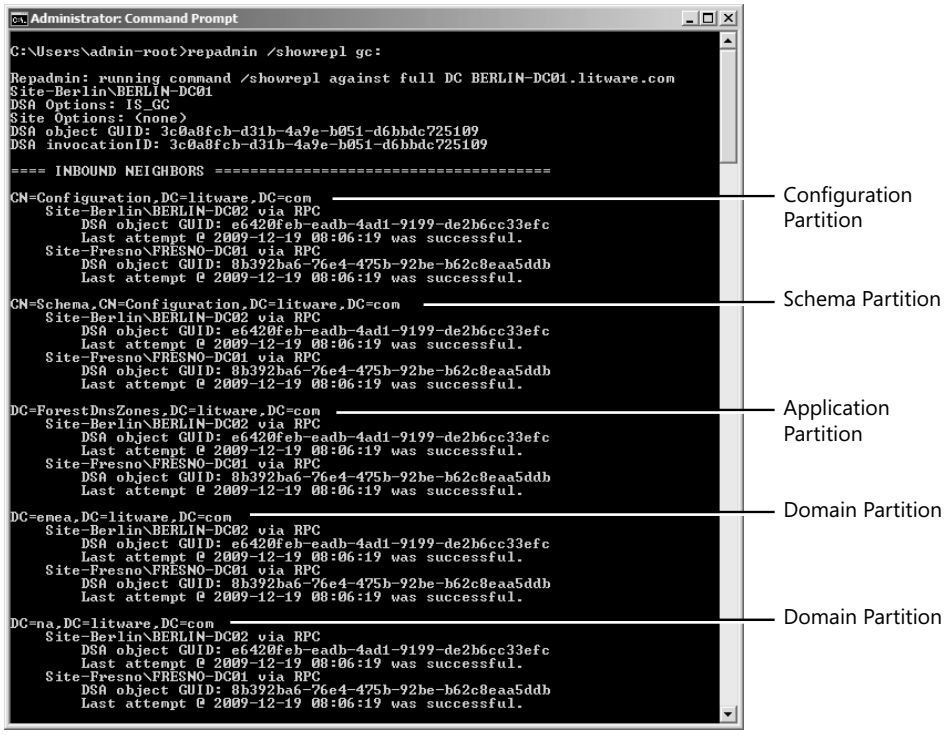

FIGURE 3-2 Using Repadmin to look at Active Directory partitions

NOTE *Additional information about Active Directory Partitions can be found in "Active Directory Logical Structure and Data Storage" at http://go.microsoft.com/fwlink/?LinkId=179859.*

The Schema Partition

Before Exchange Server 2010 can store information in Active Directory, the schema partition needs to be modified so that Exchange-related objects (such as connector or mailbox information) and attributes (such as an Exchange Mailbox server or a user object) are defined in the Active Directory schema. The schema partition stores the general layout of all Active Directory objects and its attributes. It includes two types of information:

- **Schema classes** The objects that can be created
- **Schema attributes** The properties that can be used for each object

NOTE *Each and every domain controller and global catalog server in Active Directory contains a complete replica of the schema partition. Thus it is important to plan the Exchange Server 2010 schema extension accordingly—you cannot remove a schema extension.*

INSIDE TRACK

How to Safely Extend the Schema

Ross Smith IV
Senior Program Manager, Exchange Server Product Group, Microsoft Corporation

Schema extensions are permanent. Care should be taken to ensure that a schema extension is successful because a failed schema extension may mean rebuilding the forest. The preferred way to mitigate the impact of a failed schema extension is to isolate the schema master by disabling its ability to replicate changes to other domain controllers in the forest.

This is accomplished by executing the command repadmin /options +DISABLE_OUTBOUND_REPL. When outbound replication has been halted on the schema master, you can then proceed with extending the schema. If the schema extension is successful, you can re-enable outbound replication and allow the changes to propagate throughout the forest. If, on the other hand, the schema extension fails, you simply shut down the schema master server, wipe it, and seize the schema master role on another domain controller.

Because Exchange depends on Active Directory, each released Exchange version implemented different schema versions. To identify the current Exchange schema version, the schema attribute *ms-Exchange-Schema-Version-Pt* was added. In the attribute, you can identify the schema version by looking at the *rangeUpper* attribute and identify the Exchange version according to Table 3-2.

TABLE 3-2 Exchange Schema Version Numbers

RANGEUPPER NUMBER	EXCHANGE VERSION
14622	Exchange Server 2010 or Exchange Server 2007 SP2
11116	Exchange Server 2007 SP1
10628	Exchange Server 2007

RANGEUPPER NUMBER	EXCHANGE VERSION
6870	Exchange Server 2003
4406	Exchange Server 2000 SP3
4397	Exchange Server 2000

In Figure 3-3 you can see that the Exchange Schema version is currently Exchange Server 2010 or Exchange Server 2007 Service Pack 2.

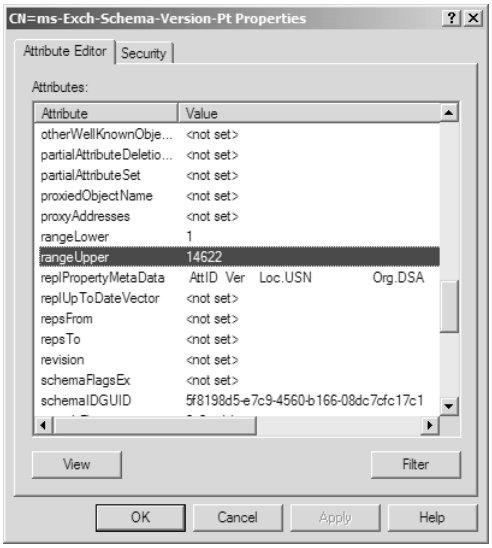

FIGURE 3-3 Checking the Exchange schema version in ADSI Edit

Using the *rangeUpper* attribute you can also identify whether a schema update was replicated successfully to the local domain controller. By connecting to the DC and then verifying that the attribute has the current value, you can be sure that the schema update has replicated.

IMPORTANT ***It is true that the Exchange Server 2010 schema extension also includes the Exchange 2003 and 2007 schema extensions. However, if you plan to ever install an Exchange 2003 or 2007 server into the organization after Exchange 2010 is deployed, you must install the older Exchange Server version as the first server and install Exchange Server 2010 afterwards. Exchange Server 2007 should include the Mailbox, Client Access Server, and Hub Transport roles. Once you have installed Exchange Server 2010, you will not be able to install Exchange 2003 or 2007 anymore!***

The Configuration Partition

The configuration partition contains configuration information for the Active Directory forest. Additionally, some distributed applications and other services store information in the configuration partition. This information in the configuration partition replicates through the entire forest so that each domain controller and global catalog has a replica of it.

The configuration partition stores each type of configuration information in separate containers. A *container* is an Active Directory object similar to an organizational unit (OU) that you use to organize other objects.

Exchange Server 2010 stores information such as global settings, address lists, connections, and so on to the configuration partition. Figure 3-4 shows you how to look at the information Exchange Server 2010 stores in the configuration partition. You can see it either using ADSI Edit or Active Directory Sites and Services. (You'll need to enable the Show Services Node.) You also need to be a member of the Organizational Management group or the View-Only Management group; thus, you must have Exchange Organizational permissions to expand below the Exchange Organization level.

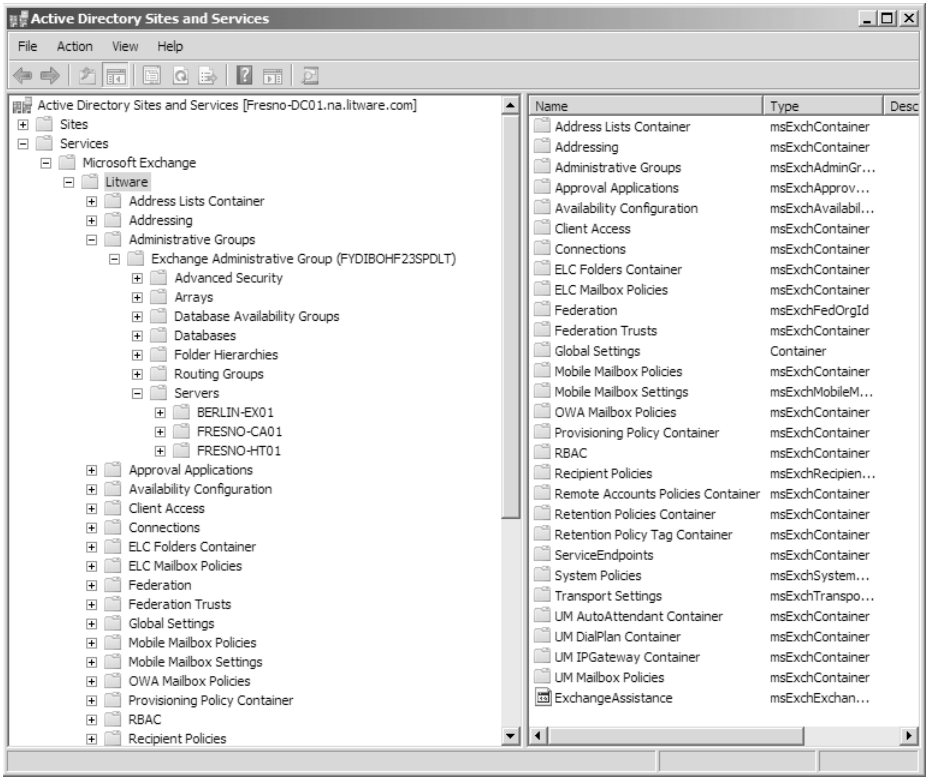

FIGURE 3-4 Exchange configuration in the configuration container

The Domain Partition

The domain partition holds domain-related information in containers as well as OUs. It includes information about users, groups, and computers in that domain. The domain partition is stored on every domain controller of that specific domain. Every global catalog server has a subset of information from every domain partition in the forest, as well as a complete copy of its own domain's objects. For example, a global catalog server in a different domain will contain information on the individual user, such as the user's display name or its SMTP addresses, but not its password.

For every Exchange-prepared domain (meaning that the Exchange Setup /PrepareDomain has been run for the domain) Exchange Server 2010 creates an OU called Microsoft Exchange System Objects in which it will store Exchange-related system objects such as the mailbox database's mailbox and public folder proxy objects.

You can verify that a domain was Exchange Domain–Prepared by looking at the *ObjectVersion* attribute on the Microsoft Exchange System Objects OU in that domain. For Exchange 2010 RTM it should read 12639, as shown in Figure 3-5.

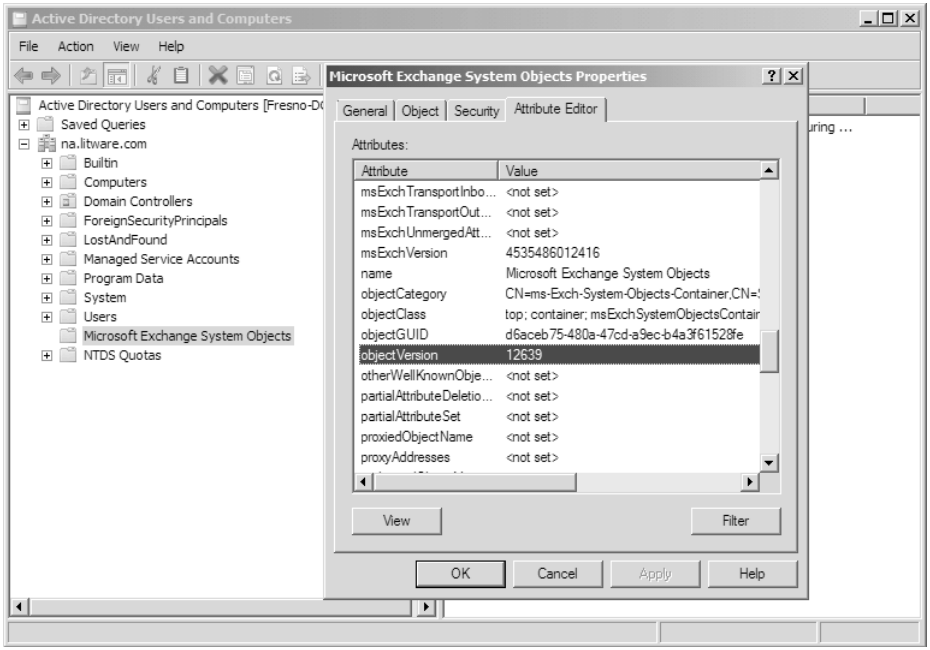

FIGURE 3-5 Identifying an Exchange-prepared domain

The Application Partition

Application partitions hold specific application data that the application requires. The main benefit of application partitions is replication flexibility. You can specify the domain controllers that hold a replica of an application partition, and these domain controllers can include a subset of domain controllers throughout the forest.

Currently the only application that uses the application partition is DNS, to store DNS zones in the partition as Active Directory–integrated DNS zone. Exchange Server 2010 does not use application partitions to store information.

Table 3-3 describes the application partitions commonly available in Active Directory.

TABLE 3-3 Application Partitions in Active Directory

APPLICATION PARTITION	DESCRIPTION
ForestDnsZones	Replicates to all DNS servers in forest
DomainDNSZones	Replicates to all DNS servers in domain (is created once you add the second DNS server to the domain)

Of course only DNS servers that run on Domain Controller can access the application partitions; thus all DNS zones that are stored in this partition are Active Directory–integrated.

NOTE ***Using ADSI Edit you do not see application partitions by default as well-known naming contexts. To look at them you need to directly address them using their distinguished names (DNs). You can identify the DN using a tool such as repadmin /showrepl.***

Active Directory Replication Impact on Exchange 2010

Active Directory replication is a crucial component of Exchange 2010. As described in previous sections, the different partitions store Exchange-related configuration data. This data is automatically replicated between Active Directory sites using Active Directory replication mechanisms.

Because Exchange 2010 relies on the replication mechanisms to work correctly, you might face delays in configuration caused by replication latency. For example, if you configure an Exchange Server in the domain emea.litware.com, but your current computer is located in na.litware.com, you will see that configuration is not immediately available in the domain emea.litware.com. You need to wait until Active Directory replication takes place and replicates the changes to the domain. Normally the replication between sites happens every 15 minutes or longer depending on your Active Directory Site Link configuration.

There are two possibilities to overcome the replication delay:

- Configure your EMC or EMS so that you directly use a domain controller located in the target domain. For example, in EMS you can set the preferred domain controller using the following cmdlet:

```
Set-ADServerSettings –PreferredServer <DC FQDN>
```

- Use Repadmin to push replication to target domain. For more information on the Repadmin tool, read the Microsoft whitepaper "Monitoring and Troubleshooting Active Directory Replication Using Repadmin" available at *http://www.microsoft.com/downloads/details.aspx?familyid=c6054092-ee1e-4b57-b175-5aabde591c5f&displaylang=en*.

Besides Active Directory replication, Active Directory sites and IP site link information are important for message routing between Exchange servers. Exchange 2010 uses the cost assignment that is part of every IP site link to determine the lowest-cost route for traffic to follow when multiple paths exist to the destination. This information is used by the Hub Transport role to decide to which Exchange Hub Transport server a message is sent to when the target Exchange Hub Transport server is not available.

More information about Active Directory sites, IP site links, and their relevance to message routing in Exchange 2010 can be found in Chapter 5, "Routing and Transport."

Active Directory Requirements

For Exchange Server 2010 Active Directory and domains must meet several requirements. Consider the following when evaluating your current Active Directory design:

- The server on which the Schema Master role runs must have at least Windows Server 2003 SP1 (32-bit or 64-bit) installed.
- You need to run Windows Server 2003 SP1 or later (32-bit or 64-bit) on global catalog servers in every Active Directory site where you plan to install Exchange Server 2010. If you still have older global catalog in your environment, it is recommended that you upgrade all your domain controllers to prevent any problems.
- Active Directory must be at least in Windows Server 2003 forest functionality mode.
- Windows Server 2008 functionality mode is supported if your Exchange organization includes Exchange 2007 and/or 2010.
- All domains that will include Exchange Server 2010 servers or recipients must be at least in Windows Server 2003 domain functional level.
- Because of the importance of the global catalog in an Exchange Server organization, you must deploy at least one global catalog in each Active Directory site that contains an Exchange 2010 server.

Single versus Multi-Forest Implementation

The following forest implementations are available:

- Single Forest
- Multi-forest

- Resource Forest
- Hybrid Forest

Figure 3-6 shows the different forest approaches and in what forest user accounts mailboxes and Exchange servers are available.

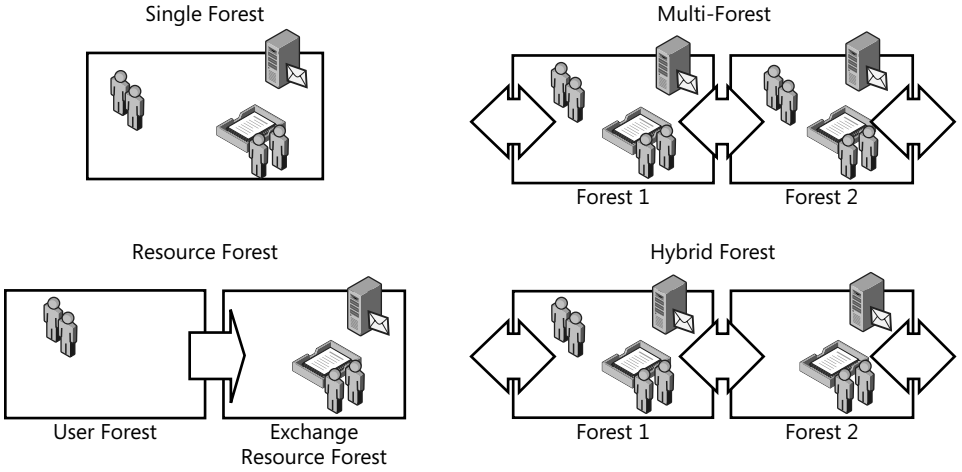

FIGURE 3-6 Exchange Forest Topologies

NOTE ***One issue that you should keep in mind before deciding your forest design is the management effort required to administer the various types of forest. Sometimes you might make a decision regarding the kind of deployment to execute only to realize in production that it's harder to manage—and perhaps even that some tools don't work in a specific configuration. The old KISS rule (keep it stupid and simple) comes to mind here, which you should always consider if you have the chance.***

Single Forest

An Active Directory forest is a set of one or more domain trees that share common configuration and schema information. A forest is a security boundary. By default, no account from outside the forest can hold security principals to access information inside the forest. A single Active Directory forest design provides the following characteristics:

- A single forest that matches the Exchange organization. (This is the most common and easiest approach.)
- No limitations to Exchange and Outlook functionality.

Multi-Forest

The multi-forest (or cross-forest) implementation consists of at least two loosely connected forests that operate independently from each other but are somewhat connected. This forest approach includes the following characteristics:

- Includes multiple Exchange organizations, often multiple SMTP addresses.
- Users are part of the different forests.
- Can include multiple service providers that administer their own forest and do not have access on another forest.
- Common when one company buys another company without integrating it into the existing Active Directory forest.
- The forests have to share a trust relationship before data can move between the forests. This is the requirement so the forests can be connected from the Exchange perspective using, for example, linked mailboxes. This approach has quite a few limitations. For example, a user cannot easily open a mailbox from another user that is located in a different forest.
- Synchronization of availability information and public folder information between forests is often very complicated.

Resource Forest

The resource forest implementation consists of one or more account forests and an Exchange forest that includes all Exchange servers, mailboxes, and distribution lists. The account forest contains the user accounts and security groups. This forest approach includes the following characteristics:

- Mailboxes are attached to disabled user accounts and associated to user accounts in the account forest.
- Administration between the account forest of the organization and the Exchange forest is separate. Most often you will create a resource forest in a hosting environment. A service provider provides the resource forest that is connected using a one-way trust to the account forest. This ensures that the service provider has no permissions in any way in the account forest and only can manage the resource forest.
- You can have better Exchange capabilities and availability with a clean resource forest that is fully controlled by the Exchange administrators.
- The concept of messaging in a cloud—such as Microsoft BPOS or the implementation of Exchange on a hosting platform—can be seen as a resource forest implementation.
- Typically reduces token bloat by moving the DL membership of a user object to a different forest.
- Because all the mailboxes are part of the same resource forest, there are no limitations to Exchange and Outlook functionality. (If this is not the situation, the implementation is called a *hybrid forest*.)

Hybrid Forest

The hybrid forest combines the concepts of a resource forest and a single forest—thus, a hybrid forest contains not only user accounts or resource mailboxes (such as user-disabled, mailbox-enabled objects) but also includes active mailboxes (mailbox-enabled user accounts where Exchange server is in the same forest). This forest approach includes the following characteristics:

- Each forest may contain a combination of enabled and disabled user accounts that are either mail enabled or mailbox enabled.
- Differs from a resource forest in that all forests have mailboxes, and you might find mail-enabled users and disabled mailbox-enabled users in the same forest.
- Contains both resource mailboxes (user-disabled mailbox-enabled objects) and active mailboxes (user-enabled mailbox-enabled objects).
- Common in a migration to or from a resource forest model or in an organization that has a resource forest for some business units but also uses the resource forest as the primary forest for other business units.
- Only the users of the same Exchange organization have no limitations to the Exchange or Outlook functionality.

NOTES FROM THE FIELD

Planning a Forest Design

Andrew Ehrensing
Principal Consultant, Microsoft Consulting Services, US Central Region

Microsoft recommends starting all designs with a single forest/single domain environment for Active Directory. On some occasions business requirements dictate a change from this model to a multi-forest model; however, this should be avoided whenever possible. Introducing multiple forests adds significant complexity and cost in both capital and operational expenses.

Single vs. Multi-Domain Implementation

After the forest design has been made, the discussion about domains follows. This discussion is only necessary if you have a multi-domain environment and your Exchange implementation will be part of a single forest design.

In such an environment, you need to decide whether you want to install all Exchange servers in a single, Exchange-dedicated domain or place Exchange in the domain where the user accounts are stored. Especially in a single forest implementation you need to consider the following domain approaches.

Single Domain

A single domain is where Exchange servers and users are located in the same domain. This approach has the following characteristics:

- Simplest implementation and thus the easiest to manage
- Centralized administration
- Used if only one domain is available

> ***NOTE*** *Always start with a Single Domain as the basic design; only move to a different domain design if forced!*

Single Exchange-Dedicated Domain

A single Exchange-dedicated domain can be found in a multi-domain forest where one domain is created just for hosting Exchange servers and managing distribution lists. This approach has the following characteristics:

- Exchange servers are maintained in their own dedicated domain. The disadvantage is that the extra domain brings the extra costs that stem from deploying and managing an additional domain.
- Exchange administration of dedicated domain can be totally independent from Active Directory administration of forest. This means that the administrators can perform all administrative tasks on the Exchange servers without requiring any administrative rights in other Active Directory domains.
- Has a special requirement for additional user domain controllers that need to be available where the Exchange servers are located.
- Common if you have your own internal service provider for Exchange.

Multi-domain

A multi-domain approach includes the Exchange servers directly in their user domains—thus, the Exchange servers are spread between the different domains. This approach has the following characteristics:

- Used if the Exchange administrations is split between different divisions or departments that own their own domains
- Used if you have a geographic domain design and thus want to add the Exchange servers to their respective regional domains

In a multi-domain environment you have to make sure that you configure the right scope in EMC and EMS. The following cmdlet configures a forest-wide scope:

```
Set-ADServerSettings -ViewEntireForest:$true
```

NOTE *If you can choose between a single- or multi-domain implementation for Exchange, it is a best practice to use a single domain. This reduces complexity dramatically and will be much easier to manage.*

NOTES FROM THE FIELD

A Mix of Single and Multi-Domain Implementations

A multinational electronics company with more than 70 Active Directory domains in the forest mixes a multi-domain and single Exchange-dedicated domain approach. The reason for running two different approaches was that parts of the company have a centralized Exchange administration, but some countries manage the Exchange servers on their own. However, issues in Exchange caused by replication latency and other problems supported the decision to move to a single Exchange-dedicated domain approach.

Planning Naming Conventions

Another area that needs to be considered is planning for naming conventions for your objects. In a large, complex organization (such as the scenario of Litware) it is easy to fail to understand what function a server performs at first glance; a good naming convention conveys the function of a server to an administrator without forcing the administrator to examine the server's properties.

Many organizations implement naming conventions for key components because of this requirement. Naming conventions can allow administrators and users to easily identify the purpose of an object without requiring input from others. Typically, the larger the organization, the more strict these naming conventions can become.

Naming conventions help you to easily identify key elements of an object. For example, a server's role, location, and purpose can be identified in environments that have grown beyond a tangible number.

Of course, some small companies out there have just a handful of servers, but what if you have hundreds of servers that you can no longer easily remember? In this situation, you should start thinking of naming conventions for your environment.

The most common naming convention is for the Display Name attribute of users and distribution groups. It might get complicated to identify a user when there is no strict convention, especially in companies that have mailboxes from all over the world. We've seen organizations in which some administrators created the mailboxes with the convention Firstname Lastname and others using Lastname Firstname. Of course this causes confusion and should be corrected.

Not only do you need to consider the Display Name—in Exchange 2010 it is recommended that you also define the following names:

- Server
- Database Availability Group (DAG)
- Database
- Active Directory Site
- Mailbox
- Distribution Groups
- …

Naming conventions vary according to the organization's requirements and depend on locations, geographic distribution, and other factors. There is no single best convention available.

To provide some guidance on naming conventions, the following examples were developed for this book. You can use them as basis to define your own conventions. Remember that you might need more objects to be defined than are described in this section; in general it is good to describe all names that are required, but not every name available.

Server Name

The server name can be used to identify the physical location of a server in terms of country, site, city, or datacenter. It is very common for the server name to also include service-specific information. This eases the process of identifying the role of the server by just looking at its name.

Because some city names are long, it is recommended to use an abbreviation such as an internal site code or the three-letter airport code. In this book, the server name includes the city (only cities with short names are used), the role that is running on it, and a number:

<cityname>-<2 letter service spec><2 digit number>

Table 3-4 describes the details about the service-specific abbreviations used to create the server name in this book.

