Microsoft Exchange Server on EMC Symmetrix Storage Systems

• Generating Restartable and Recoverable Exchange Server 2010 Copies
• Exchange Server 2010 Remote Replication and Disaster Restart
• Exchange Server 2010 Layout and Performance

Don Turner
Preface

Chapter 1  Microsoft Exchange Server

Exchange Server overview .............................................................. 20
Exchange Server history .............................................................. 22
Exchange Server 2010 server roles.................................................. 24
  The Exchange Client Access Server role................................. 24
  The Exchange Hub Transport Server role .............................. 25
  The Exchange Unified Messaging Server role....................... 26
  The Exchange Edge Transport Server role ............................. 26
  The Exchange Mailbox Server role.......................................... 26
  The Exchange directory ............................................................ 27
  The message transfer subsystem ............................................. 27
  The message property store ..................................................... 27
  Exchange Server 2010 operating system requirements...... 28
  Exchange Server 2010 network requirements ....................... 28
  Required non-core Exchange Server 2010 application
  and services ................................................................................ 29
  Exchange Server 2010 core application services—
  Mailbox Server role ............................................................... 30
  Exchange Server 2010 Information Store components .......... 33
  Extensible Storage Engine ........................................................ 33
  ESE98 log writer .................................................................... 34
  ESE98 lazy writer ................................................................... 34
  ESE98 version store ................................................................ 35
  The ESE Database Cache ......................................................... 35
  The Extensible Storage Engine database structures .......... 36
  ESE functionality improvements for Exchange Server
  2010 ......................................................................................... 38
## Contents

- Extensible Storage Engine database page structure ........................................ 40
- Exchange Server Store Instances .................................................................................. 42
  - High-availability options in Exchange up to Exchange Server 2007 ........................................ 43
  - High Availability options for Exchange Server 2010 ........................................ 48
- Backup and restore operations for Exchange Server 2010 ........................................ 55
  - Volume Shadow Copy Service for Exchange Server ........................................ 55
- Exchange Server 2010 and EMC integration ............................................................ 59

### Chapter 2  EMC Foundation Products

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>62</td>
</tr>
<tr>
<td>Symmetrix hardware and EMC Enginuity features</td>
<td>65</td>
</tr>
<tr>
<td>Symmetrix VMAX platform</td>
<td>66</td>
</tr>
<tr>
<td>EMC Enginuity operating environment</td>
<td>67</td>
</tr>
<tr>
<td>EMC Solutions Enabler base management</td>
<td>69</td>
</tr>
<tr>
<td>EMC Change Tracker</td>
<td>72</td>
</tr>
<tr>
<td>EMC Symmetrix Remote Data Facility</td>
<td>73</td>
</tr>
<tr>
<td>SRDF benefits</td>
<td>74</td>
</tr>
<tr>
<td>SRDF modes of operation</td>
<td>74</td>
</tr>
<tr>
<td>SRDF device groups and composite groups</td>
<td>75</td>
</tr>
<tr>
<td>SRDF consistency groups</td>
<td>75</td>
</tr>
<tr>
<td>SRDF terminology</td>
<td>79</td>
</tr>
<tr>
<td>SRDF control operations</td>
<td>81</td>
</tr>
<tr>
<td>Failover and failback operations</td>
<td>85</td>
</tr>
<tr>
<td>EMC SRDF/Cluster Enabler solutions</td>
<td>87</td>
</tr>
<tr>
<td>EMC TimeFinder</td>
<td>88</td>
</tr>
<tr>
<td>TimeFinder/Mirror establish operations</td>
<td>89</td>
</tr>
<tr>
<td>TimeFinder split operations</td>
<td>90</td>
</tr>
<tr>
<td>TimeFinder restore operations</td>
<td>91</td>
</tr>
<tr>
<td>TimeFinder consistent split</td>
<td>92</td>
</tr>
<tr>
<td>Enginuity Consistency Assist</td>
<td>92</td>
</tr>
<tr>
<td>TimeFinder/Mirror reverse split</td>
<td>95</td>
</tr>
<tr>
<td>TimeFinder/Clone operations</td>
<td>95</td>
</tr>
<tr>
<td>TimeFinder/Snap operations</td>
<td>98</td>
</tr>
<tr>
<td>EMC Storage Resource Management</td>
<td>101</td>
</tr>
<tr>
<td>EMC PowerPath</td>
<td>106</td>
</tr>
<tr>
<td>PowerPath/VE</td>
<td>108</td>
</tr>
<tr>
<td>EMC Replication Manager</td>
<td>115</td>
</tr>
<tr>
<td>EMC Open Replicator</td>
<td>117</td>
</tr>
<tr>
<td>EMC Virtual Provisioning</td>
<td>118</td>
</tr>
<tr>
<td>Thin device</td>
<td>118</td>
</tr>
<tr>
<td>Data device</td>
<td>118</td>
</tr>
</tbody>
</table>
Chapter 3

Creating Exchange Server 2010 Clones

Introduction ..................................................................................... 130
Overview ......................................................................................... 131
Recoverable versus restartable copies of stores ................................ 132
   Recoverable copies ..................................................................... 132
   Restartable copies .................................................................... 132
Copying mailbox and logs with Exchange Server shutdown ........ 134
   Using TimeFinder/Mirror ........................................................... 134
   Using TimeFinder/Clone ............................................................. 136
   Using TimeFinder/Snap ............................................................... 138
Copying the store using EMC Consistency technology ............ 141
   Using TimeFinder/Mirror ........................................................... 141
   Using TimeFinder/Clone ............................................................. 143
   Using TimeFinder/Snap ............................................................... 145
Copying the store using the Exchange Server VSS Interface .... 148
   Using TimeFinder/Mirror ........................................................... 148
   Using TimeFinder/Clone ............................................................. 150
   Executing TF/EIM .................................................................. 150
   Using TimeFinder/Snap ............................................................... 152
Copying the store using Replication Manager ......................... 155
Transitioning disk copies to Exchange Server 2010 clones .... 157
   Instantiating clones from consistent split or shutdown images .... 157
   Using Exchange Server and VSS to process the image .......... 160
Reinitializing the cloned environment ......................................... 162
Choosing an Exchange cloning methodology ........................... 163

Chapter 4

Backing Up Exchange Server 2010

Introduction ..................................................................................... 166
EMC Consistency technology and backup .................................... 167
Exchange Server backup functionality ........................................ 168
   Offline backups ....................................................................... 168
   Online VSS backups ............................................................... 169
   Backup considerations ........................................................... 173
Contents

Integrating EMC technologies with Exchange Server
backups ........................................................................................................ 176
VSS integration ......................................................................................... 176
Managing backup integration ...................................................................... 176
TimeFinder Exchange Integration Module .............................................. 178
Planning to use TF/EIM ........................................................................... 178
Implementing backups with TF/EIM ...................................................... 179
Using TimeFinder/Mirror with TF/EIM ................................................. 181
Using TimeFinder/Clone with TF/EIM .................................................... 183
Using TimeFinder/Snap with TF/EIM ...................................................... 185
Using Remote Replication with TF/EIM ................................................. 186
Replication Manager ................................................................................ 189
Inside Replication Manager’s Exchange Agent ..................................... 189
Preparing the Exchange Server 2010 environment .................................. 191
Installing Replication Manager ............................................................... 191
Protecting mailbox databases ................................................................. 192
Considerations for DAGs ......................................................................... 195
Mounting replicas .................................................................................... 196

Chapter 5 Restoring and Recovering Microsoft Exchange Databases

Introduction ................................................................................................. 198
EMC Consistency technology and restore ................................................. 199
Exchange Server restore functionality ...................................................... 200
Types of Exchange Server restores ......................................................... 200
Point-in-time restores ............................................................................. 200
Roll forward restores ............................................................................. 200
Integrating EMC technologies with Exchange Server
recovery ...................................................................................................... 203
VSS integration ......................................................................................... 203
TimeFinder Exchange Integration Module .............................................. 204
Planning to use TF/EIM ........................................................................... 204
Implementing restores with TF/EIM ......................................................... 204
Using TimeFinder/Mirror with TF/EIM .................................................... 205
Using TimeFinder/Clone with TF/EIM .................................................... 207
Using TimeFinder/Snap with TF/EIM ...................................................... 209
Using remote replication with TF/EIM .................................................... 211
Replication Manager ................................................................................ 213
Recovering databases ............................................................................. 213
Recovering mailbox databases ................................................................. 214
Restoring a database to the point-in-time of a replica ................................ 214
Chapter 6  Understanding Exchange Server 2010 Continuity of Operations

Introduction ................................................................................................. 234
Definitions .................................................................................................... 236  
  Continuity of Operations ........................................................................ 236
  Exchange database restart ....................................................................... 236
  Exchange database recovery ................................................................. 236
  Dependent-write consistency .................................................................. 237
  Roll forward recovery ............................................................................ 237
Design considerations for continuity of operation plans ............................. 238
  Recovery Time Objective (RTO) ............................................................ 238
  Recovery Point Objective (RPO) ............................................................ 239
  RPO and RTO ....................................................................................... 239
  Operational complexity .......................................................................... 239
  Source server activity ............................................................................ 240
  Production impact ................................................................................. 240
  Target server activity ............................................................................ 240
  Number of copies .................................................................................. 241
  Distance for solution ............................................................................. 241
  Bandwidth requirements ........................................................................ 241
  Testing the solution ............................................................................... 241
  Cost ........................................................................................................ 242
Remote replication considerations ............................................................. 243
  Propagation delay ................................................................................. 243
  Bandwidth requirements ........................................................................ 244
  Network infrastructure .......................................................................... 244
  Method of instantiation ......................................................................... 245
  Method of reinstatement ....................................................................... 245
  Change rate at the source site ............................................................... 245
  Locality of reference .............................................................................. 245
  Expected data loss ................................................................................ 246
  Failback operations ............................................................................... 246
Continuity and Replication Technologies .................................................. 248
  Application level redundancy features ................................................. 248
Array-based redundancy features ........................................................... 249
  Planning for array-based replication .................................................... 249
  SRDF/S: Single Symmetrix to single Symmetrix .................................. 250
  SRDF/A ............................................................................................... 252
  SRDF/AR single hop ............................................................................ 256
  SRDF/AR multihop ............................................................................... 258
  Integrating Exchange with array-based replication ......................... 261
  Symmetrix Remote Data Replication Cluster Enabled (SRDF/CE) ....... 262
### Appendix A  SYMCLI Group Creation Command Samples

<table>
<thead>
<tr>
<th>Example</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeFinder/Mirror example</td>
<td>330</td>
</tr>
<tr>
<td>Device group</td>
<td>330</td>
</tr>
<tr>
<td>Composite group</td>
<td>330</td>
</tr>
<tr>
<td>TimeFinder/Clone example</td>
<td>332</td>
</tr>
<tr>
<td>Device group</td>
<td>332</td>
</tr>
<tr>
<td>Composite group</td>
<td>332</td>
</tr>
<tr>
<td>TimeFinder/SNAP example</td>
<td>334</td>
</tr>
<tr>
<td>Device group</td>
<td>334</td>
</tr>
<tr>
<td>Composite group</td>
<td>334</td>
</tr>
</tbody>
</table>

### Appendix B  References

<table>
<thead>
<tr>
<th>Document</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related documents</td>
<td>338</td>
</tr>
<tr>
<td>SYMCLI</td>
<td>338</td>
</tr>
<tr>
<td>TimeFinder</td>
<td>338</td>
</tr>
<tr>
<td>SRDF</td>
<td>338</td>
</tr>
<tr>
<td>Replication Manager</td>
<td>339</td>
</tr>
<tr>
<td>Microsoft</td>
<td>339</td>
</tr>
</tbody>
</table>

### Glossary

### Index
Title                                                                                                                                Page
1 Balanced B-tree ............................................................................................... 37
2 Basic Single Copy Cluster ............................................................................. 44
3 Basic LCR deployment .................................................................................. 46
4 Basic CCR deployment .................................................................................. 47
5 DAG with 5 members .................................................................................... 50
6 DAG with a server offline for maintenance or a failed server............... 51
7 DAG with a server back online synchronizing the databases ............... 52
8 DAG extended across two Active Directory sites ..................................... 53
9 Volume Shadow Copy Service architecture ............................................. 56
10 Microsoft VSS framework communications paths .................................. 58
11 Symmetrix VMAX logical diagram .......................................................... 67
12 Basic synchronous SRDF configuration .................................................... 74
13 SRDF consistency group ........................................................................... 77
14 SRDF establish and restore control operations ........................................ 83
15 SRDF failover and failback control operations ......................................... 85
16 Geographically distributed four-node EMC SRDF/CE clusters ............ 87
17 EMC Symmetrix configured with standard volumes and BCVs .......... 89
18 ECA consistent split across multiple database-associated hosts .......... 93
19 ECA consistent split on a local Symmetrix system ................................ 94
20 Creating a copy session using the symclone command ............................ 97
21 TimeFinder/Snap copy of a standard device to a VDEV ......................... 100
22 SRM commands.......................................................................................... 102
23 PowerPath/VE vStorage API for multipathing plug-in......................... 109
24 Output of the rpowermt display command on a Symmetrix VMAX device ................................................................. 112
25 Device ownership in vCenter Server........................................................ 113
26 Virtual Provisioning components.............................................................. 119
27 Virtual LUN Eligibility Tables................................................................... 121
28 Copying a shutdown Exchange store with TimeFinder/Mirror .......... 135
29 Copying a shutdown Exchange store with TimeFinder/Clone ............ 137

Microsoft Exchange Server on EMC Symmetrix Storage Systems
Microsoft Exchange Server on EMC Symmetrix Storage Systems

Figures

30 Copying a shutdown Exchange store with TimeFinder/Snap ............. 139
31 Copying a running Exchange store with TimeFinder/Mirror ............. 142
32 Copying a running Exchange store with TimeFinder/Clone ............. 144
33 Copying a running Exchange store with TimeFinder/Snap ............. 146
34 TimeFinder/VSS backup of an Exchange store ......................... 149
35 TimeFinder/Clone VSS backup of an Exchange store .................... 151
36 TimeFinder/Snap VSS backup of an Exchange store ..................... 153
37 Replication Manager processing replica of an Exchange store ......... 155
38 ESEUTIL output for the db1.edb mailbox store ....................... 158
39 Using ESEUTIL to recover mailbox stores to a consistent state ....... 159
40 ESEUTIL showing clean shutdown state after recovery ............... 160
41 EMC VSS enabled requestors and VSS overview ....................... 170
42 TimeFinder Exchange Integration Module with TF/Mirror ............. 182
43 TimeFinder Exchange Integration Module with TF/Clone ............. 184
44 TimeFinder Exchange Integration Module with TF/Snap ............. 186
45 TimeFinder Exchange Integration Module with SRDF ................. 188
46 RM software architecture ....................................................... 189
47 Exchange Server point in time restore .................................. 201
48 Exchange Server roll forward recovery .................................. 202
49 TimeFinder/Mirror restore using TF/EIM ................................ 206
50 TimeFinder/Clone restore using TF/EIM ................................ 208
51 TimeFinder/Snap restore using TF/EIM ................................ 210
52 Restore Wizard—Replica to restore ....................................... 215
53 Restore Wizard—Objects to be restored .................................. 216
54 Restore Wizard - Restore options .......................................... 217
55 Restore Wizard—Restore warning .......................................... 218
56 Restore Wizard—Objects to be restored .................................. 221
57 Mount options dialog box configured for mailbox recovery .......... 226
58 Get-MailboxDatabase example .............................................. 227
59 Get-MailboxStatistics example ............................................ 228
60 RestoreMailbox example output ............................................ 229
61 SRDF/S replication process .................................................. 250
62 SRDF/Asynchronous replication process ................................ 253
63 SRDF/AR single-hop replication process ................................ 257
64 SRDF/AR multihop replication process ................................ 260
65 High-level view of AutoStart integration into the DAG framework ... 270
66 Round-robin approach for zoning HBA and Symmetrix fibre adapter ports .............................................. 304
67 Logical view of mirrored hypervolumes .................................. 312
68 Logical view of striped and concatenated metavolumes ............. 313
<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Exchange Server 2010 license features</td>
<td>21</td>
</tr>
<tr>
<td>2. ESE versions and page size</td>
<td>40</td>
</tr>
<tr>
<td>3. SYMCLI base commands</td>
<td>69</td>
</tr>
<tr>
<td>4. TimeFinder device type summary</td>
<td>99</td>
</tr>
<tr>
<td>5. Data object SRM commands</td>
<td>103</td>
</tr>
<tr>
<td>6. Data object mapping commands</td>
<td>103</td>
</tr>
<tr>
<td>7. File system SRM commands to examine file system mapping</td>
<td>104</td>
</tr>
<tr>
<td>8. File system SRM command to examine logical volume mapping</td>
<td>104</td>
</tr>
<tr>
<td>9. SRM statistics command</td>
<td>105</td>
</tr>
<tr>
<td>10. Comparison of cloning technologies for Exchange Server</td>
<td>163</td>
</tr>
<tr>
<td>11. Replication options</td>
<td>164</td>
</tr>
<tr>
<td>12. Replication options</td>
<td>264</td>
</tr>
<tr>
<td>13. Recommended Mailbox server performance counters—Active database copy I/O</td>
<td>282</td>
</tr>
<tr>
<td>14. Recommended Mailbox server performance counters—Active Log I/O</td>
<td>282</td>
</tr>
<tr>
<td>15. Recommended Mailbox server performance counters—Passive Database Copy I/O</td>
<td>283</td>
</tr>
<tr>
<td>16. Database and Log checkpoint depth</td>
<td>284</td>
</tr>
<tr>
<td>17. Mailbox database I/O read/write ratios</td>
<td>286</td>
</tr>
<tr>
<td>18. Memory configurations for Exchange 2010 servers based on installed server roles</td>
<td>291</td>
</tr>
<tr>
<td>19. Estimated IOPS per mailbox based on message activity and mailbox database</td>
<td>292</td>
</tr>
<tr>
<td>20. Required minimum memory per mailbox database</td>
<td>294</td>
</tr>
<tr>
<td>21. Causes of disk I/O by the Mailbox server</td>
<td>295</td>
</tr>
<tr>
<td>22. Exchange critical folder recommended maximums</td>
<td>300</td>
</tr>
<tr>
<td>23. Microsoft Exchange client hard disk recommendations</td>
<td>302</td>
</tr>
<tr>
<td>24. Microsoft Exchange I/O profiles</td>
<td>306</td>
</tr>
<tr>
<td>25. IOPS per spindle by disk speed</td>
<td>320</td>
</tr>
</tbody>
</table>
26 Sample drive count calculation based on drive speed and RAID type................................................................. 321
27 Sample drive count calculation based on drive size and capacity ........ 322
28 Number of logs generated per mailbox................................................................. 324
29 Log IOPS requirement.................................................................................. 325
This EMC Engineering TechBook describes how the EMC Symmetrix storage system operates and interfaces with Exchange Server 2010. The information in this document is based on Exchange Server 2010 on Symmetrix storage systems running Solutions Enabler version 7.1 or later and current releases of Symmetrix Enginuity microcode.

This document provides an overview of Exchange Server 2010 along with a general description of EMC products and utilities that can be used for Exchange Server administration. EMC Symmetrix storage systems and EMC software products can be used to manage Exchange Server 2010 environments and to enhance system and storage management backup/recovery and restart procedures. Using EMC products and utilities to manage Exchange Server 2010 environments can help reduce database and storage management administration, reduce system CPU resource consumption, and reduce the time required to clone, back up, recover, or restart Exchange Server 2010 environments.

As part of an effort to improve and enhance the performance and capabilities of its product lines, EMC periodically releases revisions of its hardware and software. Therefore, some functions described in this document may not be supported by all versions of the software or hardware currently in use. For the most up-to-date information on product features, refer to your product release notes.

In this document, the product name “Exchange Server 2010 SP1” may be referred to as “Exchange Server 2010.”

**Audience**

This TechBook is intended for Exchange Server systems administrators and storage management personnel responsible for managing Exchange Server systems.
This TechBook was written by Don Turner, an employee of EMC based at Brentford, United Kingdom. Don has over 15 years of experience in Windows Server, Exchange Server and EMC Storage Systems.

This document is divided into seven chapters. Topics range from general Exchange Server 2010 and EMC product descriptions in chapters 1 and 2 to detailed discussions of implementation procedures in chapters 3 through 7. Detailed examples are included to illustrate features and functions, and are only provided to clarify the topic being discussed. Not all possible combinations of solutions are covered in this document. These examples were developed for laboratory testing and may need tailoring to suit other operational environments. Any procedures outlined in this document should be thoroughly tested before production implementation.

Microsoft provides extensive documentation on Exchange Server through its website, and through the online documentation set. These resources should be the primary source for information on Exchange Server-specific commands. Updated downloadable or online versions of the documentation are available from the Exchange Server 2010 website: http://www.microsoft.com/exchange

EMC uses the following conventions for special notices.

Note: A note presents information that is important, but not hazard-related.

**Typographical conventions**
EMC uses the following type style conventions in this document:

**Normal**
Used in running (nonprocedural) text for:

- Names of interface elements (such as names of windows, dialog boxes, buttons, fields, and menus)
- Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, functions, utilities
- URLs, pathnames, filenames, directory names, computer names, filenames, links, groups, service keys, file systems, notifications

**Bold**
Used in running (nonprocedural) text for:

- Names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system calls, man pages
We’d like to hear from you!

Your feedback on our TechBooks is important to us! We want our books to be as helpful and relevant as possible, so please feel free to send us your comments, opinions and thoughts on this or any other TechBook:

TechBooks@emc.com
Preface
This chapter includes the following topics:

- Exchange Server overview ................................................. 20
- Exchange Server history...................................................... 22
- Exchange Server 2010 server roles.......................................... 24
- Exchange Server 2010 Information Store components ............ 33
- Exchange Server Store Instances........................................... 42
- Backup and restore operations for Exchange Server 2010......... 55
- Exchange Server 2010 and EMC integration............................ 59
Exchange Server overview

Exchange Server 2010 is Microsoft Corporation’s premier messaging and collaborative software. Exchange Server is part of the Windows Server System line of server products and is widely used by many enterprises using Microsoft infrastructure solutions. Microsoft Exchange not only provides a number of major features consisting of electronic mail, shared calendars and tasks, and support for mobile and web-based access to information, but also supports very large amounts of data storage.

As a messaging server platform, Exchange Server 2010 shares the following common features with other e-mail systems:

- Transfers e-mail messages to intended recipients in a reliable way, whether the recipients reside on the local server, another server in the same Exchange Server organization, or another server in an external messaging environment that is connected to the organization.
- Stores the e-mail messages in a server-based store.
- Supports various e-mail clients that are used to access or download messages.
- Gives information about recipients in the organization to the users through an address book or global address list.

At the time of publication of this document, the current version of Microsoft Exchange is 2010 SP1 or version 14.01. This document, however, will concentrate on Exchange Server 2010 SP1 will be referred to as Exchange Server 2010.

Exchange Server 2010 can be installed on the following server platforms


The active directory (AD) must be in Windows 2000 native mode or later.

All AD Domain Controllers / Global Catalog servers and AD FSMO role holders must run at least Windows Server 2003 SP1.
Exchange Server 2010 is available in two versions, Standard Edition and Enterprise Edition. The supported options are detailed in Table 1 on page 21.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Standard Edition</th>
<th>Enterprise Edition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Support</td>
<td>5 Databases</td>
<td>100 Databases</td>
</tr>
</tbody>
</table>

In Exchange Server 2010, the various functions have been devolved to specific server roles and are described later in this chapter. This TechBook will concentrate on the Mailbox server role.
Exchange Server history

The original version of Exchange Server was an X.400-based mail server that also supported the X.500 directory standard. This product replaced the Microsoft Mail software package originally acquired from Network Courier. Exchange was a client-server-based mail system that used a single database store. The Exchange 4.0 directory formed the initial foundation for Microsoft's Active Directory service, an LDAP-compliant directory server. Active Directory was later integrated into Windows 2000 as the foundation of Windows Server domains.

Exchange Server 5.0 not only introduced the new Exchange Administrator console, but also opened up “integrated” access to SMTP-based networks for the first time. Unlike Microsoft Mail (which required a stand-alone SMTP relay), Exchange Server 5.0 could, with the help of an add-in called the Internet Mail Connector, communicate directly with servers using the new SMTP mail standard. Version 5.0 was soon supplanted by the next version that would add additional mail transport options and easier back-end administration.

Exchange Server 5.5 was generally available in two editions, Standard ("5.5/S") and Enterprise ("5.5/E"). These two versions differed in database store size, mail transport connectors, and clustering capabilities. The Standard edition had a database size limitation of 16 GB (similar to that of earlier versions of Exchange Server), while the Enterprise edition had a limit of 8 TB—unlimited for all practical purposes. The Standard edition was packaged with a Site Connector, an MS Mail Connector, an Internet Mail Service (previously “Internet Mail Connector”), an Internet News Service (previously “Internet News Connector”), as well as software to interoperate with cc:Mail, Lotus Notes, and Novell GroupWise. The Enterprise edition added an X.400 connector, and interoperability software with SNADS and PROFS. The Enterprise edition could be configured for use with Microsoft Cluster Service, while Standard did not support Cluster configurations.

Exchange Server 2000, or version 6.0, overcame many of the limitations of its predecessors. For example, this new version raised the maximum number of databases to 20, and increased the number of servers in a Microsoft Cluster Service configuration from two to four.
Exchange Server 2003 version 6.5 enhanced disaster recovery. One example of this is the “Messaging Dial Tone recovery” that allowed the server to send and receive mail while the message stores were being recovered from backup. Additionally, the Microsoft Volume Shadow Copy Service (VSS) and modifications to the resource group dependencies within a Microsoft Exchange Virtual Server instance for Microsoft Cluster Service added to the highly available feature set.

Exchange Server 2007 version 7 extended the old two-role front-end back-end topology from previous versions by allocating specific roles to separate server types. Breaking down Exchange Server 2007 into five server roles provides many advantages, allowing the Exchange topology to be more flexible and highly scalable. It also allows better hardware utilization and, in conjunction with only using 64-bit hardware, allows more mailboxes to be hosted on a single server.

