

Nathan Winters, Neil Johnson, and Nicolas Blank

Microsoft®
Exchange Server 2013

Design, Deploy, and Deliver an
Enterprise Messaging Solution



Microsoft[®] Exchange Server 2013

**Design, Deploy, and Deliver
an Enterprise Messaging
Solution**

Nathan Winters

Neil Johnson

Nicolas Blank

 **SYBEX[®]**
A Wiley Brand

Acquisitions Editor: Mariann Barsolo
Development Editor: Gary Schwartz
Technical Editor: Henrik Walther
Production Editor: Liz Britten
Copy Editor: Linda Recktenwald
Editorial Manager: Pete Gaughan
Production Manager: Tim Tate
Vice President and Executive Group Publisher: Richard Swadley
Vice President and Publisher: Neil Edde
Book Designers: Maureen Forys, Happenstance Type-O-Rama; Judy Fung
Proofreader: Daniel Aull, Word One, New York
Indexer: Ted Laux
Project Coordinator, Cover: Katherine Crocker
Cover Designer: Ryan Sneed
Cover Image: ©iStockphoto.com/Kalawin

Copyright © 2013 by John Wiley & Sons, Inc., Indianapolis, Indiana
Published simultaneously in Canada

ISBN: 978-1-118-54190-6
ISBN: 978-1-118-75027-8 (ebk.)
ISBN: 978-1-118-77953-8 (ebk.)

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (877) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2013937651

TRADEMARKS: Wiley, the Wiley logo, and the Sybex logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Microsoft is a registered trademark of Microsoft Corporation. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

10987654321

Dear Reader,

Thank you for choosing *Microsoft Exchange Server 2013*. This book is part of a family of premium-quality Sybex books, all of which are written by outstanding authors who combine practical experience with a gift for teaching.

Sybex was founded in 1976. More than 30 years later, we're still committed to producing consistently exceptional books. With each of our titles, we're working hard to set a new standard for the industry. From the paper we print on, to the authors we work with, our goal is to bring you the best books available.

I hope you see all that reflected in these pages. I'd be very interested to hear your comments and get your feedback on how we're doing. Feel free to let me know what you think about this or any other Sybex book by sending me an email at nedde@wiley.com. If you think you've found a technical error in this book, please visit <http://sybex.custhelp.com>. Customer feedback is critical to our efforts at Sybex.

Best regards,

A handwritten signature in black ink, appearing to read 'Neil Edde', written in a cursive style.

Neil Edde
Vice President and Publisher
Sybex, an Imprint of Wiley

To my wife, Elizabeth: You have supported, loved, and inspired me through so many challenges. Who would have thought having said “never again” after my first book that I would now have completed two more!

—Nathan Winters

I dedicate this book to the special ladies in my life: to my wife, Mandy, for standing by me while I sacrificed time to make this book happen, and to my daughters, Anna and Lisa. I would also like to thank God for the gift of communication that makes writing a book of this nature possible.

—Nicolas Blank

I would like to dedicate this book to my family, all of whom have supported me throughout the writing process in some way. I would especially like to thank Liz for her tolerance of me writing during the early hours and Leo for always being able to make me smile.

—Neil Johnson

Acknowledgments

As you can probably imagine, writing a book is a hefty task. It requires the inspiration and coordination of many different groups of people without whom it would not be possible. Therefore, I am very grateful for this opportunity to call them out for some well-earned recognition.

Throughout this book, we have worked to ensure that you get the very best advice. This has meant working with our accumulated network of friends and colleagues, calling in favors to ensure that we take advantage of the insights of the top experts in their fields.

I would like to start by thanking Neil Johnson and Nicolas Blank personally. Both fall into the friends and colleagues category. We have worked together in various capacities over the last several years, and without their knowledge, enthusiasm, and sheer hard work, this book would absolutely never have happened.

Of course, outside of the authors, the other major driver behind a book like this is the publisher. As always, Sybex has helped us at every stage of the process with a superb team, starting with Mariann Barsolo, our acquisitions editor, who helped us shape the book and hone in on the audience, Gary Schwartz, our development editor, who put up with the random formatting and grammar that we came up with and turned it into something resembling what you see now, and Liz Britten and the copy editing team who did a crack job in getting this polished for printing while accommodating some late changes! Over all, Pete Gaughan had the job of pushing us to keep at least within some semblance of a schedule!

That leaves one very important person in the team, our technical editor, Henrik Walther. Henrik had a big part to play in ensuring the technical accuracy of the examples throughout the book. As someone with huge experience in the Exchange world, he also provided useful guidance and thoughts on the project as a whole.

As previously mentioned, one of the things that made this book possible was our close network of colleagues who contributed. I would like to give special thanks to the following group of people who contributed significant chunks—up to and including whole chapters.

- ◆ Bhargav Shukla, Director of Product Research and Innovation at KEMP Technologies
- ◆ Ruth Bacci, Exchange Consultant at Microsoft UK
- ◆ Glen Scales, Exchange MVP who works as a freelance developer and engineer specializing in Exchange development
- ◆ Steve Goodman, Exchange MVP who works as a technical architect at one of the UK's leading IT services providers
- ◆ Nic Bishop, Exchange Technical Solution Professional at Microsoft UK

I would also like to call out specifically members of the Exchange product group who provided support, guidance, and material:

- ◆ Julie Xu, Principal Group Program Manager, and Thomson Qu, Program Manager
- ◆ Astrid McClean, Senior Program Manager, Quentin Christensen, Program Manager, and Ryan Wilhelm, SDE
- ◆ Greg Taylor, Principal Program Manager Lead

—*Nathan Winters*

There are many people who helped me form ideas or simply spent time with me talking about design and Exchange in general. Among these, I would most like to thank all of my colleagues and customers, without whom my contribution to this book would have been impossible.