TABLE 3-4 Server Name Service-Specific Abbreviations

SERVICE SPECIFICATION/ ROLE (TWO LETTERS)	DESCRIPTION
EX	Multi Exchange Role Server
MB	Mailbox Server/Public Folder
HT	Hub Transport
ET	Edge Transport

SERVICE SPECIFICATION/ ROLE (TWO LETTERS)	DESCRIPTION
CA	Client Access Server
UM	Unified Messaging
DC	Domain Controller
SV	Multi Role Server

Examples for server names include:

- Dallas-EX01 (multi-role Exchange located in Dallas)
- Munich-DC01 (DC located in Munich)
- Miami-CA01 (Client Access server located in Miami)
- Berlin-ET01 (Edge Transport located in Berlin)

Database Availability Group Name

Database availability group names can include information on physical sites included in the replication (such as Berlin and Fresno or the Active Directory site names) but for most of the implementations a simple numbering is probably sufficient. In this book, the database availability group name includes DAG name and an increasing number. The convention is as follows:

DAG<2 digit number>

DAG01 and DAG02 are both examples of database availability group names.

Database Name

In Exchange versions before 2010 Mailbox, databases always were fixed to a server name. Thus it was common practice to include the Exchange server name or server information in the database name. This practice changed in Exchange 2010—a database is no longer fixed to a server. The database is no longer associated with the server, thus the database name should include other aspects such as the DAG to which the database is associated, the main location where the database is used, or the purpose of the database (such as mailbox limits).

In this book, the database name includes the DAG it is part of, the city where it is based, and an increasing number. You can argue that we did not consider site resiliency, because having the city name as part of the database name might confuse people in the event of a cross-site database failover. However, in that scenario you can clearly identify which databases are failover databases just by looking at the name. The convention is as follows:

<DAG>-<city>-<2 digit number>

Examples of database names include:

- DAG01-Munich-01
- DAG05-Dallas-05
- DAG06-Sydney-01

Active Directory Site Name

The Active Directory site name should follow the purpose of the creation for multiple Active Directory sites—namely, to define different locations, subsidiaries, or branch offices. In most cases these are the physical boundaries of a location or a datacenter, and indicate a different connectivity between them. Thus the important factor in the name for identification is the location or datacenter name. In some cases for larger companies you might consider including the site code or street name into the site name to identify it correctly.

In this book, the Active Directory Site Name includes the word *Site*, a dash, and then the city or region the site spans. The dash needs to be included because spaces are not allowed in the site name. The convention is as follows:

Site-<city>

Examples for Active Directory site names include:

- Site-Miami
- Site-Munich
- Site-Sydney

User Names

To find the users of your system it is important to define a convention for making the user names. This can follow simple rules such as last name and first name separated by a comma or space and can include other information, such as initials when needed or departmental information after the name (such as when you have two identical names). Because many names are not unique, the general rule is to add as much information as necessary to identify the user.

In this book, the user names follow the simplest user name rule: last name followed by the first name separated by a comma. The convention is as follows:

<Lastname>, <Firstname>

Examples of user names include:

- Healy, Joe
- Richardson, Shawn

Planning Namespace

Before you set up your Exchange organization, one of the most important areas that needs to be planned is your internal (organization-facing) and external (Internet-facing) namespace. A namespace is a logical structure commonly represented by one or more domain names in DNS.

Namespace planning is most important for the Client Access Server role. However, many considerations are also needed for the Hub Transport and Edge Transport roles. This section should provide the general basis for understanding the importance for namespace planning. The topics are also discussed in Chapter 4, "Client Access in Exchange 2010"; Chapter 5, "Routing and Transport"; and Chapter 7, "Edge Transport and Messaging Security."

During migration or transitions you also might need to consider special namespace requirements. These are addressed in Chapter 14, "Transitioning from Exchange Server 2003 and Exchange Server 2007."

The official Microsoft support statement for Exchange 2010 and SLD/Disjoint/Non-contiguous Namespaces can be found at *http://msexchangeteam.com/archive/2009/10/27/452969.aspx*.

Namespace Scenarios

When you implement your Exchange 2010 organization, you need to decide how your internal and external namespace will be defined. This is important because it affects the following areas:

- DNS configuration of your Exchange servers
- How your certificates are created and what names they include
- Client Access (Outlook Anywhere, Outlook Web App, POP3 and IMAP4, SMTP)

If you have multiple datacenters available where your Exchange 2010 servers are located, consider the following general advice for namespace planning:

- Plan your namespaces such that both datacenters can be active.
  - This still allows for incremental deployment.
  - You provide failover capabilities or can manually switch over a datacenter.
- Each datacenter needs the following namespaces, depending on your client connectivity capabilities:
  - Outlook Web App/OA/EWS/EAS namespace
  - POP3/IMAP4 namespace
  - RPC Client Access namespace
  - SMTP namespace
- Consider which datacenter will maintain the Autodiscover namespace.

To start planning your namespace, you need to consider the various locations of clients and servers and the physical connections they have to the Exchange servers. Typically the namespaces align with your DNS configuration.

You can choose from the following namespace-planning options:

- Consolidated data center
- Single namespace with proxy sites
- Single namespace with multiple sites
- Regional namespaces
- Multiple forests

Consolidated Data Center

This namespace scenario is the simplest one and includes a single namespace to access a single physical site where all the Exchange servers are hosted. The Contoso scenario described in Chapter 2 of this book is an example of a consolidated data center. This scenario has the following advantages:

- Only one or very few DNS records need to be managed.
- Only one or very few certificates are required for your Exchange organization.
- All users use the same URL to access the Exchange server.

This namespace scenario is configured by providing Internet access to the Client Access server by opening the relevant ports by a firewall or implementing an application layer firewall such as Forefront Threat Management Gateway in the perimeter network.

If you want to provide POP3/IMAP4, you need also to consider how the clients will send their messages using SMTP. To overcome this easily, you can configure the Hub Transport role on each Client Access server. Otherwise, you need to plan separately for message sending and message receiving namespaces.

Single Namespace with Proxy Sites

This model is based on the consolidated datacenter model but proxies the requests to the physical Mailbox server located at another site. One of the sites has one or more Internet-facing Client Access servers that proxy the requests.

This scenario has the following advantages:

- Only one or very few DNS records need to be managed.
- Only one or very few certificates are required for your Exchange organization
- All users use the same URL to access Exchange server.

The disadvantage of this model is that most users will access their mailboxes using proxying, thus accessing their data might be slower across latent WAN links.

To configure this namespace model, you need to configure the *ExternalURL* option of the Client Access server(s) at one site, and make sure that the *ExternalURL* settings on all the other

sites are configured to *$Null*. This configuration ensures that the Client Access server does not redirect the connection to the target Client Access server, but instead proxies it. Redirect means that the Client Access server forwards the connection to the target Client Access server; proxy means that the Client Access server contacts the target Client Access server and retrieves the data for the connection.

Single Namespace with Multiple Sites

This model uses a single namespace for an organization that has multiple sites. For example, the Litware scenario would be a possible candidate for this approach because the company has multiple physical sites and wants to use a single namespace. The two possible approaches to implementing a single namespace with multiple sites are with a Client Access server proxy site or an intelligent firewall:

- The Client Access server proxy site approach includes Client Access servers based in a separate Active Directory site that is used to proxy the traffic to the site where the user's mailbox is located. To configure this namespace model, you need to configure the *ExternalURL* option to the single namespace of the Client Access servers at all sites.
- The intelligent firewall approach uses an application-layer firewall such as Forefront TMG and decides during client authentication that the traffic is forwarded to the correct target site based on configured rules. To configure this namespace model, you need to configure the *ExternalURL* option to the single namespace of the Client Access servers at all sites.

This scenario has the following advantages:

- Only one or very few DNS records need to be managed.
- Only one or very few certificates are required for your Exchange organization.
- All users use the same URL to access Exchange server.

The disadvantage of this model is that you must either have an application-layer firewall that is capable of forwarding the traffic to the correct physical sites like Microsoft Forefront TMG or a Client Access server proxy site.

Regional Namespaces

This model uses one namespace for each region or site. The users will use their regional namespace to access their messages.

This scenario has the following advantages:

- The client traffic is automatically optimized based on the region or site level. (For example, if you implement a namespace based on a city, all users of that city will use the local access.)
- Performance and end-user experience are optimized.
- Failover is provided if the regional namespace is unavailable by using a different namespace (if the Mailbox server of the site is still available).

The disadvantage of this model is that you need to manage multiple DNS records as well as multiple certificates. Additionally you have multiple Internet entry points that require a firewall.

> ***NOTE*** ***The regional namespaces model is recommended if your topology includes multiple sites that have their own Internet connectivity.***

Multiple Forests

The multiple forest model uses one dedicated namespace for each forest. For example if Contoso and Litware merged, Contoso users would need to access mail.contoso.com and Litware users would use mail.litware.com to access their mailboxes. Client Access server proxy redirection between the two forests does not work, so if one forest is not available, no users would be able to access their messages.

In this model, every namespace that is implemented needs its own Internet access point, DNS record, and a certificate. Within each forest, use a regional namespace model to improve customer experience.

Disjoint Namespace

The disjoint namespace model is a special scenario for planning the namespace. You face this scenario when your primary DNS suffix on domain controllers or member servers in the domain is not the same as the DNS domain name of your Active Directory domain.

For example, you have a disjoint namespace when the Exchange Server that is part of the Litware.com domain has a primary DNS suffix of Contoso.com. This computer (as the primary DNS suffix that does not match the DNS domain name) is said to be disjoint.

You might require these namespaces to be different for several reasons. For example, if DNS management in your company is split between administrators who manage Active Directory and administrators who manage networks, you may need to have a topology with a disjoint namespace.

Microsoft Exchange 2010 has three supported scenarios for deploying Exchange in a domain that has a disjoint namespace:

- **Scenario 1** The primary DNS suffix of the domain controller is not the same as the DNS domain name. Computers that are members of the domain can be either disjoint or not disjoint.
- **Scenario 2** The Exchange servers in an Active Directory domain are disjoint, even though the domain controller is not disjoint.
- **Scenario 3** The NetBIOS domain name of the domain controller is not the same as the subdomain of the DNS domain name of that domain controller.

In Exchange 2010 you may need to configure the DNS suffix search list to include multiple DNS suffixes if you have a disjoint namespace.

In a disjoint namespace environment, you must configure the following:

- All disjoint domains need to be added to the *msds-allowedDNSSuffixes* attribute of your root domain.
- The DNS suffix search list must include all DNS suffixes, including the disjoint DNS suffixes.

As mentioned, it is required to configure every disjoint domain in the *msds-allowedDNSSuffixes* attribute of your root domain. For example, if you have the disjoint namespace contoso.com that you need to add to the Litware.com forest, configure the settings on the domain level using the Windows Server 2008 Administrative Tool ADSI Edit, as shown in Figure 3-7.

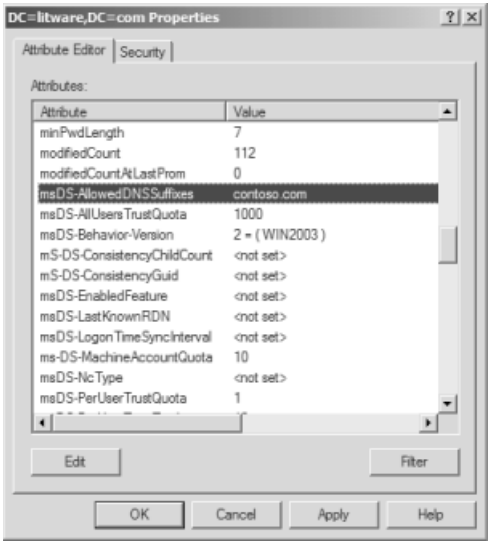

FIGURE 3-7 Configuring a disjoint namespace in the domain

Additionally, make sure that the DNS suffix search list contains all DNS namespaces that are deployed within your organization. To do this, you must configure the DNS search list for each computer in the domain that is disjoint. The list of namespaces should include not only the primary DNS suffix of the disjoint member computer and the DNS domain name, but also any additional namespaces for other servers the Exchange Servers may interoperate (such as the monitoring server).

For more information on Exchange 2010 and disjoint namespaces, see "Understanding Disjoint Namespace Scenarios" at *http://technet.microsoft.com/en-us/library/bb676377.aspx*.

NOTES FROM THE FIELD

A Disjoint Namespace Example

Carsten Allendoerfer
Head of System Services Group, Johannes Gutenberg-University Mainz, Germany

Our disjoint namespace consists of the Active Directory domain called uni-mainz.de (we have a single domain/single forest implementation) but all servers use the primary DNS suffix of zdv.uni-mainz.de. Only domain controllers use the correct suffix of the Active Directory domain.

We started to run Exchange 2010 in early beta phase and had quite a few problems caused by the disjoint namespace scenario. We still have one unsolved problem with the Active Directory Topology Discovery Service, which causes problems when the NetLogon service starts too early and no network connection is available. You can resolve this by using fixed IP addresses for the servers instead of DHCP. I assume this is a general problem in Windows Server 2008 R2 and not an Exchange 2010 problem.

Sometimes applications from Microsoft and other companies take for granted that the DNS suffixes are equal to the Active Directory Domain name. Take my advice: carefully test every application before implementing it. From what I've learned in the past 10 years running in a disjoint namespace scenario, I would never recommend it to anyone.

Single Label Domains

A single label domain (SLD) is basically a DNS domain name set equal to a NetBIOS domain name. It does not contain a suffix such as .com or .org and consists only of a single word, such as LITWARE or CONTOSO.

Before Active Directory, in Windows NT a single label domain was the basis so some companies continued to use an SLD in Active Directory. In Windows Server 2008 R2 you can no longer create SLDs—if you find an environment that still has SLDs, consider migrating to a normal namespace to prevent issues in the future.

Exchange Server 2010 supports SLDs; however, the Exchange product team does not recommend this configuration because future versions of Exchange or third-party applications might cause issues in this scenario. For that reason, you should move your organization to a normal namespace scenario.

Non-contiguous Namespaces

A non-contiguous namespace (sometimes referred as a discontiguous namespace) is a namespace where an Active Directory forest includes multiple domain trees of different names. Thus the forest is not defined hierarchically. A forest can have one or more domain trees, and these trees are defined by the DNS names. For example, contoso.com would be a domain tree in the Litware.com forest.

In Windows Server 2008 R2, you can configure multiple domain trees by using the Advanced Mode Installation in the Active Directory Domain Services Wizard (dcpromo.exe).

> ***IMPORTANT*** ***If you have similar tree names (such as litware.com and litware.de), be sure to choose different NetBIOS domain names for their respective domains. If you select the same NetBIOS names for both trees, the configuration is not supported. The general rule is that each domain must still register a unique legacy NetBIOS domain name.***

If your organization has a non-contiguous namespace scenario, DNS must be configured so that every Exchange server is able to resolve all domain names in the environment. You are also required to configure *msDS-AllowedDNSSuffixes* within the Active Directory environment for all namespaces used in the forest. For more information on how to configure *msDS-AllowedDNSSuffixes* refer to the section "Disjoint Namespace" earlier in the chapter.

Planning Certificates

This section is about certificates that are generally used by Exchange 2010 to secure communication between the servers and between the client and the servers. It explains the basics about the certificates, and then dives into the details of what types of certificates are available and what you need to know to plan the names you put into your certificates accordingly.

About Digital Certificates

A digital certificate basically is an electronic representation of users, computers, or other devices or services. A digital certificate consists of a private and a public key pair.

The private key is stored only on a computer, device, or possibly a digital ID card. In many companies these keys are kept under the same security level as the user password for Windows. The public key is used to encrypt data for you and is required by everybody that wants to security communicate with you.

A digital certificate is always issued by a CA with a private and public key pair. The process is similar to applying for a passport at your local governmental office. The governmental

office issues your passport, like the CA, and the passport you receive is like the digital certificate.

You use digital certificates in relation to sending and receiving e-mail in two areas:

- **Data encryption** Make sure the data you transmit cannot be decoded somewhere between the sender and the receiver.
- **Digital Signature** The receiver can verify that the data received was originated by you.

Types of Certificates

There are three types of certificates based on the authority that issues the certificate: self-signed certificates, Windows public key infrastructure (PKI)–generated certificates, and third-party certificates. Table 3-5 provides an overview of these types of certificates and their uses.

TABLE 3-5 Types of Digital Certificates

CERTIFICATE TYPE	DESCRIPTION
Self-signed certificates	When Exchange 2010 is installed, a new certificate is generated automatically if no computer certificate is available. This certificate is used by default to encrypt all communication inside and outside the Exchange organization. If you access your OWA using a Web browser, you need to confirm that the server's certificate is correct because you do not trust this certificate by default. Self-signed means that the computer itself acted as a CA and signed its own certificate.
Windows PKI–generated certificates	These certificates are issued by a Windows CA (such as Windows Server 2008 R2's Active Directory Certificate Service) and you can request them at no extra cost and install them immediately. Normally, they are not trusted publicly, so you need to make sure that the root certificate is imported at every server, client, and device that does not belong to your Active Directory. In your Active Directory forest, the information is distributed automatically.
Third-party certificates	This type of certificate is automatically trusted within the Internet and can be purchased by a third-party CA such as VeriSign. It is the easiest and least time-consuming way to implement certificates, but you need to buy them. Thus, you probably won't have an official certificate for every Exchange server in your environment.

You cannot use self-signed certificates for mutual TLS or Domain Security communication to and from the Internet in Exchange 2010—only Windows PKI–generated certificates or third-party certificates are supported there.

IMPORTANT ***If you decided to use Windows PKI–generated certificates for Internet messaging, you have to make sure that your partners' servers trust your root CA (by importing your root certificate).***

Working with Certificates in Exchange 2010

Exchange uses certificates to communicate securely between the different server roles. By default each Exchange server uses either the certificate issued by the domain or issues its own self-signed certificate and uses this one for communication. If you do not require secure communication to the Internet, a self-signed certificate works without issue. However, if you want to consider a secure Exchange 2010 implementation, some server roles require an independent certificate if they are communicating with the client. Table 3-6 provides an overview of which Exchange Server roles require which certificate for which purpose.

TABLE 3-6 Server Roles and Certificates Requirement

SERVER ROLE	PROTOCOL(S) THAT REQUIRES CERTIFICATE	TYPES OF CERTIFICATES REQUIRED
Hub Transport	SMTP over TLS	Windows PKI or third-party for external, self-signed for internal mail flow
Client Access Server	Outlook Web App (OWA) Exchange Web Services (EWS) Outlook Anywhere ActiveSync POP3 IMAP4 Autodiscover	Windows PKI or third-party
Edge Transport	SMTP over TLS	Windows PKI or third-party
Mailbox Server	—	Any certificate
Unified Messaging	SIP over TLS	Windows PKI or self-signed

SERVER ROLE	PROTOCOL(S) THAT REQUIRES CERTIFICATE	TYPES OF CERTIFICATES REQUIRED
Application Layer Firewall/Reverse Proxy[1]	SMTP over TLS Outlook Web App (OWA) Exchange Web Services (EWS) Outlook Anywhere ActiveSync POP3 IMAP4 Autodiscover	Windows PKI or third-party

[1] An application-layer firewall such as Microsoft TMG can be used to proxy traffic between the perimeter and internal network. For that reason it can proxy all Exchange protocols but does not require it.

Exchange 2010 certificates need to have a certain format to work correctly with the TLS protocol. Because the Edge Transport servers might have multiple domain names or service connection points (SCPs), you have two options:

- Use a single certificate on your server(s) with Subject Alternative Names (SAN) support, also known as Unified Communications Certificates.
- Use individual certificates.

NOTE ***Microsoft recommends using a SAN certification because it's simpler to administer on the servers. Unfortunately, it is also more expensive than a normal certificate if purchased from a third-party CA.***

Thus when considering certificates in Exchange 2010, you need to answer two key questions:

- Where do you want to place certificates? Do you want to use one certificate per server or a single certificate for all your servers? If you want to use a single certificate for all your servers, make sure you distinguish between internal and external servers. If you use an application-layer firewall in a perimeter network, consider implementing a separate certificate for it.
- What SAN names should the certificates have? If you use one certificate for all servers, you need to consider all SAN names that you want to add.

To plan for all the domains or host names that should be included in the certificate, Table 3-7 should provide you with a basic understanding of what is required.

TABLE 3-7 Creating Certificates for Exchange Roles

SERVER ROLE	CERTIFICATE NAME REQUIREMENTS	EXAMPLES (FOR LITWARE SCENARIO)
Hub or Edge Transport	■ Hub Transport server's FQDN ■ Domain name(s) used for TLS	▯ Berlin-HT01.Litware.com n Litware.com
Client Access Server	■ Client Access server's FQDN (for internal client proxy for ECP, OCS 2007 R2) ■ Service FQDN for OWA, ActiveSync, POP3, IMAP, and so on	▯ Fresno-CA01.Litware.com n OWA.Litware.com n Mail.Litware.com n Webmail.litware.com n Imap.litware.com n Autodiscover.litware.com
Unified Messaging	■ UM server's FQDN (for SIP over TLS)	▯ Berlin-UM01.Litware.com

NOTE ***Short names or NetBIOS names are no longer required in a certificate in Exchange 2010. However, if your users are using short names in the browser to access OWA, or if you implement the certificate on legacy Exchange Servers, you should also add the short names to the certificates.***

Consider carefully the following best practices regarding creating certificates for Exchange 2010 :

- Use SAN certificates that can cover multiple host names.
- Minimize the number of certificates. If your company's security policy permits, use only one certificate for all Client Access, Hub/Edge servers, and the application-layer firewall.
- Because Office Communication Server (OCS) 2007 requires the server name in the certificate principal name (PN) and a key of <=1024 bit keys, it is recommended that you use an additional certificate if OCS is required.
- Put only the names you need in the certificate.
- Don't list computer host names on certificate host name lists, if at all possible.
- If using a certificate for each datacenter, ensure that the Certificate Principal Name is the same on all certificates. Otherwise, Outlook Anywhere will not connect in a site failover scenario.

Planning Exchange Server 2010 Placement

This section explains the planning aspects you need to consider to plan Exchange 2010 server placement at your company's locations. It starts with considering the domain controllers because they play an important role when Exchange Server 2010 is installed and then discusses what Exchange roles should be planned at which site.

Domain Controller and Global Catalog Placement

In planning your Exchange 2010 server placement, never forget to include domain controller or global catalog servers. Because Exchange 2010 requires good communications with Active Directory to access its configuration, the logical starting point is to consider domain controllers and global catalog servers that already exist and verify whether additional servers are needed. This is especially important because Exchange 2010 won't start without communicating to a global catalog server. For that reason it is important that you consider the following areas in your planning:

- At least one global catalog (which obviously is also a domain controller) must be available in the same Active Directory site where you plan to install Exchange 2010.
- For redundancy reasons, it's always good to have at least two global catalog servers available in an Active Directory site where Exchange 2010 will be installed.
- Using 64-bit domain controllers increases the directory service performance significantly, although 32-bit domain controllers are still supported.
- As in Exchange 2007, the recommended 4:1 ratio of Exchange processor cores to global catalog processor cores (32-bit) still applies for Exchange 2010. If you have 64-bit global catalogs with enough memory to house the ntds.dit, the ratio increases to 8:1. For example, if you have two Exchange servers with four processor cores per server, you should have at least one global catalog processor core for Exchange 2010 server requests. As you cannot dedicate GCs to Exchange as the servers will deal with requests from other applications, you need to make sure to deploy sufficient GC capacity to deal with Exchange and the other applications.
- If you're planning to host Exchange servers for multiple domains at a single Active Directory site you must include domain controllers from each domain you host resources for. This ensures that a domain controller of their own domain is always referred to your Outlook clients.

IMPORTANT ***Exchange Server 2010 does not support Windows Server 2008 Read-Only-Domain Controllers (RODCs) or Read-Only Global Catalogs (ROGCs) existing in the same Active Directory site. If you are using RODCs or ROGCs, you cannot install Exchange 2010 in those Active Directory sites—you need to create separate ones for Exchange.***

Using Exchange Server 2010 on Member Servers or Domain Controllers

You must also consider on which Windows 2008 server role you want to install Exchange Server 2010: member servers or domain controllers. Even though Microsoft supports the installation of Exchange Server 2010 on domain controllers, we strongly advise against it. This is because you need to be a local administrator to manage an Exchange Server 2010 server, and local administrators will automatically receive Admin permissions on all of your domain controllers.

In some circumstances, such as branch-office situations, you may not have a choice because hardware is spare or budget is limited. We've seen situations where a single piece of hardware held everything: domain controller, Exchange Server, and file and print services. However, avoid that if possible. You can use virtualization to separate domain controllers and Exchange servers easily when using virtualization such as Windows Server 2008 R2 Hyper- and still run everything on a physical computer.

> ***NOTE*** ***As a protective feature, Dcpromo—the command to promote a Windows Server 2008 member server to a domain controller—cannot be run again after you have installed Exchange Server 2010 on a Windows 2008 member server. After Exchange Server 2010 is installed, changing the role from a member server to a domain controller or vice versa is not a Microsoft-supported scenario.***

Exchange Server Role Placement

To manage Exchange Server 2010 in a more natural way, server roles were implemented. These roles enable administrators to easily choose which features should be installed on an Exchange server. They provide the following advantages over the architecture used in Exchange versions before Exchange 2007:

- They reduce attack surface because only required roles are installed.
- They allow you to install the servers for their intended role only.
- They provide more possibilities for scalability and reliability.
- They lower complexity to reduce system outages.

In Exchange Server 2010 you can choose from five server roles: Mailbox server, Hub Transport server, Client Access, Unified Messaging server, and Edge Transport server. Figure 3-8 provides an overview of all Exchange Server 2010 roles, their functionality, and their connections.

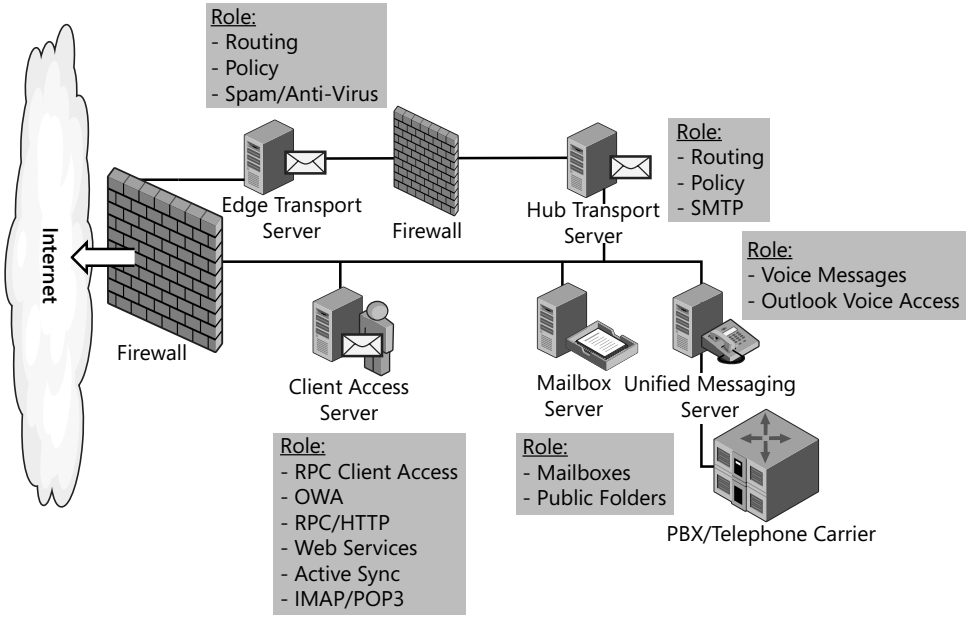

FIGURE 3-8 Exchange Server 2010 roles

As you can see in the figure, you must follow certain rules to develop a plan of which roles you place at which Active Directory sites. Table 3-8 provides an overview as well as the main planning aspects for each role. More details about the Exchange Server 2010 roles are covered in later chapters of this book.