Exchange Server 2010 version 14 was carried on from the role-based functions of Exchange Server 2007, but a number of changes to the database topology and the HA scenarios were made. It removed the need for database hosting storage groups that had been around since Exchange Server 2000. It also extended the number of supported databases from 50 to 100 in the Enterprise version.
Exchange Server 2010 server roles

In Exchange Server 2000 and 2003 there were two roles, front end and back end. In that architecture, a front-end server accepted requests from clients and proxied them to the appropriate back-end server for processing. In Exchange Server 2007 and Exchange Server 2010 five server roles can be installed and configured. The server roles can be installed and configured, for Exchange Server 2007 on a computer that is running Windows Server 2003 or Windows Server 2008. The server roles can be installed and configured for Exchange Server 2010 on a computer that is running only Windows Server 2008 or later. The server roles are described next.

The Exchange Client Access Server role

The Client Access Server (CAS) role accepts connections to the Exchange Server 2010 server from many different clients. Software clients such as Outlook Express and Eudora use Post Office Protocol version 3 (POP3) or Internet Message Access Protocol version 4rev1 (IMAP4) connections to communicate with the Exchange server. Hardware clients, such as mobile devices, use ActiveSync, POP3, or IMAP4 to communicate with the Exchange server. The CAS role must be installed in every Exchange organization. The CAS role supports the Microsoft Office Outlook Web Access and Microsoft Exchange ActiveSync client applications, Outlook Anywhere, Microsoft Entourage 2004 and Entourage 2008 for Mac, and the POP3 and IMAP4 protocols. The CAS role also provides access to free/busy data by using the Availability service and enables clients that are running Microsoft Office Outlook 2007, Outlook 2010, and certain mobile operating systems to download automatic configuration settings from the Autodiscover service. Unlike previous versions of Exchange Server where access from Outlook clients using MAPI from inside an organization firewall was directly through to the mailbox store on the Exchange Server mailbox server, in Exchange Server 2010 all access to the mailstore is through the CAS. This allows the number of mailbox per mailbox server to be increased. The CAS supports:

- **Outlook Web App**—Outlook Web App (formerly Outlook Web Access in previous versions of Exchange Server) in Exchange Server 2010 lets you access your e-mail from an expanded set of Web browsers. These include Internet Explorer 6.0 and later, Firefox, Safari, and Google’s Chrome.
Exchange ActiveSync—Exchange ActiveSync enables the synchronization of data between a mobile device and Exchange Server 2010. E-mail, contacts, calendar information, and tasks that can be run on devices that run Windows Mobile software can be synchronized.

POP3 and IMAP—In addition to supporting MAPI and HTTP clients, Exchange Server 2010 also supports POP3 and IMAP4 clients.

The Availability service—The Exchange Server 2010 Availability service improves free/busy data access for information workers by providing secure, consistent, and up-to-date free/busy data to computers that are running Outlook 2007 and later. Outlook 2007 and later uses the Autodiscover service to obtain the URL of the Availability service. The Autodiscover service resembles the Domain Name System (DNS) web service for Outlook 2007 and later. Essentially, the Autodiscover service helps Outlook 2007 and later locate various web services, such as Microsoft Exchange Unified Messaging, Offline Address Book, and Availability services.

The Autodiscover Service—The Autodiscover service enables Outlook clients and some mobile devices to receive their necessary profile settings directly from the Exchange server by using the client’s domain credentials. These settings automatically update the client with the information that is needed to create the user’s profile.

The Exchange Hub Transport Server role

The Hub Transport Server (HTS) role handles all mail flow inside the organization, applies organizational message policies, and is responsible for delivering messages to a recipient’s mailbox. Hub Transport servers provide the following functionality:

Mail flow—The HTS processes all mail that is sent inside the Exchange organization before it is delivered to a recipient’s inbox within the organization or routed to users outside the organization. There are no exceptions to this.

Categorization—The categorizer performs recipient resolution, routing resolution, and content conversion for all messages that move through the Exchange Server 2010 transport pipeline.

Routing—The HTS role determines the routing path for all messages that are sent and received in the organization.

Delivery—Messages are delivered to a recipient’s mailbox.
Unified Messaging (UM) combines voice messaging, fax, and e-mail messaging into one store, accessible from a telephone and a computer. Exchange Server 2010 Unified Messaging integrates Exchange Server with telephony networks and brings the Unified Messaging features to the core of Exchange Server.

The Edge Transport (Edge) server role is deployed in the organization's perimeter network and handles all Internet-facing mail flow, providing protection against spam and viruses.

In Exchange Server 2010, the Mailbox server role is one of several server roles that you can install and then configure on a computer. The Mailbox server role hosts mailbox and public folder databases. It also generates the offline address book. Mailbox servers provide services that calculate e-mail address policies and address lists for recipients, and enforce managed folders. Servers on which the Mailbox Server role is installed are called Mailbox Servers.

The Mailbox Server can also act as a CAS and HTS but UM and Edge must reside on separate servers. High-availability options are only available on a Mailbox Server. No other roles can participate in a HA cluster. Clustering options will be discussed later in this chapter.

Mailbox Servers in Exchange Server 2010 provides for the following functions:

- Host Mailbox databases.
- Provide e-mail storage.
- Host public folder databases.
- Calculate e-mail address policies.
- Generate address lists and offline address books (OABs).
- Conduct multi mailbox searches.
- Provide high availability and site resiliency.
- Provide context indexing.
- Provide messaging records management (MRM) and retention policies.

Exchange Server 2010 implements this base functionality through the use of the following architectural components.
The Exchange directory

The directory contains information about the end users of the Exchange messaging system and stores configuration information about the message-handling subsystem. This information is used to deliver messages to the intended recipients and route messages from one messaging server to another. Prior to Exchange Server 2000, the Exchange Directory was self-contained within each Exchange instance. With the introduction of Exchange Server 2000, the directory function was integrated into the Windows Active Directory. Exchange Server 2010 continues to utilize the Windows Active Directory and communicates to the Active Directory through an access module known as ADAccess.

The message transfer subsystem

Exchange Server 2010 mailbox servers do not transport mail in the same way as versions prior to Exchange Server 2007. That task of sending and receiving messages is now carried out by the hub transport server. All messages that are sent or received by a mailbox server are done via the hub transport server, even if the sender and recipient reside within the same mailbox store. All communication between the mailbox server and the hub transport server, within an AD site, is carried out using server-side Remote Procedure Calls (RPC). Exchange RPCs require the servers to be connected by LAN-quality bandwidth, otherwise the risk of RPC timeouts and failures is increased. All message transfers now use the AD intersite communications.

If multiple hub transport servers exist within an AD site, then the Mail Submission service, running on the hub transport server, load balances between all available hub transport servers.

The message property store

Exchange Server 2010 stores e-mail messages and other items in mailboxes and public folders using the message store or more appropriately, the Exchange Store. The Exchange Store contains message tables that the transfer subsystem uses to store messages temporarily when messages are routed from one server to another. The Exchange store relies on Extensible Storage Engine (ESE) technology to implement the messaging databases. A deeper discussion of the Exchange database technology is outlined later in this chapter.
Microsoft Exchange Server

All Exchange Server 2010 server roles have a number of operating system requirements to be installed and are:

- **64-bit computer**—All production servers for Exchange Server 2010 require a 64-bit computer. 32-bit versions are available for training and lab purposes but should not be used in a production environment.

- **Microsoft .NET Framework version 3.0**—Windows Server 2008 and later only require Microsoft .NET Framework version 3.0.

- **Microsoft Windows PowerShell (for the Exchange Management Shell)**—Windows PowerShell is the next-generation command line shell and scripting language for Windows. Exchange Server 2007 was the first major application to make use of PowerShell for deployment and administration and is continued for Exchange Server 2010.

- **Microsoft Management Console 3.0**—Used for the Exchange Management Console.

In order to complete basic message transfers within an Exchange Server 2010 organization, the network must support TCP/IP. Both Active Directory and the SMTP service require TCP/IP. The following network components are required for proper Exchange Server 2010 communication and message transfer:

- **IP and TCP**—Exchange Server 2010 requires TCP/IP to communicate with other computers on the network. Exchange Server 2010 does not support other network protocols.

- **DNS**—Exchange Server 2010 requires the Domain Name Service (DNS) to resolve the IP addresses for other hosts on a TCP/IP network, locate domain controllers and global catalog servers in an Active Directory domain, and locate e-mail servers in other messaging organizations.

- **DHCP and WINS**—Exchange Server 2010 does not require Dynamic Host Configuration Protocol (DHCP) to function; however, some of the networking clients on the TCP/IP network may require this service. DHCP is used to automatically assign an IP address to computers on a network. Windows Internet Name Service (WINS), on the other hand, is used by Microsoft Windows clients to perform NetBIOS name resolution. In network environments that contain routers that do not forward broadcasts across network segments, WINS is required to resolve...
the IP addresses for other computers on the network. Additionally, Exchange Server 2010 requires WINS when deployed in a Microsoft Cluster Service configuration.

- **Windows sockets**—Exchange Server 2010 uses Windows sockets to provide connection points for network clients connecting to services on the server. A Windows socket is an endpoint defined by an IP address combined with a port number used to identify a service implemented on a Windows Server or client.

- **Active Directory**—Active Directory provides the directory service for Exchange Server 2010.

- **Security subsystem**—Exchange Server 2010 uses the security subsystem of Windows Server 2008 to authenticate users within the Exchange organization. The security subsystem ensures that only authorized users can access mailboxes or send e-mail.

Microsoft Windows services are applications that run on Windows computers regardless of whether a user is logged in or not. A Windows service includes an executable file, a directory for storing application components, and registry settings that define the service parameters.

Exchange Server 2010 relies heavily on the operating system for network communication, security, directory services, and so forth. For example, Exchange Server 2010 requires TCP/IP and depends on the TCP/IP protocol stack and related components.

The following is a partial list for reference:

- **Event Log**—This service enables event log messages issued by Exchange services and other Windows-based programs and components to be viewed in Event Viewer.

- **NTLM Security Support Provider**—This service provides security for programs that use remote procedure calls (RPCs) and transports other than named pipes to log in to the network using the NTLM authentication protocol.

- **Remote Procedure Call (RPC)**—This service enables the RPC endpoint mapper to support RPC connections to the server. This service also serves as the Component Object Model (COM). Typical RPC clients are MAPI clients, such as Microsoft Outlook and Exchange System Manager, but internal components of System Attendant, such as DSProxy, are also RPC clients. To accept directory requests from MAPI clients and pass them on to
Microsoft Exchange Server 2010 has a number of Application Core components. A summary of the most common services for the mailbox server role are listed:

- **Microsoft Exchange System Attendant Service**—System Attendant is one of the most important services for Exchange Server 2010. It forwards directory lookups to a global catalog server for legacy Outlook clients, generates e-mail addresses and offline address books, updates free/busy information for legacy clients, and maintains permissions and group memberships for the server. The executable file is mad.exe and is located by default in the \Program Files\Microsoft\Exchange Server\V14\Bin directory.

- **Exchange Information Store Service**—The Microsoft Exchange Information Store service is another, very important component in Exchange Server 2010, because it maintains the messaging databases that contain all server-based mailboxes and public folders. The executable file of the Exchange Information Store service is Store.exe, located by default in the \Program Files\Microsoft\Exchange Server\V14\Bin directory.

- **Exchange Mail Submission**—The Exchange Mail Submission service submits messages from the Mailbox server to the Hub Transport servers. The executable file of the Exchange Mail Submission service is MSExchangeMailSubmission.exe, located by default in the \Program Files\Microsoft\Exchange Server\V14\Bin directory.

- **Exchange Mailbox Assistants**—The Exchange Mailbox Assistants performs background processing of mailboxes in the Exchange store. The executable file of the Exchange Mailbox Assistants service is MSExchangeMailboxAssistants.exe, located by default in the \Program Files\Microsoft\Exchange Server\V14\Bin directory.

- **Exchange Replication Service**—The Microsoft Exchange Replication service provides replication functionality for Mailbox server role databases and is used by local continuous replication, cluster continuous replication, and standby continuous replication. The executable file of the Exchange Replication service...
service is Microsoft.Exchange.Cluster.ReplayService.exe, located by default in the \Program Files\Microsoft\Exchange Server\V14\Bin directory. This service can only be run on a Mailbox Server role server and only if a high-availability option has been configured.

- **Exchange Search Indexer**—The Exchange Search Indexer service drives indexing of mailbox content, which improves the performance of content search. The executable file of the Exchange Search Indexer service is Microsoft.Exchange.Search.ExSearch.exe, located by default in the \Program Files\Microsoft\Exchange Server\V14\Bin directory.

- **Exchange Transport Log Search**—The Exchange Transport Log Search service provides remote search capability for Microsoft Exchange Transport log files. The executable file of the Exchange Transport Log Search service is MSExchangeTransportLogSearch.exe, located by default in the \Program Files\Microsoft\Exchange Server\V14\Bin directory.

- **Exchange Active Directory Topology Service**—The Exchange Active Directory Topology Service provides Active Directory topology information to Exchange services. The executable file of the Exchange Active Directory Topology Service service is MSExchangeADTopologyService.exe, located by default in the \Program Files\Microsoft\Exchange Server\V14\Bin directory.

- **Microsoft Exchange RPC Client Access**—The Microsoft Exchange RPC Client Access service resides on both the mailbox server and CAS and is responsible for managing client RPC connections. The executable file of the Microsoft Exchange RPC Client Access is Microsoft.Exchange.RpcClientAccess.Service.exe located by default in the \Program Files\Microsoft\Exchange Server\V14\Bin directory.

- **Microsoft Exchange Server Extension for Windows Server Backup**—The Microsoft Exchange Server Extension for Windows Server Backup service allows Windows Server Backup users to back up and recover application data for Microsoft Exchange Server. The executable file of the Microsoft Exchange Server Extension for Windows Server Backup service is wsbexchange.exe located by default in the \Program Files\Microsoft\Exchange Server\V14\Bin directory.
- **Microsoft Exchange Throttling**—The Microsoft Exchange Throttling service limits the rate of user operations. The executable file of the Microsoft Exchange Throttling service is "MSExchangeThrottling.exe" located by default in the \Program Files\Microsoft\Exchange Server\V14\Bin directory.

- **Microsoft Search (Exchange)**—The Microsoft Search (Exchange) service is an Exchange-customized version of Microsoft Search. The executable file of the Microsoft Search (Exchange) service is msftesql.exe” -Exchange -s:Exchange -f:Exchange located by default in the \Program Files\Microsoft\Exchange Server\V14\Bin directory.
Exchange Server 2010 Information Store components

Much of this discussion up to now has focused on Microsoft Exchange Client support and connectivity. This is due to the fact that the type of Exchange client(s) chosen for deployment within the organization will impact or determine which database files will ultimately store their messaging data. Exchange Server 2010 supports the use of two types of Information Stores: mailbox stores and public folder stores.

Each Information Store is a logical database that now only includes one database file. In Exchange Server 2000 and 2003, two database files were used, the database file .edb and the streaming database file .stm. The streaming database files for non-MAPI clients are no longer used, as all non-MAPI clients access their mailboxes via the CAS. The database file (.edb) contains all messages submitted to the store. In Exchange Server 2007, all MAPI (implies Microsoft Outlook) connections were direct to the store, whilst all other clients access through the CAS. In Exchange Server 2010, all connectivity to the mailbox store is through the CAS.

In order to store mailbox and public folder data, the Microsoft Exchange Information Store service employs a database technology called ESE. ESE is a multiuser, indexed sequential access method (ISAM) table manager with full data manipulation language (DML) and data definition language (DDL) capability.

Extensible Storage Engine

Today, Microsoft Exchange Server 2000, 2003, 2007, and 2010 employ version ESE98 of the Extensible Storage Engine and although the internal structure has been modified over the years, the fundamental operation of the database remains the same. ESE98 allows applications to store records and create multiple indexes across those records, allowing users to access those records in different ways. The ESE98 database engine has been derived from a technology known as the Joint Engine Technology referred to as JET Blue.

It should be noted that although this variant of JET (Blue) is often confused with JET Red, a database technology employed in Microsoft Access, the two varieties of JET represent two very different implementations.

Similar to any transactional-based relational database system, Exchange Server 2010 through ESE adheres to the four ACID properties:
Atomicity
Transactions are implemented in such a manner that either all updates for the transaction are committed or none of the updates occur.

Consistency
Only valid data is written to the database files, and changes representing transactions are managed in such a manner as to ensure that state changes are processed from one “correct” or consistent state to another similarly consistent state.

Isolation
Pending transactional changes are not reflected to other processes, including other concurrently executing transactions, until the transaction is committed.

Durability
Once a transaction has been committed, transactional integrity is maintained even in the event of a system failure.

Exchange Server 2010 provides ACID compliance with the use of three core database components within ESE98. These three components are defined as the “Log Writer,” the “Lazy Writer,” and the “Version Store.” Each of these components have unique functions but all have a role in both Exchange Server I/O transaction log and database processing.

---

**ESE98 log writer**

As Exchange Server transactions occur, ESE98 stores them in transaction log buffers that are configured in server memory. ESE98 will flush these log buffers to the transaction log disk when either a log buffer is filled or a transaction commit occurs. This I/O operation to the log device is performed sequentially and synchronously, making it a fast and efficient process. Log writes will always occur prior to the corresponding update to the database file. Exchange Server 2010 refers to this functionality as write-ahead logging.

---

**ESE98 lazy writer**

As transactions occur within the Exchange Information Store (IS), the log writer persists the update to the log and ESE98 uses buffer pool space to record “dirty pages.” A dirty page is maintained in memory and must at some stage be flushed to the database file.
The buffer pool is comprised of 8 KB buffers allocated from Windows Server memory and is managed by Exchange's dynamic buffer allocation (DBA) algorithm to cache these modified database pages. ESE98 will flush dirty pages to disk as soon as possible, although not necessarily immediately, and the mechanism used to accomplish this process is referred to as the lazy writer. As the name implies, the lazy writer will manage the process of destaging “dirty” pages to the database volumes when sufficient host server I/O bandwidth is available. The process of periodically flushing these dirty pages to the database disks using the lazy writer is referred to as database checkpointing. ESE98 will attempt to balance ESE98-generated I/O through a leveling algorithm that tries to flush dirty pages to disk or checkpoint without overloading the disk subsystem.

Each of these database technology components (the log writer, log buffers, IS buffers, the lazy writer, and the version store) allows ESE98 to safely and reliably commit changes to the databases while achieving optimum performance and scalability.

---

**ESE98 version store**

The version store provides ESE98 the ability to track and manage multiple concurrent transactions to the same database page, providing the isolation and consistency attributes required for ACID compliance. The in-memory version-store data structure stores database pages as they are modified, and ensures that each transaction references the correct page version. This implies that Exchange Server 2010 can manage concurrent session threads allowing the ability to read the original contents of a database page even though an active update transaction may not have been committed. In the event that the transaction is aborted, the version store will allow ESE98 to roll back or undo the transaction and return the data pages to their initial state.

---

**The ESE Database Cache**

Exchange Server 2003 runs on a 32-bit operating system, which limits the maximum size of the virtual address space to 4 GB. The operating system leaves only 2 GB of addressable RAM for Exchange (or 3 GB when the /3GB switch is set in the server boot.ini file). With such a limited amount of addressable RAM available, the size of the database cache must be carefully managed to allow Exchange to perform at its highest level. In Exchange Server 2003 the maximum amount of ESE cache was set to 900 MB. Even if the 64-bit version of Windows Server 2003 is used, the maximum ESE database cache size does not increase. This limitation in the ESE Database cache size is...
one of the major contributors to having a greater number of physical disk drives to support the I/O requirements for Exchange Server 2003 because more requests had to be retrieved from the database sitting on the storage subsystem, rather than be satisfied from server-side cache.

With the 64-bit architecture in Exchange Server 2007 and Exchange Server 2010, the maximum size of the database cache is no longer constrained by limits on the virtual address space. Instead, it is determined by the amount of available memory and by database I/O. For example, on a server that has 32 GB of physical RAM, ESE may increase the database cache to 16 GB if this amount is sufficient to meet its memory needs, and leave the remaining memory for system cache and other applications that are running on the server. A large database cache greatly increases performance because disk I/O is reduced and the ability to read information from memory is much faster than having to read information from a disk. Because the maximum size of the database cache is not set by default, ESE can increase the size of the database cache to consume almost all available RAM on the server if there is enough database I/O pressure to justify the increase. If other applications or the Windows Server system cache request or require memory, ESE decreases the size of the database cache as required. ESE does not increase the size of the cache unless there is enough database I/O pressure to justify the growth.

The fundamental construct of an Exchange database file is its B-tree structure. B-tree is a proven database technology that allows for optimized data access, with significant historical precedent. The key to B-tree design is a hierarchical arrangement of data pages that enforces certain structural constraints. The name is derived from the structural design, which is similar to an upside-down tree, starting with a root node and extending out to leaf nodes. The B in B-tree simply refers to “Balanced,” hence the name “B-tree.” A graphical example of a B-tree is provided in Figure 1 on page 37.

The root page (node) is the first parent of the tree, and all other nodes are linked from this page. With B-tree technology, each node can have only one parent, but parent nodes can have from zero to N child nodes under them. In this way, the ESE98 pages are arranged into tables that form a large database file used for storing Exchange data. For Exchange Server 2010 the page size was increased to 32 KB. The page size is the minimum size for reading and writing to the database; it is also the unit size used for database caching. Reading
from the disk is slower than performing operations in memory; therefore, by increasing the page size to 32 KB, ESE reduces IOPS, which increases performance by caching the larger page size in memory.

Figure 1 Balanced B-tree

Microsoft has made some key design changes to the B-tree structure over the years, in order to ensure that data access is fast and efficient. B-tree technology typically does not specify the depth and width of these B-tree structures (other than they must be balanced), which means that a B-tree can extend to an unlimited number of levels (called tree depth) and width (called the fan-out, degree, or branching factor). ESE limits the depth and fan-out of the B-tree such that every 32 KB database page of data in the database can be accessed with a consistent number of I/O operations to the database.

ESE98 enhances B-tree technology by implementing the B-tree variant that adds an additional horizontal relationship between nodes in the tree structure. This feature lets database pages point, not only to parent and child nodes, but also to the adjacent nodes. Tree depth has the greatest effect on performance. A uniform tree depth across the entire structure (every leaf node or data page is equidistant from the root node) implies database performance is consistent and predictable.

Note: This “B-tree” technology is additionally deployed using ESE98 in Microsoft Active Directory, WINS, and DNS.

From the perspective of the Exchange IS process (Store.exe), all the stored “data” appears in the form of database tables that are joined and linked to represent what the user references as their e-mail
“Inbox.” Each table comprises one B-tree that contains user data; in addition, this data may also include many secondary index B-trees that provide alternate mailbox views.

The Extensible Storage Engine (ESE) has been improved in Exchange Server 2010 to achieve the following goals:

- **Larger I/O and sequential I/O to reduce IOPS.** By increasing the size of the I/O and reducing the frequency of read/writes in Exchange Server 2010, ESE is able to increase performance. In addition, ESE can increase performance by making the data in the database more sequential, which increases the likelihood that related data is in the same vicinity in the B-tree. In all versions of Exchange Server, the data inside the database is stored in B-trees as already described, and the B-trees are then divided into pages. In Exchange Server 2007 and earlier, the data stored in the B-trees was not contiguous. In fact, in all versions of Exchange Server up to and including Exchange Server 2007, performed random I/O reads and writes to the database. This meant that related data would not be in the same vicinity on the hard disk. Non-contiguous data requires more passes to read and write to the hard disk.

- **Optimization for storage.** Disk subsystems are more efficient at handling fewer but larger I/O. In all versions of Exchange Server, the page size is the minimum read/write size and the minimum size for database caching. Coalescing I/Os refers to the process of combining database page operations into a single I/O operation, thereby producing fewer and bigger I/O operations. Increasing the average database I/O sizes through coalescing I/Os has the following benefits:
  - Increased disk use efficiency—Disks are more efficient at processing large I/Os. The more efficiently the disk is utilized, the more mailboxes can be hosted on that disk.
  - Increased cache warming rate—Cache warming is a process that helps reduce the execution times by preloading the initial queries that were executed against a database the last time the database was started. After a server restart, failover, or switchover, the larger I/O allows ESE to increase the rate at which the cache is warmed.
**Database management reduction.** One of the goals of ESE in Exchange Server 2010 was to reduce the cost of maintaining and managing a database. Database maintenance is comprised of several tasks that manage and keep the integrity of the mailbox database. Database maintenance is divided into the following:

- Store mailbox maintenance
- ESE database maintenance

In Exchange Server 2007, the ESE database maintenance was disk-intensive. For Exchange Server 2010, improvements have been made to increase performance. On large or very heavy profile servers, the store mailbox maintenance task will only last approximately 45 minutes, while ESE database maintenance, on previous versions of Exchange Server, usually took from six to eight hours per night to complete on large databases.

**Online defragmentation.** The architecture for online defragmentation has changed for Exchange Server 2010. Online defragmentation was moved out of the Mailbox database maintenance process. Online defragmentation now runs in the background 24×7. No settings to configure this feature are required as Exchange Server 2010 monitors the database as it is being used. Small changes are made over time to keep it defragmented for space and contiguity. If the database analyzes a range of pages and finds that they are not as sequential as they should be, it starts an asynchronous thread to defragment that section of the B-tree/table. Online defragmentation is also throttled so it does not have a negative impact on client performance.

**Online database scanning.** Online database scanning (also known as database checksumming) has also changed. For Exchange Server 2007 SP1, the option to use half of the online defragmentation time for this database scanning process (to ensure Exchange read every page from your database in a specific period of time to detect any corruptions) was available. This I/O is 100 percent sequential (thus easy on the disk) and on most systems equated to a check-summing rate of about 5 megabytes (MB)/sec.

In Exchange Server 2010, online database scanning simply checksums the database and performs post Exchange Server 2010 Store crash operations. Space can be leaked due to crashes, and online database scanning finds and recovers lost space. The system in Exchange Server 2010 is designed with the expectation...
that every database is fully scanned once every 3 days. A warning event is fired if the databases are not scanned. In Exchange Server 2010, there are now two modes to run online database scanning on active database copies:

- Run as the last task in the scheduled Mailbox Database Maintenance process. It can be configured as to how long it runs by changing the Mailbox Database Maintenance schedule. This option works well for smaller databases that are less than 1 terabyte (TB) in size.