I would also like to thank the following people specifically for their help:

- ◆ Matt Gossage for his help and insight on the storage and monitoring chapters
- ◆ Conrad Sidey for believing in me and making himself available to talk about pretty much anything that I needed to discuss
- ◆ Robert Gillies, Andrew Ehrensing, Ross Smith IV, Greg Taylor, Jeff Mealiffe, Ramon Infante, David Espinoza, John Rodriguez, Alexandre Costa, Michael Wilson, Brian Day, and Scott Schnoll, who all provided guidance to me about deploying or designing Exchange Server 2013

—*Neil Johnson*

I have many people to thank in writing this book, starting with the fine people at Wiley as well as my coauthors, Nathan Winters and Neil Johnson. I would like to extend my thanks to the following individuals:

- ◆ Joe Newbert for language and process in dissecting the finer points of requirements in Chapter 1
- ◆ Boris Lokhvitsky for your kindness and patience, making the mathematical world accessible to the common man in Chapter 13

—*Nicolas Blank*

About the Authors



Nathan Winters has worked in IT since graduating from the Royal College of Music (RCM) in 2003, where he studied the clarinet! His first job was at the RCM, migrating from Exchange 5.5 and Windows NT4 to Exchange and Windows Server 2003. Nathan has since worked in a variety of roles for Microsoft partners, including consultancy and practice management. He now works for Microsoft UK as a presales technical specialist. Throughout 2012 and 2013, Nathan has been a regular speaker at industry conferences, such as TechEd and Exchange Connections, both in Europe and in the United States. Before joining Microsoft, Nathan was active in the UK technical community, running the Exchange user group (MMMUG) and writing numerous articles for *Windows IT Pro* magazine and the MExchange.org website, among others. He was awarded the distinction of Microsoft MVP between 2006 and 2011. In addition to this book, Nathan has recently completed *Mastering Microsoft Lync Server 2013*, also published by Sybex/Wiley. On the rare occasions when he is not working, he enjoys wildlife photography and badminton.



Nicolas Blank is a messaging architect, author, and speaker focused on all things Exchange at NB Consult in South Africa. With over 14 years of experience with Exchange, Nicolas consults with customers on cloud-based and on-premises Exchange projects, as well as companies building Exchange-focused products. Nicolas currently holds the status of Microsoft Certified Master, Exchange 2010 and Office 365, and has received the Microsoft MVP award for Microsoft Exchange every year since 2007. Nicolas blogs regularly on Exchange and messaging topics at blankmanblog.com as well as tweeting as @nicolasblank.



Neil Johnson has worked in IT since leaving Derby University, where he studied engineering. Initially, he worked with Novell Netware and Unix but then quickly moved on to Windows NT and Exchange Server. Neil worked in a number of roles, including third-line support, field engineer, technical design authority, and systems analyst before joining Microsoft in 2006. Since joining Microsoft, Neil has led some of the largest and most complex Exchange deployments in the United Kingdom. He can often be found speaking at Microsoft internal and public events about Exchange or Exchange Online, and he is also an instructor with the rank of Microsoft Certified Solutions Master: Messaging. Neil writes for the Microsoft Exchange Team Blog (EHLO) and maintains both the Jetstress Field Guide and Exchange Client Network Bandwidth Calculator. Neil has a passion for motorsport and is a lifelong Williams F1 team supporter. In his limited spare time, Neil is a keen photographer and loves to explore the national forest woodlands in the midlands where he lives with his son, Leo, and partner, Liz.

Contents at a Glance

<i>Introduction</i>	<i>xix</i>
Chapter 1 • Business, Functional, and Technical Requirements	1
Chapter 2 • Exchange Design Fundamentals	11
Chapter 3 • Exchange Architectural Concepts	27
Chapter 4 • Defining a Highly Available Messaging Solution	59
Chapter 5 • Designing a Successful Exchange Storage Solution	79
Chapter 6 • Management	101
Chapter 7 • Exchange 2013 Hybrid Coexistence with Office 365	137
Chapter 8 • Designing a Secure Exchange Solution	159
Chapter 9 • Compliance	199
Chapter 10 • Collaborating with Exchange	231
Chapter 11 • Extending Exchange	261
Chapter 12 • Exchange Clients	299
Chapter 13 • Planning Your Deployment	321
Chapter 14 • Migrating to Exchange 2013	331
Chapter 15 • Operating and Monitoring Exchange Server 2013	355
<i>Index</i>	<i>373</i>

Contents

<i>Introduction</i>	<i>xix</i>
---------------------------	------------

Chapter 1 • Business, Functional, and Technical Requirements1

Building the Foundation for Requirements	1
Establishing Project Roles	2
Getting Started with the Exchange Design	2
Requirements as Part of a Larger Framework	3
Understanding the Types of Requirements	4
Business Requirements	4
Technical Requirements	6
Constraints	7
Assumptions	8
Requirements Elicitation	8
Summary	9

Chapter 2 • Exchange Design Fundamentals11

Introducing Design Documents	11
From Requirements to Design	11
No Single Way to Implement Exchange	12
How Much Detail Is Enough?	12
Section Guide	12
Section Index	13
Executive Summary	13
Business Requirements	14
Summary of Vision and Scope	14
Functional Specification	14
Architecture Summary	14
Compliance	15
External Publishing	15
Migration or Legacy Integration Requirements	15
Interoperation with Third-Party Applications	16
High-Availability Strategy and Requirements	16
Transport Design	17
Client Access Design	18
Mailbox Design	18
VM Requirements	19
Bandwidth Requirements	20
Exchange Solution Sizing	20
Moving Forward	24
A Living Document	24
How Do You Know When to Finish Designing?	24