TABLE 3-8 Exchange Server 2010 Roles and Planning Aspects

SERVER ROLE	DESCRIPTION	PLANNING ASPECT
Mailbox Server	Hosts your mailboxes as well as public folder databases.	You must plan to position Exchange servers at the Active Directory sites where most of the users are located or depending on your IT strategy in key regional datacenters. Because your users no longer directly connect to the Mailbox server in Exchange 2010 (only for public folder access), you need to have at least the Client Access and Hub Transport server roles available in the same Active Directory site.

SERVER ROLE	DESCRIPTION	PLANNING ASPECT
Client Access Server	This role hosts RPC Client Access; availability service and Autodiscover for Outlook 2007 or later; Exchange ActiveSync; client protocols such as MAPI, POP3, IMAP4, Outlook Web App (OWA), Outlook Anywhere and Exchange Web Services (EWS). All client traffic flows through the Client Access Server role in Exchange 2010, including MAPI connections from Outlook clients that are handled by the RPC Client Access layer.	Required in every Active Directory site where a Mailbox server is installed. The recommendation is three Client Access server processor cores per four Mailbox processor cores.
Hub Transport Server	Manages all internal message routing within the Exchange organization and hosts transport rules that can be applied to messages. It also accepts all SMTP types of messages even if they come from the user.	Required in every Active Directory site where a Mailbox server is installed. In this Active Directory site a global catalog must be available. The rule of thumb regarding sizing (with antivirus) is one Hub Transport processor core per five Mailbox server processor cores. For redundancy reasons you should have at least two Hub Transport servers in larger or critical Active Directory sites.
Edge Transport Server	Acts as a smart host and SMTP relay in your perimeter network and handles all Internet-facing mail flow. Provides anti-spam and antivirus functionality. Provides address rewriting and Transport rules to protect the internal network.	Depending on the size of your organization, you should plan at least two Edge Transport servers to provide redundancy in case of problems.
Unified Messaging Server	Connects Exchange with your telephone system or private branch exchange (PBX) to provide voice access to your mailbox.	Supports a maximum of 100 concurrent calls per server. The planning aspect should include the number of users and how they use Unified Messaging. A single Unified Messaging server can host approximately 3,000 users. You need at least one Hub Transport server available in the same site.

Exchange Server 2010 server roles can coexist on a single Exchange computer with a few rules to consider:

- The Mailbox role, Hub Transport role, Client Access role, and Unified Messaging role can coexist on a server.
- On a Mailbox server that is member of a DAG, you cannot use Windows Network Load Balancing (NLB).
- Edge Transport cannot be shared with any other server role.

In a smaller organization you will probably end up having a server that hosts multiple roles, mainly the Mailbox, Client Access, and Hub Transport roles. The larger the organization, the more dedicated those server roles will get.

NOTES FROM THE FIELD

Planning Exchange Server Roles and Placement

Joe Cirillo
Senior Engineer and Architect, Horizons Consulting, US/Central Region

When I prepare to install a new Exchange messaging system or integrate with an earlier version of Exchange I always take the same approach. Whether I am installing one physical server with multiple roles or installing individual roles on dedicated hardware I always begin by first installing the Client Access Server role.

Because the Client Access server is used by every mail client, you can fully prepare the Client Access server for client access before installing the Mailbox server role, thus ensuring that once the Mailbox server is online, users will be able to successfully connect to their mailboxes based on your preconfigured settings.

When I prepare the Client Access server, I like to do the following:

- ***If there will be a high volume of content conversion, move the %TMP% folder to a dedicated set of drive spindles for improved performance.***
- ***Replace the self-signed certificate with a public certificate, typically a SAN certificate.***
- ***Create the necessary DNS records to support the services to be used.***
- ***Configure Exchange services for the Autodiscover Service.***
- ***Enable Outlook Anywhere.***
- ***Create a Client Access Array using EMS (new in Exchange 2010).***
- ***Configure the hardware load balancer.***

After I have fully configured and tested access and functionality of the Client Access server, I install the Hub Transport role. This way I can ensure that message transport is working properly. Again, having this role installed and configured before the Mailbox server role will ensure that once I provision a mailbox to a user, that user will be able to successfully send and receive e-mail messages.

When I prepare the Hub Transport server, I like to do the following:

- *Confirm that message tracking is enabled.*
- *Change the location of the Queue Database and Queue Database Transaction Logs for improved performance.*
- *If necessary, modify the organizational level send and receive message size limits.*
- *If necessary, modify the default receive connector to accept anonymous connections.*
- *If necessary, modify the default receive connector's message size limit.*
- *If necessary, create an additional receive connector for applications that require relay access.*
- *Configure remote domains.*
- *Configure accepted domains.*
- *Configure e-mail address policies.*
- *Create send connectors.*
- *If necessary, modify send connector message size limits.*
- *Configure the hardware load balancer.*

Once the Hub and Client Access servers are in place and properly configured I can safely install the Mailbox server role, comfortable in the knowledge that my users will be able to successfully connect to their mailboxes and send and receive e-mail.

When I prepare the Mailbox server, I like to do the following:

- *Change the file location of the default database and logs.*
- *Create additional databases.*
- *If necessary, modify mailbox database settings (storage limits, deleted item retention, and so on).*
- *If necessary, create a Public Folder database.*
- *Configure the mailbox limits cache (see http://technet.microsoft.com/en-us/library/bb684892(EXCHG.80).aspx).*
- *Publish offline address books to the required distribution mechanisms (public folders and/or Exchange File Distribution on a Client Access server).*

- ***Control Outlook Access to Exchange based on client version.***
- ***Configure any settings required to enable high availability such as joining a server to a DAG.***

For designs that call for a distributed messaging infrastructure where Exchange Servers will exist in multiple locations I still follow the basic guideline on installation order of Client Access, then Hub, and then Mailbox. Again, this ensures that all my services are working (even between sites) prior to provisioning a mailbox on a Mailbox server.

Planning Network Port Requirements

When the first versions of Exchange came out, security was not a major consideration. Of course, security was of concern in 1996 but the level of Internet connectivity that systems have today, together with the threat posed by being connected to the Internet, make it quite different. Obviously, this has changed in recent years. Windows Firewall is now a main component of every Windows Server 2008 operating system. Windows Firewall basically filters inbound and outbound traffic based on firewall rules. Exchange Server 2010 creates the Windows Firewall rules to open the ports required during Exchange Setup—thus you no longer need to configure these settings manually.

Some companies want to put certain server roles into a perimeter network. A perimeter network is a network zone that is deployed between the Internet and a company's intranet as defense-in-depth and is protected by one or more firewall systems.

IMPORTANT ***When defining your firewall ports, always consider the concept of "less is more." The fewer ports you allow to open, the more secure your system will be.***

To provide an easy overview of the masses of ports, this section is organized according to the Exchange Server roles. The tables are sorted according the required ports so you can recognize which ports are used for which services or data paths.

For more information about firewall configuration see the Exchange Network Port Reference at *http://technet.microsoft.com/en-us/library/bb331973(printer).aspx*.

Mailbox Server

The Mailbox server role hosts the mailbox and public folder databases. Apart from public folders, the clients do not communicate directly with the Mailbox server but instead use the Client Access server for communication that then establishes the connection to the Mailbox server. For this reason it is not recommended to separate a Mailbox and Client Access server with a firewall.

Table 3-9 shows which ports are required for services or data paths to and from the Mailbox server role. It's important to understand that RPC traffic is always encrypted.

TABLE 3-9 Network Ports for Mailbox Role

DATA PATH	REQUIRED PORTS	ENCRYPTED BY DEFAULT?
Messaging application programming interface (MAPI) access, Availability Web service (Client Access to Mailbox server), Content indexing, Recipient Update Service RPC access (Exchange 2003 only), Microsoft Exchange Active Directory Topology Service access, Microsoft Exchange System Attendant service legacy access, Offline address book (OAB) accessing Active Directory, Recipient Update Service RPC access. RPC Endpoint mapper	135/TCP (RPC)	Yes
Mailbox Assistants, Admin remote access (remote registry), Microsoft Exchange System Attendant service legacy access (listen)	135/TCP (RPC)	No
Clustering or DAG to communicate between cluster nodes (status and activity)	135/TCP (RPC), 3343/UDP + randomly high TCP ports	No
DAG (Log shipping, seeding)	64327/TCP (customizable)	No
Active Directory access, DSAccess to Active Directory	389/TCP/UDP (LDAP), 3268/TCP (LDAP GC), 88/TCP/UDP (Kerberos), 53/TCP/UDP (DNS), 135/TCP (RPC Netlogon)	Yes
Microsoft Exchange System Attendant service legacy access to Active Directory, Recipient update to Active Directory	389/TCP/UDP (LDAP), 3268/TCP (LDAP GC), 88/TCP/UDP (Kerberos), 53/TCP/UDP (DNS), 135/TCP (RPC Netlogon)	Yes
Admin remote access (SMB/File)	445/TCP (SMB)	No

Hub and Edge Transport Servers

Exchange Server 2010 includes two roles that perform message transport functionality: Hub Transport server and Edge Transport server. You will need to consider this section when you are planning to implement an Edge Transport server role in your organization. The Hub Transport server takes care of messages that are routed within an organization; the Edge Transport server role routes messages inside and outside of the organization. For that reason Edge Transport servers are always placed in a perimeter network, whereas Hub Transport servers are always installed behind the network perimeter and belong to the corporate network.

Table 3-10 explains which ports are required for services or data paths to and from the Hub Transport and the Edge Transport server roles.

NOTE ***Because the Edge Transport server is designed to be located in the perimeter network, it is assumed that only the communication between Hub Transport and Edge Transport needs to be protected by firewalls. Of course, Edge Transport communication to the Internet also should be protected if the Edge Transport server is located in the perimeter network.***

TABLE 3-10 Network Ports for Hub and Edge Transport Servers

DATA PATH	REQUIRED PORTS	ENCRYPTED BY DEFAULT?
Hub Transport server to Hub Transport server, Hub Transport to Edge Transport and vice versa, Edge Transport to Edge Transport, Unified Messaging server to Hub Transport server and vice versa	25/TCP (TLS)	Yes
Active Directory access from Hub Transport server	389/TCP/UDP (LDAP), 3268/TCP (LDAP GC), 88/TCP/UDP (Kerberos), 53/TCP/UDP (DNS), 135/TCP (RPC Netlogon)	Yes
Microsoft Exchange EdgeSync service (from Hub to Edge)	50636/TCP (SSL)	Yes
Active Directory Rights Management Services (AD RMS) access from Hub Transport server	443/TCP (HTTPS)	Yes
Mailbox server to Hub Transport and vice versa	135/TCP (RPC)	Yes
Clients to Hub Transport server (using SMTP)	25/TCP (SMTP) or 587/TCP (SMTP)	Yes for TLS

As the table shows, encryption is the default in many situations. Hub Transport to Hub Transport is encrypted by default using the Exchange server's certificate. If no machine certificates are available for your Exchange server, the system will use self-signed certificates for encrypting the communication. The same is true for Hub Transport to Edge Transport communication.

If clients such as Windows Messaging directly communicate with the Hub Transport server, the only encryption possible is TLS over SMTP.

Client Access Server

The Client Access Server role manages all client requests and communicates directly with the Mailbox role. Therefore it is best practice not to separate Mailbox and Client Access Server roles with a firewall but instead to keep them in the same network. Remember, Microsoft doesn't support a firewall being placed between Client Access and Mailbox servers.

Table 3-11 describes which ports are required for services or data paths to and from the Client Access Server role.

TABLE 3-11 Network Ports for Client Access Servers

DATA PATH	REQUIRED PORTS	ENCRYPTED BY DEFAULT?
Active Directory access	389/TCP/UDP (LDAP), 3268/TCP (LDAP GC), 88/TCP/UDP (Kerberos), 53/TCP/UDP (DNS), 135/TCP (RPC Netlogon)	Yes
Autodiscover service, Availability service, Outlook Web App (OWA), Outlook Anywhere, Exchange ActiveSync, Client Access server to Client Access server for Exchange ActiveSync and OWA, Outlook Accessing Offline Address Book (OAB)	80/TCP or 443/TCP (SSL)	Yes
Client Access server to Client Access server for Exchange Web Services (EWS)	443/TCP (SSL)	Yes
POP3	110/TCP (TLS) or 995/TCP (SSL)	Yes (TLS/SSL)
IMAP4	143/TCP (TLS) or 993/TCP (SSL)	Yes (TLS/SSL)
Client Access server to Unified Messaging server	5060/TCP, 5061/TCP, 5062/TCP + a dynamic port	Yes
Client Access server to Exchange Server 2010 Mailbox server	135/TCP (RPC) + dynamic ports	Yes

DATA PATH	REQUIRED PORTS	ENCRYPTED BY DEFAULT?
Client Access server to a Mailbox server that is running Exchange Server 2003 or before	80/TCP, 443/TCP (SSL)	No
Client Access server to Client Access server (POP3)	995 (SSL)	Yes
Client Access server to Client Access server (IMAP4)	993 (SSL)	Yes
Remote Powershell to Client Access Server	80/TCP, 443/TCP (SSL)	Yes

When a Client Access server proxies POP3 requests to another Client Access server, the communication occurs over port 995/TCP, regardless of whether the connecting client uses POP3 and requests TLS or connects on port 995/TCP using SSL. The same applies to IMAP4 connections where Client Access Server always uses port 993/TCP to proxy requests.

NOTE ***When your Exchange 2010 Client Access server is communicating with an Exchange 2003 server, it is a best practice to use Kerberos authentication (disable NTLM and Basic authentication) and configure OWA to use forms-based authentication.***

Unified Messaging Server

The Unified Messaging server role is used to play voice messages to users using a IP gateway or a IP PBX (Private Branch eXchange). This server role communicates to all other server roles and is always placed in the organization's internal network.

Table 3-12 explains which ports are required for services or data paths to and from the Unified Messaging server role.

TABLE 3-12 Network Ports for Unified Messaging Servers

DATA PATH	REQUIRED PORTS	ENCRYPTED BY DEFAULT?
Active Directory access	389/TCP/UDP (LDAP), 3268/TCP (LDAP GC), 88/TCP/UDP (Kerberos), 53/TCP/UDP (DNS), 135/TCP (RPC Netlogon)	Yes
Unified Messaging to Mailbox server	135/TCP (RPC)	Yes
Unified Messaging to Hub Transport server	25/TCP (TLS)	Yes

DATA PATH	REQUIRED PORTS	ENCRYPTED BY DEFAULT?
Unified Messaging to Client Access server	5075/TCP, 5076/TCP, 5077/TCP	Yes
Unified Messaging to Client Access server (Play on Phone)	135/TCP (RPC)	Yes
Unified Messaging to private branch exchange (PBX)	5060/TCP, 5065/TCP, 5067/TCP (unsecured) or 5061/TCP, 5066/TCP, 5068/TCP (secured) and a dynamic TCP and UDP port	No
Unified Messaging Web Service	80/TCP, 443/TCP (SSL)	Yes

International Considerations

You need to consider certain areas when implementing Exchange 2010 in a global, heterogeneous, or multi-language environment. This section considers the most important factors of a global implementation: the language, time and message format, and message text encoding factors.

Multiple Language Support for Exchange

An important factor in international implementations to consider is the language for Exchange that should be installed. By default, Exchange Server 2010 only includes the English language for Exchange, but optionally you also can install additional language bundles.

NOTE *If you install from the Exchange 2010 DVD, most of the language packs are automatically included.*

Exchange 2010 comes with two different language bundles:

- **Language Pack for Exchange** You need this if you want to provide a localized version of the Exchange management tools (EMC and ECP) and OWA prompts for a specific language. A language pack includes the names of the default folders, user interface and layout, translated help text, and so on.
- **Unified Messaging Language Packs** You need these when you want to provide the Exchange Server 2010 Unified Messaging feature for a specific language.

Language Packs for Exchange

Depending on the Outlook Web App (OWA) languages you want to support, you need to install the respective Language Pack for Exchange. This provides localized messages for the users. It provides, for example, OWA in the local language on Client Access servers. On Mailbox servers it provides the default folder names in that language. On Hub Transport servers it provides key strings such as "Read", "Not Read", or "Undeliverable" in the local language.

The following are some general recommendations when using language packs:

- Always consider applying language packs for Exchange to all Exchange roles of that specific site. This is to prevent a mix of non-English and English strings in OWA and Outlook for users who set their language to non-English.
- You should deploy the language packs starting with your Mailbox servers.
- After installing the Exchange language packs, restart the computer to complete the installation of the language packs.
- If no Exchange language pack is deployed, English will be the only language available in Exchange and OWA, regardless of the operating system language.

You can find an overview of available language packs for Exchange 2010 at *http://technet.microsoft.com/en-us/library/dd298152.aspx*.

If you want to install the full language pack bundle for Exchange, you can download it at *http://go.microsoft.com/fwlink/?LinkId=147077*.

> ***NOTE*** ***To add a language pack for Exchange after you've installed Exchange, you need to run Setup from your CD, not by going through Control Panel. Basically you add the language pack as though you were installing a new Exchange server.***

What language packs are installed on an Exchange server? On a Client Access server that's easy to answer: You can see the language packs when logging on to OWA. On the Regional tab, under Settings, select the Language drop-down menu.

On Hub Transport or Mailbox servers, this is a bit more complicated. You have to use Regedit.exe and look into the \HKLM\Software\Microsoft\ExchangeServer\v14\Language Packs key to see what languages are installed on that specific Exchange server, as shown in Figure 3-9.

To remove a language you need to remove the entire key entry for that specific language found under HKLM\Software\Microsoft\ExchangeServer\v14\Language Packs\Client.

The language codes follow the ISO 639-1 codes (as described at *http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes*) except where more specific languages (such as zh-Hant) or specific cultures (pt-pt, for example) have been added. Be aware that removing languages directly from the Registry may cause issues in future versions of Exchange.

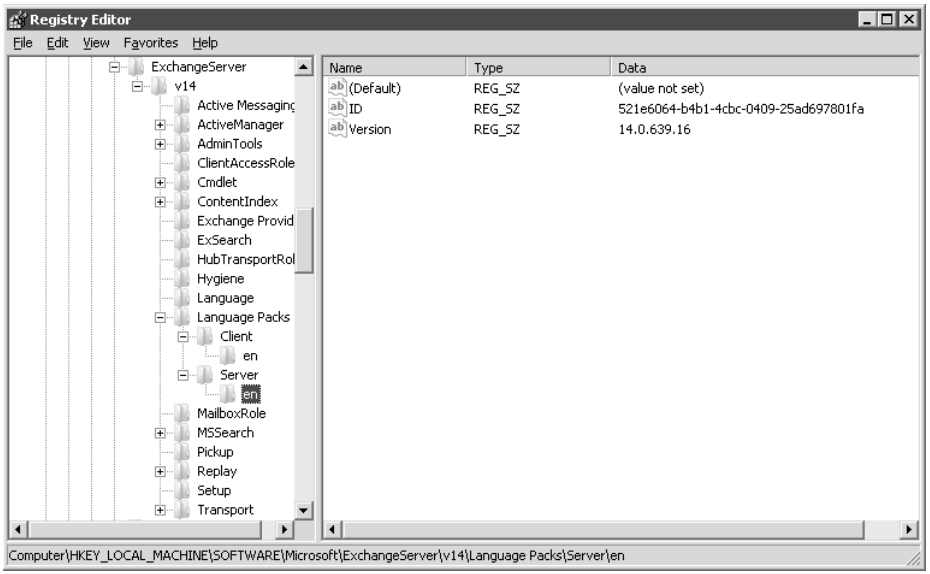

FIGURE 3-9 Identifying installed language packs in the Registry

Unified Messaging Language Packs

Unified Messaging Language Packs contain prerecorded prompts, grammar files, text to speech (TTS) data, Automatic Speech Recognition (ASR) files, and Voice Mail Preview capabilities for a specific language. They are only available for the Exchange 2010 Unified Messaging role and thus should not be installed on other server roles.

Install the same UM language packs to all Exchange UM server roles located in the same site. This will automatically provide the same language capabilities to all your users.

Because the UM language packs are continuously enhanced, you can download the latest language packs at *http://www.microsoft.com/downloads/details.aspx?FamilyID=3fdf49db-cb84-4dfe-8b8b-b30178b1a514&displaylang=en*.

Time, Time Zone, and Daylight Saving

Time zone settings on Exchange Server 2010 computers are similarly crucial to those on domain controllers in your Windows environment. If the servers run out of time synchronization, you will receive errors because Exchange assumes it is no longer working correctly. The EMS uses Kerberos when users authenticate themselves when they create a new Remote Windows PowerShell session. If Kerberos doesn't work, users won't be able to authenticate and the session cannot be created. If the time is not set correctly, EMS will fail to work. Every message is time stamped, so if message servers stamp the wrong time, this will screw up operations like message tracking.

Time is stored internally by Exchange using the Coordinated Universal Time (UTC) time zone to prevent issues from time conversions. All log files include the time in UTC format, which is sometimes confusing.

It is important to make sure you synchronize your time accordingly. Exchange Server 2010 automatically synchronizes the time within the domain if the server is a member of the domain. You therefore should configure your forest to synchronize the time. Detailed steps can be found on the Configure the Windows Time Service on the Forest Root Domain Controller Web page at *http://technet.microsoft.com/en-us/library/cc778920(WS.10).aspx*.

With Exchange servers that are not part of the domain like Edge Transport servers, you need to take care. Best practice is to configure a time server for every Edge Transport server that automatically receives the current time using the Network Time Protocol (NTP) from a server in your organization or directly from the Internet. You can use the following command to configure a NTP server:

```
net time /setsntp: <ntp server>.
```

In a multi-forest environment, make sure that the NTP servers you configured in each forest synchronize the time between them or with the same source.

The Daylight Saving Time (DST) changes every year, so you are well advised to keep your Exchange Servers during this time updated. You only need to consider this when you operate multiple Exchange Servers in different time zones, or if you're planning to do so.

NOTE ***Always enable the clock as a system icon on your taskbar; it helps prevent time issues from happening. Enabling the clock will show you the current system time whenever you log on to the server. You can immediately correct it if there is an issue.***

Message Format and Encoding

Because binary files cannot be sent directly using SMTP, they need to be encoded into a different message format. This is because SMTP messages can only consist of characters with 7 bits (or ASCII printable), meaning that you need to translate all 8-bit characters into 7-bit characters to transfer them using SMTP. This process is called *message encoding*.

Exchange Server 2010 supports the following message encoding formats:

- Uuencode (or UNIX-to-UNIX encoding) is one of the oldest encoding formats that supports encoding binary data. Three bytes of the binary file (24 bit) are divided into four of six bytes, and these six-byte values are associated with printable ASCII code. Uuencode has been widely replaced by MIME, but you still might need it if you're communicating with native UNIX SMTP servers to overcome message conversation issues.
- MIME (or Multipurpose Internet Mail Extensions) is the most common encoding format used today on the Internet. MIME nowadays is not only used by SMTP messaging,

but also by protocols such as HTTP. With MIME it is possible to exchange information about the type of messages (the content type) between the sender and the recipient of the message. MIME also defines the art of coding (Content-Transfer-Encoding).

To encode non-text elements, Exchange 2010 uses by default MIME encoding. This coding of non-ASCII characters is based on quoted-printable (QP) coding the binary data, typically using Base64-coding. As mentioned in the "Mail Client Support" section that follows, some UNIX or Linux clients cannot understand this message format and have issues with special characters that are not part of ASCII, such as German vocal or Chinese character sets. In such situations you might need to adapt the encoding format to solve the problem.

In addition to the encoding format, Exchange Server and Outlook use the Transport Neutral Encapsulation Format (TNEF) as the file format for attachments in e-mail messages. Attachments in this format often contain files called winmail.dat or win.dat. This format allows Outlook users to use some advanced features, but message programs other than Outlook cannot use it and receive an attachment called winmail.dat.

Mail Client Support

This section describes supported client and browser versions for Exchange 2010 and provides a feature overview of Microsoft Office Outlook 2003, Outlook 2007, and Outlook 2010.

Microsoft Outlook/Entourage

Several client programs supporting Exchange 2010 are available. Outlook 2010 for Windows and Entourage 2008 for Mac OS provide the most features for Exchange 2010 because they are engineered by Microsoft to work together well. Thus they include features such as MailTips that maximize the use of Exchange functionality. Because they are available in different versions, the following list provides an overview of the supported clients for Exchange Server 2010:

- Microsoft Outlook 2003 or later on Windows including the latest Service Pack
- Microsoft Entourage 2008 SP2 EWS or later on Mac OS
- Microsoft Outlook on Mac OS (2010 release)

NOTE ***Because of the functional change in Exchange 2010 whereby Outlook no longer communicates directly to the Mailbox server, Outlook 2002 (which was part of Office XP) and earlier versions cause weird issues when connected to Exchange Server 2010. I personally tested Outlook 2002 with Exchange 2010 and some features were not working correctly; consider migrating any Outlook 2002 users before moving their mailboxes to Exchange 2010. Also, Microsoft does not support Outlook 2002 or earlier for Exchange 2010.***

Each version of Outlook supports different features with Exchange Server. To receive the most server-based features, you will need to use the latest Outlook Version: Outlook 2010. Table 3-13 provides feature guidance on Exchange 2003, Exchange 2007, and Exchange 2010 and which features are available in which version of Exchange Server.