- Run in the background 24x7, the default behavior. This option works well for larger databases, up to 2 TB, where more time to checksum the databases is needed. Exchange scans the database no more than once per day and again will fire a warning event if it can’t finish scanning in a 7-day period.

Online database scanning can be configured to not run 24x7 by clearing the checkbox (if the I/O overhead during a peak period is not wanted). When this is done, the last part of the online database scanning window is used to do the database checksumming and lost space recovery.

### Extensible Storage Engine database page structure

An ESE98 database can contain up to $2^{32}$ (2 to the 32nd power) Exchange database pages or 16 TB. Technically, the practical database size is limited by the ability to back up, restore, and perform other maintenance routines on the database (online or offline defragmentation and database repairs) in a timely manner.

The page size in ESE is determined by the associated application. Table 2 on page 40 lists various ESE versions, associated applications, and page sizes.

<table>
<thead>
<tr>
<th>ESE version</th>
<th>Application</th>
<th>Data page size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESE97</td>
<td>Exchange Server 5.5</td>
<td>4 KB</td>
</tr>
<tr>
<td>ESE98</td>
<td>Exchange Server 2000</td>
<td>4 KB</td>
</tr>
<tr>
<td></td>
<td>Exchange Server 2003</td>
<td>4 KB</td>
</tr>
</tbody>
</table>
Table 2  ESE versions and page size (page 2 of 2)

<table>
<thead>
<tr>
<th>ESE version</th>
<th>Application</th>
<th>Data page size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESE98</td>
<td>Exchange Server 2007</td>
<td>8 KB</td>
</tr>
<tr>
<td>ESE98</td>
<td>Exchange Server 21010</td>
<td>32KB</td>
</tr>
<tr>
<td>ESENT</td>
<td>Active Directory, DNS, WINS</td>
<td>8 KB</td>
</tr>
</tbody>
</table>
Exchange Server Store Instances

Microsoft Exchange first introduced the concept of storage groups with the release of Exchange Server 2000 and this functionality was refined and enhanced for Exchange Server 2007. This storage group functionality represents the fundamental cornerstone in storage design as well as for server scaling with Exchange Server 2007. Although a single storage group can accommodate up to five databases as was the case with Exchange Server 2000 and 2003, in Exchange Server 2007 the maximum number of databases has risen from 20 to 50 and unlike Exchange 2000 and 2003, the maximum number of storage groups in Exchange Server 2007 is 50. However the maximum number of databases is also 50. In most environments the use of one database per storage group is used. In the case of using cluster continuous replication (CCR) or standby continuous replication (SCR) there can only be one database per storage group. A detailed explanation of the new clustering features is defined later in this chapter.

For Exchange Server 2010 the concept of storage groups for the databases removed and now the storage group functionality has been moved into the database.

An Exchange Server 2010 store is defined, therefore, as the combination of an instance in memory of the ESE (ESE98) that runs as a subprocess of Store.exe, and the set of database and log files contained within that storage group. Each store represents a separate running instance of ESE98, the database engine that drives Exchange Server 2010.

Because each store represents an independent instance of the database engine (ESE98), administrative access to those databases is now controlled at the store level.

Stores also allow the Exchange Server administrator greater flexibility and granularity in partitioning groups of end users. Users may, for example, be grouped by Exchange user profile (workload), or perhaps by mailbox quota size. As a result, careful consideration and planning should be given to store design. These decisions will ultimately impact server and backup/restore performance, and disaster/recovery design.
In Exchange 2000 and 2003, the only high-availability options were to use Microsoft Cluster Service (MSCS) and the shared disk model. In Exchange Server 2007 and 2010, the only role that can be made highly available is the Mailbox Server role. High availability for the other server roles is achieved by using more than one server in each role, which provides load balancing and a level of redundancy. The high-availability options are:

**Single copy clusters**— The shared storage cluster configurations in versions up to and including Exchange 2007 is called a single copy cluster (SCC). An SCC uses the Cluster service and shared storage to host a Clustered Mailbox Server (CMS); in Exchange Server 2000 and 2003 this was called an Exchange Virtual Server (EVS). A CMS is a logical computer that moves between physical nodes. In Exchange Server 2007, the setup automatically creates a floating network identity using the hostname and the IP address provided by the administrator. The floating network identity moves between the nodes in the cluster, based on node availability and maintenance needs. These mechanisms allow users to access their mailbox data if the storage is available and at least one of the two nodes is operational. To make failure recovery work, Exchange and the Cluster service work together to bring the clustered mailbox server online on an available node after a failure.
Figure 2  Basic Single Copy Cluster

In Exchange 2007 most of the cluster administration has been moved from the Cluster Administration console into the Exchange Server 2007 admin tools such as the Exchange Management Shell.

When installed into a cluster configuration, the installation locations of storage group data differ from those of a standard local installation. All data and transaction log files must be located on disk devices viewed by the cluster as shared disks. This is a requirement to ensure that both nodes within a given cluster are able to access all required databases within the CMS. By default, the Exchange Server 2007 instance itself will implement a resource dependency on the shared disk resources installed within the resource group. This dependency must be maintained, and any modifications to the storage group structure that may introduce an additional disk resource, such as adding a database file on a new shared LUN, must be replicated within the resource group, by adding the new disk as a resource within the group, and adding it as a dependency for the CMS.
Solutions built using EMC’s geographically dispersed cluster product SRDF®/CE for MSCS implement identical functionality and restrictions. In Figure 2 on page 44, an SRDF/CE for MSCS implementation is displayed. The dependencies as described are required to be maintained, but now additionally include the disks depending on the SRDF/CE for MSCS resource. In this manner, states for RDF devices are managed appropriately before upper-level services attempt to start.

In many ways, the implementation of SRDF/CE for MSCS is somewhat transparent as compared to a standard MSCS installation. Unlike the shared database files and logs for all the storage groups within the CMS, the Exchange Server executables are installed locally to each node within the cluster.

**Local continuous replication**—Local continuous replication (LCR) is a single-server solution that uses Exchange Server 2007 built-in technology to create and maintain a copy of a storage group on a second set of disks that are connected to the same server as the production storage group. LCR provides asynchronous log shipping, log replay, and a quick manual switch to a copy of the data. The production storage group is referred to as the active copy, and the copy of the storage group maintained on the separate set of disks is referred to as the passive copy. Figure 3 on page 46 illustrates a basic deployment of LCR.
Cluster continuous replication—Exchange CCR is a high-availability feature of Exchange 2007 that combines asynchronous log shipping and replay features of Exchange 2007 with the failover and management features provided by the Microsoft Cluster Service (MSCS) on Windows Server 2003 and Windows Failover Clustering on Windows Server 2008 and later.

The Exchange CCR cluster is a two-node Majority Node Set (MNS) failover cluster with a file share witness. The active node hosts the Clustered Mailbox Server that serves Exchange data from the active copy of the storage groups. The passive node hosts a hot standby or passive copy of the storage groups and Exchange CCR continuously ships the logs from the active to the passive copy and replays the logs into the database that resides in the passive copy of the storage group. Figure 4 on page 47 illustrates a basic Exchange CCR environment.

If Exchange Server 2007 is installed on Windows Server 2003 clustering then both of the nodes must reside on the same IP subnet. If it is installed on Windows Server 2008 or later, the cluster can span IP subnets but they must reside within the same AD site.
Exchange Server 2007 Service Pack 1 adds an additional feature designed to provide site resilience.

**Standby continuous replication**—Standby continuous replication (SCR) is a new feature introduced in Exchange 2007 SP1. As its name implies, SCR is designed for scenarios that use or enable the use of standby recovery servers. SCR extends the existing continuous replication features and enables new data availability scenarios for Exchange 2007 Mailbox servers. SCR uses the same log shipping and replay technology used by LCR and CCR to provide added deployment options and configurations. SCR can be used to replicate data from both stand-alone Mailbox servers and clustered mailbox servers. SCR can be used on any Mailbox server configuration. SCR cannot be enabled for a storage group that contains more than one database. No LCR or CCR can be enabled on the target. A source server can have multiple target servers and a target server can have multiple source servers. All Exchange target and production servers...
must be in the same Active Directory domain. An SCR server cannot exceed 50 databases. Also, as with CCR, SCR requires that the database and log file paths be the same on the source and the target.

High Availability options for Exchange Server 2010

Exchange Server 2010 includes many changes to its core architecture. It combines the key availability and resilience features of CCR and SCR, built into Exchange Server 2007, into single high availability solution which handles both onsite data replication and offsite data replication. Single copy clusters are no longer supported with Exchange Server 2010.

Mailbox servers can be defined as part of a database availability group (DAG) to provide automatic recovery at the individual mailbox database level instead of at the server level. Each mailbox database can have up to 16 copies. Other new high availability concepts are introduced in Exchange Server 2010, such as database mobility.

To summarize, the key aspects to data and service availability for the Mailbox server role and mailbox databases are:

- Exchange Server 2010 uses an enhanced version of the same continuous replication technology introduced in Exchange Server 2007. Continuous replication now operates at the database level. As Exchange Server 2010 still uses the ESE database which produces transaction logs, which are then replicated to one or more other locations and replayed into one or more copies of a mailbox database. Log shipping and seeding no longer uses Server Message Block (SMB) for data transfer. Exchange Server 2010 continuous replication uses a single administrator-defined TCP port for data transfer. In addition, Exchange Server 2010 includes built-in options for network encryption and compression for the data stream. Log shipping no longer uses a pull model, where the passive copy pulls closed log files from the active copy. Instead, the active copy pushes the log files to each configured passive copy. Seeding is no longer restricted to using only the active copy of the database. Passive copies of mailbox databases can now be specified as sources for database copy seeding and reseeding.

- As already stated, storage groups no longer exist in Exchange Server 2010. Instead, there are simply mailbox databases, mailbox database copies, and public folder databases. The primary management interfaces for Exchange databases has moved
within the Exchange Management Console from the Mailbox node under Server Configuration to the Mailbox node under Organization Configuration.

- Some Windows Failover Clustering technology is used by Exchange Server 2010, but it's now completely managed by Exchange. The need to install, build, or configure any aspects of failover clustering, by Administrators when deploying highly available Mailbox servers, has been removed.

- Each Mailbox server can host as many as 100 databases, and each database can have as many as 16 copies.

- In addition to the transport dumpster feature, introduced in Exchange Server 2007, a new Hub Transport server feature named shadow redundancy has been added. Shadow redundancy provides redundancy for messages for the entire time they are in transit. The solution involves a technique similar to the transport dumpster. With shadow redundancy, the deletion of a message from the transport database 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 back successful delivery, the message is resubmitted for delivery to that next hop.

**Database Availability Groups**

A Database Availability Group (DAG) is the base component of the high availability and site resilience framework built into Microsoft Exchange Server 2010. A DAG is a set of up to 16 Mailbox servers that provide automatic database-level recovery from failures that affect individual databases. Any server in a DAG can host a copy of a mailbox database from any other server in the DAG. When a server is added to a DAG, it works with the other servers in the DAG to provide automatic recovery from failures that affect mailbox databases, such as a disk failure or server failure.

A DAG is a boundary for mailbox database replication, database and server switchovers, failovers, and an internal Exchange Server 2010 component called Active Manager. Active Manager, which runs on every server in a DAG, manages switchovers and failovers.

Any server in a DAG can host a copy of a mailbox database from any other server in the DAG. When a server is added to a DAG, it works with the other servers in the DAG to provide automatic recovery from failures that affect mailbox databases, such as a disk failure or server failure.
A DAG is created by using the New-DatabaseAvailabilityGroup cmdlet from the Exchange Management Shell. A DAG is initially created as an empty object in Active Directory. This directory object is used to store relevant information about the DAG, such as server membership information. When the first server is added to a DAG, a failover cluster is automatically created for the DAG. This failover cluster is used exclusively by the DAG, and the cluster must be dedicated to the DAG. Use of the cluster for any other purpose is not supported.

In addition to a failover cluster being created, the infrastructure that monitors the servers for network or server failures is initiated. The failover cluster heartbeat mechanism and cluster database are then used to track and manage information about the DAG that can change quickly, such as database mount status, replication status, and last mounted location.

During creation, the DAG is given a unique name, and either assigned one or more static IP addresses.

Figure 5 on page 50 illustrates how a DAG can provide high availability for mailbox databases.
In Figure 5 on page 50, the yellow databases are active mailbox database copies and the green databases are passive mailbox database copies. In this example, the database copies are not mirrored across each server, but rather spread across multiple servers. This ensures that no two servers in the DAG have the same set of database copies, providing the DAG with greater resilience to failures, including failures that occur while other components are unavailable as a result of regular maintenance.

On a Server Failure, or for planned maintenance, the active mailbox databases have to be made active on one of the other servers within the DAG. A server switchover, which activates the copy of DB4 on another Mailbox server, SVR3. A server switchover (manual or automatic due to server failure) moves all active mailbox database copies from their current server to one or more other Mailbox servers in the DAG. The database will be moved automatically to servers that can best cope with the extra workload, unless explicitly specified in the Exchange management shell cmdlet when moving databases for routine maintenance.

In Figure 6 on page 51, there’s only one active mailbox database on SVR4 (DB4), so only one active mailbox database copy is moved to SVR3, whilst the server is undergoing maintenance or recovered from failure.
In Figure 7 on page 52 After SVR4 is available, the other members of the DAG are notified, and the copies of DB1, DB4, and DB5 hosted on SVR4 are automatically synchronized with the active copy of each database.
In Figure 8 on page 53 the DAG has been extended across two Active Directory sites, Boston and Cork. Each server in the Boston data center has a passive copy of each active database in the Boston data center and SVR6 in the Cork data center also hosts a passive copy of each database. SVR6 in the Cork data center could not only also host passive database copies but it could host all active copies, or it could host a mixture of active and passive copies.
In addition to SVR6, multiple DAG members could be deployed in the Cork data center, providing protection against additional failures. This configuration also provides additional capacity, so that if the Boston data center fails, the Cork data center can support a much larger user population.
Support for streaming backup APIs was removed from Exchange Server 2010 so the only option for creating a backup of the Exchange Server 2010 databases is by utilizing the Microsoft VSS framework. The Microsoft VSS provides a framework for using snapshot and cloning technologies with the Windows Server 2008 platform. More specifically, VSS provides services that deliver an infrastructure upon which Windows applications developed by Microsoft and third-party vendors such as EMC Corporation can leverage replication technologies for the purpose of backup and restore. For EMC, this facilitates integration of backup/restore operations with hardware-based disk mirroring technology for local (TimeFinder®) or remote (SRDF) execution.

The VSS framework has three primary goals:

- Coordinate application state, to facilitate cloning processes, including synchronizing application data spread over multiple volumes.
- Provide discovery and enumeration of snapshots or clones (called Shadow Copies).
- Provide a framework in which hardware and software vendors can plug in interoperable shadow copy creation components (called Providers).

VSS on Windows Server 2008 allows third-party vendors to implement solutions for backup/restore using Requestor applications which coordinate with the VSS framework. Application-specific Writer processes coordinate application state to facilitate the shadow copy creation. Third-party vendors may also create hardware or software providers which the VSS framework can utilize to implement product-specific technologies and features. The Windows Server’s VSS architecture is represented in Figure 9 on page 56.
VSS Providers

For a vendor’s hardware- or software-based disk mirror technology to function within the VSS framework, the vendor must develop a VSS Provider—a component that manages volumes and creates clones and snapshots according to a specific vendor’s technology and implementation. Typically, a VSS Provider is a module that manages a physical Shadow Copy and exposes that Shadow Copy to the operating system or applications.

Note: Windows Server 2008 includes a software-based provider that the OS implements as a copy-on-write software snapshot.

VSS Requestors

Backup/restore and disaster-recovery solution vendors can develop applications that make use of the VSS architecture. To do so, these vendors must develop a requestor—an automated or GUI-based process or application that requests one or more shadow copy sets from one or more volumes. The requestor is the main process that communicates with the shadow copy interface that coordinates...
activities between providers and writers. This communication lets a requestor select which components (for example, one or more storage groups) should be Shadow Copied to complete the requirements of the backup operation.

**VSS Writers**

The most important participants within the VSS framework are, arguably, the applications. An application must carefully expose and coordinate backup/recovery functions that are specific to the application’s technology, implementation, and disaster-recovery requirements. Writers respond to Shadow Copy requests to let the application prepare, freeze, and thaw I/O to ensure that no writes occur on the volume when the Provider creates the Shadow Copy.

A backup operation performed with VSS is a well-orchestrated process that involves the interaction of each component in the VSS framework. Figure 10 on page 58 details generalized flow and interaction diagram of a backup operation using VSS technology.
To support the VSS framework, an application such as Exchange must provide the Writer component. Microsoft does not support VSS operations with any version of Exchange before Exchange Server 2003.

Exchange Server 2010, paired with Windows Server 2008, is required to provide VSS support for Information Store backup and recovery for Exchange Server 2010. With Exchange Server 2010, the Writer functionality has been built into the Store process. This Writer provides the necessary support for Requestors to initiate backup and restore operations for Exchange Server 2010.
Exchange Server 2010 and EMC integration

Operationally, the primary level of integration is provided around backup/recovery and restart operations. EMC provides a number of utilities that can be used to facilitate the administration of backup and recovery procedures. These utilities are designed to reduce operating system overhead and mitigate the amount of time required to perform backup and recovery operations.

Products such as the EMC® TimeFinder Exchange Integration Module (TF/EIM) and EMC Replication Manager (RM) facilitate low-impact backup and recovery procedures that utilize disk mirror technologies to optimize execution time and enhance availability. The backup images created by these products represent a valid backup image for an Exchange Server 2010 storage group by using Exchange Server 2010’s VSS Framework. Utilizing EMC Symmetrix Remote Data Facility (SRDF), customers can create or migrate the backup images to remote arrays, providing the capability of creating disaster recovery or disaster restart solutions.

EMC provides a series of solutions based on consistency technology that can be used to provide a valid restart point for individual stores of Exchange Server 2010 instances. More importantly, this functionality allows customers to create larger federated restart solutions. These are generally represented by a number of disparate heterogeneous messaging and database environments tied together to create a business application or process.

Restart solutions provide functionality beyond standard backup/recovery processes. It is often impossible to utilize standard backup and recovery processes to create a business point of consistency across multiple environments.
This chapter introduces the EMC foundation products discussed in this document that work in Microsoft Infrastructure environments:

- Introduction .......................................................... 62
- Symmetrix hardware and EMC Enginuity features .......... 65
- EMC Solutions Enabler base management ....................... 69
- EMC Change Tracker .................................................. 72
- EMC Symmetrix Remote Data Facility ............................... 73
- EMC TimeFinder ........................................................... 88
- EMC Storage Resource Management ................................. 101
- EMC PowerPath ......................................................... 106
- EMC Replication Manager ........................................... 115
- EMC Open Replicator .................................................. 117
- EMC Virtual Provisioning ............................................. 118
- EMC Virtual LUN migration .......................................... 121
- EMC Symmetrix Fully Automated Storage Tiering (FAST) ..... 124
- EMC Symmetrix Fully Automated Storage Tiering for Virtual Pools (FAST VP) ............................................. 126
EMC offers an extensive product line of high-end storage solutions targeted to meet the requirements of mission-critical databases and applications. The Symmetrix product line includes the DMX Direct Matrix Architecture™ series and the VMAX™ Virtual Matrix™ series. EMC Symmetrix is a fully redundant, high-availability storage processor, providing nondisruptive component replacements and code upgrades. The Symmetrix system features high levels of performance, data integrity, reliability, and availability.

**EMC Enginuity™ Operating Environment** — Enginuity enables interoperation between the latest Symmetrix platforms and previous generations of Symmetrix systems and enables them to connect to a large number of server types, operating systems and storage software products, and a broad selection of network connectivity elements and other devices, ranging from HBAs and drivers to switches and tape systems.

**EMC Solutions Enabler** — Solutions Enabler is a package that contains the SYMAPI runtime libraries and the SYMCLI command line interface. SYMAPI provides the interface to the EMC Enginuity operating environment. SYMCLI is a set of commands that can be invoked from the command line or within scripts. These commands can be used to monitor device configuration and status, and to perform control operations on devices and data objects within a storage complex.

**EMC Symmetrix Remote Data Facility (SRDF®)** — SRDF is a business continuity software solution that replicates and maintains a mirror image of data at the storage block level in a remote Symmetrix system. The SRDF component extends the basic SYMCLI command set of Solutions Enabler to include commands that specifically manage SRDF.

**EMC SRDF consistency groups** — An SRDF consistency group is a collection of related Symmetrix devices that are configured to act in unison to maintain data integrity. The devices in consistency groups can be spread across multiple Symmetrix systems.

**EMC TimeFinder®** — TimeFinder is a family of products that enable LUN-based replication within a single Symmetrix system. Data is copied from Symmetrix devices using array-based resources without using host CPU or I/O. The source Symmetrix devices remain online
for regular I/O operations while the copies are created. The TimeFinder family has three separate and distinct software products, TimeFinder/Mirror, TimeFinder/Clone, and TimeFinder/Snap:

- **TimeFinder/Mirror** enables users to configure special devices, called business continuance volumes (BCVs), to create a mirror image of Symmetrix standard devices. Using BCVs, TimeFinder creates a point-in-time copy of data that can be repurposed. The TimeFinder/Mirror component extends the basic SYMCLI command set of Solutions Enabler to include commands that specifically manage Symmetrix BCVs and standard devices.

- **TimeFinder/Clone** enables users to make copies of data simultaneously on multiple target devices from a single source device. The data is available to a target’s host immediately upon activation, even if the copy process has not completed. Data may be copied from a single source device to as many as 16 target devices. A source device can be either a Symmetrix standard device or a TimeFinder BCV device.

- **TimeFinder/Snap** enables users to configure special devices in the Symmetrix array called virtual devices (VDEVs) and save area devices (SAVDEVs). These devices can be used to make pointer-based, space-saving copies of data simultaneously on multiple target devices from a single source device. The data is available to a target’s host immediately upon activation. Data may be copied from a single source device to as many as 128 VDEVs. A source device can be either a Symmetrix standard device or a TimeFinder BCV device. A target device is a VDEV. A SAVDEV is a special device without a host address that is used to hold the changing contents of the source or target device.

**EMC Change Tracker** — EMC Symmetrix Change Tracker software measures changes to data on a Symmetrix volume or group of volumes. Change Tracker software is often used as a planning tool in the analysis and design of configurations that use the EMC TimeFinder or SRDF components to store data at remote sites.

**Solutions Enabler Storage Resource Management (SRM) component** — The SRM component extends the basic SYMCLI command set of Solutions Enabler to include commands that allow users to systematically find and examine attributes of various objects on the host, within a specified relational database, or in the EMC
enterprise storage. The SRM commands provide mapping support for relational databases, file systems, logical volumes and volume groups, as well as performance statistics.

**EMC PowerPath®** — PowerPath is host-based software that provides I/O path management. PowerPath operates with several storage systems, on several enterprise operating systems and provides failover and load balancing transparent to the host application and database.
Symmetrix hardware and EMC Enginuity features

Symmetrix hardware architecture and the EMC Enginuity operating environment are the foundation for the Symmetrix storage platform. This environment consists of the following components:

- Symmetrix hardware
- Enginuity-based operating functions
- Solutions Enabler
- Symmetrix application program interface (API) for mainframe
- Symmetrix-based applications
- Host-based Symmetrix applications
- Independent software vendor (ISV) applications

All Symmetrix systems provide advanced data replication capabilities, full mainframe and open systems support, and flexible connectivity options, including Fibre Channel, FICON, ESCON, Gigabit Ethernet, and iSCSI.

Interoperability between Symmetrix storage systems enables customers to migrate storage solutions from one generation to the next, protecting their investment even as their storage demands expand.

Symmetrix enhanced cache director technology allows configurations of up to 512 GB of cache. The cache can be logically divided into 32 independent regions providing up to 32 concurrent 500 MB/s transaction throughput.

The Symmetrix on-board data integrity features include:

- Continuous cache and on-disk data integrity checking and error detection/correction
- Fault isolation
- Nondisruptive hardware and software upgrades
- Automatic diagnostics and phone-home capabilities

At the software level, advanced integrity features ensure information is always protected and available. By choosing a mix of RAID 1 (mirroring), RAID 1/0, high performance RAID 5 (3+1 and 7+1) protection and RAID 6 (6+2 and 14+2), users have the flexibility to
choose the protection level most appropriate to the value and performance requirements of their information. The Symmetrix VMAX is EMC’s latest generation of high-end storage solutions.

From the perspective of the host operating system, a Symmetrix system appears to be multiple physical devices connected through one or more I/O controllers. The host operating system addresses each of these devices using a physical device name. Each physical device includes attributes, vendor ID, product ID, revision level, and serial ID. The host physical device maps to a Symmetrix device. In turn, the Symmetrix device is a virtual representation of a portion of the physical disk called a hypervolume.

### Symmetrix VMAX platform

The EMC Symmetrix VMAX Series with Enginuity is a new entry to the Symmetrix product line. Built on the strategy of simple, intelligent, modular storage, it incorporates a new scalable fabric interconnect design that allows the storage array to seamlessly grow from an entry-level configuration into the world’s largest storage system. The Symmetrix VMAX provides improved performance and scalability for demanding enterprise storage environments while maintaining support for EMC’s broad portfolio of platform software offerings.

The Enginuity operating environment for Symmetrix version 5875 is a new, feature-rich Enginuity release supporting Symmetrix VMAX storage arrays. With the release of Enginuity 5875, Symmetrix VMAX systems deliver new software capabilities that improve capacity utilization, ease of use, business continuity and security.

The Symmetrix VMAX also maintains customer expectations for high-end storage in terms of availability. High-end availability is more than just redundancy; it means nondisruptive operations and upgrades, and being “always online.” Symmetrix VMAX provides:

- Nondisruptive expansion of capacity and performance at a lower price point
- Sophisticated migration for multiple storage tiers within the array
- The power to maintain service levels and functionality as consolidation grows
- Simplified control for provisioning in complex environments

Many of the new features provided by the new EMC Symmetrix VMAX platform can reduce operational costs for customers deploying VMware Infrastructure environments, as well as enhance...
Symmetrix hardware and EMC Enginuity features

EMC Foundation Products

functionality to enable greater benefits. This document details those features that provide significant benefits to customers deploying VMware Infrastructure environments.