Overengineering	25
Keep It Simple	25
Future Proofing	25
The Microsoft Way	25
Chapter 3 • Exchange Architectural Concepts	27
The Evolution of Exchange 2013	27
Exchange 2000/2003	28
Exchange 2007	30
Exchange 2010	34
Exchange 2013	39
Discontinued Features	42
Exchange 2013 Editions	42
Transport	42
Management	44
Role Separation	45
High Availability	54
Exchange Online Integration	57
Summary	57
Chapter 4 • Defining a Highly Available Messaging Solution	59
Defining Availability	59
Defining Availability Components	60
Defining the Cost of Downtime	62
Planning for Failure	63
Defining Terms for Availability	65
Service-Level Agreements	65
RPO and RTO	65
Defining High Availability and Disaster Recovery	66
Achieving High Availability	67
Building an Available Messaging System	69
Transport	69
Namespace Planning	69
Exchange Hybrid Deployment	72
Database Availability Group Planning	73
Summary	78
Chapter 5 • Designing a Successful Exchange Storage Solution	79
A Brief History of Exchange Storage	79
Exchange 4.0–5.5	79
Exchange 2000–2003	80
Exchange 2007	80
Exchange 2010	81
Storage Changes in Exchange 2013	82
Issue 1: Storage Capacity Increasing	82
Issue 2: Mechanical Disk IOPS Performance Not Increasing	83
Issue 3: JBOD Solutions Require Operational Maturity	85

Issue 4: Mailbox Capacity Requirements Increasing	86
Issue 5: Everything Needs to Be Cheaper	86
Storage Improvements in Exchange Server 2013	87
Automatic Database Reseed	88
Multiple Databases for Each JBOD Disk Spindle	88
Designing a Successful Exchange Storage Solution	90
Requirements Gathering	90
Making Sense of the Exchange Mailbox Server Role	
Requirements Calculator	93
Selecting the Right Storage Hardware	95
Storage Validation Using Jetstress	96
Summary	98
Chapter 6 • Management	101
Trends in Management of Platforms	101
Role-Based Access Control	102
RBAC Overview	103
Understanding the Components of the RBAC Permissions Model	104
Planning Your Management Strategy	105
Understanding Built-in Management Roles, Role Groups, and Role Association	107
Role Assignments	111
Under the Hood	112
Creating New Roles	113
Creating New Management Scopes	114
Creating and Managing Role Groups	115
Creating New Role Assignments	115
Understanding Role Assignment Policies	117
Applying Business Logic Using Unscoped Top-Level Roles	119
Reporting Effective Permissions and Cmdlet Usage	121
Understanding Split Permissions	123
Using EAC to Manage RBAC	125
Administration	127
The Exchange Management Tools	131
What's New in EAC?	132
Securing Access to EAC	134
Hybrid Deployments and EAC	135
PowerShell and Exchange Management Shell	135
Summary	136
Chapter 7 • Exchange 2013 Hybrid Coexistence with Office 365	137
What Is Exchange Hybrid?	137
High-Level Infrastructure Overview	137
Why Consider Exchange Hybrid?	140
Benefits of Exchange Online	140
Trade-offs of Exchange Online	141

Design Considerations	143
Solution Requirements	143
Solution Design	144
Proof of Concept	145
Deployment Planning and Preparation	145
Common Deployment Hurdles	150
Summary	156
Chapter 8 • Designing a Secure Exchange Solution	159
Why and What to Secure?	159
What Does Security Mean?	159
How Real Is the Threat Today?	160
What Is Necessary to Secure?	161
Handling Security Conversations	162
The Challenges	162
Trustworthy Computing	164
Designing a Secure Exchange Solution	170
Protecting against Malware and Spam	170
Protecting against Unauthorized Network Access	177
Protecting against Unauthorized Data Access	183
Security of Data in Transit	184
Security of Data at Rest	186
Security of Data in Long-Term Storage	193
Auditing and Reporting	193
Summary	197
Chapter 9 • Compliance	199
Overview of Messaging Compliance	199
Regulations	200
Designing Your Policies	203
Discussions with the Legal Department	203
Typical Requirements	203
Compliance Policy	205
Compliance Solutions	206
Exchange Functionality	206
Exchange 2013 Compliance Scenarios	209
Communication	229
Summary	229
Chapter 10 • Collaborating with Exchange	231
What Is Collaboration?	231
Basic Collaboration with Email	232
The Client Experience	232
Helping Users Learn to Collaborate	233
The Address Book: a Place to Find and Get to Know People	234
Shared Mailboxes	235