TABLE 3-13 Outlook Feature Comparison by Exchange Server Version

	SERVER INDEPENDENT	EXCHANGE 2003 (SP2)	EXCHANGE 2007 (SP1+)	EXCHANGE 2010 (RTM)
E-mail	X	X	X	X
Push e-mail	—	X	X	X
Calendar (on server)	—	X	X	X
Calendar	X	X	X	X
Free/Busy information	—	X	X	X
Free/Busy details sharing	—	—	X	X
Scheduling assistant	—	—	X	X
Contacts (on server)	—	X	X	X
Contact sharing	—	—	—	X
Calendar sharing	—	—	—	X
Calendar publishing				X
Archive access	—	—	—	X
Orgizational hierarchy	—	X	X	X
GAL access	—	X	X	X
MailTips	—	—	—	X
Conversation view	—	—	—	X
Conversation actions (ignore/move always)	—	—	—	X
UM (Voice mail)	—	—	X	X
UM preview	—	—	—	X
Protected Voice Mail	—	—	—	X
Managed folders	—	—	X	X
Tasks (on Server)	—	X	X	X
Public folders	—	X	X	X
Notes (server stored)	—	X	X	X

	SERVER INDEPENDENT	EXCHANGE 2003 (SP2)	EXCHANGE 2007 (SP1+)	EXCHANGE 2010 (RTM)
IRM protected messages	—	—	X	X
Policy management (group policy)	X			
Offline address book	—	X	X	X
AutoDiscover	—	—	X	X
OOF	—	X	X	X
External/Internal OOF	—	—	X	X
OOF Scheduling	—	—	X	X
Voting Buttons	—	X	X	X
Search folders			X	X
Search			X	X
Favorites folders			X	X
Journal (on Server)	—			
RSS Feeds (on Server)	—		X	X
Custom forms				
Custom dictionaries				
Mail rules	—	X	X	X

NOTES FROM THE FIELD

Consider Outlook RPC encryption

Ross Smith IV
Senior Program Manager, Exchange Server Product Group, Microsoft Corporation

Because Exchange Server 2010 requires Outlook traffic to be RPC encrypted, you might run into issues if you already have Outlook 2003 or Outlook 2007 in place today. By default, Outlook 2003 and Outlook 2007 do not use RPC encryption, so you will need to enable it before they're able to connect to an Exchange Server 2010. For details on how to prepare for this situation, you can find more information at http://support.microsoft.com/kb/2006508.

Outlook Web App

Exchange Server 2010 also supports various browsers not only in Outlook Web App Light but also with the Outlook Web App Premium edition that also provides rich features to browser users. OWA Premium includes features such as drag-and-drop, Junk-Mail filter configuration or voicemail configuration options that are not available in OWA Light. For Outlook Web App, the following browsers are supported:

- **Outlook Web App Premium on Microsoft Windows Vista or later** Internet Explorer 7 or later, Firefox 3.0.1 or later, Google Chrome 3.0.195.27 or later
- **Outlook Web App Premium on Apple Mac OS X** Safari 3.1 or later, Firefox 3.0.1 or later
- **Outlook Web App Premium on Linux** Firefox 3.0.1 or later
- **Outlook Web App Light** Almost any other browser or operating system

NOTE ***Even though browsers that run on operating systems other than Windows support Outlook Web App Premium, remember that it still has some limitations. For example, if you want to use the S/MIME control provided by Exchange for digital signatures or message encryption in Outlook Web App, you need to run Internet Explorer and Windows.***

A full list of browsers that support Outlook Web App can be found at *http://help.outlook.com/en-us/140/bb899685.aspx*.

IMAP and POP3 Clients

Exchange Server 2010 also provides support for IMAP and POP3 protocols. Any IMAP4/POP3 client (such Outlook Express or Thunderbird) can be used. However, you need to consider that some native IMAP or POP3 clients such as MailX have problems with the Microsoft internal content-encoding. By default Exchange 2010 converts the content of message attachments to quoted-printable (QP) format. If you client has issues reading it you might need to use a different client. Most of the Windows or Mac OS IMAP/POP3 clients do not cause issues, but for some in the area of LINUX such as MailX you need to consider this.

Additional Resources

- Windows Server 2008 R2 Components: *http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=64a5cc28-f8a1-4b30-a4a2-455c65bda8d7*
- How to disable certain Internet Protocol version 6 (IPv6) components in Windows Vista, Windows 7 and Windows Server 2008: *http://support.microsoft.com/kb/929852/*
- IPv6 for Microsoft Windows FAQ: *http://go.microsoft.com/fwlink/?LinkId=147465*

- Outlook Anywhere Scalability with Outlook 2007, Outlook 2003, and Exchange 2007: *http://technet.microsoft.com/en-us/library/cc540453(EXCHG.80).aspx*
- Exchange Load Generator 2010 Beta (64 bit): *http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=cf464be7-7e52-48cd-b852-ccfc915b29ef*
- Sysinternals BgInfo Tool: *http://technet.microsoft.com/en-us/sysinternals/bb897557.aspx*
- Active Directory Logical Structure and Data Storage: *http://go.microsoft.com/fwlink/?LinkId=179859*
- Monitoring and Troubleshooting Active Directory Replication Using Repadmin: *http://www.microsoft.com/downloads/details.aspx?familyid=c6054092-ee1e-4b57-b175-5aabde591c5f&displaylang=en*
- The official Microsoft support statement for Exchange 2010 and SLD/Disjoint/Non-contiguous Namespaces: *http://msexchangeteam.com/archive/2009/10/27/452969.aspx*
- DNS Requirements for Installing Active Directory: *http://technet.microsoft.com/en-us/library/cc739159(WS.10).aspx*
- Language packs for Exchange 2010: *http://technet.microsoft.com/en-us/library/dd298152.aspx*
- Language pack bundle for Exchange 2010: *http://go.microsoft.com/fwlink/?LinkId=147077*
- Language codes as defined in ISO 639-1 codes: *http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes*
- Exchange Server 2010 UM Language Packs: *http://www.microsoft.com/downloads/details.aspx?FamilyID=3fdf49db-cb84-4dfe-8b8b-b30178b1a514&displaylang=en*
- Configure the Windows Time Service on the Forest Root Domain Controller: *http://technet.microsoft.com/en-us/library/cc778920(WS.10).aspx*
- Wikipedia information about Microsoft Office 2010 for Mac OS X: *http://en.wikipedia.org/wiki/Microsoft_Office_2010_for_Mac*
- Exchange 2010 Outlook Web App Supported Browsers: *http://help.outlook.com/en-us/140/bb899685.aspx*

PART II

Designing Exchange Server 2010

CHAPTER 11

Designing High Availability

High availability has become a requirement for deploying most enterprise messaging environments; however, not everyone takes the time to understand everything that it involves. Many administrators have been conditioned to think that high availability means the same thing as failover clustering, or that high availability is a feature than can be enabled. Although failover clustering and network load balancing are high-availability platforms, they do not provide high availability by themselves. It is important to understand that clustering is only one piece of high availability.

Rather than being a product feature, high availability is an achievement that requires strong management, testing, and change control processes. An organization cannot achieve high availability just by implementing a product feature. The most important requirement in achieving high availability is implementing a high-availability philosophy within the organization where administrators think, evaluate, collaborate, and then perform actions that are in harmony with that goal.

Achieving High Availability

A number of barriers stand in the way of achieving high availability. For example, a poor implementation of Exchange might be one where Exchange is installed on improperly sized servers and installed without following best practices. In this case it is possible to deploy an Exchange messaging environment over a short time period. This is easy to do quickly, but a lot of important details can be missed and availability will no doubt suffer.

By contrast, in a high-availability environment the messaging system deployment is well designed. The deployment plan will be based on the information presented in Chapter 2, "Exchange Deployment Projects." During the deployment project,

organizational messaging requirements are researched. The current messaging environment is examined for inadequacies and solutions are identified. Research into how best to deploy Exchange may go on for an extended period while consultants are brought in to help build a design. Vendors are also brought in to discuss how their products will work and how they can contribute to running a highly available system. Hardware is sized and tested to meet both business and technical requirements, such as service-level agreements (SLAs), recovery point objectives, and cost considerations. Hardware will be considered that has the defined level of fault-tolerant components such as redundant memory, drives, network connections, cooling fans, power supplies, and so on.

A high-availability environment will also incorporate a significant amount of design, planning, and testing. A high-availability environment will often, but not always, include additional features, such as failover clustering and load balancing, which are designed to decrease downtime by enabling rolling upgrades and allowing for a preplanned response to failures. The messaging client software and its potential configurations can also improve availability. For example, Outlook 2003 and later offers the Exchange Cached Mode configuration that allows users to create new messages, respond to existing mail in their Inboxes, and manage their calendars (among many other tasks) even if the connection is lost to the Exchange server. Cached Exchange Mode allows users to continue working locally even though the Exchange server might be down for a short time. When the connection to the Exchange server is restored, any changes made will be synchronized. In the end, all critical business systems must be analyzed to understand the cost incurred when they are unavailable. If downtime has a significant cost, the organization should take steps to minimize downtime. This is particularly true if the cost of downtime is greater than the cost of deploying a suitable highly available solution.

The opposite of availability is downtime, both planned and unplanned. Planned downtime is the result of scheduled events, such as maintenance. Unplanned downtime is the result of unscheduled events. Events that cause unplanned downtime can be minor, such as a faulty hardware driver or a processor failure, or major, such as an earthquake, fire, or flood.

Measuring Availability

Availability is usually expressed as the percentage of time that a service is available. As an example, a requirement for 99.9 percent availability over a one-year period of 24-hour days, 7 days a week allows for only 8.75 hours of downtime, as shown in Table 11-1. In complex environments, organizations specify availability targets for each service. When dealing with an Exchange messaging environment, availability goals may be tied to specific features such as Microsoft Outlook Web App, Simple Mail Transfer Protocol (SMTP) message delivery, and Outlook MAPI connectivity. These availability targets are then turned into SLAs that hold the group operating the messaging system accountable for meeting those targets. In some cases, if those targets are not met, the salaries and bonuses of the employees and managers in the responsible group can be affected. In some instances both planned and unplanned downtime affect the overall availability target; in other environments planned downtime is exempt from the availability target. Because successfully achieving high availability includes update management to mitigate potential downtime, some planned downtime is required.

TABLE 11-1 Permitted Downtime for Specific Availability Targets

AVAILABILITY TARGET	PERMITTED DOWNTIME ANNUALLY
99 percent	87 hours, 36 minutes
99.9 percent	8 hours, 46 minutes
99.99 percent	52 minutes, 34 seconds
99.999 percent	5 minutes, 15 seconds

This bit of background should not detract from the great features provided to help achieve high availability in Exchange 2010; rather, the purpose is to provide a frame of reference as the Exchange-specific high-availability features are discussed.

Exchange 2010 High-Availability Features

Exchange 2010 builds on the solid foundation set by Exchange 2007 with regard to high availability. Exchange 2007 introduced a number of new options for availability, including cluster continuous replication (CCR), standby continuous replication (SCR), single copy cluster (SCC), and local continuous replication (LCR). Exchange 2010 introduces the Database Availability Group (DAG), which combines the best functionality available in Exchange 2007. A DAG is a group of up to 16 Exchange 2010 Mailbox servers that can each maintain up to 100 databases. A database may have up to 16 copies of each database using continuous replication.

The DAG differs from Exchange Server 2007 SP1 in the following ways:

- With CCR, there can be only two highly available copies of the database within the cluster; within the DAG there can be up to 16 copies of each database.
- With SCR, the activation process required administrative intervention; within a DAG, failover between individual database copies can happen automatically.
- With SCC, a single shared copy of the database consumes less storage but provides no redundancy. Exchange Server 2010 has no configuration that replaces this functionality, although some third-party solutions may be able to provide similar functionality by using the Third Party Replication API.
- With LCR, a single-server configuration allows two copies of a database to reside on different storage connected to the same server. No configuration in Exchange Server 2010 replaces this functionality.

Exchange 2010 provides database-level failover within the DAG. A single database failure no longer affects all mailbox databases on a server. Database failover time has also been improved since Exchange 2007. The DAG also makes it easier to implement site failover because now the DAG handles both in-site and inter-site replication.

Exchange 2010 also has improved non-mailbox high availability. Transport servers now have a feature called *shadow redundancy*, which provides redundancy for in-transit messages.

Another improvement is online mailbox moves. In previous versions of Exchange, mailboxes are moved offline which requires users to disconnect their clients in order to complete the move. Since this process impacts the users, these mailbox moves are usually scheduled during maintenance windows. Only being able to move mailboxes at night and on the weekends during a migration project does not provide enough time to complete the migration. The online mailbox move feature allows mailboxes to be moved between databases asynchronously without taking the user offline. The users will be able to maintain their connection and work while their e-mail is being moved in the background. This reduces end-user downtime and allows mailbox migrations to be performed during business hours. Online mailbox moves help improve availability for end users. More information about Exchange 2010 high-availability planning can be found in the Planning for High Availability and Site Resilience topic at *http://technet.microsoft.com/en-us/library/dd638104.aspx*.

Availability Planning for Mailbox Servers

In addition to normal IT best practices and redundant hardware, the DAG is the primary high-availability option for Exchange 2010 Mailbox servers. A DAG is a collection of servers that provides continuous replication and availability for mailbox databases, as shown in Figure 11-1.

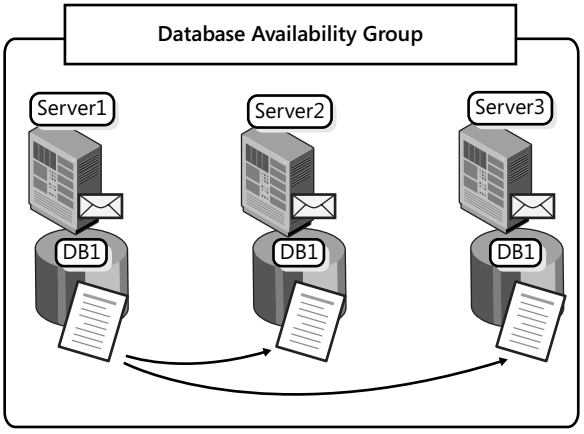

FIGURE 11-1 A Database Availability Group

Continuous replication creates a passive database copy on another Mailbox server in the DAG, and then uses asynchronous log shipping to maintain the copies.

The continuous replication process follows these steps:

1. The active transaction log is written and then closed.
2. The Microsoft Exchange Replication service replicates the closed log to servers hosting the passive database copies.

3. Because each copy of the database is identical, the Log Inspector will examine the transaction logs for the following:

   Verifies the physical integrity of the transaction log

   Verifies that the header generation is not higher than the highest generation for the current database copy

   Verifies the log header matches the generation of the file name

   Verifies the log file signature in the header matches the log file

   The transaction log is then placed in the defined transaction log directory.

4. The Information Store then validates the transaction log and then applies the logs to the database copy. The databases remain in sync.

A DAG also has the following characteristics:

- Requires the Windows failover clustering feature and uses an Enterprise version of Windows server (Windows Server 2008 or Windows Server 2008 R2), although the installation and configuration tasks occur with the Exchange Server management tools. Exchange Server does not use Windows failover clustering to handle database failover. Instead, it uses Active Manager to manage the failover process.
- Members must have the same operating system.
- You can add up to 16 servers to a single DAG and create up to 16 copies of a database. Up to 100 databases can be mounted as either a passive or active copy of the database on each server in the DAG.
- Uses an evolution of the continuous replication technology that is available in Exchange 2007.
- A DAG can be created after you install the Mailbox server. If a Mailbox server is hosting active mailbox databases, it can be added to a DAG later, it if meets the requirements.
- Allows you to move a single database between servers in the DAG without affecting other databases. Failover occurs per mailbox database, not for an entire server.
- Allows up to 16 copies of a single database on separate servers. A server can only host one copy of each database.
- Requires the database and transaction log copies for each database to be stored in the same path on all servers. For example, if you store Mailbox Database 1 in *D:\DB\Mailbox Database 1* on Dallas-MB01A, you must also store it in *D:\DB\Mailbox Database 1* on all other servers that host copies of Mailbox Database 1.
- Defines the boundary for replication, failovers, and switchovers—only servers in the DAG can host database copies. You cannot replicate database copies to Mailbox servers that are not in the same DAG.
- Does not require that all databases have the same number of copies. In a 16-node DAG, one database can have 16 copies, whereas other databases are neither redundant nor have varying number of copies.

In Exchange 2010 transaction log shipping occurs over TCP sockets as opposed to the file share (Server Message Block) used in Exchange 2007. You can view the current TCP port used for replication by running *Get-DatabaseAvailabilityGroup -Status | Format-List*. The default TCP port used for replication is 64327. This can be set using the *Set-DatabaseAvailabilityGroup -ReplicationPort* cmdlet. For this change to take effect, you need to create the Windows Firewall exceptions for the new TCP port and then restart the Microsoft Exchange Replication service on each node in the DAG. In the initial release of Exchange 2010, when you created a DAG using the EMC, the DAG was automatically configured to obtain an IP address from DHCP. To complete the configuration and assign a static IP address, you had to use the EMS. In SP1, the DAG can be configured with an IP address from within the EMC.

The target member notifies the member running the active copy of which transaction logs it expects to receive. The source member then responds by sending the required transaction log files. After the transaction logs are received from the source server, the files are placed in the target server's Inspector directory for processing. The logs are then inspected and verified for integrity and the header is inspected. After passing inspection, a transaction log is placed in the log directory on the target Mailbox server. If the transaction log does not pass inspection the target server will request it from the source up to three times before setting the mailbox database copy to Failed. When a database copy status is Failed, it will periodically attempt to copy the missing log files in order to return the database to a state of Healthy. The target Exchange server then plays the logs against the local copy of the database.

Before this transaction log shipping process can start, the database copy must first be seeded. Seeding is the process of creating a consistent database copy on a DAG member to act as a baseline that will be updated through continuous replication of the transaction log files. This can be accomplished using the following methods:

- **Automatic seeding** Automatic seeding occurs during the creation of a new database.
- **Manually copying the offline database** This method involves dismounting the database and copying the database file to the target server. If you do this, service will be interrupted while the database is dismounted.
- **Using the *Update-MailboxDatabaseCopy* cmdlet** You can use the *Update-MailboxDatabaseCopy* cmdlet in the EMS to seed a database copy.
- **Using the Update Database Copy Wizard** You can use the Update Database Copy Wizard within the EMC to seed a database copy.

Database failover occurs when the active database fails, and another copy of the database is activated on another server in the DAG. This can occur because of a number of failure types including: network, storage, and server hardware. If a entire DAG member fails, each of the active highly available databases will attempt to fail over to another configured DAG member. A switchover occurs when an administrator initiates moving an active database from one server to another.

NOTES FROM THE FIELD

Exchange High-Availability Improvements

Colin Lee
Technology Specialist, Unified Communications, Microsoft Corporation, Australia

In my opinion, Exchange 2007 is an evolutionary step in providing a complete high-availability solution with continuous replication. This provides capability for high availability, with CCR, and disaster recovery (DR), with SCR. Many customers I have worked with implemented this solution for high availability and DR with great success and were able to improve their SLA, or internal operational level agreement. As with all new technology there are areas for improvement and Microsoft continues to evolve continuous replication with Database Availability Group (DAG) in Exchange 2010. The introduction of DAGs in Exchange 2010 adds improvements that my customers requested as they were looking to improve SLAs even further. These requests are often around the active-passive nature of CCR and the ability to seamlessly failover if the disk (or raid group) the database resides on fails.

> ***NOTE*** ***In a CCR implementation with multiple storage groups an outage of a disk did not trigger a failover between the nodes and required some manual intervention to initiate a recovery, whether that be a restore from back for a DB or triggering a node failover.***

Exchange 2010 solves this issue with the capability that makes the database the unit of failover. It also helps address the perception that a passive node was sitting around idle. This is because up to 16 members can be put in a DAG, and all members can host active mailboxes. This is a powerful perception where upper management have a tendency to view "idle" servers as inefficiencies a company can do without. The following comments are from a customer that has migrated from Exchange 2007 with a CCR and SCR implementation to Exchange 2010 with a DAG that spans multiple datacenters.

"Moving to Exchange 2010 has allowed us to provide a more highly available solution to our hotels department whilst at the same time giving us (IT) increased simplicity in managing the infrastructure. We have extremely high confidence in our DAG with its ability for single database failovers as opposed to our old CCR and SCR setup. Implementing our DAG together with Datacentre Activation Coordination mode has also given us the confidence to increase our Disaster Recovery scope from a single storage group of critical mailboxes to the entire group, yet at the same time maintaining an uncomplicated recovery process."

Active Manager

Windows failover clustering is not used to replicate or manage the active database copies in a DAG; however it is used to store information for several pieces of volatile information about the DAG such as the state of active database copies. Exchange Server uses a Windows failover cluster, but there are no cluster groups for Exchange Server, and the cluster has no storage resources. In the Failover Cluster Management Console, you will see an empty cluster, as shown in Figure 11-2. Exchange 2010 does use the cluster API library functions for cluster network (heartbeating), node management, and cluster registry functions. Although Active Manager stores database information in the cluster database, it isn't accessed directly by any other components.

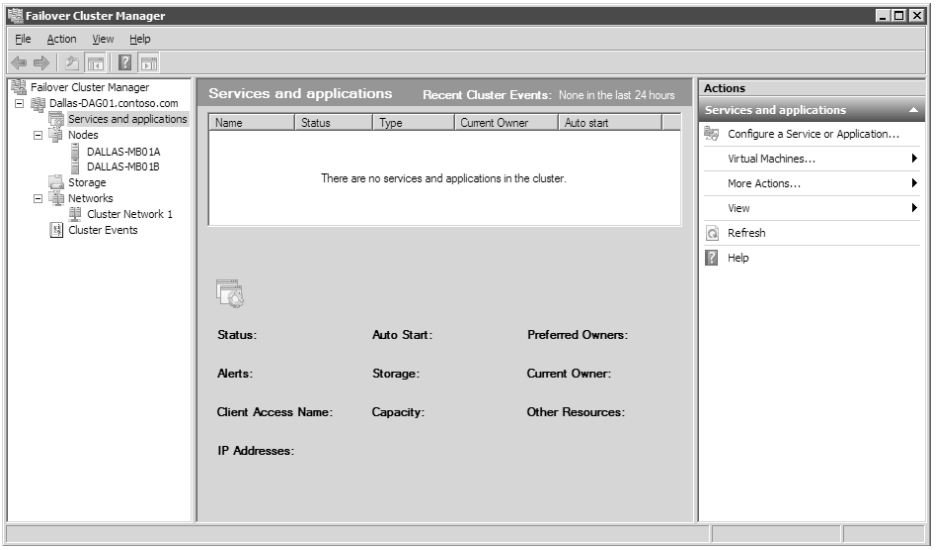

FIGURE 11-2 Windows Failover Cluster Management objects for a DAG

To manage mailbox database replication and activation Exchange 2010 includes a new component called *Active Manager,* which runs as a function of the Microsoft Exchange Replication service (MSExchangeRepl.exe). Active Manager replaces the resource model and failover management features integrated into Windows failover clustering that previous Exchange Server versions used. To simplify the architecture Active Manager runs on all Mailbox servers, even if the server is not part of a DAG.

Active Manager runs on all of the DAG members and runs as either the primary active manager (PAM) or a standby active manager (SAM). The PAM is the Active Manager in a DAG that controls which copies will be active and which will be passive. It is responsible for processing topology change notifications and reacting to server failures. The DAG member acting as the PAM is always the member that currently owns the default cluster group, as shown in Figure 11-3. In order to identify the PAM it is recommended to use *Get-DatabaseAvailabilityGroup <DAG Name> -Status | Format-List Name, PrimaryActiveManager*

rather than using the Windows Failover Clustering tools. If the server that owns the default cluster group fails, the PAM function automatically moves to the server that takes ownership of the default cluster group.

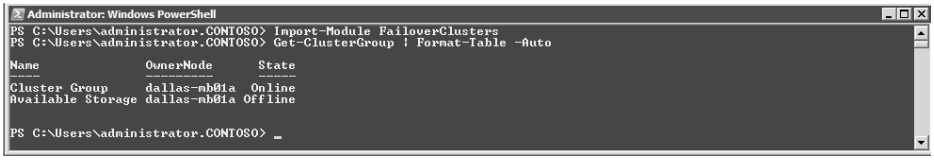

FIGURE 11-3 Identifying the DAG member that has the PAM function

If you are going to perform maintenance on the server that hosts the default cluster group, you must first manually move the PAM function to another server in the DAG, as shown in Figure 11-4, on a Windows Server 2008 R2 server. To do the same on Windows Server 2008 you run from a command prompt *cluster.exe group "Cluster Group" /MOVETO:Dallas-MB01B*.

FIGURE 11-4 Moving the PAM function

Far from having a passive role, the SAM function provides information about which server hosts the active copy of a mailbox database. The SAM detects local database and Information Store failures and reacts to them by requesting the PAM to initiate a failover when a copy is available. A SAM does not determine a failover target, nor does it update a database's location state for the PAM. Each SAM accesses the state of the active database copy in order to answer any request for where the active copy is from other Exchange components like the Hub Transport of Client Access servers. The PAM also performs the functions of the SAM role on the local system.

SP1 includes *StartDagServerMaintenance.ps1,* a script that you use to take a computer out of service. The script moves active databases off of the server and blocks databases from activating on that server. It will also ensure that all critical DAG support functionality is moved to another server, and blocked from moving back. The *StopDagServerMaintenance.ps1* script is then used to complete the operation and remove the blocks and allow databases to be activated on that node.

Adding Database Copies

Creating a database availability group is just the first step in making a database highly available. A database that exists on one of the DAG members must be set up with additional copies on other DAG members. Some databases may require more copies than others.

When creating a database copy, you can specify the following details:

- The name of the database you are copying.
- The name of the Mailbox server that will host the database copy.
- The amount of time (in minutes) to delay log replay. This sets how long to wait before the transaction logs are committed to the database copy. Setting the value for replay lag time to 0 disables the log replay delay.
- The amount of time (in minutes) for log truncation delay. This controls how long to wait before truncating committed transaction logs. Setting the value for truncation lag time to 0 disables the log truncation delay.
- An activation preference number. This represents the activation preference order of a database copy when multiple databases have the same copy queue length after a failure or outage of the active copy,
- The seed copy server. This server will be used to copy the seed database and content indexing information to the new copy. Although this is specified when creating a new database copy, replication always occurs from the active database to each of the copies.

Creating databases copies should be done according to a high-availability plan. A high-availability plan should be created that identifies the level of redundancy required for your environment. If JBOD (Just a Bunch of Disks) will be used to store database files, additional copies of the database should exist on other servers to sustain a disk failure.

You can add database copies using the *Add-MailboxDatabaseCopy* cmdlet or you can use the Add Mailbox Database Copy Wizard in the EMC.