Figure 11 on page 67 illustrates the architecture and interconnection of the major components in the Symmetrix VMAX storage system.

![Symmetrix VMAX logical diagram](ICO-IMG-000752)

**Figure 11   Symmetrix VMAX logical diagram**

---

**EMC Enginuity operating environment**

EMC Enginuity is the operating environment for all Symmetrix storage systems. Enginuity manages and ensures the optimal flow and integrity of data through the different hardware components. It also manages Symmetrix operations associated with monitoring and optimizing internal data flow. This ensures the fastest response to the user's requests for information, along with protecting and replicating data. Enginuity provides the following services:

- Manages system resources to intelligently optimize performance across a wide range of I/O requirements.
- Ensures system availability through advanced fault monitoring, detection, and correction capabilities and provides concurrent maintenance and serviceability features.
EMC Foundation Products

- Offers the foundation for specific software features available through EMC disaster recovery, business continuity, and storage management software.
- Provides functional services for both Symmetrix-based functionality and for a large suite of EMC storage application software.
- Defines priority of each task, including basic system maintenance, I/O processing, and application processing.
- Provides uniform access through APIs for internal calls, and provides an external interface to allow integration with other software providers and ISVs.
EMC Solutions Enabler base management

The EMC Solutions Enabler kit contains all the base management software that provides a host with SYMAPI-shared libraries and the basic Symmetrix command line interface (SYMCLI). Other optional subcomponents in the Solutions Enabler (SYMCLI) series enable users to extend functionality of the Symmetrix systems. Three principle sub-components are:

- Solutions Enabler SYMCLI SRDF, SRDF/CG, and SRDF/A
- Solutions Enabler SYMCLI TimeFinder/Mirror, TimeFinder/CG, TimeFinder/Snap, TimeFinder/Clone
- Solutions Enabler SYMCLI Storage Resource Management (SRM)

These components are discussed later in this chapter.

SYMCLI resides on a host system to monitor and perform control operations on Symmetrix storage arrays. SYMCLI commands are invoked from the host operating system command line or via scripts. SYMCLI commands invoke low-level channel commands to specialized devices on the Symmetrix called gatekeepers. Gatekeepers are very small devices carved from disks in the Symmetrix that act as SCSI targets for the SYMCLI commands.

SYMCLI is used in single command line entries or in scripts to monitor and perform control operations on devices and data objects toward the management of the storage complex. It also monitors device configuration and status of devices that make up the storage environment. To reduce the number of inquiries from the host to the Symmetrix systems, configuration and status information is maintained in a host database file.

Table 3 lists the SYMCLI base commands discussed in this document.

<table>
<thead>
<tr>
<th>Command</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>symdg</td>
<td></td>
<td>Performs operations on a device group (dg)</td>
</tr>
<tr>
<td>create</td>
<td></td>
<td>Creates an empty device group</td>
</tr>
<tr>
<td>delete</td>
<td></td>
<td>Deletes a device group</td>
</tr>
<tr>
<td>rename</td>
<td></td>
<td>Renames a device group</td>
</tr>
</tbody>
</table>
### Table 3  SYMCLI base commands (page 2 of 3)

<table>
<thead>
<tr>
<th>Command</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>symdg (cont)</td>
<td>release</td>
<td>Releases a device external lock associated with all devices in a device group</td>
</tr>
<tr>
<td></td>
<td>list</td>
<td>Displays a list of all device groups known to this host</td>
</tr>
<tr>
<td></td>
<td>show</td>
<td>Shows detailed information about a device group and any gatekeeper or BCV devices associated with the device group</td>
</tr>
<tr>
<td>symcg</td>
<td></td>
<td>Performs operations on a composite group (cg)</td>
</tr>
<tr>
<td></td>
<td>create</td>
<td>Creates an empty composite group</td>
</tr>
<tr>
<td></td>
<td>add</td>
<td>Adds a device to a composite group</td>
</tr>
<tr>
<td></td>
<td>remove</td>
<td>Removes a device from a composite group</td>
</tr>
<tr>
<td></td>
<td>delete</td>
<td>Deletes a composite group</td>
</tr>
<tr>
<td></td>
<td>rename</td>
<td>Renames a composite group</td>
</tr>
<tr>
<td></td>
<td>release</td>
<td>Releases a device external lock associated with all devices in a composite group</td>
</tr>
<tr>
<td></td>
<td>hold</td>
<td>Hold devices in a composite group</td>
</tr>
<tr>
<td></td>
<td>unhold</td>
<td>Unhold devices in a composite group</td>
</tr>
<tr>
<td></td>
<td>list</td>
<td>Displays a list of all composite groups known to this host</td>
</tr>
<tr>
<td></td>
<td>show</td>
<td>Shows detailed information about a composite group, and any gatekeeper or BCV devices associated with the group</td>
</tr>
<tr>
<td>symld</td>
<td></td>
<td>Performs operations on a device in a device group</td>
</tr>
<tr>
<td></td>
<td>add</td>
<td>Adds devices to a device group and assigns the device a logical name</td>
</tr>
<tr>
<td></td>
<td>list</td>
<td>Lists all devices in a device group and any associated BCV devices</td>
</tr>
<tr>
<td></td>
<td>remove</td>
<td>Removes a device from a device group</td>
</tr>
<tr>
<td></td>
<td>rename</td>
<td>Renames a device in the device group</td>
</tr>
<tr>
<td></td>
<td>show</td>
<td>Shows detailed information about a device in a the device group</td>
</tr>
<tr>
<td>Command</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>symbcv</td>
<td></td>
<td>Performs support operations on BCV pairs</td>
</tr>
<tr>
<td>list</td>
<td></td>
<td>Lists BCV devices</td>
</tr>
<tr>
<td>associate</td>
<td></td>
<td>Associates BCV devices to a device group – required to perform operations on the BCV device</td>
</tr>
<tr>
<td>disassociate</td>
<td></td>
<td>Disassociates BCV devices from a device group</td>
</tr>
<tr>
<td>associate -rdf</td>
<td></td>
<td>Associates remotely attached BCV devices to a SRDF device group</td>
</tr>
<tr>
<td>disassociate -rdf</td>
<td></td>
<td>Disassociates remotely attached BCV devices from an SRDF device group</td>
</tr>
</tbody>
</table>
EMC Change Tracker

The EMC Symmetrix Change Tracker software is also part of the base Solutions Enabler SYMCLI management offering. Change Tracker commands are used to measure changes to data on a Symmetrix volume or group of volumes. Change Tracker functionality is often used as a planning tool in the analysis and design of configurations that use the EMC SRDF and TimeFinder components to create copies of production data.

The Change Tracker command (symchg) is used to monitor the amount of changes to a group of hypervolumes. The command timestamps and marks specific volumes for tracking and maintains a bitmap to record which tracks have changed on those volumes. The bitmap can be interrogated to gain an understanding of how the data on the volume changes over time and to assess the locality of reference of applications.
EMC Symmetrix Remote Data Facility

The Symmetrix Remote Data Facility (SRDF) component of EMC Solutions Enabler extends the basic SYMCLI command set to enable users to manage SRDF. SRDF is a business continuity solution that provides a host-independent, mirrored data storage solution for duplicating production site data to one or more physically separated target Symmetrix systems. In basic terms, SRDF is a configuration of multiple Symmetrix systems whose purpose is to maintain multiple copies of logical volume data in more than one location.

SRDF replicates production or primary (source) site data to a secondary (target) site transparently to users, applications, databases, and host processors. The local SRDF device, known as the source (R1) device, is configured in a partner relationship with a remote target (R2) device, forming an SRDF pair. While the R2 device is mirrored with the R1 device, the R2 device is write-disabled to the remote host. After the R2 device synchronizes with its R1 device, the R2 device can be split from the R1 device at any time, making the R2 device fully accessible again to its host. After the split, the target (R2) device contains valid data and is available for performing business continuity tasks through its original device address.

SRDF requires configuration of specific source Symmetrix volumes (R1) to be mirrored to target Symmetrix volumes (R2). If the primary site is no longer able to continue processing when SRDF is operating in synchronous mode, data at the secondary site is current up to the last committed transaction. When primary systems are down, SRDF enables fast failover to the secondary copy of the data so that critical information becomes available in minutes. Business operations and related applications may resume full functionality with minimal interruption.

Figure 12 on page 74 illustrates a basic SRDF configuration where connectivity between the two Symmetrix is provided using ESCON, Fibre Channel, or Gigabit Ethernet. The connection between the R1 and R2 volumes is through a logical grouping of devices called a remote adapter (RA) group. The RA group is independent of the device and composite groups defined and discussed in “SRDF device groups and composite groups” on page 75.
SRDF benefits

SRDF offers the following features and benefits:

- High data availability
- High performance
- Flexible configurations
- Host and application software transparency
- Automatic recovery from a component or link failure
- Significantly reduced recovery time after a disaster
- Increased integrity of recovery procedures
- Reduced backup and recovery costs
- Reduced disaster recovery complexity, planning, testing, etc.
- Supports Business Continuity across and between multiple databases on multiple servers and Symmetrix systems.

SRDF modes of operation

SRDF currently supports the following modes of operation:

- *Synchronous mode (SRDF/S)* provides real-time mirroring of data between the source Symmetrix system(s) and the target Symmetrix system(s). Data is written simultaneously to the cache of both systems in real time before the application I/O is completed, thus ensuring the highest possible data availability.
Data must be successfully stored in both the local and remote Symmetrix systems before an acknowledgment is sent to the local host. This mode is used mainly for metropolitan area network distances less than 200 km.

- **Asynchronous mode (SRDF/A)** maintains a dependent-write consistent copy of data at all times across any distance with no host application impact. Applications needing to replicate data across long distances historically have had limited options. SRDF/A delivers high-performance, extended-distance replication and reduced telecommunication costs while leveraging existing management capabilities with no host performance impact.

- **Adaptive copy mode** transfers data from source devices to target devices regardless of order or consistency, and without host performance impact. This is especially useful when transferring large amounts of data during data center migrations, consolidations, and in data mobility environments. Adaptive copy mode is the data movement mechanism of the Symmetrix Automated Replication (SRDF/AR) solution.

### SRDF device groups and composite groups

Applications running on Symmetrix systems normally involve a number of Symmetrix devices. Therefore, any Symmetrix operation must ensure all related devices are operated upon as a logical group. Defining device or composite groups achieves this.

A device group or a composite group is a user-defined group of devices that SYMCLI commands can execute upon. Device groups are limited to a single Symmetrix system and RA group (a.k.a. SRDF group). A composite group, on the other hand, can span multiple Symmetrix systems and RA groups. The device or composite group type may contain R1 or R2 devices and may contain various device lists for standard, BCV, virtual, and remote BCV devices. The `symdg/symld` and `symcg` commands are used to create and manage device and composite groups.

### SRDF consistency groups

An SRDF consistency group is a collection of devices defined by a composite group that has been enabled for consistency protection. Its purpose is to protect data integrity for applications that span multiple
RA groups and/or multiple Symmetrix systems. The protected applications may comprise multiple heterogeneous data resource managers across multiple host operating systems.

An SRDF consistency group uses PowerPath or Enginuity Consistency Assist (SRDF-ECA) to provide synchronous disaster restart with zero data loss. Disaster restart solutions that use consistency groups provide remote restart with short recovery time objectives. Zero data loss implies that all completed transactions at the beginning of a disaster will be available at the target.

When the amount of data for an application becomes very large, the time and resources required for host-based software to protect, back up, or run decision-support queries on these databases become critical factors. The time required to quiesce or shut down the application for offline backup is no longer acceptable. SRDF consistency groups allow users to remotely mirror the largest data environments and automatically split off dependent-write consistent, restartable copies of applications in seconds without interruption to online service.

A consistency group is a composite group of SRDF devices (R1 or R2) that act in unison to maintain the integrity of applications distributed across multiple Symmetrix systems or multiple RA groups within a single Symmetrix. If a source (R1) device in the consistency group cannot propagate data to its corresponding target (R2) device, EMC software suspends data propagation from all R1 devices in the consistency group, halting all data flow to the R2 targets. This suspension, called tripping the consistency group, ensures that a dependent-write consistent R2 copy of the database up to the point in time that the consistency group tripped.

Tripping a consistency group can occur either automatically or manually. Scenarios in which an automatic trip would occur include:

- One or more R1 devices cannot propagate changes to their corresponding R2 devices
- The R2 device fails
- The SRDF directors on the R1 side or R2 side fail

In an automatic trip, the Symmetrix system completes the write to the R1 device, but indicates that the write did not propagate to the R2 device. EMC software intercepts the I/O and instructs the Symmetrix to suspend all R1 source devices in the consistency group from propagating any further writes to the R2 side. Once the suspension is
complete, writes to all of the R1 devices in the consistency group continue normally, but they are not propagated to the target side until normal SRDF mirroring resumes.

An explicit trip occurs when the command `symrdf -cg suspend` or `split` is invoked. Suspending or splitting the consistency group creates an on-demand, restartable copy of the database at the R2 target site. BCV devices that are synchronized with the R2 devices are then split after the consistency group is tripped, creating a second dependent-write consistent copy of the data. During the explicit trip, SYMCLI issues the command to create the dependent-write consistent copy, but may require assistance from PowerPath or SRDF-ECA if I/O is received on one or more R1 devices, or if the SYMCLI commands issued are abnormally terminated before the explicit trip.

An EMC consistency group maintains consistency within applications spread across multiple Symmetrix systems in an SRDF configuration, by monitoring data propagation from the source (R1) devices in a consistency group to their corresponding target (R2) devices as depicted in Figure 13 on page 77. Consistency groups provide data integrity protection during a rolling disaster.

![Figure 13 SRDF consistency group](image)

**Figure 13** SRDF consistency group
A consistency group protection is defined containing volumes X, Y, and Z on the source Symmetrix. This consistency group definition must contain all of the devices that need to maintain dependent-write consistency and reside on all participating hosts involved in issuing I/O to these devices. A mix of CKD (mainframe) and FBA (UNIX/Windows) devices can be logically grouped together. In some cases, the entire processing environment may be defined in a consistency group to ensure dependent-write consistency.

The rolling disaster described previously begins, preventing the replication of changes from volume Z to the remote site.

Since the predecessor log write to volume Z cannot be propagated to the remote Symmetrix system, a consistency group trip occurs.

A ConGroup trip holds the write that could not be replicated along with all of the writes to the logically grouped devices. The writes are held by PowerPath on UNIX/Windows hosts and by IOS on mainframe hosts (or by ECA-RDA for both UNIX/Windows and mainframe hosts) long enough to issue two I/Os to all of the Symmetrix systems involved in the consistency group. The first I/O changes the state of the devices to a suspend-pending state.

The second I/O performs the suspend actions on the R1/R2 relationships for the logically grouped devices which immediately disables all replication to the remote site. This allows other devices outside of the group to continue replicating, provided the communication links are available. After the relationship is suspended, the completion of the predecessor write is acknowledged back to the issuing host. Furthermore, all writes that were held during the consistency group trip operation are released.

After the second I/O per Symmetrix completes, the I/O is released, allowing the predecessor log write to complete to the host. The dependent data write is issued by the DBMS and arrives at X but is not replicated to the R2(X).

When a complete failure occurs from this rolling disaster, the dependent-write consistency at the remote site is preserved. If a complete disaster does not occur and the failed links are activated again, the consistency group replication can be resumed. EMC recommends creating a copy of the dependent-write consistent image while the resume takes place. Once the SRDF process reaches synchronization the dependent-write consistent copy is achieved at the remote site.
SRDF terminology

This section describes various terms related to SRDF operations.

Suspend and resume operations

Practical uses of suspend and resume operations usually involve unplanned situations in which an immediate suspension of I/O between the R1 and R2 devices over the SRDF links is desired. In this way, data propagation problems can be stopped. When suspend is used with consistency groups, immediate backups can be performed off the R2s without affecting I/O from the local host application. I/O can then be resumed between the R1 and R2 and return to normal operation.

Establish and split operations

The establish and split operations are normally used in planned situations in which use of the R2 copy of the data is desired without interfering with normal write operations to the R1 device. Splitting a point-in-time copy of data allows access to the data on the R2 device for various business continuity tasks. The ease of splitting SRDF pairs to provide exact database copies makes it convenient to perform scheduled backup operations, reporting operations, or new application testing from the target Symmetrix data while normal processing continues on the source Symmetrix system.

The R2 copy can also be used to test disaster recovery plans without manually intensive recovery drills, complex procedures, and application service interruptions. Upgrades to new versions can be tested or changes to actual code can be made without affecting the online production server. For example, modified server code can be run on the R2 copy of the database until the upgraded code runs with no errors before upgrading the production server.

In cases where an absolute real-time copy of the production data is not essential, users may choose to split the SRDF pair periodically and use the R2 copy for queries and report generation. The SRDF pair can be re-established periodically to provide incremental updating of data on the R2 device. The ability to refresh the R2 device periodically provides the latest information for data processing and reporting.

Failover and failback operations

Practical uses of failover and failback operations usually involve the need to switch business operations from the production site to a remote site (failover) or the opposite (failback). Once failover
occurs, normal operations continue using the remote (R2) copy of synchronized application data. Scheduled maintenance at the production site is one example of where failover to the R2 site might be needed.

Testing of disaster recovery plans is the primary reason to temporarily fail over to a remote site. Traditional disaster recovery routines involve customized software and complex procedures. Offsite media must be either electronically transmitted or physically shipped to the recovery site. Time-consuming restores and the application of logs usually follow. SRDF failover/failback operations significantly reduce the recovery time by incrementally updating only the specific tracks that have changed; this accomplishes in minutes what might take hours for a complete load from dumped database volumes.

**Update operation**

The update operation allows users to resynchronize the R1s after a failover while continuing to run application and database services on the R2s. This function helps reduce the amount of time that a failback to the R1 side takes. The update operation is a subset of the failover/failback functionality. Practical uses of the R1 update operation usually involve situations in which the R1 becomes almost synchronized with the R2 data before a failback, while the R2 side is still online to its host. The -until option, when used with update, specifies the target number of invalid tracks that are allowed to be out of sync before resynchronization to the R1 completes.

**Concurrent SRDF**

Concurrent SRDF means having two target R2 devices configured as concurrent mirrors of one source R1 device. Using a Concurrent SRDF pair allows the creation of two copies of the same data at two remote locations. When the two R2 devices are split from their source R1 device, each target site copy of the application can be accessed independently.

**R1/R2 swap**

Swapping R1/R2 devices of an SRDF pair causes the source R1 device to become a target R2 device and vice versa. Swapping SRDF devices allows the R2 site to take over operations while retaining a remote mirror on the original source site. Swapping is especially useful after failing over an application from the R1 site to the R2 site. SRDF swapping is available with Enginuity version 5567 or later.
Data Mobility

Data mobility is an SRDF configuration that restricts SRDF devices to operating only in adaptive copy mode. This is a lower-cost licensing option that is typically used for data migrations. It allows data to be transferred in adaptive copy mode from source to target, and is not designed as a solution for DR requirements unless used in combination with TimeFinder.

Dynamic SRDF

Dynamic SRDF allows the creation of SRDF pairs from non-SRDF devices while the Symmetrix system is in operation. Historically, source and target SRDF device pairing has been static and changes required assistance from EMC personnel. This feature provides greater flexibility in deciding where to copy protected data.

Dynamic RA groups can be created in a SRDF switched fabric environment. An RA group represents a logical connection between two Symmetrix systems. Historically, RA groups were limited to those static RA groups defined at configuration time. However, RA groups can now be created, modified, and deleted while the Symmetrix system is in operation. This provides greater flexibility in forming SRDF-pair-associated links.

SRDF control operations

This section describes typical control operations that can be performed by the Solutions Enabler symrdf command.

Solutions Enabler SYMCLI SRDF commands perform the following basic control operations on SRDF devices:

- **Establish** synchronizes an SRDF pair by initiating a data copy from the source (R1) side to the target (R2) side. This operation can be a full or incremental establish. Changes on the R2 volumes are discarded by this process.
- **Restore** resynchronizes a data copy from the target (R2) side to the source (R1) side. This operation can be a full or incremental restore. Changes on the R1 volumes are discarded by this process.
- **Split** stops mirroring for the SRDF pair(s) in a device group and write-enables the R2 devices.
- **Swap** exchanges the source (R1) and target (R2) designations on the source and target volumes.
Failover switches data processing from the source (R1) side to the target (R2) side. The source side volumes (R1), if still available, are write-disabled.

Failback switches data processing from the target (R2) side to the source (R1) side. The target side volumes (R2), if still available, are write-disabled.

Establishing an SRDF pair

Establishing an SRDF pair initiates remote mirroring—the copying of data from the source (R1) device to the target (R2) device. SRDF pairs come into existence in two different ways:

- At configuration time through the pairing of SRDF devices. This is a static pairing configuration discussed earlier.
- Anytime during a dynamic pairing configuration in which SRDF pairs are created on demand.

A full establish (symrdf establish –full) is typically performed after an SRDF pair is initially configured and connected via the SRDF links. After the first full establish, users can perform an incremental establish, where the R1 device copies to the R2 device only the new data that was updated while the relationship was split or suspended.

To initiate an establish operation on all SRDF pairs in a device or composite group, all pairs must be in the split or suspended state. The symrdf query command is used to check the state of SRDF pairs in a device or composite group.

When the establish operation is initiated, the system write-disables the R2 device to its host and merges the track tables. The merge creates a bitmap of the tracks that need to be copied to the target volumes discarding the changes on the target volumes. When the establish operation is complete and the SRDF pairs are in the synchronized state. The R1 device and R2 device contain identical data, and continue to do so until interrupted by administrative command or unplanned disruption. Figure 14 on page 83 depicts SRDF establish and restore operations.
The establish operation may be initiated by any host connected to either Symmetrix system, provided that an appropriate device group has been built on that host. The following command initiates an incremental establish operation for all SRDF pairs in the device group named MyDevGrp:

```
symrdf –g MyDevGrp establish –noprompt
```

### Splitting an SRDF pair

When read/write access to a target (R2) device is necessary, the SRDF pair can be split. When the split completes, the target host can access the R2 device for write operations. The R2 device contains valid data and is available for business continuity tasks or restoring data to the R1 device if there is a loss of data on that device.

While an SRDF pair is in the split state, local I/O to the R1 device can still occur. These updates are not propagated to the R2 device immediately. Changes on each Symmetrix system are tracked through bitmaps and are reconciled when normal SRDF mirroring operations are resumed. To initiate a split, an SRDF pair must already be in one of the following states:

- Synchronized
- Suspended
- R1 updated
- SyncInProg (if the –symforce option is specified for the split – resulting in a set of R2 devices that are not dependent-write consistent and are not usable)
The split operation may be initiated from either host. The following command initiates a split operation on all SRDF pairs in the device group named MyDevGrp:

```
symrdf -g MyDevGrp split -noprompt
```

The `symrdf split` command provides exactly the same functionality as the `symrdf suspend` and `symrdf rw_enable R2` commands together. Furthermore, the split and suspend operations have exactly the same consistency characteristics as SRDF consistency groups. Therefore, when SRDF pairs are in a single device group, users can split the SRDF pairs in the device group as shown previously and have restartable copies on the R2 devices. If the application data spans multiple Symmetrix systems or multiple RA groups, include SRDF pairs in a consistency group to achieve the same results.

**Restoring an SRDF pair**

When the target (R2) data must be copied back to the source (R1) device, the SRDF restore command is used (see Figure 14 on page 83). After an SRDF pair is split, the R2 device contains valid data and is available for business continuance tasks (such as running a new application) or restoring data to the R1 device. Moreover, if the results of running a new application on the R2 device need to be preserved, moving the changed data and new application to the R1 device is another option.

Users can perform a full or incremental restore. A full restore operation copies the entire contents of the R2 device to the R1 device. An incremental restore operation is much faster because it copies only new data that was updated on the R2 device while the SRDF pair was split. Any tracks on the R1 device that changed while the SRDF pair was split are replaced with corresponding tracks on the R2 device. To initiate a restore, an SRDF pair must already be in the split state. The restore operation can be initiated from either host. The following command initiates an incremental restore operation on all SRDF pairs in the device group named MyDevGrp (add the `-full` option for a full restore).

```
symrdf -g MyDevGrp restore -noprompt
symrdf -g MyDevGrp restore -noprompt -full
```
The restore operation is complete when the R1 and R2 devices contain identical data. The SRDF pair is then in a synchronized state and may be reestablished by initiating the following command:

```
symrdf -g MyDevGrp establish
```

## Failover and failback operations

Having a synchronized SRDF pair allows users to switch data processing operations from the source site to the target site if operations at the source site are disrupted or if downtime must be scheduled for maintenance. This switchover from source to target is enabled through the use of the failover command. When the situation at the source site is back to normal, a failback operation is used to reestablish I/O communications links between source and target, resynchronize the data between the sites, and resume normal operations on the R1 devices as shown in Figure 15 on page 85, which illustrates the failover and failback operations.

![SRDF failover and failback control operations](ICO-IMG-000004)

### Figure 15  SRDF failover and failback control operations

The failover and failback operations relocate the processing from the source site to the target site or vice versa. This may or may not imply movement of data.

#### Failover

Scheduled maintenance or storage system problems can disrupt access to production data at the source site. In this case, a failover operation can be initiated from either host to make the R2 device read/write-enabled to its host. Before issuing the failover, all applications services on the R1 volumes must be stopped. This is
because the failover operation makes the R1 volumes read-only. The following command initiates a failover on all SRDF pairs in the device group named MyDevGrp:

```
symrdf -g MyDevGrp failover -noprompt
```

To initiate a failover, the SRDF pair must already be in one of the following states:

- Synchronized
- Suspended
- R1 updated
- Partitioned (when invoking this operation at the target site)

The failover operation:

- Suspends data traffic on the SRDF links
- Write-disables the R1 devices
- Write-enables the R2 volumes

### Failback

To resume normal operations on the R1 side, a failback (R1 device takeover) operation is initiated. This means read/write operations on the R2 device must be stopped, and read/write operations on the R1 device must be started. When the failback command is initiated, the R2 becomes read-only to its host, while the R1 becomes read/write-enabled to its host. The following command performs a failback operation on all SRDF pairs in the device group named MyDevGrp:

```
symrdf -g MyDevGrp failback -noprompt
```

The SRDF pair must already be in one of the following states for the failback operation to succeed:

- Failed over
- Suspended and write-disabled at the source
- Suspended and not ready at the source
- R1 Updated
- R1 UpdInProg

The failback operation:

- Write-enables the R1 devices.
- Performs a track table merge to discard changes on the R1s.
- Transfers the changes on the R2s.
- Resumes traffic on the SRDF links.
- Write-disables the R2 volumes.
EMC SRDF/Cluster Enabler solutions

EMC SRDF/Cluster Enabler (SRDF/CE) for MSCS is an integrated solution that combines SRDF and clustering protection over distance. EMC SRDF/CE provides disaster-tolerant capabilities that enable a cluster to span geographically separated Symmetrix systems. It operates as a software extension (MMC snap-in) to the Microsoft Cluster Service (MSCS).