Creating and Managing Shared Mailboxes	236
Automatic Mailbox Mapping	237
Accessing Shared Mailboxes from Mobile Devices	237
Resource Mailboxes	238
Implementing Resource Mailboxes	238
Public Folders	240
Structure of Modern Public Folders	241
Distribution Groups	242
Site Mailboxes	245
Implementing Site Mailboxes	247
SharePoint 2013 Prerequisites	247
Configuring the SharePoint Server	248
Preparing the Exchange 2013 Server	255
Creating and Configuring a Connection from SharePoint to Exchange	256
Configuring the Connection from Exchange to SharePoint	259
Summary	259
Chapter 11 • Extending Exchange	261
Accessing Exchange Programmatically	261
Where Do I Start?	262
Taking EWS for a Test Drive without Writing Any Code	263
How Do You Connect Your Code to Exchange?	263
Where Do You Run Your Code?	263
Considerations for the Cloud	263
Choosing the Right API for Exchange Development in Exchange 2013	264
Other Exchange APIs	268
Exchange Web Services in Exchange 2013	268
EWS Managed API	269
Web Services Description Language Proxy Objects	269
Raw SOAP	269
Connection and Authentication	270
Accessing Mailbox Data	274
Searching for Items	278
In-Place eDiscovery in Exchange 2013	279
Creating Items Using Exchange Web Services	283
Other EWS features	286
Migrating a CDO 1.2 VBS Script to a PowerShell EWS Managed API Script	290
Connecting to the Target Exchange Mailbox	290
Establishing a Connection to the Mailbox's Contacts Folder	290
Filtering the Contents of the Contacts Folder for Those That Contain a Photo	291
Downloading the Contact Photo Attachment	291
Mail Apps for Outlook and the Outlook Web App	292
How Mail Apps Work	293
JavaScript API for Office	294
Permission Levels in Mail Apps	294
Using Exchange Web Services within Mail Apps	295

Getting Started with a Mail App 296
 Installing a Mail App 296
 Best Practices When Writing EWS Code 296
 Exchange, the Microsoft Stack, and Other Third-Party Products 297
 Summary 297

Chapter 12 • Exchange Clients 299

Types of Exchange Client 299
 Messaging Application Programming
 Interface (MAPI/RPC) 300
 Exchange Web Services 300
 POP/IMAP 301
 Web Browsers 302
 Exchange ActiveSync 303
 Collaboration Data Objects 304
 Why Does Client Choice Matter? 305
 User Experience 305
 Supportability 306
 Regulatory Compliance 309
 Organization Security Compliance 309
 Performing a Client Inventory 310
 Messaging API (MAPI/RPC) 310
 Web Clients (EWS, EAS, and OWA) 311
 POP3 and IMAP4 312
 Scripting 313
 Design Considerations 313
 Supportability 313
 Security 314
 Client Performance 315
 Network Usage 317
 Exchange 2013 User Throttling 318
 Summary 319

Chapter 13 • Planning Your Deployment 321

Exchange 2013 Information Resources 321
 Required Documentation 321
 Preparing Active Directory 322
 Extending the Schema 322
 Creating or Updating the Exchange Organization 323
 Preparing or Updating Active Directory Domains 323
 Designing a Rollout Process 323
 Installing into an Existing Organization 324
 SMTP Considerations for Existing Organizations 325
 Certificate Considerations 325
 Choosing a Load Balancer 326
 Making the Choice 326

Deploying Operating System-Based Antivirus Programs	327
Firewalls and Exchange	327
Publishing Exchange to the Internet.	328
Preparing Clients	328
Preproduction Load Testing.	329
User Acceptance Testing	329
Summary.	330
Chapter 14 • Migrating to Exchange 2013.	331
Inter-Org Migrations	331
Outlook Client Reconfiguration	331
Availability Data Sharing	332
Global Address List Synchronization	332
Public Folder Data Synchronization	333
Mail Flow	333
Mailbox Permissions	334
Mobile Device Reconfiguration	334
External URL Publishing.	335
Exchange Application Integration	335
Offline Address Book.	336
Distribution Groups	336
Intra-Org Migrations	336
Outlook Client Reconfiguration	337
Availability Data Sharing	337
Global Address List Synchronization	337
Public Folder Data Synchronization	337
Mail Flow and Mailbox Permissions	337
Mobile Device Reconfiguration	338
External URL Publishing.	338
Exchange Application Integration	338
Offline Address Book	338
Distribution Groups	338
Moving Mailboxes	338
Mailbox Replication Service	339
Preparing for Inter-Org Mailbox Moves	340
Storage Capacity	342
Content Indexing	343
Modern Public Folder Data Migration	343
Intra-Org Migration to Exchange Server 2013	345
Foreign Systems	346
Lotus Notes	346
Novell GroupWise	347
Other IMAP	347
Legacy Exchange Migrations	348
Version-to-Version Upgrade	348
Double-Hop Inter-Org Migration.	349

Migrating to Office 365	349
Migrating to Exchange Server 2010	349
Common Migration Problems	349
Failure to Get Business Support	350
Insufficient Planning	350
Incorrect End-User Expectations	351
Seamless vs. Velocity	351
Application Integration	352
Compliance	353
Migration Improvements in Exchange 2013	353
Batch Moves	353
Migration Endpoints	353
Summary	354
Chapter 15 • Operating and Monitoring Exchange Server 2013	355
Monitoring	356
Alerting	357
Reporting	358
Types of System Availability	358
Trending	358
Inventory	365
Monitoring Enhancements in Exchange 2013	367
Managed Availability	367
Workload Management	369
Summary	371
<i>Index</i>	373

Introduction

This book came about after several conversations among the authors that focused on the common problems we found in discussing Exchange architecture with clients. We all felt that while Exchange is a very mature product, many people, from small in-house shops to the largest consultancies, often fall into similar traps when designing, deploying, and delivering a new Exchange-based messaging platform. To this end, we wanted to capture our conversations and experiences with customers and share them more widely in the hope of helping others avoid the common pitfalls we've seen.