Lagged Database Copies

One of the options available when configuring mailbox database copies is to configure a lag time of up to 14 days. This lag time is the time that the transaction logs will be held before being committed to the database copy. By delaying committing the logs to a database copy, you have the capability to recover the copy to a point in time using the copy rather than having to pull data from tape-based backup media.

Lagged database copies are deployed to protect from logical corruption. Database logical corruption and store logical corruption are the two types of logical corruption that can occur in the Exchange database.

If you use multiple database copies and Single Item Recovery, only the extremely rare catastrophic store logical corruption case remains unaddressed. In the following scenarios lagged database copies can be used to recover data:

- Recovering a deleted item from within 14 days outside the retention period
- Recovering to a point in time because of virus outbreak

You should deploy lagged copies to mitigate a specific risk and lagged copies are usually not needed if you are also deploying a third-party backup solution. Lagged copies should not

be treated as another high-availability database copy and should not be activated for the following reasons:

- You lose your point-in-time recoverability.
- You lose your backup copy.
- Page patching is not processed on lagged copies.
- Lagged copies take a long time to bring online as transaction logs are applied.

Lagged copies have storage implications as enough space must be available to store the transaction logs for lag period. However, rather than just meeting those requirements, it is best practice to have at least enough room for three additional days of transaction logs, to provide for potential truncation failures or periods of excessive log file generation. More information on planning for and recovering Exchange 2010 is covered in Chapter 12, "Backup, Restore, and Disaster Recovery."

Continuous Replication

Block Mode Introduced in Exchange 2010 Service Pack 1 (SP1), continuous replication–block mode reduces the exposure of data loss on failover by replicating all logs writes to the passive database copies in parallel to writing them locally. In other words, block mode replicates the transactions to the database copies as they are being written to the active local transaction log files. Enabling and disabling block mode is done automatically by the log copy process by database. Block mode will automatically become active when continuous replication file mode is up-to-date with the database copies. The replication transport is the same when granular replication is enabled or disabled.

The benefit of block mode is that it can dramatically reduce the latency between the active copy and the passive copy while also reducing the possibility of data loss during a failover and the time it takes to perform a switchover.

DAG Networks

A DAG network is a set of subnets that can be configured for replication or MAPI communication. Exchange supports the use of a single network adapter and path for DAG members. However, to provide network redundancy as well as the ability to separate replication and MAPI communication, multiple network adapters and networks (subnets) are recommended. After the network hardware is in place and configured and windows failover clustering has detected the changes, these additional physical networks can be configured by setting up additional DAG networks within Exchange.

Consider the following criteria when designing the network for a DAG deployment:

- Each DAG can have only one MAPI network. This network must provide connectivity to other Exchange servers, Active Directory, and DNS.
- Each DAG member must have at least one network adapter that is able to communicate with all other DAG members.
- Each DAG member's MAPI network must be able to communicate with each of the DAG node's MAPI network interfaces.
- Each DAG member must have the same number of networks.

- Each DAG can have zero or more replication networks.
- Regardless of location, each DAG member cannot have round-trip return network latency greater than 250 milliseconds (ms).
- DAG networks support Internet Protocol Version 4 (IPv4) and IPv6. IPv6 is supported only when IPv4 is also used; a pure IPv6 environment isn't supported.
- APIPA addresses (including manually assigned addresses from the APIPA address range) aren't supported for use by DAGs.
- Each DAG member's replication network must be able to communicate with every other DAG member's replication network.
- There should be no direct routing to allow heartbeat traffic from the replication network on one DAG member to the MAPI network on another DAG node, or vice versa.
- Each DAG requires a minimum of one IP address on the MAPI network. Additional IP addresses are required when the MAPI network is extended across multiple subnets. The DAG requires an IP address on each subnet it will be active on.
- When Internet SCSI (iSCSI) is used for storage, these networks should not be used for replication. This keeps replication communication from interfering with storage operations. It is a best practice to manually disable the iSCSI network from being used by the DAG and by the cluster. For more information see "Managing Database Availability Groups" under the DAG Networks and iSCSI Networks subheading at *http://technet.microsoft.com/en-us/library/dd298065.aspx*.

A DAG network can be configured in a couple different ways. The previous list suggested having at least two networks defined: one network dedicated for MAPI communication and one network dedicated for replication, as shown in Figure 11-5. If all of the replication networks go offline or fail the MAPI network will be used for replication.

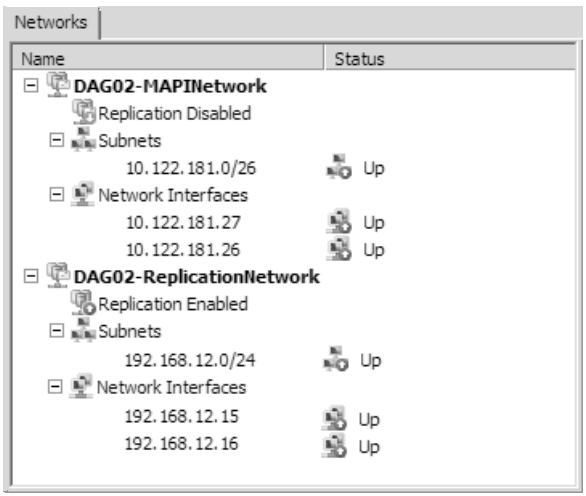

FIGURE 11-5 DAG network configuration

Database Failover Process

When a highly available mailbox database failure occurs the PAM will attempt to perform a failover of the database. Before attempting to select a suitable copy to activate the attempt copy last logs (ACLL) process occurs. ACLL makes remote procedure calls (RPCs) to each DAG node that hosts a copy of the mailbox database that is being activated. This call requests to see whether the servers are available and healthy and determines the *LogInspectorGeneration* value for the database copy. The last active mailbox database copy is used to copy any missing log files to the copy selected by Active Manager for activation. If the ACLL process fails to retrieve all of the missing log files, the configured *AutoDatabaseMountDial* value is consulted. The *AutoDatabaseMountDial* value has the following three potential values:

- **BestAvailability** This value allows the database to be automatically mounted if the copy queue length is less than or equal to 12. The copy queue length is the number of logs that the passive copies recognize and have not been replicated. When the copy queue length is less than or equal to 12, Exchange Server attempts to replicate the remaining logs to the passive copies and mount the database. This is the default value.
- **GoodAvailability** This value allows the database be automatically mounted immediately after a failover if the copy queue length is less than or equal to six. When the copy queue length is less than or equal to six, Exchange Server attempts to replicate the remaining logs to the passive copy and mount the database.
- **Lossless** This value does not allow a database to mount automatically until all logs generated on the active copy have been copied to the passive copy.

If the number of lost logs is within the configured *AutoDatabaseMountDial* value, Exchange Server mounts the database. If the number of lost logs falls outside the configured *AutoDatabaseMountDial* value, Exchange Server does not mount the database until either missing log files are recovered or an administrator manually mounts the database and accepts that the loss of data is larger than the *AutoDatabaseMountDial* setting. You use the *Set-MailboxServer* cmdlet to configure the *AutoDatabaseMountDial* setting for each DAG node.

It may seem counterintuitive to list the Best Availability as allowing for 12 missing transaction logs, and Good Availability as only allowing 6. In this case, availability is referring to the database being mounted and available, not to the possibility of lost data. In most enterprise environments, data loss is less acceptable than the loss of service. You must decide whether to keep the database available by allowing it to mount despite potential data loss or to leave it unavailable and wait for manual recovery of missing log files.

Mailbox Database Activation

When an active database failure occurs, Active Manager uses a set of selection criteria to determine which copy should be activated. It would make sense that Active Manager attempts to locate the best database copy to perform the quickest failover that is least likely to lose data. Active Manager uses a complex sorting system to determine which copy to make active.

When a failover occurs, Active Manager uses several sets of selection criteria to determine which database copy to activate. During the process for selecting the best copy to activate, Active Manager will:

1. Enumerate all the available copies.
2. Remove any copies on unreachable servers.
3. Sort available copies by how up to date they are.
4. Use the activation preference if a tiebreaker is necessary.

For more information on selection process see "Understanding Active Manager" at *http://technet.microsoft.com/en-us/library/dd776123.aspx*.

Exchange 2010 SP1 provides the *RedistributeActiveDatabases.ps1* script that provides thee ways to balance active database copies. The first option, switch parameter *-BalanceDbsByActivationPreference*, just activates the copy that has the lowest *ActivationPreference* value without taking into account Active Directory site balance. The second option, switch parameter *–BalanceDbsIgnoringActivationPreference,* attempts to balance active copies across the DAG, as shown in Figure 11-6. The third option, *-BalanceDbsBySiteAndActivationPreference,* attempts to keep active databases balanced between Active Directory sites. The version of the script included in SP1 won't move databases to less preferred copies to achieve site balance, but it will log a warning. The script will attempt to minimize an active copy imbalance during the redistribution process; this will help prevent a single node from being overwhelmed with active copies during this process.

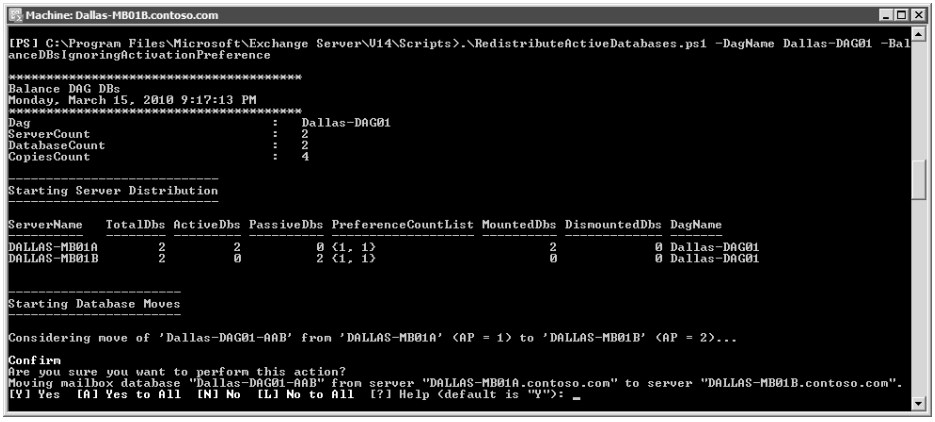

FIGURE 11-6 Running *RedistributeActivateDatabases.ps1*

Controlling Database Activation

In large environments you may want to limit which servers can host an active database in the event of a failure so that a database is not brought online in a secondary datacenter if you are performing maintenance on a server or the database is a lagged copy. A database activation policy can be set on the Mailbox server, or only the database copy can be

configured to not activate. When setting this on the Mailbox server using *Set-MailboxServer ServerName –DatabaseCopyAutoActivationPolicy,* the following three policies are available:

- **Blocked** No database can be automatically activated.
- **IntrasiteOnly** This prevents database failovers from copies that are not in the same Active Directory site.
- **Unrestricted** This allows any server in the DAG to be for database activation. This is the default configuration.

These policies only affect how Active Manager calculates where to activate database copies. An administrator can manually mount the database on a server that has the activation policy set to Blocked. The server auto activation policy is usually used during periods of maintenance when you do not want a database copy to be automatically activated on a specific server.

The second way to control database activation is to suspend database activation on a specific copy of the database. This can be done by running *Suspend-MailboxDatabaseCopy <Database Name>\<Server Name> -ActivationOnly,* as shown in Figure 11-7. Suspending activation for a specific database copy should be done on copies that you do not want to be activated automatically, such as lagged database copies.

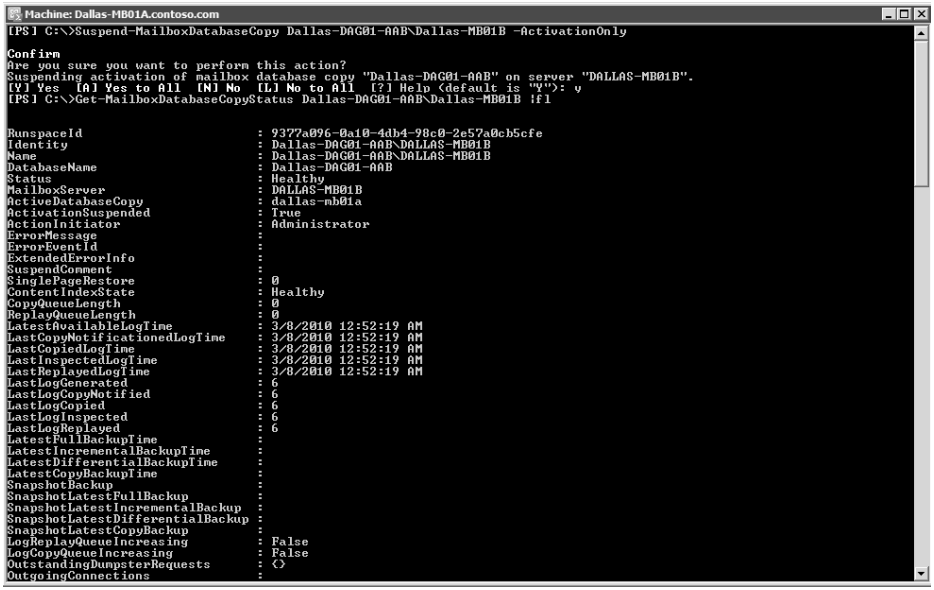

FIGURE 11-7 Suspending activation on a database copy

Unlike setting an activation policy on the Mailbox server, suspending activation on a database copy cannot be mounted directly by an administrator, as shown in Figure 11-8. However, this block can be reset in two ways: when the database copy is reseeded or if replication is suspended and then resumed.

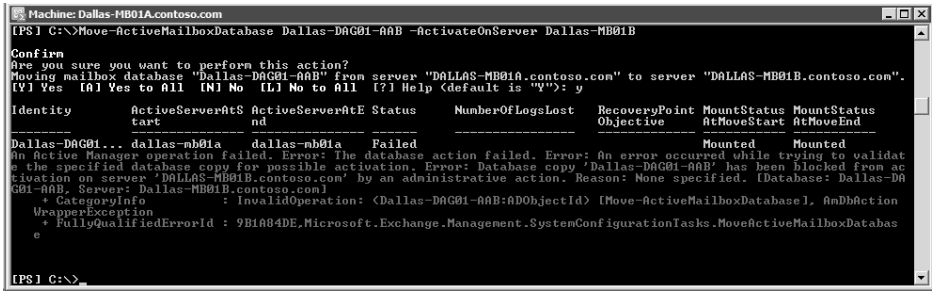

FIGURE 11-8 Attempting to activate a database copy when activation is blocked

Transport Dumpster

In case failure occurs and some transaction logs are not replicated to the passive copy, the transport dumpster is used to redeliver any recently delivered e-mail. If a database failure occurs, a request is made to the Hub Transport servers to redeliver any lost e-mail messages.

The transport dumpster only retains e-mail that has already been delivered. The local submission queue withholds any pending outgoing e-mail. After the transaction logs containing the e-mail message are replicated to and inspected by each DAG member with a copy of the database, the Hub Transport server purges the message from the dumpster.

The transport dumpster is enabled by default. Transport dumpster can be configured by using the *Get-TransportConfig* cmdlet using the following two properties:

- **MaxDumpsterSizePerDatabase** This setting defines the maximum size of the transport dumpster queue per database and is set globally for the entire Exchange organization. The recommended size is 1.5 times the maximum message size that can be sent. For example, if the maximum size for messages is 20 MB, this parameter should be set to 30 MB.
- **MaxDumpsterTime** This is the time for which the transport dumpster retains a message if the message is not purged for exceeding the maximum dumpster size. The default is set to seven days.

Managing Database Copies

You can use a number of cmdlets to manage database copies. Understanding the function of each is essential to being able to manage database copies. The following cmdlets are available:

- **Add-MailboxDatabaseCopy** This cmdlet is used to create a passive copy of an existing mailbox database on another DAG member.
- **Remove-MailboxDatabaseCopy** This cmdlet is used to delete a passive copy of an existing mailbox database.
- **Update-MailboxDatabaseCopy** This cmdlet updates or seeds a passive database copy. This is useful in situations in which seeding was not performed when the copy was created, or an error has caused the passive copy to be diverged from the active copy.

- **Suspend-MailboxDatabaseCopy** This cmdlet suspends continuous replication to the specified database copy.
- **Resume-MailboxDatabaseCopy** This cmdlet resumes continuous replication to the specified database copy that was previously suspended.
- **Set-MailboxDatabaseCopy** This cmdlet is used to configure the activation preference, replay lag time, and truncation lag time.
- **Get-MailboxDatabaseCopy** This cmdlet is used to retrieve information about the mailbox copy, such as the activation preference, replay lag time, and truncation lag time.
- **Get-MailboxDatabaseCopyStatus** This cmdlet is used to retrieve information about the health of the mailbox database copy.

Obtaining detailed information about the status of the database copies is important. One way to do this is with the *Get-MailboxDatabaseCopyStatus* cmdlet. Figure 11-9 shows the output of *Get-MailboxDatabase* | *Get-MailboxDatabaseCopyStatus* | *Format-List*. The two properties that are of immediate interest are the Context Index State and the Status, which ideally are Healthy. Also, be sure to note the *CopyQueueLength* because this is the number of transaction log files that have not been successfully copied to the passive copies. By adding the –*ConnectionStatus* parameter, additional details about the replication networks is shown, such as listing the networks being used for log replication and seeding.

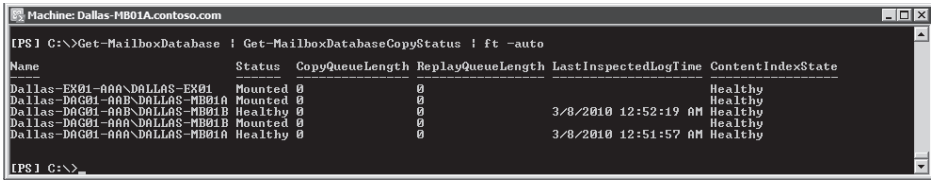

FIGURE 11-9 Running *Get-MailboxDatabaseCopyStatus*

Other potential states for database copies exist in addition to Healthy. Table 11-2 summarizes all of the possible copy status states that you may encounter.

TABLE 11-2 Database Copy Status

COPY STATUS	DESCRIPTION
ActivationSuspended	The database copy has been manually blocked from activation.
DisconnectedAndHealthy	The database copy has become disconnected from the active database copy. When it was disconnected it was in the Healthy state. This status may be reported during DAG network failures between the source copy and the target database copy.

COPY STATUS	DESCRIPTION
DisconnectedAndResynchronizing	The database copy is disconnected from the active database copy. When it was disconnected it was in the Resynchronizing state. This status may be reported during DAG network failures between the source copy and the target database copy.
Dismounted	The active copy is offline and not accepting client connections.
Dismounting	The active copy is going offline and terminating client connections.
Failed	The database copy is in a Failed state and it isn't able to copy or replay log files. In this state, the system will periodically check whether the problem that caused the copy status to change to Failed has been resolved and attempt to automatically resume.
FailedAndSuspended	The Failed and Suspended states have been set simultaneously by the system because a failure was detected, and resolution of the failure explicitly requires administrator intervention.
Healthy	The database copy is successfully copying and replaying log files.
Initializing	The system is verifying that the database and log stream are in a consistent state. This state occurs when a database copy is created; when the Microsoft Exchange Replication service is starting; and during transitions from *Suspended, ServiceDown, Failed, Seeding,* or *SinglePageRestore* to another state.
Mounted	The active copy is online and accepting client connections.
Mounting	The active copy is coming online and not yet accepting client connections.
Resynchronizing	The database copy and its log files are being compared with the active database copy to check for divergence.
Seeding	The database copy is being seeded, the content index for the mailbox database copy is being seeded, or both are being seeded. After seeding is successful, the copy status changes to Initializing.

COPY STATUS	DESCRIPTION
SeedingSource	The database copy is being used as a source for a database copy seeding operation.
ServiceDown	The Microsoft Exchange Replication service is not running on the server that hosts the mailbox database copy.
SinglePageRestore	This state indicates that a single page restore operation is occurring on the database copy.
Suspended	The database copy is in a Suspended state as a result of an administrator manually suspending the database copy by running the *Suspend-MailboxDatabaseCopy* cmdlet.

In some instances, such as during maintenance, you many need to suspend and resume continuous replication activity for a database copy. The transaction logs do not truncate the active mailbox database copy when one or more passive copies are suspended. During an extended maintenance period this may result in a large number of transaction logs accumulating in your transaction log directory. In these cases, you may opt to remove the affected passive database copy instead of suspending it. When the maintenance is complete, you can re-add the passive database copy.

Designing and Configuring DAGs

When deploying a CCR environment in Exchange 2007, the sizing was straightforward—the databases were running on one node or the other. In Exchange 2010, which offers you the ability to have 16 members with up to 1,600 databases, sizing and designing the layout is far more complex. The obvious rule is that the more servers you have in a DAG the more options you have for laying out your database copies efficiently and resiliently. Consider the implications of a three-copy, six-server DAG versus two DAGs with three servers and three copies of each database. More servers in a single DAG give you more flexibility in creating copies and to balancing load. To illustrate, if a single server fails with three active databases in a three-member DAG, the two remaining servers need to service the load from the first server, as shown in Figure 11-10.

As compared to two 3-member DAGs, a 6-member DAG can more effectively spread the results of failure across multiple servers as well as to sustain more member failures.

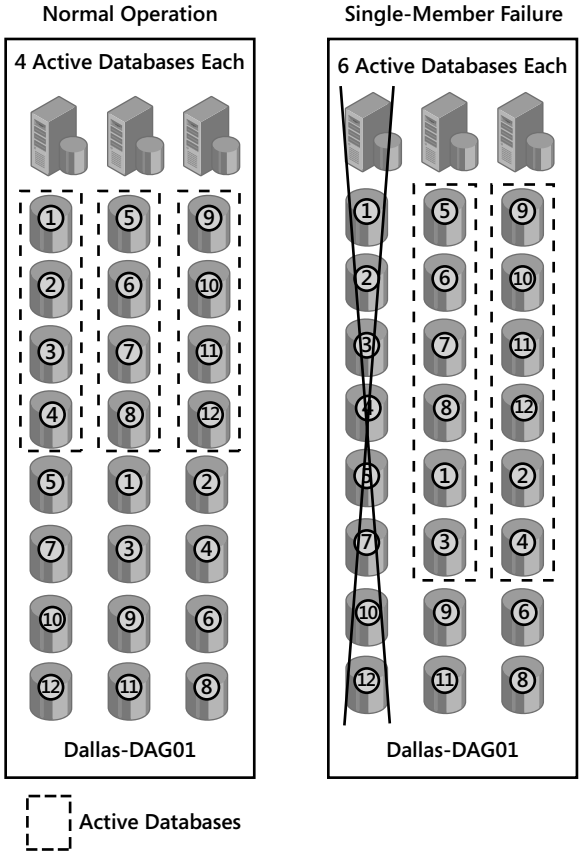

FIGURE 11-10 Three-node DAG failover

In Figure 11-10 the DAG was designed to sustain a single-node failure; if more than one member was down at least two databases would be offline. Simply adding a member to a DAG does not automatically enable it to sustain multiple failures, as Figure 11-11 shows. Here, servers are configured to mirror each other in a four-member DAG. If either A and B or C and D fail, a large number of databases will be unavailable. This configuration provides no better member redundancy than having two 2-member DAGs.

You should design the databases copies with the worst-case failure needed to meet your agreed-upon SLAs. The following two rules apply for redundancy:

1. One-member failure requires two or more high-availability copies, two or more servers, and a witness server.
2. Two-member failure requires three or more high-availability copies, four or more servers, and a witness server.

Rather than mirroring database copies on two servers it is better to stripe copies across the members or create copies randomly across the DAG to reduce the likelihood of a low number of failures causing outages for databases.

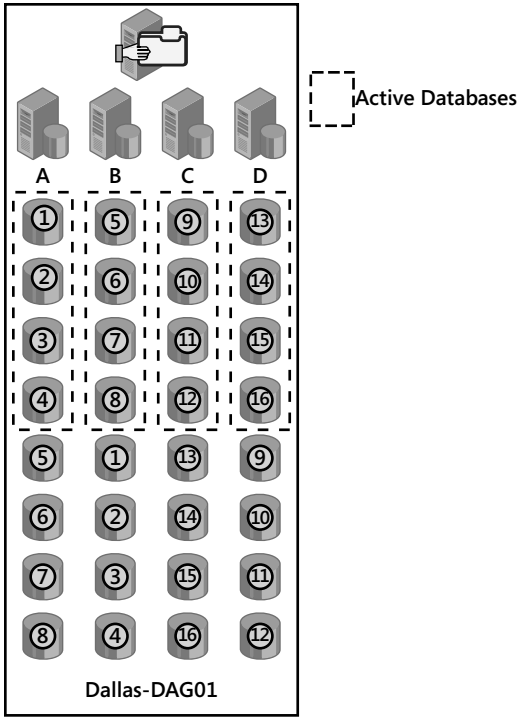

FIGURE 11-11 A four-node mirrored configuration

When determining the copy design plan for the worst case, ensure that the members can handle all of the hosted database copies becoming active. If you plan on oversubscribing the members, you can set a maximum number of simultaneous active databases on each member to ensure that more copies than the server can handle do not come online by using the *Set-MailboxServer* cmdlet with the *-MaximumActiveDatabase* parameter. When the Mailbox server has reached the maximum, no additional database mounts will be successful. If the Active Manager attempts to mount a database on the server the mount will fail and Active Manager will attempt to mount the database copy on another member if one is available. Also, as usage profiles change over time it is important to periodically evaluate the appropriate level of oversubscription and whether the number of active database copies should be modified to accommodate for hardware and usage changes.

Over the course of time, when maintenance is performed active mailbox databases may end up active on servers that they were not intended for. As part of routine maintenance activities remember to activate the database copies across the DAG. You may also use *RedistributeActiveDatabases.ps1,* which is included in SP1, to automatically load-balance active database copies across DAG members.

Deciding the number and location of database copies also involves the storage infrastructure and the operational maturity of your IT department. Assuming the operational challenges can be overcome, you should consider a few best practices when choosing whether to use RAID (Redundant Array of Independent Disks) or JBOD as summarized in Table 11-3.

TABLE 11-3 Choosing Between RAID and JBOD in a Single-Site Deployment

NUMBER OF COPIES	STORAGE OPTIONS
Two high availability	RAID
Three or more high availability	RAID or JBOD
One active and one lagged copy	RAID

When a large number of databases are hosted on each server in a DAG, disk management can become complicated, especially when you are using JBOD storage. Only 23 drive letters are available to mount additional disk drives—A and B are reserved and most likely the operating system is installed on C. When planning a DAG that will require a number of volumes, it is a best practice to use volume mount points rather than drive letters. Volume mount points allow volumes to be mounted as directories rather than drive letters. For example, you may want to mount a 1-TB volume in D:\Databases\Dallas-MB01 to store the Dallas-MB01 database files. You could then mount another 1-TB volume in C:\Databases\Dallas-MB-02 for storing the Dallas-MB02 database files. This way you are no longer constrained by the number of drive letters available.