SRDF/CE achieves this capability by exploiting SRDF disaster restart capabilities. SRDF allows the MSCS cluster to have two identical sets of application data in two different locations. When cluster services are failed over or failed back, SRDF/CE is invoked automatically to perform the SRDF functions necessary to enable the requested operation.

Figure 16 on page 87 illustrates the hardware configuration of two, four-node, geographically distributed EMC SRDF/CE clusters using bidirectional SRDF.
The SYMCLI TimeFinder component extends the basic SYMCLI command set to include TimeFinder or business continuity commands that allow control operations on device pairs within a local replication environment. This section specifically describes the functionality of:

- TimeFinder/Mirror—General monitor and control operations for business continuance volumes (BCV)
- TimeFinder/CG—Consistency groups
- TimeFinder/Clone—Clone copy sessions
- TimeFinder/Snap—Snap copy sessions

Commands such as syminr and symbcv perform a wide spectrum of monitor and control operations on standard/BCV device pairs within a TimeFinder/Mirror environment. The TimeFinder/Clone command, symclone, creates a point-in-time copy of a source device on nonstandard device pairs (such as standard/standard, BCV/BCV). The TimeFinder/Snap command, symsnap, creates virtual device copy sessions between a source device and multiple virtual target devices. These virtual devices only store pointers to changed data blocks from the source device, rather than a full copy of the data. Each product requires a specific license for monitoring and control operations.

Configuring and controlling remote BCV pairs requires EMC SRDF business continuity software discussed previously. The combination of TimeFinder with SRDF provides for multiple local and remote copies of production data.

Figure 17 on page 89 illustrates application usage for a TimeFinder/Mirror configuration in a Symmetrix system.
**TimeFinder/Mirror establish operations**

A BCV device can be fully or incrementally established. After configuration and initialization of a Symmetrix system, BCV devices contain no data. BCV devices, like standard devices, can have unique host addresses and can be online and ready to the host(s) to which they are connected. A full establish operation must be used the first time the standard devices are paired with the BCV devices. An incremental establish of a BCV device can be performed to resynchronize any data that has changed on the standard since the last establish operation.

**Note:** When BCVs are established, they are inaccessible to any host.

Symmetrix systems allow up to four mirrors for each hypervolume. The mirror positions are commonly designated M1, M2, M3, and M4. An unprotected BCV can be the second, third, or fourth mirror position of the standard device. A host, however, logically views the Symmetrix M1/M2 mirrored devices as a single device.

To assign a BCV as a mirror of a standard Symmetrix device, the `symmir establish` command is used. One method of establishing a BCV pair is to allow the standard/BCV device-pairing algorithm to arbitrarily create BCV pairs from multiple devices within a device group:

```
symmir -g MyDevGrp establish -full -noprompt
```

With this method, TimeFinder/Mirror first checks for any attach assignments (specifying a preferred BCV match from among multiple BCVs in a device group). TimeFinder/Mirror then checks if there are
any pairing relationships among the devices. If either of these previous conditions exists, TimeFinder/Mirror uses these assignments.

---

**TimeFinder split operations**

Splitting a BCV pair is a TimeFinder/Mirror action that detaches the BCV from its standard device and makes the BCV ready for host access. When splitting a BCV, the system must perform housekeeping tasks that may require a few milliseconds on a busy Symmetrix system. These tasks involve a series of steps that result in separation of the BCV from its paired standard:

- I/O is suspended briefly to the standard device.
- Write pending tracks for the standard device that have not yet been written out to the BCV are duplicated in cache to be written to the BCV.
- The BCV is split from the standard device.
- The BCV device status is changed to ready.

**Regular split**

A regular split is the type of split that has existed for TimeFinder/Mirror since its inception. With a regular split (before Enginuity version 5568), I/O activity from the production hosts to a standard volume was not accepted until it was split from its BCV pair. Therefore, applications attempting to access the standard or the BCV would experience a short wait during a regular split. Once the split was complete, no further overhead was incurred.

Beginning with Enginuity version 5568, any split operation is an instant split. A regular split is still valid for earlier versions and for current applications that perform regular split operations. However, current applications that perform regular splits with Enginuity version 5568 actually perform an instant split.

By specifying the `-instant` option on the command line, an instant split with Enginuity versions 5x66 and 5x67 can be performed. Since version 5568, this option is no longer required because instant split mode has become the default behavior. It is beneficial to continue to supply the `-instant` flag with later Enginuity versions, otherwise the default is to wait for the background split to complete.
**Instant split**

An instant split shortens the wait period during a split by dividing the process into a foreground split and a background split. During an instant split, the system executes the foreground split almost instantaneously and returns a successful status to the host. This instantaneous execution allows minimal I/O disruptions to the production volumes. Furthermore, the BCVs are accessible to the hosts as soon as the foreground process is complete. The background split continues to split the BCV pair until it is complete. When the `-instant` option is included or defaulted, SYMCLI returns immediately after the foreground split, allowing other operations while the BCV pair is splitting in the background.

The following operation performs an instant split on all BCV pairs in `MyDevGrp`, and allows SYMCLI to return to the server process while the background split is in progress:

```
symmir -g MyDevGrp split -instant -noprompt
```

The following `symmir` query command example checks the progress of a split on the composite group named `MyConGrp`. The `-bg` option is provided to query the status of the background split:

```
symmir -cg MyConGrp query -bg
```

**TimeFinder restore operations**

A BCV device can be used to fully or incrementally restore data on the standard volume. Like the full establish operation, a full restore operation copies the entire contents of the BCV devices to the standard devices. The devices upon which the restore operates may be defined in a device group, composite group, or device file. For example:

```
symmir -g MyDevGrp -full restore -noprompt
symmir -cg MyConGrp -full restore -noprompt
symmir -f MyFile -full -sid 109 restore -noprompt
```

The incremental restore process accomplishes the same thing as the full restore process with a major time-saving exception. The BCV copies to the standard device only new data that was updated on the BCV device while the BCV pair was split. The data on the corresponding tracks of the BCV device also overwrites any changed tracks on the standard device. This maximizes the efficiency of the resynchronization process. This process is useful, for example, if,
after testing or validating an updated version of a database or a new application on the BCV device is completed, a user wants to migrate and utilize a copy of the tested data or application on the production standard device.

**Note:** An incremental restore of a BCV volume to a standard volume is only possible when the two volumes have an existing TimeFinder relationship.

### TimeFinder consistent split

TimeFinder consistent split allows you to split off a dependent-write consistent, restartable image of an application without interrupting online services. Consistent split helps to avoid inconsistencies and restart problems that can occur when splitting an application-related BCV without first quiescing or halting the application. Consistent split is implemented using Enginuity Consistency Assist (ECA) feature. This functionality requires a TimeFinder/CG license.

### Enginuity Consistency Assist

The Enginuity Consistency Assist (ECA) feature of the Symmetrix operating environment can be used to perform consistent split operations across multiple heterogeneous environments. This functionality requires a TimeFinder/CG license and uses the --consistent option of the symmir command.

Using ECA to consistently split BCV devices from the standards, a control host with no database or a database host with a dedicated channel to gatekeeper devices must be available. The dedicated channel cannot be used for servicing other devices or to freeze I/O. For example, to split a device group, execute:

```
symmir -g MyDevGrp split --consistent --noprompt
```

**Figure 18 on page 93** illustrates an ECA split across three database hosts that access devices on a Symmetrix system.
Device groups or composite groups must be created on the controlling host for the target application to be consistently split. Device groups can be created to include all of the required devices for maintaining business continuity. For example, if a device group is defined that includes all of the devices being accessed by Hosts A, B, and C (see Figure 18), then all of the BCV pairs related to those hosts can be consistently split with a single command.

However, if a device group is defined that includes only the devices accessed by Host A, then the BCV pairs related to Host A can be split without affecting the other hosts. The solid vertical line in Figure 18 represents the ECA holding of I/Os during an instant split process, creating a dependent-write consistent image in the BCVs.

Figure 19 on page 94 illustrates the use of local consistent split with a database management system (DBMS).
When a **split** command is issued with ECA from the production host, a consistent database image is created using the following sequence of events shown in **Figure 19 on page 94**:

1. The device group, device file, or composite group identifies the standard devices that hold the database.
2. SYMAPI communicates to Symmetrix Enginuity to validate that all identified BCV pairs can be split.
3. SYMAPI communicates to Symmetrix Enginuity to open the ECA window (the time within Symmetrix Enginuity where the writes are deferred), the instant **split** is issued, and the writes are released by closing the window.
4. ECA suspends writes to the standard devices that hold the database. The DBMS cannot write to the devices and subsequently waits for these devices to become available before resuming any further write activity. Read activity to the device is not affected unless attempting to read from a device with a write queued against it.
5. SYMAPI sends an instant **split** request to all BCV pairs in the specified device group and waits for the Symmetrix to acknowledge that the foreground split has occurred. SYMAPI then communicates with Symmetrix Enginuity to resume the write or close the ECA window.
6. The application resumes writing to the production devices.

The BCV devices now contain a restartable copy of the production data that is consistent up until the time of the instant **split**. The production application is unaware that the **split** or
suspend/resume operation occurred. When the application on the secondary host is started using the BCVs, there is no record of a successful shutdown. Therefore, the secondary application instance views the BCV copy as a crashed instance and proceeds to perform the normal crash recovery sequence to restart.

When performing a consistent split, it is a good practice to issue host-based commands that commit any data that has not been written to disk before the split to reduce the amount of time on restart. For example on UNIX systems, the sync command can be run. From a database perspective, a checkpoint or equivalent should be executed.

**TimeFinder/Mirror reverse split**

BCVs can be mirrored to guard against data loss through physical drive failures. A reverse split is applicable for a BCV that is configured to have two local mirrors. It is generally used to recover from an unsuccessful restore operation. When data is restored from the BCV to the standard device, any writes that occur while the standard is being restored alter the original copy of data on the BCVs primary mirror. If the original copy of BCV data is needed again at a later time, it can be restored to the BCVs primary mirror from the BCVs secondary mirror using a reverse split. For example, whenever logical corruption is reintroduced to a database during a recovery process (following a BCV restore), both the standard device and the primary BCV mirror are left with corrupted data. In this case, a reverse split can restore the original BCV data from a BCVs secondary mirror to its primary mirror.

This is particularly useful when performing a restore and immediately restarting processing on the standard devices when the process may have to be restarted many times.

**Note:** Reverse split is not available when protected restore is used to return the data from the BCVs to the standards.

**TimeFinder/Clone operations**

Symmetrix TimeFinder/Clone operations using SYMCLI can create up to 16 copies from a source device onto target devices. Unlike TimeFinder/Mirror, TimeFinder/Clone does not require the traditional standard-to-BCV device pairing. Instead, TimeFinder/Clone allows any combination of source and target
devices. For example, a BCV can be used as the source device, while another BCV can be used as the target device. Any combination of source and target devices can be used. Additionally, TimeFinder/Clone does not use the traditional mirror positions the way that TimeFinder/Mirror does. Because of this, TimeFinder/Clone is a useful option when more than three copies of a source device are desired.

Normally, one of the three copies is used to protect the data against hardware failure.

The source and target devices must be the same emulation type (FBA or CKD). The target device must be equal in size to the source device. Clone copies of striped or concatenated metavolumes can also be created providing the source and target metavolumes are identical in configuration. Once activated, the target device can be instantly accessed by a target’s host, even before the data is fully copied to the target device.

TimeFinder/Clone copies are appropriate in situations where multiple copies of production data is needed for testing, backups, or report generation. Clone copies can also be used to reduce disk contention and improve data access speed by assigning users to copies of data rather than accessing the one production copy. A single source device may maintain as many as 16 relationships that can be a combination of BCVs, clones and snaps.

Clone copy sessions

TimeFinder/Clone functionality is controlled via copy sessions, which pair the source and target devices. Sessions are maintained on the Symmetrix system and can be queried to verify the current state of the device pairs. A copy session must first be created to define and set up the TimeFinder/Clone devices. The session is then activated, enabling the target device to be accessed by its host. When the information is no longer needed, the session can be terminated. TimeFinder/Clone operations are controlled from the host by using the `symclone` command to create, activate, and terminate the copy sessions.

Figure 20 on page 97 illustrates a copy session where the controlling host creates a TimeFinder/Clone copy of standard device DEV001 on target device DEV005, using the `symclone` command.
The `symclone` command is used to enable cloning operations. The cloning operation happens in two phases: creation and activation. The creation phase builds bitmaps of the source and target that are later used during the activation or copy phase. The creation of a symclone pairing does not start copying of the source volume to the target volume, unless the `-precopy` keyword is used.

For example, to create clone sessions on all the standards and BCVs in the device group `MyDevGrp`, use the following command:

```
symclone -g MyDevGrp create -noprompt
```

The activation of a clone enables the copying of the data. The data may start copying immediately if the `-copy` keyword is used. If the `-copy` keyword is not used, tracks are only copied when they are accessed from the target volume or when they are changed on the source volume.

Activation of the clone session established in the previous `create` command can be accomplished using the following command:

```
symclone -g MyDevGrp activate -noprompt
```

**New Symmetrix VMAX TimeFinder/Clone features**

Solutions Enabler 7.1 and later introduce the ability to clone from Thick to Thin devices using TimeFinder/Clone. Thick to thin TimeFinder/Clone allows application data to be moved from standard Symmetrix volumes to Virtually Provisioned storage within the same array. For some workloads Virtually Provisioned volumes offer advantages with allocation utilization, ease of use and performance through automatic wide striping. Thick to Thin TimeFinder/Clone provides an easy way to move workloads that benefit from Virtual Provisioning into that storage paradigm.
Migration from Thin devices back to fully provisioned devices is also possible. The source and target of the migration may be of different protection types and disk technologies offering versatility with protection schemes and disk tier options. Thick to Thin TimeFinder Clone will not disrupt hosts or internal array replication sessions during the copy process.

**TimeFinder/Snap operations**

Symmetrix arrays provide another technique to create copies of application data. The functionality, called TimeFinder/Snap, allows users to make pointer-based, space-saving copies of data simultaneously on multiple target devices from a single source device. The data is available for access instantly. TimeFinder/Snap allows data to be copied from a single source device to as many as 128 target devices. A source device can be either a Symmetrix standard device or a BCV device controlled by TimeFinder/Mirror, with the exception being a BCV working in clone emulation mode. The target device is a Symmetrix virtual device (VDEV) that consumes negligible physical storage through the use of pointers to track changed data.

The VDEV is a host-addressable Symmetrix device with special attributes created when the Symmetrix system is configured. However, unlike a BCV which contains a full volume of data, a VDEV is a logical-image device that offers a space-saving way to create instant, point-in-time copies of volumes. Any updates to a source device after its activation with a virtual device, causes the pre-update image of the changed tracks to be copied to a save device. The virtual device’s indirect pointer is then updated to point to the original track data on the save device, preserving a point-in-time image of the volume. TimeFinder/Snap uses this copy-on-first-write technique to conserve disk space, since only changes to tracks on the source cause any incremental storage to be consumed.

The `symsnap create` and `symsnap activate` commands are used to create source/target Snap pair.
Table 4 summarizes some of the differences between devices used in TimeFinder/Snap operations.

<table>
<thead>
<tr>
<th>Device</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual device</td>
<td>A logical-image device that saves disk space through the use of pointers to track data that is immediately accessible after activation. Snapping data to a virtual device uses a copy-on-first-write technique.</td>
</tr>
<tr>
<td>Save device</td>
<td>A device that is not host-accessible but accessed only through the virtual devices that point to it. Save devices provide a pool of physical space to store snap copy data to which virtual devices point.</td>
</tr>
<tr>
<td>BCV</td>
<td>A full volume mirror that has valid data after fully synchronizing with its source device. It is accessible only when split from the source device that it is mirroring.</td>
</tr>
</tbody>
</table>

**Snap copy sessions**

TimeFinder/Snap functionality is managed via copy sessions, which pair the source and target devices. Sessions are maintained on the Symmetrix system and can be queried to verify the current state of the devices. A copy session must first be created—a process which defines the Snap devices in the operation. On subsequent activation, the target virtual devices become accessible to its host. Unless the data is changed by the host accessing the virtual device, the virtual device always presents a frozen point-in-time copy of the source device at the point of activation. When the information is no longer needed, the session can be terminated.

TimeFinder/Snap operations are controlled from the host by using the `symsnap` command to create, activate, terminate, and restore the TimeFinder/Snap copy sessions. The TimeFinder/Snap operations described in this section explain how to manage the devices participating in a copy session through SYMCLI.

Figure 21 on page 100 illustrates a virtual copy session where the controlling host creates a copy of standard device DEV001 on target device VDEV005.
The `symsnap` command is used to enable TimeFinder/Snap operations. The snap operation happens in two phases: creation and activation. The creation phase builds bitmaps of the source and target that are later used to manage the changes on the source and target. The creation of a snap pairing does not copy the data from the source volume to the target volume. To create snap sessions on all the standards and BCVs in the device group `MyDevGrp`, use the following command.

```
symsnap -g <MyDevGrp> create -noprompt
```

The activation of a snap enables the protection of the source data tracks. When protected tracks are changed on the source volume, they are first copied into the save pool and the VDEV pointers are updated to point to the changed tracks in the save pool. When tracks are changed on the VDEV, the data is written directly to the save pool and the VDEV pointers are updated in the same way.

Activation of the snap session created in the previous `create` command can be accomplished using the following command.

```
symsnap -g <MyDevGrp> activate -noprompt
```
EMC Storage Resource Management

The Storage Resource Management (SRM) component of EMC Solutions Enabler extends the basic SYMCLI command set to include SRM commands that allow users to discover and examine attributes of various objects on a host or in the EMC storage enterprise.

Note: The acronym for EMC Storage Resource Management (SRM) can be easily confused with the acronym for VMware Site Recovery Manager. To avoid any confusion, this document always refers to VMware Site Recovery Manager as VMware SRM.

SYMCLI commands support SRM in the following areas:

- Data objects and files
- Relational databases
- File systems
- Logical volumes and volume groups
- Performance statistics

SRM allows users to examine the mapping of storage devices and the characteristics of data files and objects. These commands allow the examination of relationships between extents and data files or data objects, and how they are mapped on storage devices. Frequently, SRM commands are used with TimeFinder and SRDF to create point-in-time copies for backup and restart.

Figure 22 on page 102 outlines the process of how SRM commands are used with TimeFinder in a database environment.
EMC Solutions Enabler with a valid license for TimeFinder and SRM is installed on the host. In addition, the host must also have PowerPath or use ECA, and must be utilized with a supported DBMS system. As discussed in the Section “TimeFinder split operations” on page 90, when splitting a BCV, the system must perform housekeeping tasks that may require a few seconds on a busy Symmetrix system. These tasks involve a series of steps (shown in Figure 22 on page 102) that result in the separation of the BCV from its paired standard:

1. Using the SRM base mapping commands, first query the Symmetrix system to display the logical-to-physical mapping information about any physical device, logical volume, file, directory, and/or file system.

2. Using the database mapping command, query the Symmetrix to display physical and logical database information.

3. Next, use the database mapping command to translate:
   - The devices of a specified database into a device group or a consistency group, or
   - The devices of a specified table space into a device group or a consistency group.

4. The BCV is split from the standard device.
Table 5 lists the SYMCLI commands used to examine the mapping of data objects.

### Table 5: Data object SRM commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>symrslv</td>
<td>pd</td>
<td>Displays logical to physical mapping information about any physical device.</td>
</tr>
<tr>
<td></td>
<td>lv</td>
<td>Displays logical to physical mapping information about a logical volume.</td>
</tr>
<tr>
<td></td>
<td>file</td>
<td>Displays logical to physical mapping information about a file.</td>
</tr>
<tr>
<td></td>
<td>dir</td>
<td>Displays logical to physical mapping information about a directory.</td>
</tr>
<tr>
<td></td>
<td>fs</td>
<td>Displays logical to physical mapping information about a file system.</td>
</tr>
</tbody>
</table>

SRM commands allow users to examine the host database mapping and the characteristics of a database. The commands provide listings and attributes that describe various databases, their structures, files, table spaces, and user schemas. Typically, the database commands work with Oracle, Informix, SQL Server, Sybase, Microsoft Exchange, SharePoint Portal Server, and DB2 LUW database applications.

Table 6 lists the SYMCLI commands used to examine the mapping of database objects.

### Table 6: Data object mapping commands (page 1 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>symrdb</td>
<td>list</td>
<td>Lists various physical and logical database objects: Current relational database instances available table spaces, tables, files, or schemas of a database Files, segments, or tables of a database table space or schema</td>
</tr>
<tr>
<td></td>
<td>show</td>
<td>Shows information about a database object: table space, tables, file, or schema of a database, File, segment, or a table of a specified table space or schema</td>
</tr>
<tr>
<td></td>
<td>rdb2dg</td>
<td>Translates the devices of a specified database into a device group.</td>
</tr>
<tr>
<td></td>
<td>rdb2cg</td>
<td>Translates the devices of a specified table space into a composite group or a consistency group.</td>
</tr>
</tbody>
</table>
The SYMCLI file system SRM command allows users to investigate the file systems that are in use on the operating system. The command provides listings and attributes that describe file systems, directories, and files, and their mapping to physical devices and extents.

Table 7 lists the SYMCLI command that can be used to examine the file system mapping.

### Table 7. File system SRM commands to examine file system mapping

<table>
<thead>
<tr>
<th>Command</th>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>symhostfs</td>
<td>list</td>
<td>Displays a list of file systems, files, or directories.</td>
</tr>
<tr>
<td></td>
<td>show</td>
<td>Displays more detail information about a file system or file system object.</td>
</tr>
</tbody>
</table>

SYMCLI logical volume SRM commands allow users to map logical volumes to display a detailed view of the underlying storage devices. Logical volume architecture defined by a Logical Volume Manager (LVM) is a means for advanced applications to improve performance by the strategic placement of data.

Table 8 lists the SYMCLI commands that can be used to examine the logical volume mapping.

### Table 8. File system SRM command to examine logical volume mapping (page 1 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>symvg</td>
<td>deport</td>
<td>Deports a specified volume group so it can be imported later.</td>
</tr>
<tr>
<td></td>
<td>import</td>
<td>Imports a specified volume group.</td>
</tr>
<tr>
<td></td>
<td>list</td>
<td>Displays a list of volume groups defined on the host system by the logical volume manager.</td>
</tr>
</tbody>
</table>
### Table 8  File system SRM command to examine logical volume mapping (page 2 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>symvg (cont.)</td>
<td>rescan</td>
<td>Rescans all the volume groups.</td>
</tr>
<tr>
<td></td>
<td>show</td>
<td>Displays more detail information about a volume group.</td>
</tr>
<tr>
<td></td>
<td>vg2cg</td>
<td>Translates volume groups to composite groups.</td>
</tr>
<tr>
<td></td>
<td>vg2dg</td>
<td>Translates volume groups to device groups.</td>
</tr>
<tr>
<td>symlv</td>
<td>list</td>
<td>Displays a list of logical volumes on a specified volume group.</td>
</tr>
<tr>
<td></td>
<td>show</td>
<td>Displays detail information (including extent data) about a logical volume.</td>
</tr>
</tbody>
</table>

SRM performance statistics commands allow users to retrieve statistics about a host’s CPU, disk, and memory.

**Table 9** lists the statistics commands.

### Table 9  SRM statistics command

<table>
<thead>
<tr>
<th>Command</th>
<th>Argument</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>symhost</td>
<td>show</td>
<td>Displays host configuration information.</td>
</tr>
<tr>
<td></td>
<td>stats</td>
<td>Displays performance statistics.</td>
</tr>
</tbody>
</table>
EMC PowerPath

EMC PowerPath is host-based software that works with networked storage systems to intelligently manage I/O paths. PowerPath manages multiple paths to a storage array. Supporting multiple paths enables recovery from path failure because PowerPath automatically detects path failures and redirects I/O to other available paths. PowerPath also uses sophisticated algorithms to provide dynamic load balancing for several kinds of path management policies that the user can set. With the help of PowerPath, systems administrators are able to ensure that applications on the host have highly available access to storage and perform optimally at all times.

A key feature of path management in PowerPath is dynamic, multipath load balancing. Without PowerPath, an administrator must statically load balance paths to logical devices to improve performance. For example, based on current usage, the administrator might configure three heavily used logical devices on one path, seven moderately used logical devices on a second path, and 20 lightly used logical devices on a third path. As I/O patterns change, these statically configured paths may become unbalanced, causing performance to suffer. The administrator must then reconfigure the paths, and continue to reconfigure them as I/O traffic between the host and the storage system shifts in response to usage changes.

Designed to use all paths concurrently, PowerPath distributes I/O requests to a logical device across all available paths, rather than requiring a single path to bear the entire I/O burden. PowerPath can distribute the I/O for all logical devices over all paths shared by those logical devices, so that all paths are equally burdened. PowerPath load balances I/O on a host-by-host basis, and maintains statistics on all I/O for all paths. For each I/O request, PowerPath intelligently chooses the least-burdened available path, depending on the load-balancing and failover policy in effect. In addition to improving I/O performance, dynamic load balancing reduces management time and downtime because administrators no longer need to manage paths across logical devices. With PowerPath, configurations of paths and policies for an individual device can be changed dynamically, taking effect immediately, without any disruption to the applications.
PowerPath provides the following features and benefits:

- **Multiple paths, for higher availability and performance**—PowerPath supports multiple paths between a logical device and a host bus adapter (HBA, a device through which a host can issue I/O requests). Having multiple paths enables the host to access a logical device even if a specific path is unavailable. Also, multiple paths can share the I/O workload to a given logical device.