Who Should Read This Book

This book is aimed at those among you who are going to be, or are, working closely with Exchange in a design and deployment capacity. You could be working for an in-house IT department as the messaging lead or as part of the messaging team. You could equally be a consultant working for one of the myriad of Microsoft partners who specialize in Exchange. That doesn't necessarily mean that you have to consider yourself an Exchange specialist, however. The point of this book is to help you get it right when you come to run your Exchange project. We appreciate that you won't necessarily be doing this day in and day out, because we all know that IT departments do a messaging upgrade every two to five years. A secondary audience is the architect community: those of you who are supervising the Exchange project as a program of work but are not necessarily involved in every day-to-day aspect.

What You Will Learn

Instead of focusing, as so many books do, on the "click Next" type guidance, we felt it was far more important to teach how to think about an Exchange project. Of course, we have plenty of technical material in the book, and we've made a point to call out where to find more, whether it's on TechNet or on third-party sites. Likewise, we have called on colleagues in the Exchange product group on many occasions to help us give not only the view from the field but also the thought processes behind the creation of Exchange; that is, how it was envisioned.

What's Inside?

This book is very straightforward in structure. In essence, it was conceived as a series of essays on the topics outlined. As such, is it not necessary to read the book from cover to cover, though some may find that useful. We have tried to lay things out in a manner that would make the most sense.

Chapter 1: Business, Functional, and Technical Requirements The goal of this chapter is to help you address and answer questions from the people around you in the form of a common language. *Requirements* are essential for implementing Exchange 2013 successfully. Exchange brings a huge number of features to the table. How do you choose which features to implement and how specifically should they be implemented? The answer is requirements!

Chapter 2: Exchange Design Fundamentals In Chapter 1, we introduce requirements elicitation as we grapple with the nuances of the different types of requirements and how to distill those into a readable form. In this chapter, our goal is to transform those requirements into design decisions. We examine the structure of the design document and delve into the content that goes into each section, which includes a discussion of sizing Exchange. We cover the concept that a good design document is not a purely technical record, nor is it purely a business- or project-based one. A well-written design document is intended for many audiences, from the technical implementers on various teams right up to the CEO.

Chapter 3: Exchange Architectural Concepts The aim of this chapter is to define the concepts upon which Exchange is built and then to extrapolate those concepts into the design choices that have resulted in the Exchange 2013 version. If you are a messaging consultant or administrator faced with upgrading from an earlier version of Exchange, then the history section in this chapter will help you address the architectural changes and features required to guide your customer through an upgrade to Exchange 2013. Knowing which features have changed, which have been discontinued, and which have been deemphasized is a critical skill for messaging administrators and consultants. We walk through each of the major functional areas of Exchange 2013 to get you up to speed with the latest in Exchange best practice.

Chapter 4: Defining a Highly Available Messaging Solution When eliciting requirements, desired availability is often one of the first topics to be raised. This chapter first seeks to define high availability (HA). We then look at which components help provide an HA solution before getting into depth about the configuration-related and operational practices required to ensure a high level of uptime.

Chapter 5: Designing a Successful Exchange Storage Solution Over the last 16 years, the adoption of Microsoft Exchange Server, which began in earnest with Exchange 5.5, has expanded dramatically. Email has become a pervasive application, and it is now a primary communication medium for most organizations. Over this time, Exchange storage has also undergone an evolution. Back in 1996, the focus was primarily on making the best use of costly hard disk capacity. In recent years, however, the focus has shifted toward being able to make use of larger and, most significantly, cheaper disks. This chapter covers the history of Exchange storage and then examines how to approach Exchange storage design in Exchange Server 2013, including the data you will need, the tools that are available, and how to validate the solution that you propose.

Chapter 6: Management Some of the biggest elements to consider when planning a new system are the management and operation of that system. After all, it is these elements that together form the majority of the cost of ownership. In this chapter, we focus on the management tools that are available for Exchange. In particular, we cover concepts such as Role-Based Access Control, which enables granular delegation of permissions to administrators. We also address new Exchange features, such as the move to a web-based management interface.

Chapter 7: Exchange 2013 Hybrid Coexistence with Office 365 Exchange hybrid is the term used when an Exchange organization running in a customer or partner datacenter is connected to Microsoft's Office 365. This configuration can provide an extremely rich coexistence feature set that allows mailboxes to be hosted on-premises or in Office 365, and the end-user experience remains virtually the same. This chapter discusses why you may want to evaluate Exchange hybrid, identifies the design considerations, and provides some tips from real-world deployments to help make sure that your Exchange hybrid project is a success.

Chapter 8: Designing a Secure Exchange Solution As a messaging consultant, you'll find that some of the most complex areas in designing a Microsoft Exchange solution are those relating to security. This chapter provides you with useful insights to prepare you for the awkward questions from your security officer and enable you to elicit the security requirements in a concise manner, thus avoiding the lengthy and soul-destroying security workshops and the spectrum of confusion that will arise from that disparity between the vision and the reality!

Chapter 9: Compliance This chapter takes you through the discussions that are needed about compliance. We introduce the regulations that businesses face, cover the conversations you need to have with the legal representatives from your organization, discuss the resulting policies that should be set, and then examine what Exchange functionality is available to implement those policies.

Chapter 10: Collaborating with Exchange Exchange is often defined as a groupware product. From the outset, it was intended to be a collaborative software suite. Although it has evolved over time, the fundamental purpose of Exchange remains to help people work better together. In this chapter, we discuss exactly what makes Exchange a collaborative product; that is, what is in the current version of Exchange that helps users collaborate right out of the box? Moreover, when you add in the full suite of Office Server products, how does this improve the end-user experience and allow users to get better value out of Exchange and its related products?