Using mount points introduces a problem: if the drive that contains the mount points fails, you lose connectivity to all of the other drives. The best practice is to protect the volume that contains the mount points using RAID to reduce the likelihood of a single disk failure taking the entire server offline.

NOTES FROM THE FIELD

JBOD Impact on Operations and Risk Discussion

Arno Zwegers
Infrastructure Architect, Avanade Netherlands

Since the early days of Exchange, administrators have had servers that include storage-level redundancy. Usually this is a hardware-based RAID system where the data is stored across multiple disks. Although failure of one disk may affect performance, it does not affect the availability of the mailboxes. Now, with the ability to store the data on multiple servers, you can use JBOD as the storage technology on the DAG members themselves. The availability of the data is no longer handled by the underlying storage infrastructure, but by Exchange. However, the availability of the operating system and applications must be ensured with a RAID storage solution. This is important; otherwise, the failure of one disk may mean that the server must be rebuilt. A server rebuild takes time, and that means an increased risk of data loss for all mailbox database copies on the failed server because there is now one less copy of all those mailbox databases.

JBOD changes the way administrators will have to operate the servers. Monitoring and signaling of problems becomes even more important to handle quickly and efficiently. This also changes the process required to complete a failed disk replacement.

When a disk fails on an Exchange server with RAID protecting the operating system and databases and the administrator is notified, he or she will have to replace the failed disk. Depending on the RAID system used, replacing the disk will have to happen quickly if no online spare is available, or it can wait if an online spare is available. Rebuilding a RAID set will consume system resources and may impact performance. Many administrators prefer to rebuild the RAID set during a maintenance window; however, with an online spare this process starts immediately.

In this situation the administrator will perform two actions: replace the failed disk with a new one and monitor the status of the rebuilding process, noting when the rebuild is completed. During this process the availability of Exchange itself has not changed—it is unaware of what happened on the storage level.

Now consider an Exchange 2010 server with RAID protecting the operating system and Exchange databases stored on JBOD, where each disk is a separate volume within Windows. (This is important because people may interpret JBOD only as "without RAID" and then create a single Windows volume across all the disks, which increases risk significantly when using JBOD.) The monitoring mechanism for failed disks also needs to be updated, because it will need to understand and report on the database copy status when one or more copies are unavailable or no longer exist.

When a disk fails and the administrator responds, he or she will have to perform more actions than when using a RAID system. First, the failed disk is replaced and formatted, and then the Exchange databases copy needs to be restarted. Finally, the administrator must monitor the status of the database replication.

The administrator will have to consider how to re-create the database copies on the replaced disk. If the failed disk contained 1 TB of data, this amount of data will have to be copied to the replaced disk. This can be done by creating a new copy of the databases and transferring 1 TB of data over the network or by placing a copy of the database files onto the drive by means of USB-based storage or by restoring from a backup. This consideration is important because even on a 1-GB network connection, the copy may take more than two and a half hours to complete, and when copied across a 100-MB WAN connection this may take more than 24 hours.

JBOD reduces hardware costs, but it increases risk, even in the scenario where the DAG has three or more copies of the data. The time during copy re-creation increases risk, because during that time fewer are copies available; however, it can be argued that this risk is similar to the risk while this is a failed disk in the RAID set.

The fundamental change is where the data redundancy is handled. Administrators are used to RAID, which has been used for a long time. The additional activities that have to be performed within Exchange to provide redundancy are new. Without additional integration, many monitoring systems will not be able to effectively understand this new redundancy model. Confusion regarding how to handle failures increases the likelihood that an administrator may not identify the problem or respond quickly enough to the failure.

During the design process the risk of the operational excellence and the time it takes to reseed are important factors to consider in determining whether JBOD is a viable solution for you.

Availability Planning for Client Access Servers

Unlike the Mailbox server role and to some extent the Transport server roles, the Client Access Server role does not have any inherent high-availability functionality built in. That does not mean that it was designed without high availability in mind—it just requires other modalities to provide high availability. A separate product or feature is required to provide this functionality. The following sections cover choosing and configuring the best solution depending on deployment requirements.

Client Access Load Balancing and Failover Solutions

To provide Client Access high availability requires multiple Client Access servers to be deployed in the same Active Directory site. As mentioned, there is no integrated mechanism to provide load balancing and failover capabilities if a host becomes unavailable or overloaded. However, a variety of products are available that fill this need. Because the Client Access servers provide so many services with a number of different connections types—from OWA to MAPI to Web Services—three types of Client Access server traffic actually need to be load balanced:

- Traffic from internal networks
- Traffic from external (Internet) networks
- Traffic from other Client Access Servers (proxy)

Affinity

Some Exchange communications are *stateful*, meaning the application requires that the communication context be maintained with the same host until the session is completed. This is common in conversations that we have daily. If a co-worker asks what the deadline is for your project and then you walk into another co-worker's office and say "Wednesday," she will likely have no idea that you were answering John's question. This is similar to how

a stateful program works: It expects to continue communication with the same context until the conversation is completed. Other protocols are stateless, such as HTTP, where state information is lost between client requests. In the case of multiple, load-balanced hosts, affinity is a mechanism to direct subsequent calls to the host that answered the initial request.

It is important to understand the different types of affinity and how they are used. The Client Access server uses a number of protocols that will need to be load balanced, including HTTP and RPC. Remember some Client Access server protocols require affinity and some do not.

EXISTING COOKIES

Existing cookie affinity uses cookie information transmitted during typical client/server sessions. This type of affinity is only useful for protocols using HTTP and thus not an option for any RPC communication. OWA using forms-based authentication is an example of an application that does use existing or application cookies.

LOAD BALANCER COOKIES

Using load balancer cookies is similar to using existing cookies except that the load balancer creates the cookie and does not rely on any existing cookies. As with existing cookies, this is only usable with HTTP. Additionally, the client must support the addition of the load balancer–generated cookie. Exchange ActiveSync, Outlook Anywhere, and some Exchange Web Services do not support this capability. However, Outlook Web App, Exchange Control Panel, and Remote Windows PowerShell are good candidates for this type of affinity.

SOURCE IP

Source IP is perhaps the most common and widely supported type of affinity. With Source IP affinity, the load balancer records a client's IP address and the initial destination host. All subsequent traffic from that source IP will continue to go to the same destination host for a period of time. However, source IP load balancing has two main drawbacks.

First, affinity breaks when clients change their IP addresses. If you have an environment where this happens frequently, such as mobile clients roaming between wireless networks, this will cause issues. Users may experience symptoms such as having to re-authenticate.

Second, if you have an environment where many clients share the same source IP, such as when a device performing Network Address Translation (NAT) is used, the load will not be evenly distributed because all clients behind the NAT will be routed to the same destination IP address.

SSL SESSION ID

SSL session ID is generated when establishing an SSL encrypted session. The SSL session ID has a big advantage over source IP affinity: It can uniquely identify clients sharing the same source IP address. Another advantage is that there is no requirement to decrypt the SSL traffic. This is a hard requirement for using client CA because renegotiating the SSL session ID

puts additional overhead on the server. Directing traffic to the same server saves processing time and prevents performance impacts.

SSL session ID does not work well with all clients. Some browsers and mobile devices, such as Microsoft Internet Explorer 8.0, create a new SSL session for each browser process. Therefore, every time a user creates a new e-mail message, a separate window opens, which creates a new SSL session. The exception to this is when users use client CA. The same SSL session ID is used for all communication to a specific host.

Outlook Anywhere and some mobile clients also open several Client Access Server sessions. Each session receives a different SSL session ID, so each session could end up being connected to a different server. As discussed earlier, this is not a problem because Windows Server 2008 network load balancing can correlate the RPC_IN_DATA and RPC_OUT_DATA; however, it does cause additional overhead and can negatively impact server performance.

Selecting a Load Balancer Type

To lower cost and complexity, you should select a single load-balancing solution that works for each type of traffic. A large number of load-balancing options are available on the market; it is important to make an informed choice. Consider the following criteria during the decision-making process:

- **Features** Does the load balancer have features such as SSL offloading that you will use now and in the future?
- **Manageability** How easy is the solution to configure and maintain?
- **Failover detection** Does the solution support advanced detection (service awareness) or simple ping (host awareness)?
- **Affinity** What options does the solution support to keep client connections returning to the same host?
- **Cost** How much will it cost to implement the solution?
- **Scale** How does the solution work as the number of hosts increases?

Load balancers can be categorized into four distinct categories: Software Load Balancers, Hardware Load Balancers, Intelligent Firewalls, and Round Robin DNS. The following sections discuss each of these categories.

SOFTWARE LOAD BALANCING

Windows Network Load Balancing (NLB) has been part of the Windows Server operating system since Windows NT 4.0. Of course, a lot has changed since its early days. NLB can scale to 32 hosts on Windows Server 2008 R2, but the practical limit for Exchange is 8 hosts based on documentation provided about Microsoft's internal deployment experience. One advantage of NLB is that it is relatively inexpensive to implement.

One disadvantage of NLB is that you cannot use it combined with Windows Clustering. If you are trying to configure an all-in-one server that has the Mailbox role and Client Access

Server role, and you are using DAGs, you must use a non-Windows network load-balancing solution for client access. Another drawback is that NLB only supports source IP affinity or no affinity. This may limit its ability to effectively load balance across all of the Client Access protocols. NLB also has no built-in intelligence to test server health or functionality before sending traffic to a host. If the IIS service has stopped on one Client Access server, NLB will continue to send traffic to that node, unless it is reconfigured to stop. This can be partially overcome when NLB is deployed along with Microsoft System Center Operations Manager 2007 R2 and the NLB Management Pack, which may be an option for people that already use Operations Manager.

Other software-based load balancers are installed on a separate server or other hardware. These solutions are often more similar to hardware load balancers or application firewalls than the functionality of NLB.

HARDWARE LOAD BALANCERS

If you need to support more than eight nodes in your Active Directory site, you must consider a hardware load balancer. Having a dedicated piece of specialized hardware allows for the best performance and a considerable number of features. Most hardware load balancers support multiple affinity types, and even allow for the ability to fall back if one type fails. Typically, hardware load balancers support more advanced node health checks. These range from simple ping tests to measuring response times to custom Web pages. More expensive solutions also provide hardware redundancy, further eliminating any single points of failure.

Probably the biggest disadvantage is the cost of deploying a hardware solution. However, for large-scale deployments, this is typically the solution selected.

APPLICATION FIREWALLS

Application (Intelligent) firewalls, such as Microsoft Threat Management Gateway (TMG) or Forefront Unified Access Gateway (UAG), are similar to the hardware load balancer solution, but can also provide additional security features. For example, with Active Directory Domain Services (AD DS) security groups, you can control what time of the day groups of users can access OWA.

One disadvantage is that with this great power comes great complexity. These solutions require more testing and more administration and operational support compared to the other solutions. Another disadvantage is that these do not perform RPC load balancing; in order to do this another solution is also required.

DNS ROUND ROBIN

DNS round robin uses DNS's ability to map multiple hosts to a common name. For example, if you have three Client Access servers the DNS A record entries would look like this:

```
mail.litwareinc.com      192.168.1.2
mail.litwareinc.com      192.168.1.3
mail.litwareinc.com      192.168.1.4
```

The first client to request *mail.litwareinc.com* would have the IP address of 192.168.1.2 returned. The second request would have 192.168.1.3 returned, and the third request would have 192.168.1.4 returned. The fourth request would have the first IP address returned again, and the pattern would continue. The main advantage of this is that it has very little or no cost to implement and it's very easy to configure.

Unfortunately, the limitations of DNS round robin limit its use to lab environments and very small implementations. These limitations include no support for affinity, which requires the application to maintain affinity. For example, a Web browser navigating to *webmail .contoso.com* will actually use the IP address the DNS server returns from the name resolution query. Internet Explorer will have this DNS entry cached for about 30 minutes. If the server became unavailable during that cache period, the Web browser could not be automatically redirected to the new server. Because of this caching, the Web browser will attempt to reach an unavailable server until its cache expires. DNS round robin also does not have any health checks or dead node removal. In the preceding example, if 192.168.1.3 becomes unavailable, DNS will continue to return the down host's IP address every third request unless it is manually reconfigured. Another problem is that if multiple clients share the same local DNS server as in a LAN environment, all of those clients will use the same IP address that is cached by the local DNS server; if most of the clients are from the same location, the load will be very balanced across the servers. Finally, changes to DNS can take time to propagate. If a new Client Access server is added to DNS, it will be underutilized until the record propagates fully.

GLOBAL SERVER LOAD BALANCING

Global server load balancing (GSLB), or wide-area load balancing, is a more sophisticated version of DNS round robin available from some hardware load balancer vendors. This solution is typically deployed as a hardware device or even as a feature of a hardware load balancer. This type of load balancing uses DNS to load-balance client connectivity between sites based on a number of factors such as location of the client, response time of the servers, availability of the servers, custom weights, and more. GSLB is typically used in multiple site configurations to provide load balancing between sites. To provide full site redundancy the GSLB device should be located outside of either of the load-balanced sites or deployed in multiple sites. One way to use the GSLB is to load-balance Autodiscover to ensure that it is available even during a single site outage. In Figure 11-12, Autodiscover. constoso.com is set up for GSLB—all traffic will be sent to the IP address for the Denver Autodiscover service. In the event of a failure of Denver, the GSLB device can send all traffic for Autodiscover.contoso.com to the second site.

The GSLB device will accept DNS requests from the client and then return the appropriate IP address based on the rules defined. The TTL for the returned IP address is set low to ensure that changes are received by the client as quickly as possible. As with DNS round robin, because GSLB relies on DNS client resolution, its functionality is limited when the client DNS resolution is uncontrolled.

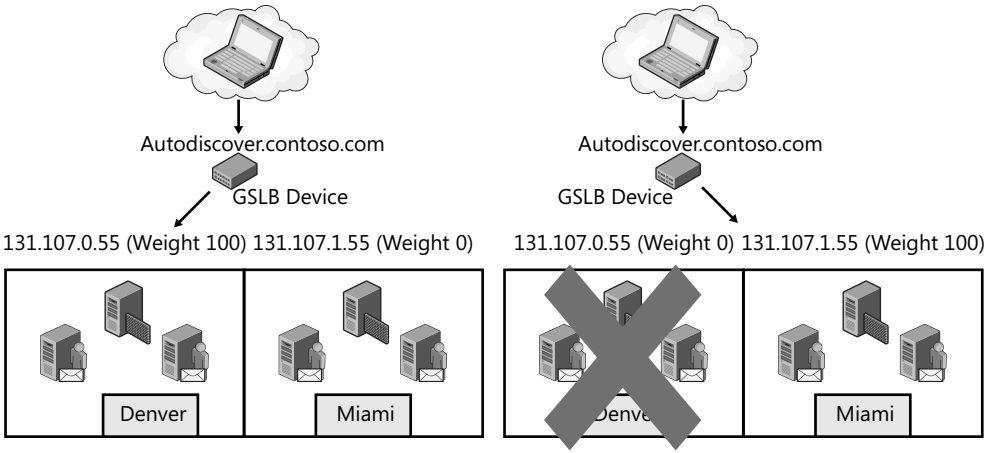

FIGURE 11-12 Using GSLB for the Autodiscover server

LOAD BALANCING SUMMARY

As you can see, you have a variety of solutions to choose from, depending on business requirements and budget. Table 11-4 combines affinity, load balancing, and other considerations when choosing a solution for load balancing.

TABLE 11-4 Load Balancer Comparison

TYPE	COST	SCALE	AFFINITY	BENEFITS	DRAWBACKS
Hardware Load Balancing	High	High	All Types	■ Automatic Failover ■ Can be used with Windows Failover Clusters ■ Service Health Checking	■ Cost ■ Complex
Application (Intelligent) Firewall	Medium	Medium	Source IP Cookie	■ SSL Bridging ■ Enhanced Security ■ AD Authentication ■ Service Health Checking	■ Complex

TYPE	COST	SCALE	AFFINITY	BENEFITS	DRAWBACKS
Software Load Balancing	Low	Low	Source IP	■ Inexpensive ■ Easy to configure	■ Limited Scale ■ Cannot be used with Windows Failover Clusters ■ No Service Health Checking
DNS Round Robin	Low	Low	Random	■ Easy to configure	■ Manual failover ■ Unpredictable traffic ■ Long failover time

Table 11-5 summarizes the configuration needed to support all of the Client Access Server protocols. If the load balancer is used to terminate the SSL certificates, the traffic between the load balancer and the Client Access server will be unencrypted; thus, the unencrypted port is used. Each of the services can be provided with separate load-balanced IP addresses to apply different load-balancing policies to each. For more information about configuring certificates and the internal and external URLs for your Client Access servers see Chapter 4, "Client Access in Exchange 2010."

TABLE 11-5 Load-Balancing Client Access Services

CLIENT ACCESS SERVICE	PROTOCOL	TCP PORT(S)	NOTES
Exchange ActiveSync	HTTP	80/443	Persistence: Source IP
IMAP4	IMAP4	143/993	
Outlook Anywhere	HTTP	80/443	Persistence: Source IP
Outlook Web App	HTTP	80/443	Persistence: Cookie or Source IP
POP3	POP3	110/994	
RPC Client Access	RPC	RPC Ports	Persistence: Source IP

By default the Outlook client will make a connection to the RPC Endpoint Mapping Service on TCP/IP port 135 on the server to negotiate a dynamic RPC port above TCP 1024 for usage. If no firewalls or load balancers are between the clients and servers this is usually not an issue. You can reduce the number ports that need to be load

balanced by modifying the Client Access servers to scope down the ports that are required. You must make three modifications:

1. Modify the registry to statically set the MAPI TCP/IP port on all of the Client Access servers.
    1. Open the Registry editor and then select *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSExchangeRpc\ParametersSystem*.
    2. Add a DWORD named TCP/IP Port.
    3. Set the value of TCP/IP port to selected port number.
    4. Close the Registry editor.
2. Modify *X:\Program Files\Microsoft\Exchange Server\V14\Bin Microsoft.Exchange.Addressbook.Service.exe.config* file to statically assign the Address Book (NSPI) and Referral Service (RFR) TCP/IP port on all of the Client Access servers.
    1. Open *X:\Program Files\Microsoft\Exchange Server\V14\Bin Microsoft.Exchange.Addressbook.Service.exe.config* in Notepad or another text editor.
    2. In the *<appSettings>* section locate the line that has *<add key="RpcTcpPort" value="0" />* and then change the 0 to the selected TCP/IP port.
    3. Save the file and close Notepad.
    4. Restart the Client Access server.
3. Modify the registry to statically set the MAPI TCP/IP port on all of the Mailbox servers hosting public folders.
    1. Open the Registry editor and then select *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSExchangeRPC\ParametersSystem*.
    2. Add a DWORD named TCP/IP Port.
    3. Set the value of TCP/IP port to selected port.
    4. Close the Registry editor.
    5. Restart the Mailbox server.

After the load balancer is configured, certificates need to be applied and the internal and external URLs need to be set on each of the Client Access servers.

Creating a Client Access Server Array

Using a load-balancing product will allow you to load-balance connectivity across the Client Access servers for all communication types. To represent the RPC Client Access load-balanced cluster in a single Active Directory site a Client Access array is created. Then the name and IP address for the network load-balanced cluster must be added into the local Domain Name System (DNS). For example, you could add an A record for Dallas-Caa01.contoso.com that points to 10.1.1.25. After adding the DNS record, you can create the Client Access array and assign it to an Active Directory site using the *New-ClientAccessArray* cmdlet. If mailbox databases are already created in the Active Directory site, you must assign the Client Access array to each of the mailbox databases in the site using the *Set-MailboxDatabase* cmdlet

with the *RpcClientAccessServer* parameter. To avoid this extra step, you should create the Client Access server array prior to installing any Mailbox servers into the Active Directory site.

A Client Access array exists in a single Active Directory site. Therefore, you need to create a Client Access array in each Active Directory site that will have load-balanced Client Access servers. Also, the Client Access array cannot match the DNS name for the external Outlook Anywhere host name or Outlook will attempt to the Client Access array via RPC before falling back to HTTPS. Because the Client Access server array name is used only for RPC access, any certificates obtained to support Client Access connectivity (OWA, Outlook Anywhere, and so on) don't need to have the Client Access array name included—RPC communications do not use certificates. For a full discussion of configuring certificates and the internal and external URLs for your Client Access servers, see Chapter 4, "Client Access in Exchange 2010."

When you put together a Client Access server array with a DAG, a redundant configuration is born. Figure 11-13 shows how an Outlook client will maintain connectivity when a mailbox database failover occurs. The client computer maintains connectivity to the same node in the Client Access server array based on the configuration of the load balancer and that Client Access server will connect to the second Mailbox server to maintain connectivity to the mailbox.

The other scenario where the Client Access server handles a failure is illustrated in Figure 11-14. When the Client Access server fails, the load balancer will reconnect the client computer to another Client Access server in the Client Access server array. The new Client Access Server will then connect to the Mailbox server with the active copy of the database so that the client computer will continue to be connected to the user's mailbox.

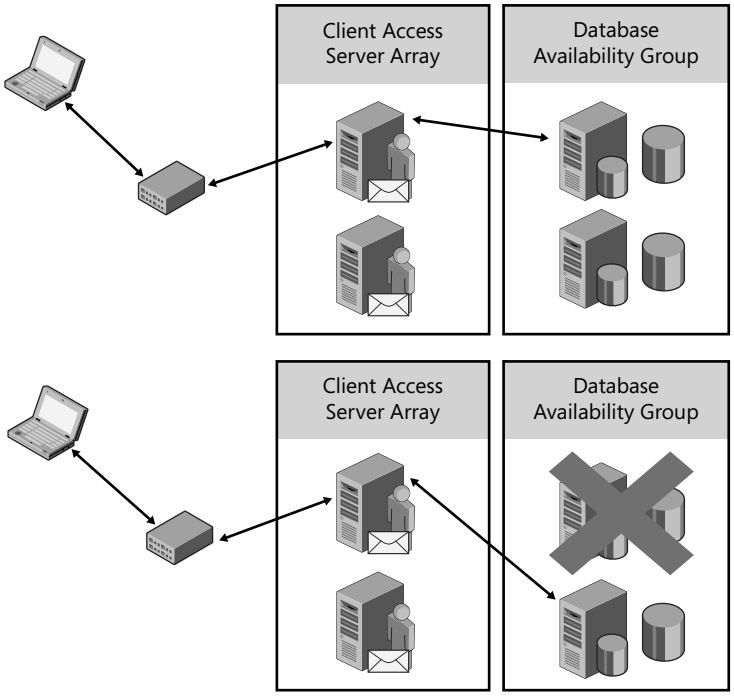

FIGURE 11-13 Client connectivity to the Client Access server during a mailbox copy failover

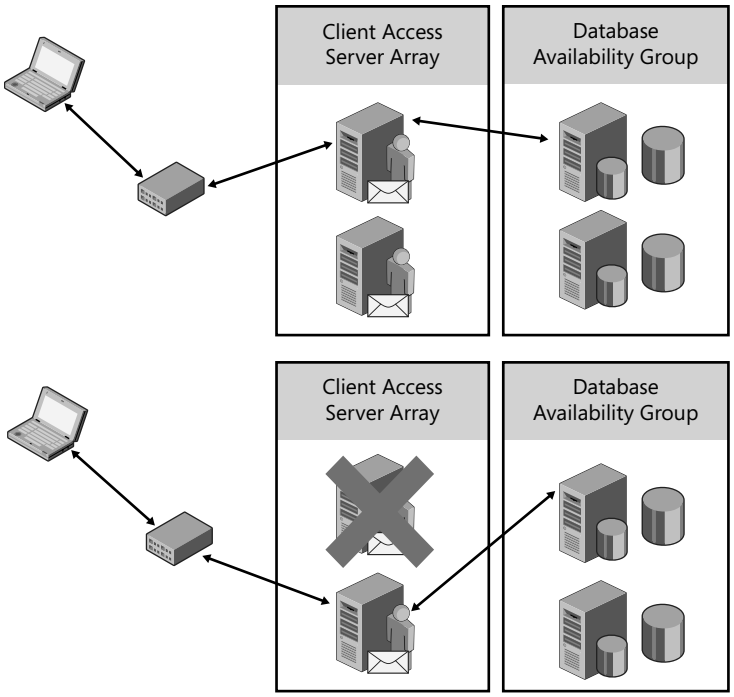

FIGURE 11-14 Client connectivity to the Client Access Server during a Client Access server failover

Availability Planning for Transport Servers

Within the Exchange organization, it is important to deploy multiple transport servers to provide message path redundancy. Deploying multiple Hub Transports in each Active Directory site automatically provides redundancy and load balancing for message delivery. Deploying multiple Edge Transport servers will also provide incoming and outgoing SMTP redundancy.

Shadow Redundancy

Exchange Server 2010 includes the shadow redundancy feature, which provides redundancy for messages for the entire time they are in transit. This is in addition to the transport dumpster. With one form of shadow redundancy, the message deletion from the transport queue is delayed until the transport server verifies that all of the next hops for that message have completed delivery. If any of the next hops fail before reporting successful delivery, the transport server resubmits the message for delivery to that next hop. If the next hop server does not support shadow redundancy, the message will be sent to the next hop and a shadow copy of the message will not be retained.

Shadow redundancy provides the following benefits:

- It eliminates the reliance on the state of the transport server queues. If redundant message paths exist, the state of any transport server isn't relevant. If a transport server fails, you can simply remove it from production without worrying about emptying its queues or losing messages currently in transit.
- If maintenance needs to be performed on the transport server the server can be brought offline without the risk of losing messages in transit.
- It reduces the need for hardware redundancy for transport servers for messages in transit.
- It consumes less bandwidth than other forms of redundancy that create duplicate copies of messages on multiple servers. With shadow redundancy the only added network traffic is the discard status being communicated between transport servers.
- It provides resilience and simplifies recovery from a transport server failure because messages still in transit within the Exchange organization are protected by the previous Exchange 2010 transport server.

***IMPORTANT** Shadow redundancy does not protect messages in the transport dumpster, which is essential in being able to recover messages in the case of a DAG member failure.*

One form of shadow redundancy is implemented by extending the SMTP protocol. These service extensions allow SMTP hosts to negotiate shadow redundancy support and communicate the discard status for shadowed messages.