- **Dynamic multipath load balancing**—Through continuous I/O balancing, PowerPath improves a host’s ability to manage heavy I/O loads. PowerPath dynamically tunes paths for performance as workloads change, eliminating the need for repeated static reconfigurations.

- **Proactive I/O path testing and automatic path recovery**—PowerPath periodically tests failed paths to determine if they are available. A path is restored automatically when available, and PowerPath resumes sending I/O to it. PowerPath also periodically tests available but unused paths, to ensure they are operational.

- **Automatic path failover**—PowerPath automatically redirects data from a failed I/O path to an alternate path. This eliminates application downtime; failovers are transparent and non-disruptive to applications.

- **Enhanced high availability cluster support**—PowerPath is particularly beneficial in cluster environments, as it can prevent interruptions to operations and costly downtime. PowerPath’s path failover capability avoids node failover, maintaining uninterrupted application support on the active node in the event of a path disconnect (as long as another path is available).

- **Consistent split**—PowerPath allows users to perform TimeFinder consistent splits by suspending device writes at the host level for a fraction of a second while the foreground split occurs. PowerPath software provides suspend-and-resume capability that avoids inconsistencies and restart problems that can occur if a database-related BCV is split without first quiescing the database.

- **Consistency Groups**—Consistency groups are a composite group of Symmetrix devices specially configured to act in unison to maintain the integrity of a database distributed across multiple SRDF arrays controlled by an open systems host computer.
EMC PowerPath/VE delivers PowerPath Multipathing features to optimize VMware vSphere virtual environments. With PowerPath/VE, you can standardize path management across heterogeneous physical and virtual environments. PowerPath/VE enables you to automate optimal server, storage, and path utilization in a dynamic virtual environment. With hyper-consolidation, a virtual environment may have hundreds or even thousands of independent virtual machines running, including virtual machines with varying levels of I/O intensity. I/O-intensive applications can disrupt I/O from other applications and before the availability of PowerPath/VE, load balancing on an ESX host system had to be manually configured to correct for this. Manual load-balancing operations to ensure that all virtual machines receive their individual required response times are time-consuming and logistically difficult to effectively achieve.

PowerPath/VE works with VMware ESX and ESXi as a multipathing plug-in (MPP) that provides enhanced path management capabilities to ESX and ESXi hosts. PowerPath/VE is supported with vSphere (ESX4) only. Previous versions of ESX do not have the PSA, which is required by PowerPath/VE.

PowerPath/VE installs as a kernel module on the vSphere host. PowerPath/VE will plug in to the vSphere I/O stack framework to bring the advanced multipathing capabilities of PowerPath - dynamic load balancing and automatic failover - to the VMware vSphere platform (Figure 23 on page 109).
At the heart of PowerPath/VE path management is server-resident software inserted between the SCSI device-driver layer and the rest of the operating system. This driver software creates a single "pseudo device" for a given array volume (LUN) regardless of how many physical paths on which it appears. The pseudo device, or logical volume, represents all physical paths to a given device. It is then used for creating virtual disks, and for raw device mapping (RDM), which is then used for application and database access.
PowerPath/VE’s value fundamentally comes from its architecture and position in the I/O stack. PowerPath/VE sits above the HBA, allowing heterogeneous support of operating systems and storage arrays. By integrating with the I/O drivers, all I/Os run through PowerPath and allow for it to be a single I/O control and management point. Since PowerPath/VE resides in the ESX kernel, it sits below the Guest OS level, application level, database level, and file system level. PowerPath/VE’s unique position in the I/O stack makes it an infrastructure manageability and control point - bringing more value going up the stack.

**PowerPath/VE features**

PowerPath/VE provides the following features:

* Dynamic load balancing - PowerPath is designed to use all paths at all times. PowerPath distributes I/O requests to a logical device across all available paths, rather than requiring a single path to bear the entire I/O burden.

* Auto-restore of paths - Periodic auto-restore reassigns logical devices when restoring paths from a failed state. Once restored, the paths automatically rebalance the I/O across all active channels.

* Device prioritization - Setting a high priority for a single or several devices improves their I/O performance at the expense of the remaining devices, while otherwise maintaining the best possible load balancing across all paths. This is especially useful when there are multiple virtual machines on a host with varying application performance and availability requirements.

* Automated performance optimization - PowerPath/VE automatically identifies the type of storage array and sets the highest performing optimization mode by default. For Symmetrix, the mode is SymmOpt (Symmetrix Optimized).

* Dynamic path failover and path recovery - If a path fails, PowerPath/VE redistributes I/O traffic from that path to functioning paths. PowerPath/VE stops sending I/O to the failed path and checks for an active alternate path. If an active path is available, PowerPath/VE redirects I/O along that path. PowerPath/VE can compensate for multiple faults in the I/O channel (for example, HBAs, fiber-optic cables, Fibre Channel switch, storage array port).
Monitor/report I/O statistics - While PowerPath/VE load balances I/O, it maintains statistics for all I/O for all paths. The administrator can view these statistics using rpowermt.

Automatic path testing - PowerPath/VE periodically tests both live and dead paths. By testing live paths that may be idle, a failed path may be identified before an application attempts to pass I/O down it. By marking the path as failed before the application becomes aware of it, timeout and retry delays are reduced. By testing paths identified as failed, PowerPath/VE will automatically restore them to service when they pass the test. The I/O load will be automatically balanced across all active available paths.

PowerPath/VE management

PowerPath/VE uses a command set, called rpowermt, to monitor, manage, and configure PowerPath/VE for vSphere. The syntax, arguments, and options are very similar to the traditional powermt commands used on all the other PowerPath Multipathing supported operating system platforms. There is one significant difference in that rpowermt is a remote management tool.

Not all vSphere installations have a service console interface. In order to manage an ESXi host, customers have the option to use vCenter Server or vCLI (also referred to as VMware Remote Tools) on a remote server. PowerPath/VE for vSphere uses the rpowermt command line utility for both ESX and ESXi. PowerPath/VE for vSphere cannot be managed on the ESX host itself. There is neither a local nor remote GUI for PowerPath on ESX.

Administrators must designate a Guest OS or a physical machine to manage one or multiple ESX hosts. rpowermt is supported on Windows 2003 (32-bit) and Red Hat 5 Update 2 (64-bit).

When the vSphere host server is connected to the Symmetrix system, the PowerPath/VE kernel module running on the vSphere host will associate all paths to each device presented from the array and associate a pseudo device name (as discussed earlier). An example of this is shown in Figure 24 on page 112, which shows the output of rpowermt display host=x.x.x.x dev=emcpower0. Note in the output that the device has four paths and displays the optimization mode (SymmOpt = Symmetrix optimization).
As more VMAX Engines or Symmetrix DMX directors become available, the connectivity can be scaled as needed. PowerPath/VE supports up to 32 paths to a device. These methodologies for connectivity ensure all front-end directors and processors are utilized, providing maximum potential performance and load balancing for vSphere hosts connected to the Symmetrix VMAX/DMX storage arrays in combination with PowerPath/VE.

### PowerPath/VE in vCenter Server

PowerPath/VE for vSphere is managed, monitored, and configured using rpowermt as discussed in the previous section. This CLI-based management is common across all PowerPath platforms and presently, there is very little integration at this time with VMware management tools. However, LUN ownership is presented in the GUI.
Figure 25 on page 113 shows a number of different devices owned by PowerPath. A set of claim rules are added to the vSphere PSA, which enables PowerPath/VE to manage supported storage arrays. As part of the initial installation process and claiming of devices by PowerPath/VE, the system must be rebooted. Nondisruptive installing is discussed in the following section.

**Nondisruptive installation of PowerPath/VE using VMotion**

Installing PowerPath/VE on a vSphere host requires a reboot. Just as with other PowerPath platforms, either the host must be rebooted or the I/O to applications running on the host must be stopped. In the case of vSphere, the migration capability built into the hypervisor allows members of the cluster to have PowerPath/VE installed without disrupting active virtual machines.

VMware VMotion technology leverages the complete virtualization of servers, storage, and networking to move an entire running virtual machine instantaneously from one server to another. VMware VMotion uses the VMware cluster file system to control access to a virtual machine’s storage. During a VMotion operation, the active memory and precise execution state of a virtual machine is rapidly transmitted over a high-speed network from one physical server to another and access to the virtual machines’ disk storage is instantly switched to the new physical host. It is therefore advised, in order to eliminate any downtime, to use VMotion to move all running virtual machines off the ESX host server before the installation of
PowerPath/VE. If the ESX host server is in a fully automated High Availability (HA) cluster, put the ESX host into maintenance mode, which will immediately begin migrating all of the virtual machines off the ESX host to other servers in the cluster.

As always, it is necessary to perform a number of different checks before evacuating virtual machines from an ESX host to make sure that the virtual machines can actually be migrated. These checks include making sure that:

- VMotion is properly configured and functioning.
- The datastores containing the virtual machines are shared over the cluster.
- No virtual machines are using physical media from their ESX host system (that is, CD-ROMs, USB drives)
- The remaining ESX hosts in the cluster will be able to handle the additional load of the temporarily migrated virtual machines.

Performing these checks will help to ensure the successful (and error-free) migration of the virtual machines. Additionally, this due diligence will greatly reduce the risk of degraded virtual machine performance resulting from overloaded ESX host systems. For more information on configuring and using VMotion, refer to VMware documentation.

This process should be repeated on all ESX hosts in the cluster until all PowerPath installations are complete.
EMC Replication Manager

EMC Replication Manager is an EMC software application that dramatically simplifies the management and use of disk-based replications to improve the availability of user’s mission-critical data and rapid recovery of that data in case of corruption.

Note: All functionality offered by EMC Replication Manager is not supported in a VMware Infrastructure environment. The EMC Replication Manager Support Matrix available on Powerlink® (EMC’s password-protected customer- and partner-only website) provides further details on supported configurations.

Replication Manager helps users manage replicas as if they were tape cartridges in a tape library unit. Replicas may be scheduled or created on demand, with predefined expiration periods and automatic mounting to alternate hosts for backups or scripted processing. Individual users with different levels of access ensure system and replica integrity. In addition to these features, Replication Manager is fully integrated with many critical applications such as DB2 LUW, Oracle, and Microsoft Exchange.

Replication Manager makes it easy to create point-in-time, disk-based replicas of applications, file systems, or logical volumes residing on existing storage arrays. It can create replicas of information stored in the following environments:

- Oracle databases
- DB2 LUW databases
- Microsoft SQL Server databases
- Microsoft Exchange databases
- UNIX file systems
- Windows file systems
- VMware file systems

The software utilizes Java-based client-server architecture. Replication Manager can:

- Create point-in-time replicas of production data in seconds.
- Facilitate quick, frequent, and non-destructive backups from replicas.
Mount replicas to alternate hosts to facilitate offline processing (for example, decision-support services, integrity checking, and offline reporting).

Restore deleted or damaged information quickly and easily from a disk replica.

Set the retention period for replicas so that storage is made available automatically.

Replication Manager has a generic storage technology interface that allows it to connect and invoke replication methodologies available on:

- EMC Symmetrix™ arrays
- EMC CLARiiON™ arrays
- EMC VNX™ arrays
- HP StorageWorks arrays

Replication Manager uses Symmetrix API (SYMAPI) Solutions Enabler software and interfaces to the storage array’s native software to manipulate the supported disk arrays. Replication Manager automatically controls the complexities associated with creating, mounting, restoring, and expiring replicas of data. Replication Manager performs all of these tasks and offers a logical view of the production data and corresponding replicas. Replicas are managed and controlled with the easy-to-use Replication Manager console.
EMC Open Replicator

EMC Open Replicator enables distribution and/or consolidation of remote point-in-time copies between EMC Symmetrix and qualified storage systems such as the EMC VNX storage arrays. By leveraging the high-end Symmetrix storage architecture, Open Replicator offers unmatched deployment flexibility and massive scalability.

Open Replicator can be used to provide solutions to business processes that require high-speed data mobility, remote vaulting and data migration. Specifically, Open Replicator enables customers to:

- Rapidly copy data between Symmetrix, VNX and third-party storage arrays.
- Perform online migrations from qualified storage to Symmetrix arrays with minimal disruption to host applications.
- Push a point-in-time copy of applications from Symmetrix arrays to a target volume on qualified storage arrays with incremental updates.
- Copy from source volumes on qualified remote arrays to Symmetrix volumes.

Open Replicator is tightly integrated with the EMC TimeFinder and SRDF family of products, providing enterprises with highly flexible and lower-cost options for remote protection and migration. Open Replicator is ideal for applications and environments where economics and infrastructure flexibility outweigh RPO and RTO requirements. Open Replicator enables businesses to:

- Provide a cost-effective and flexible solution to protect lower-tier applications.
- Reduce TCO by pushing or pulling data from Symmetrix DMX systems to other qualified storage arrays in conventional SAN/WAN environments.
- Create remote point-in-time copies of production applications for many ancillary business operations such as data vaulting.
- Obtain cost-effective application restore capabilities with minimal RPO/RTO impact.
- Comply with industry policies and government regulations.
Virtual Provisioning™ (commonly known as Thin Provisioning) was released with the 5773 Enginuity operating environment. Virtual Provisioning allows for storage to be allocated/accessed on-demand from a pool of storage servicing one or many applications. This type of approach has multiple benefits:

- Enables LUNs to be “grown” into over time with no impact to the host or application as space is added to the thin pool
- Only delivers space from the thin pool when it is written to, that is, on-demand. Overallocated application components only use space that is written to—not requested.
- Provides for thin-pool wide striping and for the most part relieves the storage administrator of the burden of physical device/LUN configuration

Virtual Provisioning introduces two new devices to the Symmetrix. The first device is a thin device and the second device is a data device. These are described in the following two sections.

### Thin device

A thin device is a “Host accessible device” that has no storage directly associated with it. Thin devices have pre-configured sizes and appear to the host to have that exact capacity. Storage is allocated in chunks when a block is written to for the first time. Zeroes are provided to the host for data that is read from chunks that have not yet been allocated.

### Data device

Data devices are specifically configured devices within the Symmetrix that are containers for the written-to blocks of thin devices. Any number of data devices may comprise a data device pool. Blocks are allocated to the thin devices from the pool on a round robin basis. This allocation block size is 768K.

Figure 26 on page 119 depicts the components of a Virtual Provisioning configuration:
Symmetrix VMAX specific features

Solutions Enabler 7.1 and later introduce two new features to Symmetrix Virtual Provisioning - thin pool write balancing and zero space reclamation. Thin pool write balancing provides the ability to automatically rebalance allocated extents on data devices over the entire pool when new data devices are added. Zero space reclamation allows users to reclaim space from tracks of data devices that are all zeros.

Thin pool write rebalance

Thin pool write rebalancing for Virtual Provisioning pools extends the functionality of the Virtual Provisioning feature by implementing a method to normalize the used capacity levels of data devices within a virtual data pool after new data drives are added or existing data drives are drained. This feature introduces a background optimization task to scan the used capacity levels of the data devices within a virtual pool and perform movements of multiple track groups from the most utilized pool data devices to the least used pool data devices.
The process can be scheduled to run only when changes to virtual pool composition make it necessary and user controls exist to specify what utilization delta will trigger track group movement.

Zero space reclamation

Zero space reclamation or Virtual Provisioning space reclamation provides the ability to free, also referred to as "de-allocate," storage extents found to contain all zeros. This feature is an extension of the existing Virtual Provisioning space de-allocation mechanism. Previous versions of Enginuity and Solutions Enabler allowed for reclaiming allocated (reserved but unused) thin device space from a thin pool. Administrators now have the ability to reclaim both allocated/unwritten extents as well as extents filled with host-written zeros within a thin pool. The space reclamation process is nondisruptive and can be executed with the targeted thin device ready and read/write to operating systems and applications.

Starting the space reclamation process spawns a back-end disk director (DA) task that will examine the allocated thin device extents on specified thin devices. A thin device extent is 768 KB (or 12 tracks) in size and is the default unit of storage at which allocations occur. For each allocated extent, all 12 tracks will be brought into Symmetrix cache and examined to see if they contain all zero data. If the entire extent contains all zero data, the extent will be de-allocated and added back into the pool, making it available for a new extent allocation operation. An extent that contains any non-zero data is not reclaimed.
EMC Virtual LUN migration

This feature offers system administrators the ability to transparently migrate host visible LUNs from differing tiers of storage that are available in the Symmetrix VMAX. The storage tiers can represent differing hardware capability as well as differing tiers of protection. The LUNs can be migrated to either unallocated space (also referred to as unconfigured space) or to configured space, which is defined as existing Symmetrix LUNs that are not currently assigned to a server-existing, not-ready volumes-within the same subsystem. The data on the original source LUNs is cleared using instant VTOC once the migration has been deemed successful. The migration does not require swap or DVR space, and is nondisruptive to the attached hosts or other internal Symmetrix applications such as TimeFinder and SRDF. Figure 27 on page 121 shows the valid combinations of drive types and protection types that are available for migration.

<table>
<thead>
<tr>
<th>Drive Type</th>
<th>Flash</th>
<th>Fibre Channel</th>
<th>SATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>Fibre Channel</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>SATA</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protection Type</th>
<th>RAID 1</th>
<th>RAID 6</th>
<th>RAID 6</th>
<th>Un-Protected</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID 1</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>RAID 6</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>Un-Protected</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>x</td>
</tr>
</tbody>
</table>

Figure 27 Virtual LUN Eligibility Tables

The device migration is completely transparent to the host on which an application is running since the operation is executed against the Symmetrix device; thus the target and LUN number are not changed and applications are uninterrupted. Furthermore, in SRDF environments, the migration does not require customers to re-establish their disaster recovery protection after the migration.

The Virtual LUN feature leverages the newly designed virtual RAID architecture in Enginuity 5874 and later, which abstracts device protection from its logical representation to a server. This powerful approach allows a device to have more simultaneous protection types such as BCVs, SRDF, Concurrent SRDF, and spares. It also enables
seamless transition from one protection type to another while servers and their associated applications and Symmetrix software are accessing the device.

The Virtual LUN feature offers customers the ability to effectively utilize SATA storage - a much cheaper, yet reliable, form of high capacity storage. It also facilitates fluid movement of data across the various storage tiers present within the subsystem - the realization of true “tiered storage in the box.” Thus, Symmetrix VMAX becomes the first enterprise storage subsystem to offer a comprehensive “tiered storage in the box,” ILM capability that complements the customer’s tiering initiatives. Customers can now achieve varied cost/performance profiles by moving lower priority application data to less expensive storage, or conversely, moving higher priority or critical application data to higher performing storage as their needs dictate.

Specific use cases for customer applications enable the moving of data volumes transparently from tier to tier based on changing performance (moving to faster or slower disks) or availability requirements (changing RAID protection on the array). This migration can be performed transparently without interrupting those applications or host systems utilizing the array volumes and with only a minimal impact to performance during the migration.

The following sample commands show how to move two LUNs of a host environment from RAID 6 drives on Fibre Channel 15k rpm drives to Enterprise Flash drives. The new symmigrate command, which comes in EMC Solutions Enabler 7.0, is used to perform the migrate operation. The source Symmetrix hypervolume numbers are 200 and 201, and the target Symmetrix hypervolumes on the Enterprise Flash drives are A00 and A01.

1. A file (migrate.ctl) is created that contains the two LUNs to be migrated. The file has the following content:
   
   200 A00
   201 A01

2. The following command is executed to perform the migration:

   symmigrate -sid 1261 -name <ds_mig> -f <migrate.ctl> 
establish

   The ds_mig name associated with this migration can be used to interrogate the progress of the migration.
3. To inquire on the progress use the following command:

```bash
symmigrate -sid 1261 -name <ds_mig> query
```

The two host accessible LUNs are migrated without having to impact application or server availability.

**Virtual LUN VP mobility**

Virtual LUN technology offers manual control of data mobility between storage tiers within a VMAX array. Virtual LUN data movement can nondisruptively change the drive type (capacity, rotational speed) and the protection method (RAID scheme) of Symmetrix logical volumes. Enginuity 5875 brings a further enhancement to this feature by allowing thin device movement from one thin pool to another thin pool.

Open systems, mainframe, and IBM i (formerly iSeries) volumes, as well as meta volumes, can be migrated with Virtual LUN technology. Open system thin devices and thin meta volumes can be migrated to alternative thin pools. Migrations between thin pools can include changes to RAID scheme and drive type. This feature provides tiered storage capabilities within a single Symmetrix array. Virtual LUN data movement is transparent to hosts and applications and does not affect local and remote disaster recovery protection during and after the data movement.

Virtual LUN VP migrations are session based - each session may contain multiple devices to be migrated at the same time. There may also be multiple concurrent migration sessions. At the time of submission a migration session name is specified. This session name is subsequently used for monitoring and managing the migration.

While an entire thin device will be specified to be migration, only thin device extents that are allocated will be relocated. Thin device extents that have been allocated, but not written to (for example, pre-allocated tracks), will be relocated but will not cause any actual data to be copied.

New extent allocations that occur as a result of a host write to the thin device during the migration will be satisfied from the migration target pool.
EMC Symmetrix Fully Automated Storage Tiering (FAST)

With the release of EMC Enginuity 5874, EMC now offers the first generation of Fully Automated Storage Tiering technology. EMC Symmetrix VMAX Fully Automated Storage Tiering (FAST) for standard provisioned environments automates the identification of data volumes for the purposes of allocating or re-allocating application data across different performance tiers within an array. FAST proactively monitors workloads at the volume (LUN) level in order to identify "busy" volumes that would benefit from being moved to higher-performing drives. FAST will also identify less "busy" volumes that could be relocated to higher-capacity drives, without existing performance being affected. This promotion/demotion activity is based on policies that associate a storage group to multiple drive technologies, or RAID protection schemes, based on the performance requirements of the application contained within the storage group. Data movement executed during this activity is performed nondisruptively, without affecting business continuity and data availability.

The primary benefits of FAST include:

- Automating the process of identifying volumes that can benefit from Enterprise Flash Drives and/or that can be kept on higher-capacity, less-expensive drives without impacting performance
- Improving application performance at the same cost, or providing the same application performance at lower cost. Cost is defined as space, energy, acquisition, management and operational expense.
- Optimizing and prioritizing business applications, which allows customers to dynamically allocate resources within a single array
- Delivering greater flexibility in meeting different price/performance ratios throughout the lifecycle of the information stored

Management and operation of FAST are provided by SMC, as well as the Solutions Enabler Command Line Interface (SYMCLI). Also, detailed performance trending, forecasting, alerts, and resource utilization are provided through Symmetrix Performance Analyzer.
(SPA). EMC Ionix™ ControlCenter® provides the capability for advanced reporting and analysis to be used for charge back and capacity planning.
EMC Symmetrix Fully Automated Storage Tiering for Virtual Pools (FAST VP)

FAST VP builds incrementally upon the functionality of FAST v1, adding support for thin devices and sub-LUN data movement. FAST v1 moved application data only at the volume (LUN) level. Entire devices were promoted or demoted between tiers based on overall device performance. FAST VP adds finer granularities of performance measurement and data movement. The data from a single thin device under FAST control can be spread across multiple tiers. FAST is free to relocate individual chunks of a thin device, based on performance data gathered at the sub-extent level.

FAST VP is released with Enginuity 5875 and requires Solutions Enabler 7.2.

FAST VP will maximize the benefits of in-the-box tiered storage by optimizing cost vs. performance requirements by placing the right thin data extents, on the right tier, at the right time. The FAST VP system allows a storage administrator to decide how much SATA/FC/Flash capacity is given to a particular application and then automatically place the appropriate busiest thin data extents on the desired performance tier and the least busy thin data extents on a capacity tier. The administrator’s input criteria are assembled into FAST policies. The FAST VP system uses policy information to perform extent data movement operations within two or three disk tiers in the VMAX array. Because the unit of analysis and movement is measured in thin extents this sub-LUN optimization is extremely powerful and efficient. FAST VP made available in Enginuity 5875 is an evolution of the existing FAST and EMC Optimizer technology.

There are two components of FAST VP - the FAST controller and the Enginuity 5875 microcode. The microcode is responsible for collection of performance statistics, at both the LUN and sub-LUN level. The FAST controller is responsible for analyzing performance data collected by the microcode.

The resulting data analysis generates a data movement policy that contains promotion and demotion thresholds for each tier included in a FAST policy.

The microcode applies this movement policy to all thin devices under FAST VP control to determine the appropriate tier for the data and will generate and execute movement requests to relocate thin extents to the appropriate tier.
Management and operation of FAST VP are provided by SMC, as well as the Solutions Enabler Command Line Interface (SYMCLI). Also, detailed performance trending, forecasting, alerts, and resource utilization are provided through Symmetrix Performance Analyzer (SPA). Ionix ControlCenter provides the capability for advanced reporting and analysis to be used for charge back and capacity planning.
Creating Exchange Server 2010 Clones

This chapter includes the following topics:

- Introduction ................................................................. 130
- Overview ........................................................................... 131
- Recoverable versus restartable copies of stores.............. 132
- Copying mailbox and logs with Exchange Server shutdown... 134
- Copying the store using EMC Consistency technology .... 141
- Copying the store using the Exchange Server VSS Interface..... 148
- Copying the store using Replication Manager ..................... 155
- Transitioning disk copies to Exchange Server 2010 clones .... 157
- Reinitializing the cloned environment............................... 162
- Choosing an Exchange cloning methodology .................. 163
Creating Exchange Server 2010 Clones

Introduction

This chapter describes the Exchange Server 2010 cloning process using various EMC products. Determining which replication products to use depends on the customer's requirements and environment. Products such as the TimeFinder Integration Module for Exchange Server, the TimeFinder CLI tools themselves, and Replication Manager provide an easy method of copying Exchange Server 2010 mailbox stores in a single Symmetrix array.

In an Exchange environment prior to Exchange Server 2010, the smallest operational unit that could be replicated in a consistent manner was that of an Exchange storage group. In Exchange Server 2010 the smallest unit is now the Exchange store. As discussed in Chapter 1 an Exchange store is comprised of the combination of both a mailbox store and a transaction log area. It is the combination of both the mailbox store and the transaction log that is typically required for restart or recovery processing. Therefore, in this chapter, the cloning mechanisms will discuss the functionality as it may be applied to one, or many, stores.