Chapter 11: Extending Exchange As a messaging platform, Exchange has grown over the years into a highly competent and cohesive system with a vast range of functionality. Nevertheless, that doesn't preclude the possibility of extending Exchange's functionality or leveraging it through other systems. This chapter examines the use of Exchange as a messaging platform to build upon, and it touches on where it resides within the Microsoft catalog of products. We begin by investigating the concepts and capabilities Exchange provides to developers for creating custom solutions, and we discuss the thought processes behind integrating Exchange with other Microsoft and non-Microsoft systems.

Chapter 12: Exchange Clients One of the reasons behind the success of Exchange Server is that it supports many client types. It is possible to connect to an Exchange mailbox from pretty much any operating system. The experience provided by clients varies so dramatically that end users may not even realize that they are using Exchange Server. This chapter addresses ways in which clients differ and how they may impact your design and deployment approach for Exchange Server 2013. We also discuss some features enhanced in Exchange 2013 that help protect your Exchange 2013 servers from rogue clients.

Chapter 13: Planning Your Deployment Deploying any version of Exchange for the first time can be a daunting task. Each version of Exchange has architectural considerations that are different from previous versions. Prerequisites may even change between service packs

as well as best practices relating to both the deployment and operation of Exchange. This chapter will help you to understand what makes Exchange 2013 different from other versions of Exchange, and it will help to ensure that you achieve a successful rollout.

Chapter 14: Migrating to Exchange 2013 Exchange migrations are just like any other type of solution deployment—they require practical planning to ensure success. This chapter outlines some of the more important aspects for migration planning and some common problems that you may experience in the field.

Chapter 15: Operating and Monitoring Exchange Server 2013 Why monitor and report on your Exchange service? Exchange monitoring, reporting, and alerting are fundamentally about one thing—keeping your messaging infrastructure running sufficiently to meet your service availability targets. Given this fact, it often surprises us that project teams will go to great lengths designing highly available clustered solutions, with overly complex redundant components, and then assume that installing an operations-monitoring product with its out-of-box configuration will be sufficient to keep things running. This chapter takes you through what you really must do to ensure successful monitoring and operations of Exchange.

Hardware and Software Requirements

You have a variety of options to test out the concepts in this book. You can go and start an Exchange deployment project—only kidding! Seriously, however, much of this book discusses concepts and thought processes rather than actual step-by-step technical procedures. Of course, those exist too. In order to immerse yourself into the actual technology, build a lab and get an Office 365 trial tenant. If you want to explore the basic functionality of Exchange, then an Office 365 tenant is one of the simplest ways to get up and running, because this allows you to test out the vast majority of client-side functionality and much of the administrative side without the need of servers. If the actual underlying workings of Exchange are important to you, then an on-premises lab is a necessity. In this case, much can be achieved on a single, well-specified machine. For example, a lot of the lab work for this book was created on a Dell T7500 workstation with five hard drives and 24 GB of RAM—a fairly lowly specified box these days!

How to Contact the Authors

We welcome feedback from you about this book. Obviously, it's always nice to get messages about what you liked about the book, but we also welcome suggestions for improvements that we could make in future editions. You can reach Nathan by writing to nathan@clarinathan.co.uk, Nicholas at nicholas@nbconsult.co.za, or Neil at neil.johnson@microsoft.com. If you are looking for information about Nathan's future articles or would like to discuss a speaking engagement, visit Nathan's blog at www.nathanwinters.co.uk.

Sybex strives to keep you supplied with the latest tools and information you need for your work. Please check www.sybex.com/go/exchangedesigndeploy, where we'll post additional content and updates that supplement this book should the need arise.

Chapter 1

Business, Functional, and Technical Requirements

The goal of this chapter is to help you address and answer questions from the people around you in the form of a common language. *Requirements* are essential for implementing Exchange 2013 successfully.

Exchange can be a daunting product to contemplate, with over 20 million lines of code. There are many, many features from which to choose, though some have not changed significantly between Exchange versions (address book generation, for example). Furthermore, there's a discussion of how these features are to be implemented and the entire best practices conversation, which comes with the territory. How do you choose which features to implement and which to leave behind? The answer is requirements. And how do you decide which best practice to apply? Again, the answer is requirements.

Building the Foundation for Requirements

Requirements elicitation can sometimes be seen as boring, tedious, and overly complex. This perception can often derail this most critical part of a new project. Requirements elicitation is traditionally associated with software engineering, which implies a long list of requirements to satisfy the discipline of creating or modifying software. With the exception of writing scripts, most administrators who wish to implement Exchange don't need to know or care about the difference between a functional and a business requirement, since they're not creating software from scratch. However, we still need to capture the essence of "why" we are taking certain actions, as well as the "what" and the "how" we are doing them.

This chapter is particularly important for the Exchange administrator or consultant who may have been tasked with installing, upgrading, or migrating to Exchange for the first time in a formal manner and who doesn't know where to start. Even if you have successfully implemented Exchange, this chapter will still be of tremendous value to you if this is the first time that you are the one documenting a design.

Requirements are the core of an Exchange project. Based on the requirements, a host of other documentation items can be affected. These may include the following:

- ◆ Vision and scope document
- ◆ The Exchange 2013 design
- ◆ Testing plan
- ◆ Migration plan
- ◆ The bill of materials required to implement Exchange

- ◆ Test case documentation
- ◆ Adjustments to the disaster recovery plan (DRP)

A good place to start is to learn how to identify and document requirements correctly and with enough detail to satisfy people from different parts of IT and within the business as a whole. A bad place to begin is by installing Exchange on the basis of a diagram only. Since we are in IT, we often start with a diagram of something and then wind up making design changes on the fly.