The protocol implementation of shadow redundancy works between Exchange 2010 transport servers. In the following scenario, a message is sent from an Exchange 2010 mailbox out to the Internet from a Hub Transport through an Edge Transport server, as shown in Figure 11-15. In this case the message flow follows these stages:

1. Hub delivers the message to Edge1:
   a. Hub opens an SMTP session with Edge1.
   b. Edge1 advertises shadow redundancy support.
   c. Hub notifies Edge1 to track discard status.
   d. Hub submits the message to Edge1.
   e. Edge1 acknowledges receipt of the message and registers Hub1 to receive discard information for the message.
   f. Hub moves the message to the shadow queue for Edge1 and marks Edge1 as the primary server. Hub becomes the shadow server.
2. Edge1 delivers the message to the next hop:
   a. Edge1 submits message to a third-party e-mail server.
   b. The third-party e-mail server acknowledges the message's receipt.
   c. Edge updates the discard status for the message as delivery complete.

3. If the message is delivered successfully, when Hub queries Edge1 for discard status:
   a. At end of each SMTP session with Edge1, Hub queries Edge1 for the discard status on messages previously sent. If Hub has not sent any other messages to Edge1, it will open an SMTP session with Edge1 to query for the discard status after five minutes and will fail over three failures or 15 minutes. This time can be configured using *Set-TransportConfig* with the *ShadowHeartbeatTimeoutInterval* parameter. The number of retries can be configured by running *Set-TransportConfig -ShadowHeartbeatRetryCount*.
   b. Edge1 checks the local discard status and sends back the list of messages registered to Hub1 that have been delivered and then removes the discard information.
   c. Hub deletes the delivered messages from its shadow queue.
4. If the message delivery fails, then Hub queries Edge1 for discard status and resubmits the message:
   a. If Hub cannot contact Edge1, Hub resumes the primary role and resubmits the messages in the shadow queue to another available transport server, Edge2.
   b. The resubmitted messages are delivered to Edge2, and the workflow starts from step 1.

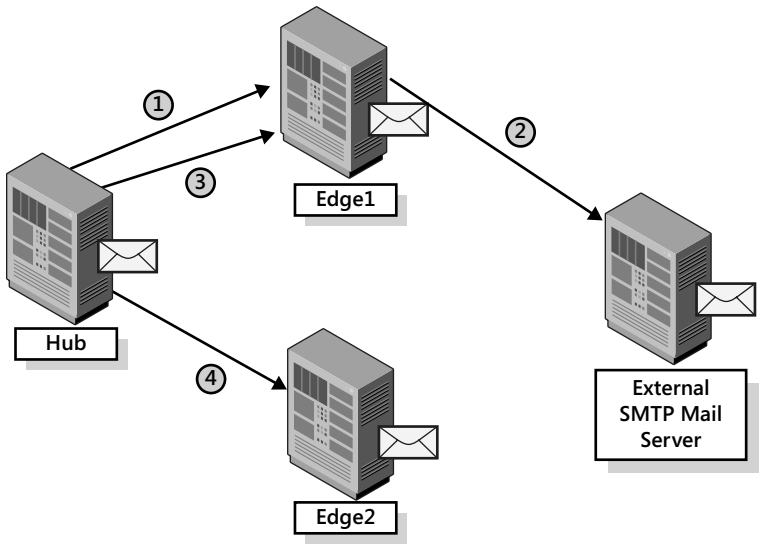

FIGURE 11-15 Transport shadow redundancy

The Shadow Redundancy Manager (SRM) is the core component of a Transport server responsible for managing shadow redundancy. The SRM is responsible for maintaining the shadow server for all of its primary messages. The SRM is also responsible for maintaining the following information for all the shadow messages in its shadow queues:

- Determining when the shadow server should take ownership of shadow messages, thus making it the primary server
- Maintaining the list and checking primary server availability for each shadow message

- Processing discard notifications from primary servers
- Removing the shadow messages from the database once after receiving the discard notification
- Sending the discard status to the shadow servers

Shadow redundancy does not require any sort of configuration. When multiple transport servers are deployed they will automatically negotiate the use of shadow redundancy. When multiple Hub Transport servers are deployed in each Active Directory site each e-mail message will exist in two places while in transit. Because each message exists in two locations you may consider deploying Hub Transport servers without RAID-protected disks because the in transit e-mail messages will exist on another server and not need to be recovered. It is not always advantageous to deploy transport servers without redundant storage for the message queue as shadow redundancy does not protect e-mail messages in the transport dumpster. In configurations with a multi-site DAG as well as others that consistently maintains a number of e-mail messages in the transport dumpster because of transaction log replication latency you should store the message queue on redundant storage to reduce the probability of losing transport dumpsters data. You can determine the number of items in the transport dumpster by viewing the *Dumpster Item Count* counter on the *MSExchangeTransport Dumpster* performance object using Performance Monitor or by trending this counter using a solution like Microsoft System Center Operations Manager.

To reduce the likelihood of a server failure causing a loss of e-mail, the Mailbox Submission service on a DAG member first attempts to load-balance submission requests across other Hub Transport servers in the same Active Directory site. If the Hub Transport role is installed on the DAG member and it cannot submit messages to any other Hub Transport server in the site, it will fall back to the local Hub Transport server.

Inbound E-mail Redundancy

Another form of shadow redundancy called *delayed acknowledgement* is used in scenarios when a transport server receives a message from a mail server that doesn't support shadow redundancy. Rather than immediately confirming receipt of the message from the submitting service, it delays sending an acknowledgement until it has confirmed that the message has been successfully delivered.

For inbound e-mail delivery with Edge or Hub Transport servers, the typical way to provide redundancy is to use an MX record for each of the e-mail servers accessible for e-mail delivery. MX records are weighted records in DNS that point to the e-mail servers responsible for receiving mail for a domain. The MX records with a lower weighting will be attempted before higher-weighted records. Records that have the same weight will be load balanced. Using MX records to provide this redundancy is part of the way SMTP was designed, so this configuration is often sufficient. In some instances where large numbers of SMTP servers are deployed, you may choose to use network load balancing to have more control over the inbound SMTP traffic, but load balancing should never be used inside the Exchange organization or against the Default Receive Connector on each Hub Transport server. Load balancing and redundancy are built in to the transport service.

NOTE *More information about MX records and how they are used can be found in RFC 2821.*

Planning Cross-site Failovers

The high-availability improvements in Exchange 2010 make it even easier to deploy cross-site failover solutions without a need for third-party network and storage solutions. The secondary site can be used to handle primary site outages resulting from maintenance or other, more serious failures. Even with the improvements in Exchange 2010, careful planning must be done to successfully deploy and maintain a multi-site deployment.

Cross-site DAG Considerations

The primary building block of a cross-site solution is the cross-site DAG. Extending a DAG between sites does have a couple requirements, including the following:

- Fewer than 250 milliseconds of latency between all DAG members. To ensure consistent DAG operations there should be minimal latency.
- At least one domain controller in each site. Exchange requires a domain controller in each site it is deployed; for redundancy at least two should be deployed.
- At least one Client Access server in each site. To provide client connectivity to both sites at least one Client Access server must be deployed; for redundancy at least two should be deployed.
- At least one Hub Transport server in each site. To provide e-mail transport to both sites at least one Hub Transport must be deployed; for redundancy at least two should be deployed.
- Consider the impact on supporting services to a failover. The appropriate number and configure of Client Access servers, Hub Transport, Edge Transport, Unified Messaging server roles, and domain controllers must be located at each site to support the maximum number of active mailboxes.
- In the case of a complete datacenter failure:
  - Quorum must be reestablished. To mount databases, a quorum must be established within the cluster. If a majority of the members, including the file share witness, are unavailable the DAG must be manually reconfigured to reestablish quorum.
  - Manual switchover process. To bring up the second site, the administrator must manually initiate the switchover. A complete datacenter switchover is not something to consider lightly from a business process standpoint. Requiring manual intervention was put in place to ensure that an administrator has to make the decision to initiate a full datacenter switchover.

Cross-site Considerations for Client Access and Transport

When you deploy non-Mailbox servers to support a cross-site failover, you might come across several issues, including Domain Name System (DNS) entries for Outlook Web App, Outlook Anywhere, and Autodiscover. Inbound e-mail (MX) must be redirected to reflect the secondary site's IP addresses. These record changes should be automated to provide the quickest return to service. Until the clients that connect to these services have the new addresses they will fail. These changes can be improved by deploying DNS servers in multiple locations or by using third-party global-server load balancing. If you are using a hosted anti-spam or archiving service these services must be redirected to the new site.

Proper namespace planning is needed for the failover process to run smoothly. To do this you must consider each datacenter as being active and choose a unique set of names for each Exchange service. This includes OWA, Post Office Protocol version 3 (POP3), Internet Message Access Protocol version 4 (IMAP4), Exchange Web Services, and Outlook Anywhere; however, it cannot include Autodiscover. Having this number of names requires that you configure certificates to reflect the names that each site uses. To do this, ensure that the certificates contain all required host names for services in both datacenters or use a wildcard certificate. If you choose to use separate certificates for each datacenter, you must ensure that each certificate has the same certificate principal name. To reduce the impact on Outlook connections, you must run *Set-OutlookProvider EXPR -CertPrincipalName msstd:<certificate principal name>*. For more information on namespace planning see Chapter 4, "Client Access in Exchange 2010."

NOTES FROM THE FIELD

Client Access Namespace and the Impact to High Availability and Site Resiliency

Gary A. Cooper
Senior Systems Architect, Horizons Consulting, Inc., United States

In previous versions of Exchange Server, when thinking of high availability and site resiliency, we often thought only of how to protect the mailbox database and how to make it available in another datacenter in the event that something happened to your primary copy. Although database availability and the DAG are still important factors in Exchange Server 2010, it is now equally important to consider the Client Access Server role and the overall namespace design and its impact on your high availability and site resiliency plan.

To account for the impact the namespace design has on availability, it is helpful to think about the different switchover/failover (*over) scenarios and the impact those *over scenarios have on all of the client connectivity types that your organization needs to support. When the namespace design has been drawn out, I recommend deploying the design in a lab environment so that the *over scenarios can be played out and the client types supported by the organization can be fully tested to gain the impact on users. It is important to note whether the client will continue to run without

interruption or will experience a brief disconnect and then automatically reconnect. Possibly, the client will reconnect, but only after a timeout value has been exceeded (for example: DNS resolver cache expiring). During the testing phase, you can also work out any intervention steps you must take to ensure a smoother transition during a failure.

After you have fully tested the client impact, it is important to document the results both for your design documentation and so that you can articulate the results to both your senior management and to the user community at large. In this way, you can set everyone's expectations properly and avoid confusion in the event that the unthinkable disaster happens.

*To visualize the different scenarios, it is often helpful to build a chart that allows you to track the success or failure of each client connection type given specific *over scenarios.*

	HIGH AVAILABILITY (SINGLE-SITE AND SINGLE-NAMESPACE)		SITE RESILIENCY (TWO-SITE AND TWO-NAMESPACE)	
CLIENT TYPE	**SWITCHOVER**	**FAILOVER**	**SWITCHOVER**	**FAILOVER**
OWA	No user impact (Success)	No user impact (Success)	No user impact (Success)	No user impact (Success)
Exchange ActiveSync 5/6	No user impact (Success)	No user impact (Success)	Client failure and profile must be manually updated (Failure)	No user impact (Success)
Exchange ActiveSync 6.1+	No user impact (Success)	No user impact (Success)	No user impact (Success)	No user impact (Success)
Outlook 2007/2010 (Outlook Anywhere)	Short client disconnect and reconnect (Success)	Short client disconnect and reconnect (Success)	No user impact (Success)	No user impact (If EXPR matches certificate CN) (Success)
Outlook 2007/2010 (Internal RPC)	Short client disconnect and reconnect (Success)	Short client disconnect and reconnect (Success)	No user impact (Success)	No user impact (If EXPR matches certificate CN) (Success)
POP3/IMAP4	No user impact (Success)	No user impact (Success)	Client failure and profile must be manually updated (Failure)	No user impact (Success)

Cross-site Switchover

Deploying a DAG across two sites can allow database copies to exist in two locations and provide site resiliency. This allows a single mailbox database to fail over and switch over to the secondary site. The client software will react to the changes in one of two possible ways when the active mailbox database is moved from one site to another. Understanding these reactions is important to ensuring that you perform the correct type of failover for your needs:

- The Client Access server will directly connect to the Mailbox server.
- The client will be redirected to connect to the second site, as shown in Figure 11-16.

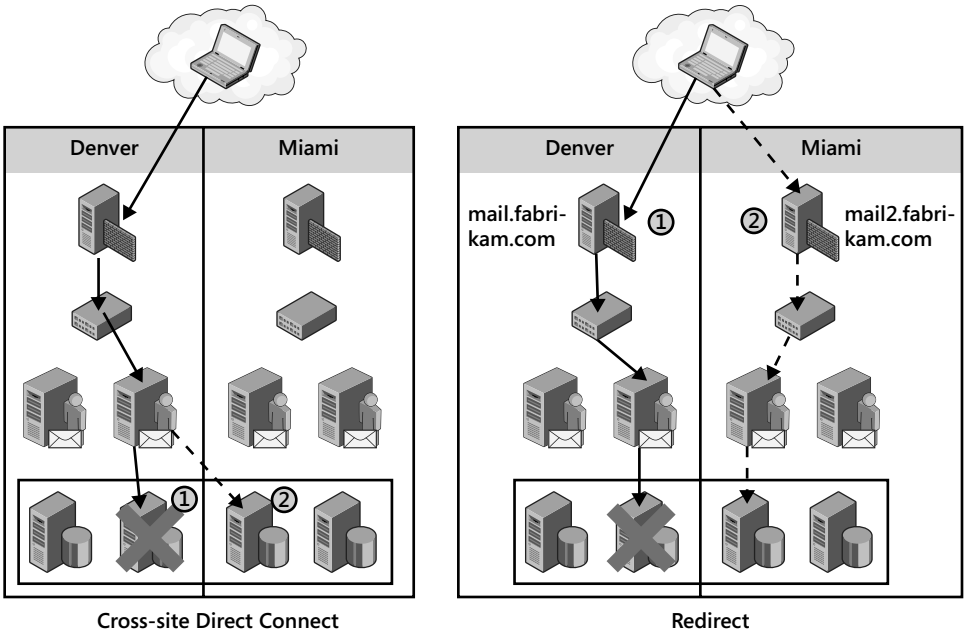

FIGURE 11-16 Comparing cross-site connections and redirect

Exchange 2010 SP1 includes functionality to control the connection behavior of Outlook when a cross-site database failover or switchover occurs. By default, Outlook will connect across from the primary Client Access server to the activated Mailbox server for temporary cross-site situations. Alternatively, the administrator can prevent all cross-site connections. Temporary and permanent cross-site moves are differentiated by the administrator explicitly resetting the database copy activation preference.

In the initial release of Exchange 2010, the default behavior is to perform a direct connect from the Client Access server array in the first datacenter to the mailbox hosting the active copy in the second datacenter. Redirection will only occur when the *RPCClientAccessServer* property is changed on the mailbox database. In SP1, you can choose to enable or disable cross-site direct connect and define an activation preference for a database.

The new SP1 behavior is based on the following three properties:

- Home server property in Outlook
- Preferred database site (*RPCClientAccessServer*)
- Active database site

Cross-site direct connect happens in the following scenarios:

- If the Outlook profile home server value, preferred database site, and mounted database site are the same, Outlook will connect (or stay connected) to the Client Access server array and that will connect to the Mailbox server cross-site.
- If the Outlook profile array site is the same as the preferred database site, and the mounted database site is different and cross-site connections are allowed, Outlook will connect (or stay connected) to the Client Access server array and will connect to the Mailbox server cross-site.
- If the Outlook profile home server property value is the same as the mounted database site, and different than the preferred database site, Outlook will connect (or stay connected) directly through the to the Client Access server array to the Mailbox server cross-site. This happens when you change the activation preference.

Redirection happens in the following scenarios:

- If the Outlook profile home server property value is different, and the preferred and mounted database sites are the same, the RPC Client Access service must redirect Outlook to the preferred and mounted database site and update the Outlook profile.
- If the Outlook profile home server property value is the same as the preferred database site, and the mounted database site is different, the Client Access server will redirect Outlook to the mounted database site if cross-site connections are not allowed.

Using cross-site direct connect is often suitable when a single mailbox server is undergoing maintenance or there are other temporary issues that will be resolved in a short period of time. Redirection may be needed when multiple systems or the entire datacenter will undergo maintenance. Performing a redirection switchover will force the clients to reconnect to the secondary site and allow maintenance to be completed. If redirection is used to switch over, it will also be done to perform the switchback to allow the clients to reconnect to the primary site. To enable cross-site direct connect, run *Set-DatabaseAvailabilityGroup <DAG Name> -AllowCrossSiteRpcClientAccess: $true* from the EMS. Conversely, to disable cross-site direct connect, run *Set-DatabaseAvailabilityGroup <DAG Name> -AllowCrossSiteRpcClientAccess: $false* from the EMS. To determine whether cross-site direct connect is enabled, run *Get-DatabaseAvailabilityGroup <DAG Name>* | *Format-List* as shown in Figure 11-17.

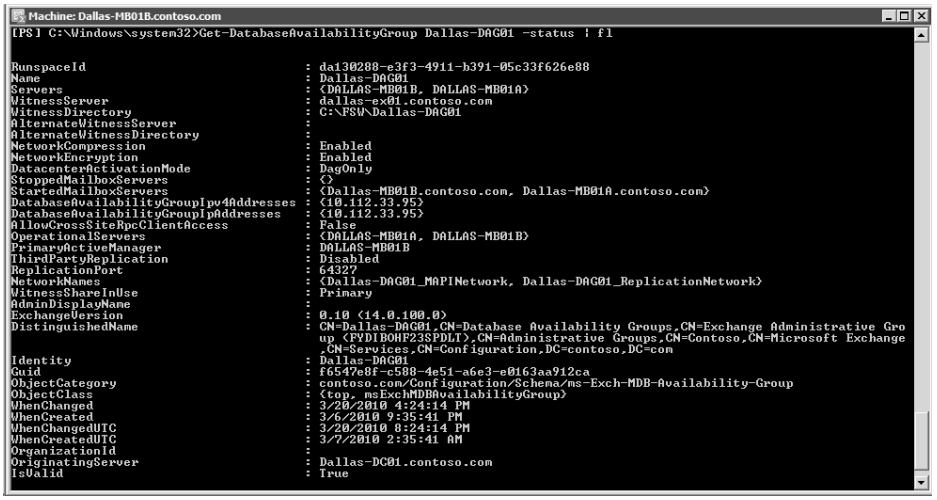

FIGURE 11-17 Retrieving the cross-site direct connect setting

Handling Datacenter Failures

To prepare for activating a secondary site in the case of a primary site failure, you must enable datacenter activation coordination (DAC) mode on the DAG by running *Set-DatabaseAvailabilityGroup <DAG Name> -DatacenterActivationMode:DagOnly*. Also in preparation you should also set the alternate witness server and alternate witness directory for a server available in the second site. This allows an administrator to activate the site even if a majority DAG members remain unavailable in the failed site, and it prevents split-brain scenarios. The Active Directory site defines the datacenter boundaries; therefore, to enable DAC mode, the DAG must span at least two sites. A datacenter failure is a catastrophic event because such a failure requires an administrator to make the decision to perform a full datacenter switchover, because the process is not automatic. The datacenter switchover process includes the following steps:

1. Evaluate the situation and then decide to perform a datacenter switchover.
2. Configure the DAG to remove the primary site's servers from the Windows Failover Cluster, but retain them in the DAG. This is done by running *Stop-DatabaseAvailabilityGroup <DAG Name> –ActiveDirectorySite <Primary Site Name> -ConfigurationOnly* in the primary site, if possible.
3. Configure the DAG to use an alternate witness server and restore the functionality in the secondary site. To do this, first stop the cluster service on each of the secondary site's DAG's servers, and then run *Restore-DatabaseAvailabilityGroup <DAG Name> -ActiveDirectorySite <Secondary Site Name>*.

4. Start the cluster service on each of the servers in the DAG in the secondary site. The remaining Active Managers will then coordinate mounting databases in the secondary site.
5. Adjust DNS records, if necessary, for Simple Mail Transfer Protocol (SMTP), OWA, Autodiscover, and Outlook Anywhere. These adjustments can be done manually or automatically using a third-party global-server load balancer.

After the primary site is recovered you may choose to perform a switchover to the primary site. This process includes the following steps:

1. Evaluate the situation and decide to perform a datacenter failback. Verify that the primary datacenter is capable of hosting Exchange services.
2. Reconfigure the DAG to add the DAG members in the primary datacenter back into the failover cluster by running *Start-DatabaseAvailabilityGroup <DAG Name> –ActiveDirectorySite <Primary Site Name>*.
3. Configure the DAG to use the primary site's witness server by running *Set-DatabaseAvailabilityGroup <DAG Name> –WitnessServer <Primary Site Witness Server>*.
4. Manually reseed or allow replication to update the primary datacenter's database copies, depending on the state of the primary site copy.
5. Schedule downtime for the mailbox databases and then dismount them.
6. Move databases back to the primary datacenter by running *Move-ActiveMailboxDatabase <Database> –ActivateOnServer <Server in Primary Site>*, and then mount the databases in the primary datacenter.
7. Adjust DNS records, if necessary, for Simple Mail Transfer Protocol (SMTP), OWA, Autodiscover, and Outlook Anywhere. These adjustments can be done manually or automatically using a third-party global-server load balancer.

In Exchange Server 2010 DAC mode tasks are available to restore service in a standby datacenter while a minority of the DAG members are available. Prior to SP1, DAC mode was limited to at least three members in the DAG. In that three-node DAG, two members needed to be in the primary datacenter (Active Directory site). In SP1, DAC mode has been improved to support a two-member DAG with a member in each datacenter. As with all DAGs with an even number of members, this implementation requires a witness server to provide the additional vote to obtain quorum.

Cross-site Best Practices

You can use the best practices described in this section to ensure a successful, highly available, multiple-site configuration. First, you can reduce failover times by lowering the Time to Live (TTL) on DNS records for the Client Access server array, Client Access server URLs, and SMTP records. A low TTL reduces the time it takes DNS clients to discover the DNS entries

that point to the secondary site. If any client computers that use DNS services are outside of your control, such as a regional ISP, be sure to verify that these services will honor any TTLs set—this will impact service availability for these users. By default a DAG is configured to only compress and encrypt transaction log shipping across different subnets. To take advantage of network compression between sites, you must manually enable intersubnet compressing and encryption.

Never wait until a failure occurs to ensure that everything works as designed. You should continually monitor and verify that all messaging-system components are functioning properly. This is done by monitoring all aspects of the Exchange Server environment to ensure that it is functioning normally, and that mailbox data is successfully replicating to the secondary site in a timely manner. You should also schedule periodic switchover tests to provide an additional level of preparation and to validate the configuration and operation of the cross-site switchover process. Switchover tests are usually coordinated events where the primary servers are shut down cleanly to reduce the possibility of data loss. When performing these drills be sure to verify that you are not missing steps that would be required in a real switchover scenario where the primary datacenter becomes unavailable.

You should also follow a change management process to ensure that each Mailbox server in the DAG, each Client Access server, and each Hub Transport server are configured identically with the same updates applied. Doing so reduces the possibility of incompatibilities and unexpected behavior if a *over occurs.

Provide adequate bandwidth for replication traffic. Replication is always from source to target; therefore, multiple copies in the remote site means more bandwidth is required. To reduce the amount of bandwidth needed you should be sure that compression is enabled on the log shipping traffic for the DAG. The Exchange 2010 Mailbox calculator can be used to help estimate the bandwidth required.

Finally, you should have each DAG node connected to multiple networks. These multiple networks provide communication redundancy between DAG nodes and segregate MAPI and replication communications. To reduce network congestion and potential communications problems, you should not allow the DAG networks to route between each other. For example, you would not allow the replication network to communicate with the MAPI network or vice versa. This communication should be blocked by the network equipment, with a router or a firewall.

Multi-Site Storage Architecture

You must consider a number of factors when determining the hardware needed to support your highly available Exchange deployment, as discussed in detail in Chapter 13, "Hardware Planning for Exchange Server 2010." Having multiple database copies requires storing data on multiple disks; this reduces the requirement for having RAID-protected storage because the data is redundantly stored. Deployment decisions for RAID or JBOD should be based on cost, performance, IT operational maturity, and required resilience. To provide for storage failures, redundancy is either provided by having additional database copies or by using RAID on the storage. Table 11-6 summarizes instances when RAID or JBOD should be considered.

TABLE 11-6 Choosing Between RAID and JBOD

	2 HIGH-AVAILABILITY COPIES	3 + HIGH-AVAILABILITY COPIES	2 + HIGH-AVAILABILITY COPIES / DATACENTER	1 LAGGED COPY	2 + HIGH-AVAILABILITY COPIES AND 1 + LAGGED COPIES / DATACENTER
Primary Datacenter	RAID	RAID or JBOD	RAID or JBOD	RAID	RAID or JBOD
Secondary Datacenter	RAID	RAID or JBOD	RAID or JBOD	RAID	RAID or JBOD

Risk Mitigation

Achieving high availability requires that risks are identified and addressed. Many organizations employ risk management practices to capture and address potential disruptions to business processes. These practices usually consist of the following phases:

- **Identification** This phase includes the documentation of areas of risk within the business. These range from loss of a large customer and the associated revenue all the way to a disaster that destroys a company datacenter.
- **Assessment** This phase includes the analysis of the identified risks to determine the probability and the impact of each.
- **Mitigation** This phase includes creating a plan for mitigating each potential risk. The mitigation plans for each risk fall into the following three categories:
  - **Acceptance** This is done when a risk is accepted, usually because the probability of occurrence is so low it doesn't require mitigation or the cost outweighs the consequences of the risk. A risk that might fall into this category is the probability of datacenters that are 20 miles apart being affected by the same tornado. Although this is possible, the likelihood is so small that is acceptable.
  - **Transference** This is done when the risk is mitigated by obtaining insurance or by outsourcing the risk to others to manage. A risk that might fall into this category is outsourcing inbound anti-spam and antivirus services to Microsoft Exchange Hosted Services to handle inbound e-mail.
  - **Reduction** This is done when the risk can be managed to a point where it is less probable or can be recovered from quickly. A risk that might fall into this category is deploying a cross-site DAG in two datacenters to reduce the likelihood that a single site failure can cause a messaging system outage.
- **Implementation** This phase includes putting the risk mitigation into practice.
- **Review** This phase evaluates the risk mitigation plan to verify that it has addressed the identified risks and to evaluate whether any new risks have been introduced.