In Exchange Server 2010 DAG deployments, the ability to take a copy of either the active or any passive copy of the data is available in both Replication Manager and TimeFinder Integration Module for Exchange Server.

All storage-based technologies used to create cloned Exchange stores depend on some form of restart or recovery processing. The mechanisms used by EMC clone processing are not unique and utilize tried and tested methodologies.

A store cloning process typically includes some or all of the following steps, depending on the copying mechanism selected and the desired usage of the store clone:

- Preparing the array for replication.
- Conditioning the source environment.
- Making a copy of the store and log volumes.
- Resetting the source database.
- Presenting the target mailbox store and log copy to another server and use the /RecoverServer switch.
- Conditioning the target store copy.
Overview

There are many choices when cloning data with EMC array-based replication software. Each software product has differing characteristics that affect the final deployment. A thorough understanding of the options available leads to an optimal replication choice.

A Exchange Server 2010 store can be in one of three data states when it is being copied:

- Shutdown
- Processing normally
- Conditioned using Exchange Server VSS

Depending on the data state of the store at the time it is copied, the store copy may be restartable or recoverable. This chapter begins with a discussion of recoverable and restartable Exchange store clones. It then describes various approaches to data replication using EMC software products and how the replication techniques can be used in combination with the different store data states to facilitate the cloning process. Following that, store clone usage considerations are discussed along with descriptions of the procedures used to deploy store clones.
Creating Exchange Server 2010 Clones

Recoverable versus restartable copies of stores

The Symmetrix-based replication technologies described in this section can create two types of store copies, recoverable or restartable. A significant amount of confusion exists between these two types of database copies; a clear understanding of their differences is critical to ensure the appropriate application of each method when a cloned Exchange Server environment is required.

Recoverable copies

A recoverable store copy is one in which logs can be applied to the store data state and the store is rolled forward to a point in time after the store copy was created. A recoverable Exchange Server store copy is intuitively easy for administrators to understand since maintaining recoverable copies in the form of backups is an important administrative function. In the event of a failure of the production store, the ability to recover the mailbox stores not only to the point in time when the last backup was taken, but also to roll forward subsequent transaction log files up to the point of failure, is a key feature of the Exchange Server environment.

In general, recoverable copies of Exchange Server store must utilize the VSS subsystem.

Restartable copies

If a disk mirror copy of a running Exchange Server store is created using EMC consistency technology without utilizing a utility that implements VSS backup functionality, the copy is to be considered a restartable copy. This means that when the store is attached as a restartable copy, Exchange Server will perform crash recovery. First, all transactions that were recorded as committed and written to the transaction log files, but which may not have had corresponding data pages written to the data files, are rolled forward. Second, when the redo phase is complete, Exchange Server enters the undo phase where it looks for database changes that were recorded (for instance, dirty page flushed by a lazy write), but which were never actually committed by a transaction. These changes are rolled back or undone. The state attained is often referred to as a transactionally consistent point in time. It is essentially the same process that the Exchange Server would undergo should the server have suffered an unanticipated interruption such as a power failure.
**Note:** Log replay recovery using subsequent transaction logs to a point in time after the store copy was created is supported on a Exchange Server 2010 restartable store copy by EMC. While not explicitly supported by Microsoft, it is possible to create copies which are referred to as “Hot Splits” of Exchange store and then manipulate them for restart and roll forward. This process is documented in the EMC TimeFinder Exchange Integration Module.
Creating Exchange Server 2010 Clones

Copying mailbox and logs with Exchange Server shutdown

It is possible to take a copy of an Exchange Server store while the Exchange Server instance is offline, or the store is unmounted. Taking a copy after the store has been shut down normally will ensure a clean copy for streaming to tape or for fast startup of the cloned store. Copies of running stores are in unknown transactional data states.

Note that a cold copy of an Exchange Server store does not constitute a valid Exchange Server backup, as no record of a backup event is ever made by the Exchange Server instance. Subsequently, there is no truncation of the Exchange transaction logs, as is typical of a valid backup process. Nor is there any explicit validation of the consistency of the mailbox and transaction log files, as is typical of any VSS backup scenario; however, vendors such as EMC may include this operation as is the case with the EMC TimeFinder Exchange Integration Module (TF/EIM) or Replication Manager (RM).

The primary method of creating cold copies of Exchange Server stores is through the use of EMC’s local replication product, TimeFinder. TimeFinder is used by Replication Manager to make store copies. Replication Manager facilitates the automation and management of store clones. Additionally, TF/EIM or RM facilitates the creation of copies based on TimeFinder functionality.

TimeFinder operations are implemented in three different forms: TimeFinder/Mirror, TimeFinder/Clone, and TimeFinder/Snap. These were discussed in general terms in Chapter 2. Here, they are utilized in an Exchange store context.

Using TimeFinder/Mirror

TimeFinder/Mirror is an EMC Symmetrix array implementation that allows an additional hardware mirror to be attached to a source volume. The additional mirror is a specially designated volume in the Symmetrix configuration called a business continuance volume (BCV). The BCV is synchronized to the source volume through a process referred to as an establish. While the BCV is established, it is “not ready” to all hosts to which it may be presented. At an appropriate time, the BCV can be split from the source volume to create a complete point-in-time copy of the source data that can be used for different purposes, including backup, decision support, and regression testing.
Groups of BCVs are managed together using SYMCLI device or composite groups. Solutions Enabler commands are executed to create SYMCLI groups for TimeFinder/Mirror operations. If the store spans devices in more than one Symmetrix, a composite group is used. Figure 28 on page 135 depicts the necessary steps to make a database copy of a cold Exchange Server 2010 store using TimeFinder/Mirror.

1. Establish BCVs to Standard devices
2. Dismount Exchange Server databases
3. Split BCVs from Standard devices
4. Mount Exchange Server databases

Figure 28  Copying a shutdown Exchange store with TimeFinder/Mirror

1. Establish the BCVs to the standard devices. This operation occurs in the background and should be executed before the synchronization of the BCV copy:

```
symmir -g <device_group> establish -full -noprompt
```

Note that the first iteration of the establish needs to be a full synchronization. Subsequent iterations by default are incremental if the `-full` keyword is omitted. Once the command is issued, the array begins the synchronization process using only Symmetrix resources. Since this operation occurs independently from the host, the process must be interrogated to see when it completes. The command to interrogate the synchronization process is:
Creating Exchange Server 2010 Clones

symmir -g <device_group> verify

This command will return a 0 return code when the synchronization operation is complete. Alternatively, synchronization can be verified using the following:

symmir -g <device_group> query

After the volumes are synchronized, the split command can be issued at any time.

2. Once BCV synchronization is complete, the store needs to be brought down in order to make a copy of a cold database. Exchange Server management tools may be used to take the store offline by dismounting the stores or by shutting down the Exchange Server instance.

3. When the store is deactivated, split the BCV mirrors using the following command:

symmir -g <device_group> split -noprompt

The split command takes a few seconds to process. The database copy on the BCVs is now ready for further processing.

4. The source store or Exchange databases can now be mounted and made available to users once again.

Using TimeFinder/Clone

TimeFinder/Clone is an EMC software product that copies data internally in the Symmetrix array. A TimeFinder/Clone session is created between a source data volume and a target volume. The target volume needs to be equal to or greater in size than the source volume. The source and target for TimeFinder/Clone sessions can be any hypervolumes in the Symmetrix configuration.

TimeFinder/Clone devices are managed together using SYMCLI device or composite groups. Solutions Enabler commands are executed to create SYMCLI groups for TimeFinder/Clone operations. If the store spans more than one Symmetrix, a composite group is used. Examples of these commands can be found in Figure 29 on page 137, which depicts the necessary steps to make a copy of a cold Exchange Server store onto BCV devices using TimeFinder/Clone.
1. Create TimeFinder Clone
2. Dismount Exchange Server Databases
3. Activate TimeFinder Clone
4. Mount Exchange Server Databases

Figure 29 Copying a shutdown Exchange store with TimeFinder/Clone

1. The first action is to create the TimeFinder/Clone pairs. The following command creates the TimeFinder/Clone pairings and protection bitmaps. No data is copied or moved at this time:

   symclone -g <device_group> create -noprompt

   Unlike TimeFinder/Mirror, the TimeFinder/Clone relationship is created and activated when it is needed. No prior synchronization of data is necessary. After the TimeFinder/Clone session is created, it can be activated consistently.

2. Once the create command has completed, the database needs to be shut down to make a cold disk copy of the database. Exchange Server management tools may be used to take the store offline or shut down the instance.

3. With the store down, the TimeFinder/Clone can now be activated:

   symclone -g <device_group> activate -noprompt
After the activate, the database copy provided by TimeFinder/Clone is immediately available for further processing, even though the copying of data may not have completed.

4. The source store and databases can now be mounted and made available to users once again.

As of Enginuity 5772 and later, TimeFinder/Clone devices are no longer subject to a direct Copy on First Write (COFW) penalty. In these later revisions of Enginuity, an Asynchronous Copy on First Write (ACOFW) is implemented, such that update operations from the production host occur immediately, and Enginuity manages the transfer of the original data to the Clone device in the background. This represents a significant increase in performance to the production server as compared to earlier implementations of TimeFinder/Clone. Clone devices at Enginuity 5772 and later, however, suffer the Copy on Access (COA) penalty for access from the host attached to the Clone devices. COA means that if a track on a TimeFinder/Clone volume is accessed before it has been copied, it must first be copied from the source volume to the target volume.

In earlier versions of Enginuity, stores copied using TimeFinder/Clone are subject to COFW and COA penalties. The COFW penalty means that if a track is written to the source volume and it has not yet been copied to the target volume, it must first be copied to the target volume before the write from the host is acknowledged. COA refers to the copy process that needs to be executed when data on the Clone device is referenced by an alternate host, where the data has not been previously copied to the Clone device itself.

---

**Note:** Clone devices which are fully copied prior to being used will not incur any Copy on Write or Copy on Access activities as they are fully independent devices once activated.

---

**Using TimeFinder/Snap**

TimeFinder/Snap enables users to create complete copies of their data while consuming only a fraction of the disk space required by the original copy.
TimeFinder/Snap is an EMC software product that maintains space-saving pointer-based copies of disk volumes using VDEVs and SAVDEVs. The VDEVs contain pointers either to the source data (when it is unchanged) or to the SAVDEVs (when the data has been changed).

TimeFinder/Snap devices are managed together using SYMCLI device or composite groups. Solutions Enabler commands are executed to create SYMCLI groups for TimeFinder/Snap operations. If the store spans more than one Symmetrix, a composite group is used. Examples of these commands can be found in Appendix B, “References.”

Figure 30 on page 139 depicts the necessary steps to make a copy of a cold Exchange Server store using TimeFinder/Snap:

1. **Create Snap Session**
2. **Dismount Exchange Databases**
3. **Activate Snap Session**
4. **Mount Exchange Databases**

**Figure 30  Copying a shutdown Exchange store with TimeFinder/Snap**

1. The first action is to create the TimeFinder/Snap pairs. The following command creates the TimeFinder/Snap pairings and protection bitmaps. No data is copied or moved at this time:

   `symsnap -g <device_group> create -noprompt`
2. Once the create operation has completed, the store needs to be shut down to make a cold TimeFinder/Snap of the Exchange environment. Exchange Server management tools may be used to take the store offline or shut down the instance, whichever is most applicable.

3. With the store dismounted, the TimeFinder/Snap copy can now be activated:

   `symsnap -g <device_group> activate -noprompt`

   After the activate, the pointer-based store copy on the VDEVs is available for further processing.

4. The source store and database can now be mounted and made available to users.

   As of Enginuity 5772 and later, TimeFinder/Snap devices are no longer subject to a direct COFW penalty. In these later revisions of Enginuity an Asynchronous Copy on First Write (ACOFW) operation is implemented, such that update operations from the production host occur immediately, and Enginuity manages the transfer of the original data to the Snap device in the background. This represents a significant increase in performance to the production server as compared to earlier implementations of TimeFinder/Snap. Snap devices at Enginuity 5772 and later, however, suffer the COA penalty for access from the alternate host attached to the Snap devices. stores copied using TimeFinder/Snap in earlier version of Enginuity are subject to a COFW penalty while the snap is activated. The COFW penalty means that if a track is written to the source volume and it has not yet been copied to the snap save area, it must first be copied to the save area before the write from the host is acknowledged. COA refers to the copy process that needs to be executed when data on the Snap device is referenced by an alternate host, where the data has not been previously copied to the save area.
Copying the store using EMC Consistency technology

The replication of a running Exchange store requires a copying technique that ensures data consistency between the database and transaction log files while the store is fully online and servicing users. The VMAX store copying technique uses EMC Consistency technology in the form of a consistency group (CG) combined with an appropriate data copy process like TimeFinder/Mirror, TimeFinder/Clone, or TimeFinder/Snap. TimeFinder/CG allows for the running store copy to be created in an instant through use of the –consistent keyword on the split or activate commands. The image created in this way is in a dependent-write consistent data state and can be utilized as a restartable copy of the store.

Exchange Server environments enforce a principle of dependent-write I/O. That is, no dependent-write will be issued until the predecessor write that it is dependent on has completed. This type of programming discipline is used to coordinate mailbox store and log updates within a store and allows Exchange Server environments to be restartable in event of a power failure. Dependent-write consistent data states are created when database management systems are exposed to power failures. Using EMC Consistency technology options during the store cloning process also creates a store copy that has a dependent-write consistent data state. See for more discussion on EMC Consistency technology.

Exchange Server 2010 stores can be copied while they are running and processing transactions. The following sections describe how to copy a running Exchange Server 2010 store using TimeFinder technology.

Using TimeFinder/Mirror

TimeFinder/Mirror is an EMC software product that allows an additional hardware mirror to be attached to a source volume. The additional mirror is a specially designated volume in the Symmetrix configuration called a business continuance volume, or BCV. The BCV is synchronized to the source volume through a process called an establish. While the BCV is established, it is not ready to all hosts. At an appropriate time, the BCV can be split from the source volume to create a complete point-in-time copy of the source data that can be used for different purposes, including backup, decision support, and regression testing.
Creating Exchange Server 2010 Clones

Groups of BCVs are managed together using SYMCLI device or composite groups. Solutions Enabler commands are executed to create SYMCLI groups for TimeFinder/Mirror operations. If the store spans more than one Symmetrix, a composite group is used. Examples of these commands can be found in

Figure 31 on page 142 depicts the necessary steps to make a database copy of a running Exchange Server store using TimeFinder/Mirror:

1. Establish BCVs to Standard devices
2. Consistent split BCVs from STDs

Figure 31  Copying a running Exchange store with TimeFinder/Mirror

1. The first action is to establish the BCVs to the standard devices. This operation occurs in the background and should be executed before the synchronization of the BCV copy:

   symmir -g <device_group> establish -full -noprompt

   Note that the first iteration of the establish needs to be a full synchronization. Subsequent iterations are incremental and do not need the -full keyword. Once the command is issued, the array begins the synchronization process using only Symmetrix resources. Since this operation occurs independently from the host, the process must be interrogated to see when it completes. The command to interrogate the synchronization process is:

   symmir -g <device_group> verify
This command will return a 0 return code when the synchronization operation is complete. Alternatively, synchronization can be verified using the following:

\texttt{symmir -g <device\_group> query}

2. When the volumes are synchronized, the split command can be issued:

\texttt{symmir -g <device\_group> split -consistent -noprompt}

The \texttt{-consistent} keyword tells the Symmetrix to use Enginuity Consistency Assist (ECA) to momentarily suspend writes to the disks while the split is being processed. The effect of this is to create a point-in-time copy of the store on the BCVs. It is similar to the image created when there is a power outage that causes the server to crash. This image is a restartable copy—a term which has been defined previously. The store copy on the BCVs is then available for further processing.

Since there was no specific coordination between the Exchange Server state and the execution of the consistent split, the copy is taken independent of the store activity. In this way, EMC Consistency technology can be used to make point-in-time copies of multiple systems atomically, resulting in a consistent point in time with respect to all applications and stores included in the consistent split.

Using TimeFinder/Clone

TimeFinder/Clone is an EMC software product that copies data internally in the Symmetrix array. A TimeFinder/Clone session is created between a source data volume and a target volume. The target volume needs to be equal to or greater in size than the source volume. The source and target for TimeFinder/Clone sessions can be any hypervolumes in the Symmetrix configuration.

TimeFinder/Clone devices are managed together using SYMCLI device or composite groups. Solutions Enabler commands are executed to create SYMCLI groups for TimeFinder/Clone operations. If the database spans more than one Symmetrix, a composite group is used. Examples of these commands can be found in Figure 32 on page 144. It depicts the necessary steps to make a copy of a running Exchange Server database onto BCV devices using TimeFinder/Clone:
1. Create Clone session
2. Consistent Activate Clone session

**Figure 32** Copying a running Exchange store with TimeFinder/Clone

1. The first action is to create the TimeFinder/Clone pairs. The following command creates the TimeFinder/Clone pairings and protection bitmaps. No data is copied or moved at this time:

   ```
   symclone –g <device_group> create -noprompt
   ```

   Unlike TimeFinder/Mirror, the TimeFinder/Clone relationship is created and activated when it is needed. No prior copying of data is necessary.

2. After the TimeFinder/Clone relationship is created, it can be activated consistently:

   ```
   symclone –g <device_group> activate
   ```

   The `-consistent` keyword tells the Symmetrix to use Enginuity Consistency Assist (ECA) to momentarily suspend writes to the source disks while the TimeFinder/Clone is being activated. The effect of this is to create a point-in-time copy of the store on the target volumes. It is a copy similar in state to that created when there is a power outage resulting in a server crash. This copy is a
Copying the store using EMC Consistency technology

restartable copy—a term which has been defined previously. After the activate command, the store copy on the TimeFinder/Clone devices is available for further processing.

Since there was no specific coordination between the store state and the execution of the consistent split, the copy is taken independent of the store activity. In this way, EMC Consistency technology can be used to make point-in-time copies of multiple systems atomically, resulting in a consistent point in time with respect to all applications and databases included in the consistent split.

As of Enginuity 5772 and later, TimeFinder/Clone devices are no longer subject to a direct COFW penalty. In these later revisions of Enginuity, an ACOFW is implemented, such that update operations from the production host occur immediately, and Enginuity manages the transfer of the original data to the Clone device in the background. This represents a significant increase in performance to the production server as compared to earlier implementations of TimeFinder/Clone. Clone devices at Enginuity 5772 and later, however, suffer the COA penalty for access from the host attached to the Clone devices. COA means that if a track on a TimeFinder/Clone volume is accessed before it has been copied, it must first be copied from the source volume to the target volume.

In earlier versions of Enginuity, stores copied using TimeFinder/Clone are subject to COFW and Copy on Access COA penalties. The COFW penalty means that if a track is written to the source volume and it has not yet been copied to the target volume, it must first be copied to the target volume before the write from the host is acknowledged. COA refers to the copy process which needs to be executed when data on the Clone device is referenced by an alternate host, where the data has not been previously copied to the Clone device itself.

Note: Clone devices which are fully copied prior to being used will not incur any Copy on Write or Copy on Access activities as they are fully independent devices once activated.

TimeFinder/Snap enables users to create complete copies of their data while consuming only a fraction of the disk space required by the original copy.
TimeFinder/Snap is an EMC software product that maintains space-saving, pointer-based copies of disk volumes using VDEVs and SAVDEVs. The VDEVs contain pointers either to the source data (when it is unchanged) or to the SAVDEVs (when the data has been changed).

TimeFinder/Snap devices are managed together using SYMCLI device or composite groups. Solutions Enabler commands are executed to create SYMCLI groups for TimeFinder/Snap operations. If the database spans more than one Symmetrix, a composite group is used. Examples of these commands can be found in

Figure 33 on page 146 depicts the necessary steps to make a copy of a running Exchange Server 2010 store using TimeFinder/Snap:

Figure 33  Copying a running Exchange store with TimeFinder/Snap

1. The first action is to create the TimeFinder/Snap pairs. The following command creates the TimeFinder/Snap pairings and protection bitmaps. No data is copied or moved at this time:

   `symsnap -g <device_group> create -noprompt`

   After the TimeFinder/Snap is created, all the pointers from the VDEVs point at the source volumes. No data has been copied at this point. The snap can be activated consistently using the consistent activate command.

2. Once the create operation has completed, the activate command can be executed with the `-consistent` option. Execute the following command to perform the consistent snap:
Copying the store using EMC Consistency technology

Creating Exchange Server 2010 Clones

```
symsnap -g <device_group> activate -consistent -noprompt
```

The `-consistent` keyword tells the Symmetrix to use Enginuity Consistency Assist (ECA) to momentarily suspend writes to the disks while the activate command is being processed. The effect of this is to create a point-in-time copy of the store on the VDEVs. It is similar to the state created when there is a power outage that causes the server to crash. This image is a restartable copy—a term which has been defined previously. The store copy on the VDEVs is available for further processing.

Since there was no specific coordination between the store state and the execution of the consistent split, the copy is taken independent of the Exchange activity. In this way, EMC Consistency technology can be used to make point-in-time copies of multiple systems atomically, resulting in a consistent point in time with respect to all applications and databases included in the consistent split.

Exchange stores copied using TimeFinder/Snap in earlier version of Enginuity are subject to a COFW penalty while the snap is activated. The COFW penalty means that if a track is written to the source volume and it has not yet been copied to the snap save area, it must first be copied to the save area before the write from the host is acknowledged. COA refers to the copy process which needs to be executed when data on the Snap device is referenced by an alternate host, where the data has not been previously copied to the save area.
Creating Exchange Server 2010 Clones

Copying the store using the Exchange Server VSS Interface

Exchange Server 2010 running on Windows Server 2008 supports the VSS programmatic interface that provides the capability to use split mirror disk technology while an Exchange Server store is online and creates a recoverable copy on the copied devices. During this process, the store is fully available for reads. Write operations are suspended during the VSS split operation. When a transaction issues a commit, it is suspended until the VSS operation has been issued and the subsequent disk mirror is split.

In contrast to the executions of the previous section, there is no need to specify the -consistent option to any TimeFinder command, as Exchange Server through the VSS framework ensures store consistency is maintained.

The copy made in this manner is considered a valid backup of the target stores(s). Exchange Server 2010 will acknowledge that a valid backup has been completed and facilitates transaction log pruning.

Because the VSS framework is a programmatic interface, it cannot be called directly by a user. Custom applications are typically created by storage vendors such as EMC that interface between the VSS framework and the storage array technology. These applications are referred to as VSS Requestors. EMC provides a number of products that support this mechanism for creating valid backups for EMC Symmetrix. These products include the graphical Replication Manager and the command line interface-based TimeFinder Exchange Integration Module (TF/EIM).

In the following examples, TF/EIM is used to demonstrate the various forms of TimeFinder functionality. These examples assume execution of the TF/EIM product on the backup server.

Using TimeFinder/Mirror

TimeFinder/Mirror is an EMC software product that allows an additional hardware mirror to be attached to a source volume. The additional mirror is a specially designated volume in the Symmetrix configuration called a business continuance volume, or BCV. The BCV is synchronized to the source volume through a process called an establish. While the BCV is established, it is not ready to all hosts. At an appropriate time, the BCV can be split from the source volume to create a complete point-in-time copy of the source data that can be used for multiple purposes, including backup, decision support, and regression testing.
Creating Exchange Server 2010 Clones

Executing TF/EIM

The process detailed in Figure 34 on page 149 demonstrates the necessary steps required to create a VSS enabled split-mirror backup of a Exchange Server 2010 store using TF/EIM to manage TimeFinder/Mirror operations:

1. Establish BCVs to Standard devices
2. Execute TF/EIM backup command
   2.1 Validate files and devices for use
   2.2 Create Shadow copy
   2.3 Unmask Clones to Backup Host
   2.4 Import Shadow copy and run ESEUTIL

Figure 34  TimeFinder/Mirror VSS backup of an Exchange store

1. Using TF/EIM, it is possible to preestablish the TimeFinder/Mirror BCV devices, although if no preestablished devices exist, TF/EIM will also manage the synchronization of mirror states. If, preestablish is required, the first action is to establish the BCVs to the standard devices. As this operation occurs in the background, the preestablish can be useful in situations where the TF/EIM backup process is required to execute in a more specific time frame. Since in a preestablished environment, TF/EIM will not need to wait for the resynchronization to occur; thus, when executed, the VSS backup execution will proceed immediately.

2. To establish the BCV devices the following command may be executed:

   symmir -g <device_group> establish -full -noprompt

   Note that the first iteration of the establish needs to be a full synchronization. Subsequent iterations are incremental and do not need the -full keyword. Once the command is issued, the array begins the synchronization process using only Symmetrix
resources. Since this is asynchronous to the host, the process must be interrogated to see when it is finished. The command to interrogate the synchronization process is:

```bash
symmir -g <device_group> verify
```

This command will return a 0 return code when the synchronization operation is complete.

3. When the volumes are synchronized, the TF/EIM backup operation may be executed:

**Note:** TF/EIM does not actually use the device group to manage its internal operations, rather it examines relationships that have been created between source and disk mirrors by administrators as a result of their use of various SYMCLI device group operations.

```bash
Exbackup2010 backup -s <server> -vss -sg <store> -bcd <location of backup document> -wmd <location of metafile>
```

If the preestablish process were to be utilized, the `-preestablish` option should be utilized in the TF/EIM backup call.

---

**Using TimeFinder.Clone**

TimeFinder/Clone is an EMC software product that copies data internally in the Symmetrix array. A TimeFinder/Clone session is created between a source data volume and a target volume. The target volume needs to be equal to or greater in size than the source volume. The source and target for TimeFinder/Clone sessions can be any hypervolumes in the Symmetrix configuration.

TimeFinder/Clone devices are managed together using SYMCLI device or composite groups. Solutions Enabler commands are executed to create SYMCLI groups for TimeFinder/Clone operations. If the store spans more than one Symmetrix, a composite group is used. Examples of these commands can be found in

---

**Executing TF/EIM**

The TimeFinder Exchange Integration Module provides support for both the Symmetrix Clone Emulation mode, and native Clone operations, to facilitate backup operations for an Exchange Server
Copying the store using the Exchange Server VSS Interface

Creating Exchange Server 2010 Clones

store. Native Clone operations only require that a Clone session has been created. Clone Emulation mode further requires that the Clone devices are fully preoccupied prior to the TF/EIM execution.