Documenting requirements is thus a critical part of the design process, as we will explore later in this chapter. In summary, this chapter will equip you with the tools to ensure that “why,” “what,” and “how” are addressed and documented adequately, without requiring a degree in software engineering in order to do so.

Establishing Project Roles

Establishing roles at the outset of a project of significant magnitude is critically important. Most of you reading this book fall into one of two camps:

Camp 1: You are a highly respected individual with demonstrated competence in your field. You tend to be external to the companies you consult with, even if you are part of a multi-year project or outsource contract.

Camp 2: You are responsible for messaging within your company, or you may be a project manager who needs background reading for your first Exchange project. You have just been tasked with delivering an implementation of the newest version of Exchange.

Whichever camp you fall into, either external or internal to the organization, your attitude about such a project determines its success or failure. Your mindset needs to be that of a consultant toward a client. If you work inside the company, this may be harder to adopt; however, the methods within this chapter work equally well with either camp.

Getting Started with the Exchange Design

Often we are asked to review a design, which may be technically brilliant, architecturally sound, and mindful of the newest features introduced in the latest service pack. However, designs are very often written without capturing the essence—or even the reason for the existence—of the design. The essence or the reason for existence for an Exchange design is documented by capturing the requirements.

Exchange designs and implementations are often driven by a version’s new features list, as opposed to having captured and wrestled with requirements and extrapolated the features necessary to satisfy the needs of the business. Having a solid set of requirements to work against, among other reasons, makes your technology choices defensible, since designing or building Exchange without a solid set of requirements is like going food shopping without a list and putting anything that suits you into the shopping cart.

Documenting requirements also makes a document much easier to review. A structured document may have one section summarizing the security requirements and then a separate section encompassing the technical detail on how those security requirements are manifested as configuration detail or product features. This would allow the security or compliance officer to sign off on that portion of the document without having to wade through the storage design, unless of course the storage design captured a secure requirement.

When reviewing designs, we often see that the author has discussed the technology and then made a statement that a feature or technology should be implemented in a certain way. For example, the author may wish to implement Database Availability Groups (DAGs), which were introduced with Exchange 2010, and allow databases to be highly available. The author may say something like, “DAGs are great, so we’re going to implement a DAG.” However, the “why” of implementing a DAG is not captured. The question of why you are recommending the implementation of DAGs and which requirement you are fulfilling must be answered. Your reasons for choosing technology should always be clearly documented.

Most designs that we reviewed have little or no longevity. That is, if you were to review your own design in five years’ time, would you understand the “why” of the design? Why did you choose to implement a DAG in the manner that you did, and so forth? Keep this in mind when eliciting requirements and as a continual thread throughout your document. When your design document is reviewed, arguing that the reasons behind the design were not enumerated because you didn’t think it through or were ignorant of other options available at the time is not adequate for explaining why the “why” part of documentation is missing.

Requirements as Part of a Larger Framework

If you are a member of a consulting organization, then you will be quite familiar with the following. If you are doing this for the first time, then this section should be considered a primer as to where requirements fit into a larger framework.

There are several available methodologies from which to choose, including the Microsoft Solutions Framework. Irrespective of which methodology you choose, the steps are often quite similar:

Envisioning Phase This is the “thinking phase” of the project where you and others work to blue sky the project. In its simplest form, this involves the critical people considering the version of Exchange to be implemented and addressing such issues as *how* things are being done today and *why* they need to be done differently.

Requirements Definition Phase This phase captures the requirements, which is the focus of this chapter.

Design Phase This phase molds the requirements into something deployable and practical.

Testing Phase This is a standard phase of larger projects, and it is based entirely on the defined requirements. If the requirements are clear, the tests can be written well before the design is complete.

Deployment Phase This phase implements the design.

The first three phases will naturally generate a document set, which at minimum is similar to the following:

Vision and Scope Document This document captures the reason for the existence of the project, and it defines the business’s vision of the technology to be implemented. It also defines what is in and out of scope.

Functional or Technical Specification Document This can be separate from the design document. It defines the business requirements and lists the derived functional or technical requirements. Often, the consultant will use the scoping meeting to document the basic scope of the project and then try to derive the business and functional or technical

requirements for further clarification. We will cover this process in much greater depth shortly.

Design Document This document captures the resulting design.

Depending on the methodology selected, many other documents, such as testing plans, will also be expected.

If you are doing this for the first time, the level of detail required for a large project may overwhelm you. Nevertheless, as a consultant, it is likely that you are indeed working with this level of detail. While there is a wealth of material available about the available methodologies, the basic problem that requirements are often captured poorly, or not at all, remains.

Understanding the Types of Requirements

Classical requirements elicitation is a very deep and mysterious discipline, unless you're a business analyst, systems analyst, or the like. The aim of this chapter is not to address classical requirements elicitation from a systems analysis point of view, since that would only help you become a better systems analyst. The goal of this chapter is to determine what kinds of requirements are important for your Exchange design and to give you the ammunition you need to defend your technology choices.

You may be tempted to wrestle with the nuances among some of the more esoteric requirements. At minimum, however, you need to examine the business and technical requirements. We will take a moment to define each of these later.

The purists among you may question why we don't break technical requirements out into functional and nonfunctional requirements. *Functional requirements* describe what a system is required to do, while *nonfunctional requirements* describe how the system behaves. As mentioned at the beginning, this chapter is focused on eliciting the requirements necessary to implement existing software, not on writing software from the ground up.