Not only should risk management be practiced at the business level, but it must also be performed for IT solutions, such as the Exchange messaging environment. As you perform risk identification for your messaging environment you may list disk failure, server motherboard failure, loss of Internet connectivity, security breaches, site failures, and employee mistakes as risks. The assessment and mitigation process may create a list similar to the one in Table 11-7.

TABLE 11-7 Exchange Risk Mitigation

RISK	MITIGATION
Mailbox Server Disk Failure	Reduction: Use a RAID configuration or rely on DAG replication.
Server Motherboard Failure	Reduction: Use a DAG for Mailbox servers and deploy multiple Transport and Client Access servers.
DNS Server Failure	Reduction: Deploy multiple DNS servers and configure servers to use them.
Domain Controller Failure	Reduction: Deploy multiple domain controllers in each site.
Network Device Failure	Reduction: Deploy redundant network devices.
Loss of Internet connectivity	Reduction: Add additional Internet providers. Transference: Host servers in a colocation facility.
Security Breaches	Reduction: Good update management; implement intrusion detection and prevention systems. Transference: Outsource security to an experienced third-party provider.
Site Failures	Reduction: Deploy a failover site.
Employee Mistakes	Reduction: Provide training for employees and automate many common tasks.

One of the best ways to mitigate risk is to periodically test any disaster avoidance or recovery practices that have been put into place. This allows these measures to be tested and refined in a controlled environment, and in the end reduces risk. Often small details can be overlooked in a plan that cause delays in the recovery. For some organizations the primary datacenter is colocated in the same facility as the office space. In a situation where the primary facility is no longer viable and the IT systems are operational in the secondary datacenter, the users will still need another location to work. The processes and procedures for accessing the new location and notifying customers must also be worked out.

These fire drills also provide the opportunity to teach the employees the importance the business places on recovery and reinforces the mind-set to work toward that goal during all of their day-to-day responsibilities.

Pulling It All Together

The following sections review how each of this book's case studies implement their high-availability Exchange 2010 environment.

Contoso Case Study

The first case study is from Chapter 2, as shown in Figure 11-18.

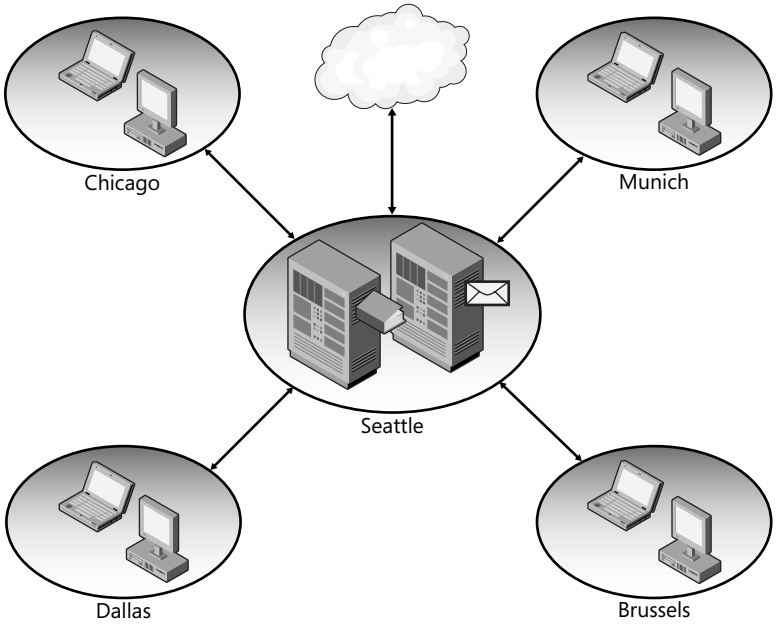

FIGURE 11-18 Contoso logical architecture

In light of high-availability requirements, Contoso proposes deploying a two-server high-availability solution. This two-server solution has both servers running Client Access, Hub Transport, and Mailbox server roles and is configured in a DAG. A file server in the site is the witness server. Although Contoso needs to purchase a third-party load-balancing solution because NLB is not supported on the same servers running Windows Failover Clustering, the money saved by purchasing half the number of servers more than makes up for the cost of the hardware load balancer, as shown in Figure 11-19.

The administrator creates an RPC Client Access array object named outlook.contoso.com and ensures that the each mailbox database has the *RpcClientAccessServer* property set to that value for the Outlook clients. The internal and external URLs also need to be updated with the load-balanced FQDN, including the *AutoDiscoverServiceInternalURI*. Because there are no proxy sites, all services should use the load-balanced FQDN. The *ExternalHostname* for Outlook Anywhere should be configured to match the certificate principle name, mail.consoto.com. An administrator will configure the load balancer with two virtual IP addresses (VIPs) that load balances both servers. The administrator will then create a DNS A record entry for mail.contoso.com and outlook.contoso.com, both pointing to separate VIPs on the load balancer.

The Contoso IT staff members have decided to deploy RAID-protected storage for each server to ensure adequate data resiliency in the event of a disk failure.

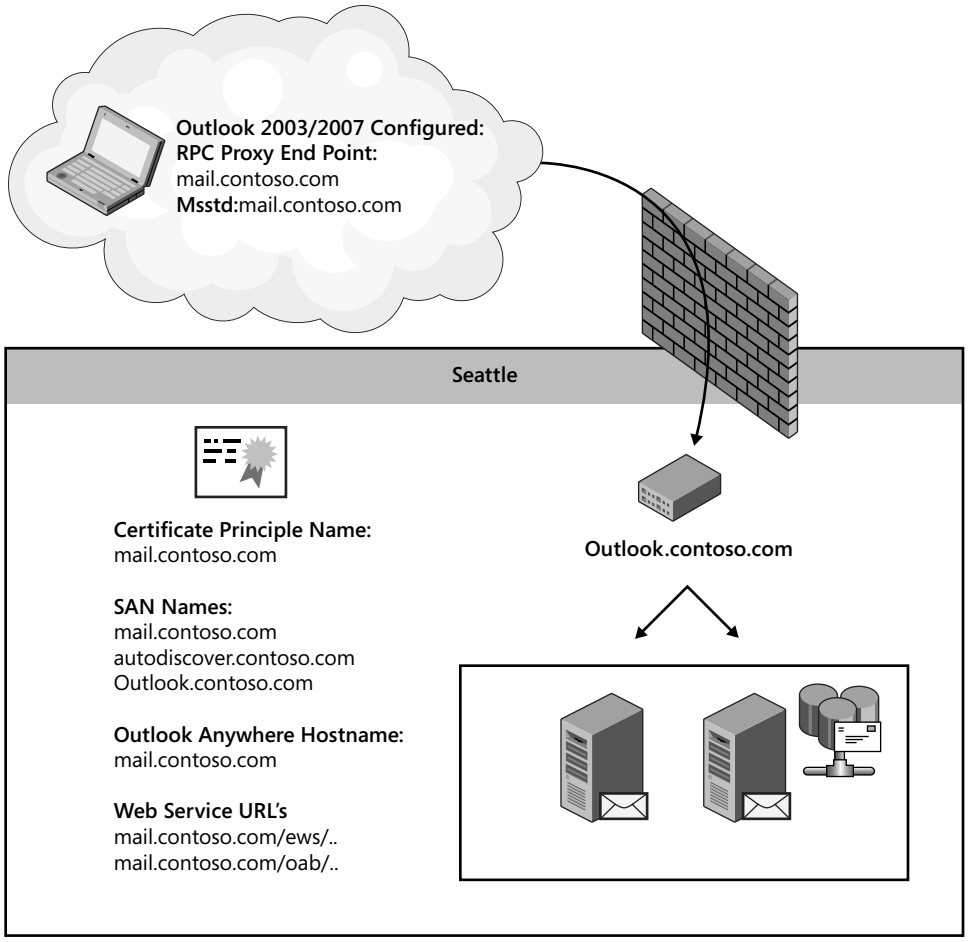

FIGURE 11-19 Proposed Contoso architecture

Fabrikam Case Study

The second, more complex case study from Chapter 2 is Fabrikam. As shown in Figure 11-20, Fabrikam has two main datacenters that will host Exchange services for all of their other sites. The Mailbox servers on the Denver site will host mailboxes for users in Phoenix, Portland, and Denver. The Mailbox servers on the Miami site will host mailboxes for user located in London, Toronto, and Miami.

In the Denver and Miami datacenters Fabrikam has deployed two hardware load-balanced Client Access servers, and the Mailbox servers are configured in a DAG. The network between the datacenters is adequate to host all client traffic both for everyday operations and in the event of a failover.

One of the key requirements for the Exchange 2010 deployment was providing a site-resilient solution. Therefore, Fabrikam has decided to migrate from their two Exchange 2007 Single Copy Clusters to a single DAG that spans both sites as shown in Figure 11-21.

In case of a Denver site failover, DAC mode is enabled and an alternate witness server is configured for a server on the Miami site.

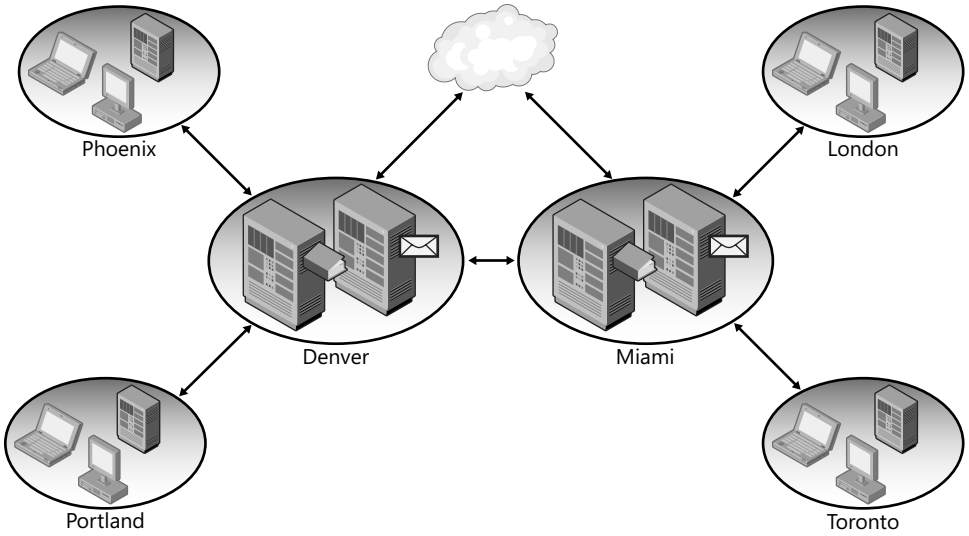

FIGURE 11-20 Fabrikam logical view

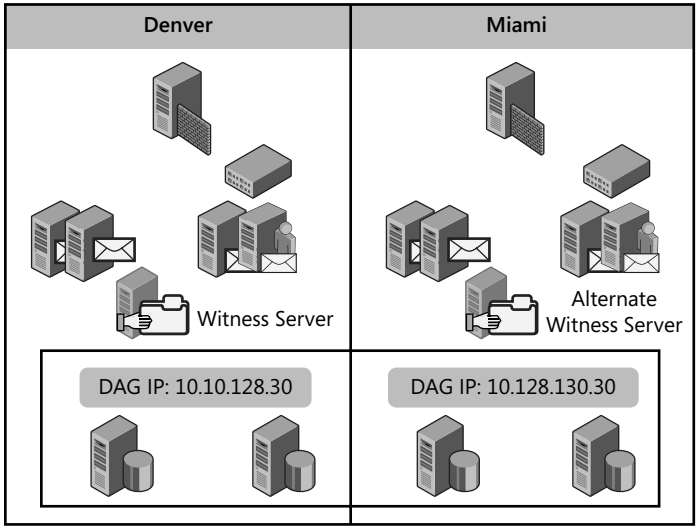

FIGURE 11-21 The Fabrikam high-availability deployment configuration

The Fabrikam Exchange 2007 deployment team followed best practice and chose to use a separate namespace for each Active Directory site. Fabrikam users in Denver access OWA using *https://mail.denver.fabrikam.com/owa,* whereas users in Miami use *https://mail.miami.fabrikam.com/owa*. To reduce confusion and support requests, Fabrikam has decided to consolidate the namespace and only instruct users to use *https://mail.fabrikam.com* for OWA, EWS, and IMAP communication. They have chosen to use GSLB to load-balance

mail.fabrikam.com and autodiscover.fabrikam.com and configure the GSLB to send client computer connections to the site that is geographically closest. For example, a user connecting over the Internet from Atlanta will be directed to the IP address for the Miami Client Access servers; a user connecting from Fresno will be directed to the Denver Client Access servers. In the event that a client is connected to a site that does not host the active copy of the user's mailbox, the Client Access server will be able to use the connectivity between the datacenters because the DAG has been configured to allow cross-site direct connect.

Fabrikam has approximately 7,000 mailboxes evenly distributed between the two sites. Twenty-four databases will be created, and each will handle almost 300 mailboxes, all with a 1-GB storage quota. The database high-availability plan defines that each database will have two copies on the primary site and one copy on the secondary site. Table 11-8 summarizes the database high-availability plan. The plan does not provide for three mailbox database copies at each site; therefore, the team has chosen to deploy RAID-protected storage for all of their mailbox database storage.

TABLE 11-8 Fabrikam's Mailbox Database Copy High-Availability Plan

	DENVER-MB01A	DENVER-MB01B	MIAMI-MB01A	MIAMI-MB01B
DAG01-DB1	**Active**	Copy 1	Copy 2	No Copy
DAG01-DB2	Copy 1	**Active**	No Copy	Copy 2
DAG01-DB3	No Copy	Copy 2	**Active**	Copy 1
DAG01-DB4	Copy 2	No Copy	Copy 1	**Active**
DAG01-DB5	**Active**	Copy 1	Copy 2	No Copy
DAG01-DB6	Copy 1	**Active**	No Copy	Copy 2
DAG01-DB7	No Copy	Copy 2	**Active**	Copy 1
DAG01-DB8	Copy 2	No Copy	Copy 1	**Active**
DAG01-DB9	**Active**	Copy 1	Copy 2	No Copy
DAG01-DB10	Copy 1	**Active**	No Copy	Copy 2
DAG01-DB11	No Copy	Copy 2	**Active**	Copy 1
DAG01-DB12	Copy 2	No Copy	Copy 1	**Active**
DAG01-DB13	**Active**	Copy 1	Copy 2	No Copy
DAG01-DB14	Copy 1	**Active**	No Copy	Copy 2
DAG01-DB15	No Copy	Copy 2	**Active**	Copy 1
DAG01-DB16	Copy 2	No Copy	Copy 1	**Active**
DAG01-DB17	**Active**	Copy 1	Copy 2	No Copy
DAG01-DB18	Copy 1	**Active**	No Copy	Copy 2
DAG01-DB19	No Copy	Copy 2	**Active**	Copy 1
DAG01-DB20	Copy 2	No Copy	Copy 1	**Active**

	DENVER-MB01A	DENVER-MB01B	MIAMI-MB01A	MIAMI-MB01B
DAG01-DB21	**Active**	Copy 1	Copy 2	No Copy
DAG01-DB22	Copy 1	**Active**	No Copy	Copy 2
DAG01-DB23	No Copy	Copy 2	**Active**	Copy 1
DAG01-DB24	Copy 2	No Copy	Copy 1	**Active**

Fabrikam uses a hosted service for all inbound e-mail traffic. The hosted service provides redundancy and an SLA that meets the business and technical requirements for Fabrikam's Exchange 2010 deployment project. To provide redundancy for inter- and intra-site message transport, two Hub Transport servers will be deployed to each site.

Litware Case Study

The last case study is for the global company Litware, Inc., as shown in Figure 11-22.

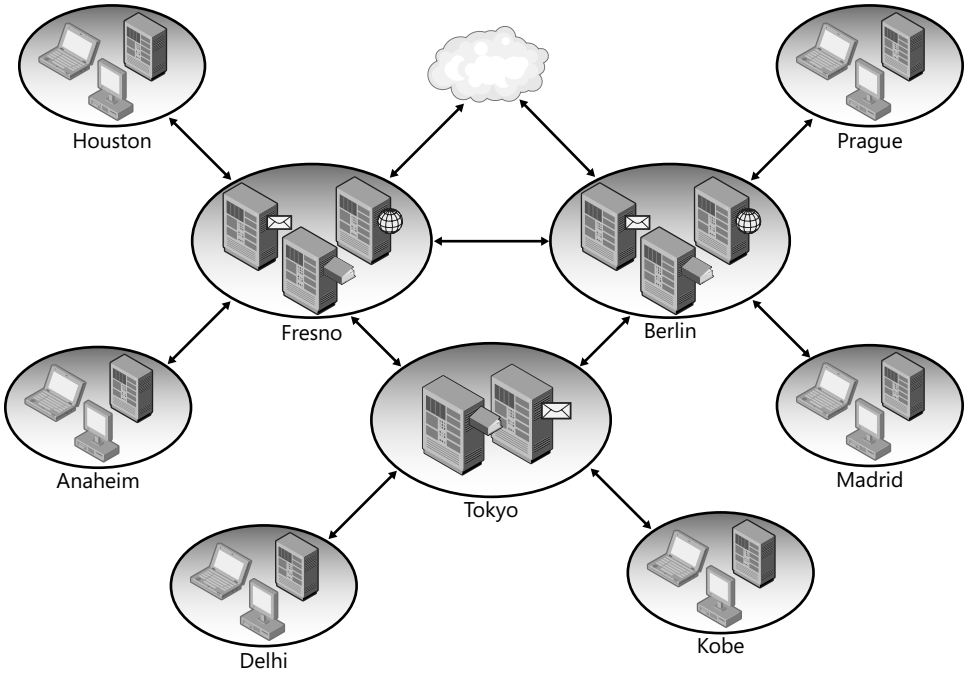

FIGURE 11-22 Litware, Inc., logical view

To reduce network costs, Litware uses regional namespaces to ensure that client traffic does not traverse the company WAN. Litware will deploy a high-availability solution within the regional datacenter, either with a software or hardware load balancer. Only the three hub sites—Fresno, Berlin, and Tokyo—have Exchange servers. The spoke sites, such as Kobe, Prague, and Madrid, will not have Exchange servers locally. Fresno, Berlin, and Tokyo will replace *Region* with the region-specific information. Tokyo, for example, will use *Tokyo.litwareinc.com* as their URL for OWA.

To provide e-mail message ingress and egress traffic to the Internet, the Edge Transport servers in Fresno and Berlin will all have a single MX record published using GSLB. When a request for the MX record for fabrikam.com is processed by GSLB, it will return the IP address for the MX record for the Edge Transport server closest to the sender. For example, if a SMTP server in Leipzig is sending an e-mail to Jeff@Fabrikam.com, GSLB will return the IP address for one of the Edge Transport servers in Berlin.

The Exchange deployment team at Litware decided to deploy their DAG using JBOD and without backups, which requires a 12-node DAG to be deployed in Fresno, Berlin, and Tokyo. To support the DAG in each location, nine Client Access servers are needed. This number was determined by using the CPU core ratio of four Mailbox server CPU cores for every three Client Access server CPU cores, will be deployed. Also, using the CPU core ratio of five Mailbox server CPU cores for every one CPU core on a Hub Transport server that is also running antivirus, five Hub Transport servers will be deployed at each location, as shown in Figure 11-23.

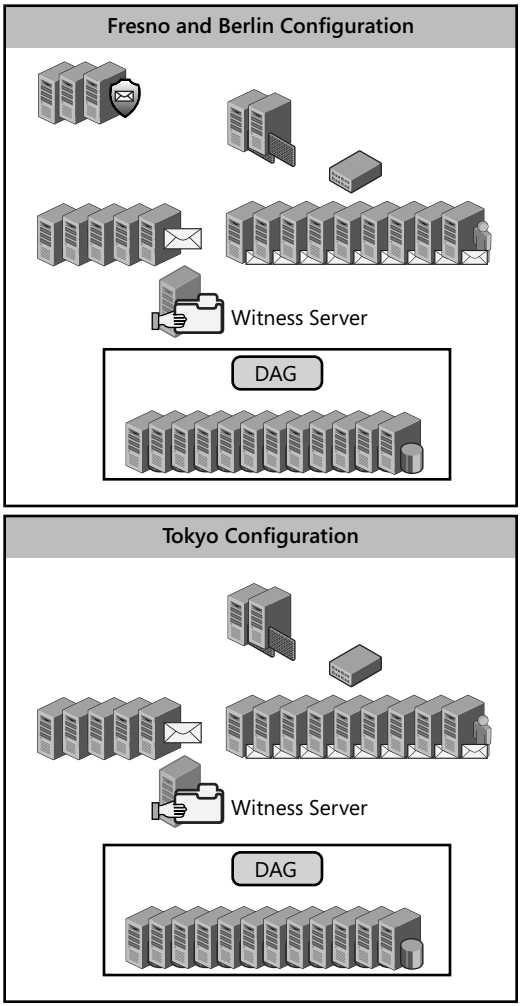

FIGURE 11-23 Litware high-availability deployment configuration

To meet Litware's business and technical requirements, each mailbox database will have three current copies and one database copy lagged for 14 days. This configuration provides enough data redundancy to eliminate having to RAID database storage and provides a lagged copy in case a point-in-time copy is required. Because of the large number of mailbox copies required, the copies will be distributed randomly across the DAG nodes and each month the Exchange administers will run an automated report to examine server load and copy distribution to determine whether adjustments need to be made to the copy distribution.

Additional Resources

- Simple Mail Transfer Protocol: *http://www.ietf.org/rfc/rfc2821.txt*
- Exchange 2010 Help: Planning for High Availability and Site Resilience: *http://technet.microsoft.com/en-us/library/dd638104.aspx*
- Exchange 2010 Help: Understanding Active Manager: *http://technet.microsoft.com/en-us/library/dd776123.aspx*

Index

Symbols and Numbers

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

About the Authors

SIEGFRIED JAGOTT works as a Principal Consultant and Team Lead for the Microsoft Messaging and Collaboration team at Siemens AG, located in Munich, Germany. He is part of the Siemens central architecture team that works closely with Microsoft to plan future enhancements of not only Windows and Exchange but also other products such as System Center Virtual Machine Manager (SC VMM). He has been involved in Microsoft technology adoption programs since Exchange 2000 and has been working with Microsoft Exchange since Exchange Server 4.0. In the past 15 years he has been involved in planning, designing, and implementing some of the world's largest Windows and Exchange infrastructures for various international customers, including Siemens itself.

Besides this, Siegfried is a frequent writer for various international magazines such as the *Windows IT Pro Magazine* and speaks at conferences about Windows- and Exchange-related topics. He is coauthor of *MCITP: Microsoft Exchange Server 2007 Messaging Design and Deployment Study Guide: Exams 70-237 and 70-238* (Sybex, 2008), *MCTS: Windows Server 2008 Applications Infrastructure Configuration Study Guide: Exam 70-643* (Sybex, 2008) and the Microsoft course MOC 10135: Configuring, Managing, and Troubleshooting Microsoft Exchange Server 2010. In his spare time he likes to go skiing in the Alps and travel around the world to scuba dive. Siegfried lives in a small town called Rednitzhembach in southern Germany, where he is currently building a new house. He holds an MBA and a Diploma in Management from Open University in England and has been a Microsoft Certified Systems Engineer (MCSE) since 1997.

JOEL STIDLEY has been working in the IT field for 15 years, and he has been a computer fanatic for much longer. Joel has been working with Microsoft Exchange since the initial Exchange Server 5.0 beta release. He has also led an engineering team to create a shared Exchange 2000 hosting platform before Microsoft released guidance on how to do so. Since the release of Exchange 2000 Server Service Pack 3, he has participated in the Microsoft Exchange JDP and TAP programs. Currently, he is the Principal

Systems Architect at Terremark Worldwide, where he works with a variety of existing and future technologies related to virtualization, directory services, storage, and messaging. In his 10 years with Terremark, he has filled a number of key roles, including technical lead for creating and operationalizing the company's Infinistructure virtualization platform. He also is a Microsoft MVP, blogger, and author or coauthor of several other technical books, including *Professional Windows PowerShell for Exchange Server 2007 Service Pack 1* (Wrox, 2008), *MCTS: Microsoft Exchange Server 2007 Configuration Study Guide: Exam 70-236* (Sybex, 2009), *MCTS: Windows Server 2008 Applications Infrastructure Configuration Study Guide: Exam 70-643* (Sybex, 2008), and the Microsoft course MOC 10135: Configuring, Managing, and Troubleshooting Microsoft Exchange Server 2010. Joel started an Exchange community and blog Web site called *ExchangeExchange.com* in 2004 to provide a place for others interested in Exchange to share information.

Contributing Authors

ANDY SCHAN started out as an electronics technologist, working on ground aviation systems and radar installations, including the Distant Early Warning System in the high arctic; from there he moved to the installation, care, and feeding of linear electron accelerators for physics research and cancer treatment. Andy's IT career began more than 12 years ago with migrating MS Mail to Exchange 4.0 for an international software company spanning 30 Exchange sites worldwide. Since then, he has worked with every major Exchange release and was the team lead for EDS Canada's Messaging and Active Directory Engineering team. He also drove the first Canadian federal government migration of Exchange 5.5 to Exchange Server 2003, where he experienced the joys of synchronizing a 275,000-object directory between Exchange 5.5, Active Directory and Novell. In addition to numerous large-scale migration projects he has participated in the Exchange Server 2007 and Exchange Server 2010 TAP programs, contributed to an Exchange Server 2007 book, and deployed a message classification and policy solution to 125,000 users in a large military environment. He has also worked on a number of identity management projects, including Active Directory migrations, federation implementations, and digital rights management for environments as large as 525,000 seats, and has spoken at NetPro's Directory Experts Conference and Quest's The Experts Conference. He is currently working as Senior Consultant for Titus International in Ottawa, Canada. In his downtime, Andy can be found on Xbox Live.

JEFFREY ROSEN has a Master of Business Administration from Case Western Reserve Weatherhead School of Management specializing in Information Systems. He is a Microsoft Certified Architect and Microsoft Certified Master, and holds a MCSE specializing in messaging and security. He began his career working with Microsoft Mail and Novell Netware. Jeffrey has been working for Microsoft Consulting Services for 11 years, working on large and complex Exchange deployments. He is a coauthor of *Professional Windows PowerShell for Exchange Server 2007 Service Pack 1* (Wrox, 2008) and *Microsoft PowerShell, VBScript & JScript Bible* (Wiley, 2009). In his spare time, you may catch him on Xbox Live.