If your project is small, the lines between functional, nonfunctional, and business and technical requirements may blur and add unnecessary complexity. Unless you find a compelling reason to include functional and nonfunctional requirements, we suggest that you focus exclusively on the business and technical requirements. Once your project reaches a certain level of complexity, however, you will need to define technical requirements in much more detail. Thus, you will then break out the functional and nonfunctional requirements aspects of the project.

Business requirements may also be captured separately in a vision document. Consulting organizations will be familiar with this procedure, and they will require completion of a specific document set in order to capture this level of detail.

You may argue that you have been given a requirement, and that sounds something like, "We need to upgrade." This statement is, in itself, only half a requirement, and it is an insufficient rationale for a business today to invest in your project. Now let's examine our two chosen requirements in more detail.

Business Requirements

In this section, we are going to discuss business requirements in the context of our upcoming Exchange implementation. This is the "why" part of your requirements. The project sponsor or management team member provides the business drivers for the project. You want to be sure that you don't get stuck in "analysis paralysis," since business requirements tend to be broad statements lacking the detail expected in technical requirements.

For our purposes, businesses tend to have a few simple drivers. These tend to fall into the category of decreasing costs, increasing/retaining revenue, or decreasing risk. A good set of business requirements should address all of these drivers if possible. You may not often be in the position of being able to drive sales up by, say, 40 percent, but you are certainly able to reduce risk by implementing a well-thought-out high-availability strategy for email, if email is a critical business function.

Let's look at a few examples of business requirements:

- ◆ Revenue requirement: A company may choose to implement a mobility app to increase productivity of its sales staff.
- ◆ Risk requirement: A company needs to protect itself from a failed audit because of a lack of support for its existing version of Exchange.
- ◆ Risk requirement: A company has made a strategic decision to migrate their technology base from Lotus Notes and Domino to Microsoft Exchange and Microsoft SharePoint due to in-house development moving to .NET based languages.

We will examine business requirements in the context of a sample customer, *XYZ Bank*.

BUSINESS REQUIREMENTS FOR A SAMPLE EXCHANGE UPGRADE

XYZ Bank has retained you to design and implement a replacement for its aging messaging system. XYZ's current messaging system is implemented in Exchange 2003, which was state of the art at the time. XYZ has an extensive branch network, with many individual Exchange 2003 servers across the country.

When it was implemented, email was not considered to be a critical application. XYZ is growing fast, adding a branch every two to three months. XYZ has also made it known that it intends to list itself on the stock exchange, which will subject XYZ to regular security and process audits.

The bank has recently decided to use email as one of the primary tools for communicating with its customers. XYZ currently limits mailboxes to a few hundred MBs. Because of this limit, employees are forced to move email data to PST files on desktops and file shares, exposing XYZ to risk from theft and corruption. Even with stringent restrictions, a number of branches are complaining that email performance is decreasing, even though the number of users on the respective servers remains the same.

Because of its critical nature, XYZ would like email to be centralized in a datacenter alongside its existing critical banking applications, with a similar level of redundancy and availability. Of course, XYZ is concerned about cost, and it wishes to explore a number of storage options before committing to purchasing a new Storage Area Network (SAN) solely for Exchange's use. Finally, XYZ has stated that it would like similar messaging functionality as provided by Exchange 2003 on day one of implementation while reserving the right to add features in the future.

This story is quite typical. It includes a mixture of requirements, including a clue that the bank is researching later versions of Exchange and that it is aware that several storage options are available. This is an indication that the bank has a number of well-defined feature-based requests.

We can discern a number of business-specific requests from this scenario:

- ◆ Replace the unsupported Microsoft Exchange 2003 platform with a currently supported Exchange 2013 environment.
- ◆ Increase the availability of the email environment to match the XYZ bank standard.
- ◆ Design for future growth.
- ◆ Allow for auditing administrative activity with the ability to demonstrate such processes.

Notice that the requirements are broad and contain little technical detail. The business requirements captured as part of a design along with an executive summary, for example, allow management and key staff to assimilate quickly the reasons why the Exchange upgrade is being deployed without getting bogged down in technical detail. In our example, XYZ concentrates on mitigating risk (support, availability, and auditing), while also supporting the stated goals of the business (expansion and increased communication).

Technical Requirements

Staying with the theme of combining technical requirements with functional and nonfunctional requirements, they are quite different from business requirements. Technical requirements are the “what” and “how” parts of requirements. Furthermore, business requirements are written as broad statements, while technical requirements are designed to be precise statements. Technical requirements should be simple lists that are both individual and granular. One of the biggest causes of confusion for implementers is ambiguity in technical specifications. Adding too much explanation within the requirements can cloud the specification.

When dealing with a product like Microsoft Exchange, we take a lot of functionality for granted, and so we should. However, there is a fine line to walk in terms of writing specifications. Let’s take, for example, the last line from our XYZ Bank scenario:

Finally, XYZ has stated that it would like similar messaging functionality as provided by Exchange 2003 on day one of implementation while reserving the right to add features in the future.

We could take for granted how to interpret this statement. However, it is actually quite subjective and needs clarification. For example, taken alone, we do not know who or what XYZ is, what “similar messaging functionality as provided by Exchange 2003” means, and what types of features could be added in the future. There are two possible paths of interpretations of this statement: One is broad interpretation based on our knowledge of Microsoft Exchange, and the other is “analysis paralysis.”

An example of analysis paralysis would be to specify Exchange functionality as follows:

- ◆ A user must be able to generate an email.
- ◆ A user must be able to display the contents of an email.
- ◆ A user must be able to share their calendar with another user.
- ◆ Calendar sharing must support granular rights.