Figure 35 on page 151 depicts the necessary steps to make a VSS backup of a Exchange Server 2010 store onto BCV devices using TimeFinder/Clone.

1. Pre-copy Clones to Standard devices
2. Execute TF/EIM backup command
   2.1 Validate files and devices for use
   2.2 Create Shadow copy
   2.3 Unmask Clones to Backup Host
   2.4 Import Shadow copy and run ESEUTIL

As of Enginuity 5772 and later, TimeFinder/Clone devices are no longer subject to a direct COFW penalty. In these later revisions of Enginuity, an Asynchronous Copy on First Write (ACOFW) is
Creating Exchange Server 2010 Clones

implemented, such that update operations from the production host occur immediately, and Enginuity manages the transfer of the original data to the Clone device in the background. This represents a significant increase in performance to the production server as compared to earlier implementations of TimeFinder/Clone. Clone devices at Enginuity 5772 and later, however, suffer the COA penalty for access from the host attached to the Clone devices. COA means that if a track on a TimeFinder/Clone volume is accessed before it has been copied, it must first be copied from the source volume to the target volume.

In earlier versions of Enginuity, stores copied using TimeFinder/Clone are subject to COFW and COA penalties. The COFW penalty means that if a track is written to the source volume and it has not yet been copied to the target volume, it must first be copied to the target volume before the write from the host is acknowledged. COA refers to the copy process which needs to be executed when data on the Clone device is referenced by an alternate host, where the data has not been previously copied to the Clone device itself.

**Note:** Clone devices which are fully copied prior to being used will not incur any Copy on Write or Copy on Access activities as they are fully independent devices once activated.

Using TimeFinder/Snap

TimeFinder/Snap enables users to create complete copies of their data while consuming only a fraction of the disk space required by the original copy.

TimeFinder/Snap is an EMC software product that maintains space-saving pointer-based copies of disk volumes using VDEVs and SAVDEVs. The VDEVs contain pointers either to the source data (when it is unchanged) or to the SAVDEVs (when the data has been changed).

TimeFinder/Snap devices are managed together using SYMCLI device or composite groups. Solutions Enabler commands are executed to create SYMCLI groups for TimeFinder/Snap operations. If the database spans more than one Symmetrix, a composite group is used. Examples of these commands can be found in Appendix A, “SYMCLI Group Creation Command Samples.”
Executing TF/EIM

Figure 36 on page 153 depicts the necessary steps to make a VSS backup of an Exchange Server store using TimeFinder/Snap.

1. The first action is to create the TimeFinder/Snap pairs. The following command creates the TimeFinder/Snap pairings and protection bitmaps. No data is copied or moved at this time:

   ```
   symsnap –g <device_group> create -noprompt
   ```

   Unlike TimeFinder/Mirror, the snap relationship is created and activated when it is needed. No prior copying of data is necessary. The create operation establishes the relationship between the standard devices and the VDEVs, and it also creates the protection metadata.

2. After the snap is created, the TF/EIM backup operation may be executed:

   ```
   Exbackup2010 backup –s <server> -vss –sg <Storage group> -bcd <location of backup document> -wmd <location of metafile> -snap
   ```

   In this example, the target production store is provided after the –sg command line option. The –wmd and –bcd command line options specify the location where TF/EIM will store the VSS metadata files.
As of Enginuity 5772 and later, TimeFinder/Snap devices are no longer subject to a direct COFW penalty. In these later revisions of Enginuity, an ACOFW operation is implemented, such that update operations from the production host occur immediately, and Enginuity manages the transfer of the original data to the Snap device in the background. This represents a significant increase in performance to the production server as compared to earlier implementations of TimeFinder/Snap. Snap devices at Enginuity 5772 and later, however, suffer the COA penalty for access from the alternate host attached to the Snap devices.

Stores copied using TimeFinder/Snap in earlier version of Enginuity are subject to a COFW penalty while the snap is activated. The Copy on First Write penalty means that if a track is written to the source volume and it has not yet been copied to the snap save area, it must first be copied to the save area before the write from the host is acknowledged. Copy on Access refers to the copy process which needs to be executed when data on the Snap device is referenced by an alternate host, where the data has not been previously copied to the save area.
Copying the store using Replication Manager

EMC Replication Manager (RM) can be used to manage and control the TimeFinder copies of Exchange Server stores. The RM product has a GUI and command line and provides the capability to:

- Auto-discover the standard volumes holding the database
- Identify the path name for all data and transaction log file locations

Using this information, RM can set up TimeFinder Groups with BCVs, Clones or VDEVs, schedule TimeFinder operations, and manage the creation of database copies, terminating older versions as needed.

Figure 37 on page 155 demonstrates the steps performed by Replication Manager using TimeFinder to create a store copy using VSS that can be used for multiple purposes:

1. In the first step, Replication Manager maps the store locations of all the data files and logs in all Symmetrix devices. Note that the dynamic nature of this activity will handle the situation when extra volumes are added to the database. The procedure will not have to change.
2. Replication Manager then establishes the BCVs to the standard volumes in the Symmetrix, or creates the necessary sessions when using Clone or Snap devices. Replication Manager polls the progress of the establish until the BCVs are synchronized and then moves on to the next step.

3. Replication Manager executes an Exchange Server VSS call to execute the backup process.

4. The VSS framework will then coordinate the necessary TimeFinder operation using the hardware provider to either activate the Snap or Clone, or split the BCV devices.

5. Next, the VSS framework informs the Exchange VSS writer that the split operation has completed, and all normal operations are resumed.

6. Replication Manager copies the VSS metadata files from the source host to an RM holding area for use later in a restore.

7. Finally, the target devices are mounted on amount host, where an ESEUTIL verification will take place.
Transitioning disk copies to Exchange Server 2010 clones

Transforming an Exchange store copy into a usable state, requires the coordination of a number of subsystems. As discussed in Chapter 1, the implementation of an Exchange infrastructure requires integration within a Windows Active Directory instance. Therefore, to be able to allow connectivity to the Exchange instance, Active Directory information would need to be replicated to an alternate environment. The details and requirements for such a configuration are beyond the scope of this TechBook.

Other scenarios that are able to utilize cloned instances of Exchange stores do exist, but all possible scenarios are not covered in this TechBook.

It is possible to utilize a copy of an Exchange store created using any of the various forms of replication discussed earlier in this chapter. How a store copy can be enabled for use depends on how the copy was created. A store copy created using either the Exchange Server VSS processing or as a consistent split, will in general represent a “Dirty Shutdown” state. That is, the individual database files will require recovery processes to return to a transactionally consistent state.

In the following discussion, the use of a consistent split copy of a running storage group will be utilized. The copy created with store shutdown may not require any specific recovery processes, since it will have been left in a clean shutdown state.

This section details how to use the ESEUTIL utility to recover the copied store, and how to utilize the store copy created using EMC Consistency technology.

Instantiating clones from consistent split or shutdown images

In most cases, when creating a copy of a store and utilizing it on a different server, the Exchange Management Console on the target Exchange server must be made aware of the new database that is coming under its control.

Verification of the state of the mailbox stores will be necessary to ensure that they have a clean shutdown state. In Figure 38 on page 158, the output of an ESEUTIL file dump is displayed, as the store copy was created using Consistent Split operations. The state is shown as “Dirty Shutdown.”
Creating Exchange Server 2010 Clones

To recover the mailbox store to a clean state, it is necessary to use the stores logs to resolve the transactional state. Again, ESEUTIL can be used to replay the transaction logs into the mailbox stores. In Figure 39 on page 159, a recovery of the mailbox stores is initiated, using the original Exchange Transaction Logs, and specifying the new mailbox store location.
Figure 39  Using ESEUTIL to recover mailbox stores to a consistent state

ESEUTIL will perform all necessary roll forward/roll back recovery on the storage group databases. Since the Write Ahead Logging functionality of Exchange Server has been maintained, the resulting cloned copy will return to a transactional consistent state, and become a fully independent clone of the production instance. Figure 40 on page 160 shows the clean state now returned by the ESEUTIL file dump function.
Creating Exchange Server 2010 Clones

Once recovered, the mailbox stores are able to be configured for use within the Recovery Storage Group. Using Exchange Management Console, it is necessary to define the new locations for the database locations.

Once the copied storage group has been correctly copied, processed, and configured into a Recovery Database (RDB), the cloned instance of the source Exchange mailbox is available.

Using Exchange Server and VSS to process the image

The VSS framework for an Exchange Server environment is specifically targeted at creating a valid backup and restore mechanism for a production instance. It is not specifically targeted for facilitating the creation of scenarios such as RDB or other cloning functions. It is possible, however, to utilize the VSS created image to create a RDB, or other cloning environment.
Unlike shutdown or Consistent Split images, copies created using the VSS framework include changes managed by the Windows environment. Specifically, the VSS framework will set two NTFS volume attributes that will need to be reversed prior to using the copy. The two relevant attributes are the READ_ONLY and HIDDEN parameters for the relevant volumes.

To revert these attributes, it will be necessary to use the DISKPART utility. Further information on the usage of the DISKPART utility, and the method to reset the changes made by the VSS subsystem, may be found on the Microsoft Support website at http://support.microsoft.com/. All other operations for utilizing the copied data would be similar to those outlined earlier.

⚠️ CAUTION ⚠️

Using the mirror devices to instantiate a database clone will modify the state of the files on those devices. Once a VSS backup image has been modified in this way, it will no longer be possible to restore this specific instance back to the production system, and it should no longer be considered a “backup” image.
Reinitializing the cloned environment

In many cases, it may be necessary to reinitialize the clone environment. To facilitate this process, it is necessary to reverse some of the processes that were executed to create the database clone using the specific process required.

A number of steps are required to remove the clone database from the target server, so that a reinitialization may be processed.

1. Unmount the RDB database from the target server. This may be executed using the Exchange System Manager UI.
2. Unmount the volumes from the target server using appropriate symntctl commands.
3. Reestablish BCV relationship or re-create sessions as appropriate.
4. Execute the appropriate split functionality as initially used.
5. Remount the volumes using appropriate symntctl commands.
6. Remounting the cloned mailbox stores in the same manner as originally processed.
Choosing an Exchange cloning methodology

Each of the replication technologies described in previous sections has pros and cons with respect to its applicability to solve a given business problem. The following matrix in Table 10 provides a contrast to the different methods that can be used and the differing attributes of those methods.

Table 10 Comparison of cloning technologies for Exchange Server

<table>
<thead>
<tr>
<th></th>
<th>TF/Snap</th>
<th>TF/Clone</th>
<th>TF/Mirror</th>
<th>Replication Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of simultaneous copies</td>
<td>15</td>
<td>16</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Production impact</td>
<td>None—ACOFW with &gt;72 code</td>
<td>COA ACOFW with &gt;72 code</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Scripting</td>
<td>Required</td>
<td>Required</td>
<td>Required</td>
<td>Automated</td>
</tr>
<tr>
<td>Exchange copy required a long time</td>
<td>Not recommended</td>
<td>Recommended</td>
<td>Recommended</td>
<td>Recommended</td>
</tr>
<tr>
<td>High write usage to copy</td>
<td>Not recommended</td>
<td>Recommended</td>
<td>Recommended</td>
<td>Recommended</td>
</tr>
</tbody>
</table>

COFW = Copy on First Write
ACOFW = Asynchronous Copy on First Write (>72 code)
COA = Copy on Access

Table 11 on page 164 details examples of the choices you might make for Exchange cloning based upon Table 10 on page 163.
Creating Exchange Server 2010 Clones

Table 11  Replication options

<table>
<thead>
<tr>
<th>System requirements</th>
<th>Replication choice</th>
</tr>
</thead>
<tbody>
<tr>
<td>The application on the source volumes is very performance-sensitive and the slightest degradation will cause responsiveness of the system to miss SLAs</td>
<td>TimeFinder/Mirror Replication Manager</td>
</tr>
<tr>
<td>Space and economy are a real concern. Multiple copies are needed and retained only for a short period of time, with performance not critical.</td>
<td>TimeFinder/Snap Replication Manager</td>
</tr>
<tr>
<td>More than two simultaneous copies need to be made. The copies will live for up to a month.</td>
<td>TimeFinder/Clone</td>
</tr>
<tr>
<td>Multiple copies are being made, some with production mount. The copies are reused in a cycle, terminating the oldest first.</td>
<td>Replication Manager</td>
</tr>
</tbody>
</table>
This chapter includes the following topics:

- Introduction ................................................................. 166
- EMC Consistency technology and backup ..................... 167
- Exchange Server backup functionality .......................... 168
- Integrating EMC technologies with Exchange Server backups 176
- TimeFinder Exchange Integration Module ...................... 178
- Replication Manager ...................................................... 189
Introduction

To mitigate the possibility of complete or partial data loss of a production database, it is necessary to implement a backup and recovery process that ensures that a separate, logically valid copy of the database is saved. If the production database is rendered unusable, the logically valid copy may be restored to the production system. This necessity holds true for systems that use other high-availability mechanisms to ensure database redundancy or service continuity or both.

Most production Exchange environments have availability requirements that demand access 24 hours a day, seven days a week. Due to these requirements, all backup operations must be processed online with minimal impact to the production system. As mailbox sizes grow and increase database and server storage requirements, the backup window increases. This begins to interrupt Online Maintenance, patch management, and other environmental tasks. As these issues becoming more prevalent, online streaming backups become less feasible.

This chapter describes Exchange Server 2010 backup processes, and how messaging administrators can leverage EMC technology in a backup strategy to:

- Reduce production system impact during backup processing
- Create consistent online backups
- Integrate Exchange Server backup/recovery processes
EMC Consistency technology and backup

In Chapter 3, “Creating Exchange Server 2010 Clones,” the use of EMC Consistency technology was described as a method to create restartable copies of a source Exchange Server storage group. This type of technology is noninvasive to the production system operations and occurs entirely at the storage array level. The restartable images represent dependent-write consistent images of all related objects that have been defined within the consistency group. This technology is sometimes referred to as a crash consistent split or a hot split.

It is possible to maintain these images locally or stream the images to longer term media such as tape. However, these copies of the Exchange databases do not utilize the Exchange Server’s VSS Writer in conjunction with EMC’s consistent splits. This leaves the Exchange system in a state where the logs are not truncated and the usage of these consistent split images is limited. This also means the supportability of the image changes from an EMC and Microsoft perspective such that they are only viable for Restart purposes. These limitations are acceptable for environments where the consistent split images are specifically used for replication or other disaster restart functionality.

EMC recommends that customers continue to utilize backup technologies to ensure that the environmental requirements are met. Those environmental requirements are log truncation, logical corruption recoverability, and regulatory compliance.
Exchange Server backup functionality

Exchange Server 2010 only provides the Exchange VSS Writer to facilitate database backup processing.

This chapter will detail integration with VSS backup functionality.

Exchange Server 2010 supports two basic types of backups: offline and online. As discussed earlier, offline backups do not support the service uptime requirements of modern implementations, so will only be discussed briefly in this chapter.

Offline backups

For offline backups, the procedure is simple: One or more databases are taken offline. In the process of being taken offline the databases naturally become consistent by flushing all remaining writes in the database cache into the database files. Once the databases are offline, physical copies of the database files and logs can be made. This can be accomplished with a file system copy, a traditional backup application, or a disk mirroring backup application. All of these methods are allowed and supported by Microsoft for offline backups. Prior to physically copying the database and log files, it is necessary to check whether the files are in a consistent state.

To validate that the database files being backed up are in the same set, verify that the output of the signature line matches exactly for both files. ESEUTIL can be used to interrogate the headers of the database EDB within a storage group.

The following steps may then be followed to create a clean shutdown copy of the data files.

1. Validate that the database files being backed up are consistent as detailed in. The output will state “Clean Shutdown” if the database is consistent.

2. Make a physical copy of the database file.
   The copy of the database file can be made through any number of ways including disk mirroring or disk cloning.

3. Make a physical copy of the log files
   The copy of the log files can be made through any number of ways including disk mirroring or disk cloning.
It is recommended that all log files are copied and truncated from production volumes except E0n.log. This will allow roll-forward recovery to occur if necessary.

**Online VSS backups**

VSS backups are a little more complex due to the stringent requirements of the VSS framework and requirements to validate the backup; yet they are relatively simple to use in production and operate on a daily basis considering that the backup is being taken from a live system without impact to running clients. The main components, the different types of VSS backups, the steps to accomplish a backup, and EMC’s integration are described here.

There are four major components dictated by the VSS framework, the VSS Service, the VSS Writer, the VSS Requestor, and the VSS provider. All four components are required to perform a VSS backup. Figure 41 on page 170 details the inter-relationship of these VSS components.
The core of the system is the VSS Service which Microsoft built into Windows Server 2003 and later. This service is a broker that handles communication between the other VSS components and guarantees that each component follows the framework guidelines.

As a component of several server applications including Exchange Server 2010 and earlier, Microsoft has included a VSS Writer. The writer is the component responsible for managing the application being backed up. This includes telling the application to quiesce its data stores, close its last open log file, and suspend all I/O. This also includes the writer’s responsibility to pass along the application name, files, and so forth, inside an XML file.

The most visible component, the VSS Requestor is not provided by Microsoft. The requestor is typically built into applications such as the EMC TF/ EIM and EMC RM. These third-party requestor
applications may then add functionality such as scheduling, cataloging, and integration with advanced storage functionality. The requestor is responsible for coordinating the creation of the shadow copy.

The last component is the provider; Microsoft has included a provider that will not be reviewed in-depth. This provider is the system provider and is built into Windows 2003 and earlier, but is not generally used by software or hardware requestors due to the flexibility a custom provider allows. These vendor-created providers are specific to the hardware or software they are designed to manage and are based on rules laid out by Microsoft. These rules are in place to ensure that the requestor can work with any given provider and that the VSS service can properly broker the requests.

All of the components are used in concert to create shadow copies of the application being targeted. For Exchange, this means that several specific types of backups can be created. These backups are restricted first by those operations that the framework allows, secondly by what the Exchange writer can create/restore, and thirdly by what the requestor and provider support. However, the framework itself can be very flexible in what it offers and can support a range of different scenarios that may or may not be suitable for any given application. The Exchange writer has implemented a large portion of these features in the VSS framework over time. As these features are implemented in the writer, backup applications (requestors) need to be updated to support the additional features. These supported scenarios vary by version of Windows Server, Exchange Server, and the requestor application.

Exchange Server 2010 supports two shadow copy methods. These are Full Shadow copies or Copy Shadow copies.

Full Shadow Copies are full replicas of the database files and logs. The term “Full Shadow Copy” indicates that the logs will be truncated through VSS after checksum verification has been completed on the storage group files. These full replicas can be obtained through a disk mirroring technique where the production spindles and the backup spindles are in a pre-established mirrored relationship and the backup spindles are split at the time the shadow copy is created, resulting in a fully independent and complete copy on the backup spindles. The full replicas can also be obtained through a cloning technique where the clone device states are created at the time of execution, but where backup spindles receive a consistent copy of the data from the production spindles at a later time.
Copy Shadow Copies are also full replicas of the database files and logs. The term “Copy Shadow Copy” indicates the logs will not be truncated as a result of this backup operation. These full replicas can be obtained through a disk mirroring technique where the production spindles and the backup spindles are in a mirrored relationship and the backup spindles are split at the time the shadow is created. The full replicas can also be obtained through a cloning technique where the backup spindles receive a copy of the data from the production spindles at the time the shadow is created.

Exchange Server 2010 enables a feature that allows throttling of the consistency check procedure. This means that the consistency check procedure can be configured to pause for one second every X (user specified) number of I/Os. This allows the ESEUTIL process to slow down and not overload an under configured system.

In general, the steps required to create a VSS shadow copy are:

1. The Requestor application creates a VSS Backup Component with the VSS Service to be used throughout the backup session. The Requestor contacts the writer to obtain the writer metadata document.

2. The Writer creates and delivers this document to the requestor. The document includes the files to be backed up, the restore method, and any exclusions that are necessary.

   The writer prepares the data in whatever way is appropriate, such as completing all open transactions, rolling transaction logs, and flushing caches. When the data is prepared for shadow copy creation, the writer notifies the Volume Shadow Copy Service.

3. The VSS initiates the “commit” shadow copy phase.

4. The VSS tells the writers to quiesce their data and temporarily freeze the application I/O write requests (I/O read requests are still possible) for the several seconds required to create the shadow copy of the volume or volumes. The application freeze is not allowed to take longer than 10 seconds. The VSS flushes the file system buffer and then freezes the file system, which ensures that file system metadata is written and that the data is written in a consistent order.

5. The VSS tells the provider to create the shadow copy (a maximum of 10 seconds).
6. The VSS thaws the file system. After the shadow copy is created, the VSS releases the writers from their temporary inactive phase and all queued write I/Os are completed.

7. The VSS queries the writers to confirm that write I/Os were successfully held during shadow copy creation.

8. If the writes were not successfully held (meaning that the shadow copy data is potentially inconsistent), the shadow copy is deleted and the requestor is notified.

9. The requestor can retry the process (go back to step 1) or notify the administrator to retry at a later time.

10. If the copy is successful, the VSS gives the location information for the shadow copy back to the requestor.

11. The Requestor mounts the copy to a separate host and performs a consistency check using the ESEUTIL utility.

Backup considerations

Other Exchange objects to back up include the offline address book and the Windows registry. These are both important components of a mailbox server that are not included in Exchange Server backups. The offline address book can be backed up through a file system backup or copy mechanism. The Windows registry should be backed up through a system state backup or registry export.

Content indexing is another important factor to account for. Backups of the content index are not required as they can be rebuilt. It is important to ensure that building the content index is done when processing is low. The initial build of the index is resource-intensive and may conflict with other processes. Once the content index is built, regular indexing results will be in little to no overhead and should not conflict with any other processes.

Online maintenance (OLM) is an important task that needs to occur in all Exchange Server 2010 environments. As an Exchange Server environment matures or as its capacity grows, the importance of online maintenance also grows. All Exchange Store-focused online database maintenance functions such as recovery item cleanup are the same in Exchange Server 2010 as they were in Exchange Server 2007.
Only ESE functions, online defragmentation, and database checksumming have changed and are described in “ESE functionality improvements for Exchange Server 2010” on page 38. All other aspects of database maintenance still have to be carried out and the option to run continuously or at scheduled times is available.

OLM is responsible for a number of important tasks including:

- Purging Indices
- Tombstone maintenance
- Dumpster Cleanup
- Public Folder Expiry
- Age Folder Tombstone
- Folder Conflict Aging
- Update Server Versions
- Cleanup Secure Folders
- Cleanup Deleted Mailboxes
- Check Messages Table

In most environments, OLM cannot complete all tasks in a single pass, due to the amount of time required to process the content within the databases. In most cases it will need to run several times over the course of one or two weeks to complete. To do this, OLM makes note of which task is running at the end of the window. The currently running task will be allowed to run to completion. When the next window starts, the next task is processed. If all of the tasks do not complete after two weeks, the environment should be examined and adjustments made to allow OLM to complete all tasks within this two week time frame. There are several indicators of the online maintenance activity.

- Windows Application Event log as event ID 700. This event indicates OLD is beginning a full pass on the specified database. This event will be seen on every time the OLD is run.
- Windows Application Event log as event ID 701. This event indicates OLD completed a full pass on the specified database. This event will be seen on every time the OLD is run.
- Windows Application Event log as event ID 1206. This event indicates OLM is starting a cleanup of items past retention date for Item Recovery on the specified database.
- Windows Application Event log as event ID 1207. This event indicates OLM is starting a cleanup of items past retention date for Item Recovery is complete for database.

- Windows Application Event log as event ID 1221. This event indicates that the OLD process has nnMB of free space on a specified database after the OLD has terminated.

It is important to note that the only OLM task that will be paused by backups is the Online Defragmentation task.

**Online defragmentation**

The intention for online defragmentation is to free up pages in the database by compacting records onto the fewest number of pages possible (manages database white space), thus reducing the amount of I/O necessary for transactional processing. The ESE database engine accomplishes this by “crawling” or scanning the database metadata (information in the database that describes tables in the database) for every database page within each table and attempts to move records onto logically ordered pages. Earlier in this discussion, it was noted that Exchange Server 2010 database technology leverages B-Tree binary structures, which implies that the manner in which these database pages are physically stored on disk is quite different from their logical organization within the “balanced tree.” As a result, the online defragmentation will begin at the farthest page to the right and begin compressing records to the left-most pages. This does not necessarily mean the pages are in order, but the movement across the physical file is logical in nature.

In Exchange Server 2010, the architecture for online defragmentation has changed. Online defragmentation was moved out of the Mailbox database maintenance process. Online defragmentation now runs in the background 24×7. there is no requirement to configure any settings for this feature as Exchange monitors the database as it’s being used, and small changes are made over time to keep it defragged for space and contiguity. If the database analyzes a range of pages and finds that they aren’t as sequential as they should be, it starts an async thread to defragment that section of the B-tree/table. Online defragmentation is also throttled so it doesn’t have a negative impact on client performance.
Integrating EMC technologies with Exchange Server backups

EMC has several tools to back up Exchange Server including TF/EIM and RM. These tools generally vary in functionality based on supported storage platforms.

To reduce complexity, EMC recommends minimizing the number of LUNs used for each storage group. In Exchange Server 2010, EMC best practice is to utilize two LUNs for each storage group, one LUN for transaction log and system files and a secondary LUN for database. These LUNs should be mounted using individual drive letters or as mount points to a separate volume. If using mount points, the drives should be mounted to a directory that resides on a non-system drive. If restoration procedures require the flexibility to restore a single database while the other databases are online, multiple database disks can be used; however, this configuration is not recommended due to the impact of both performance and complexity. It is also acceptable to run both the database and logs on the same LUN if the Exchange storage design require this. VMAX has always been able to cope with the randomness of the databases and the sequentialness of the logs to reside on the same physical disks.

VSS integration

The TF/EIM and RM are the two main applications, EMC has provided, to facilitate backups using the VSS framework. TF/EIM and RM will be covered in depth in this version of this TechBook.

- EMC NetWorker®, RepliStor®, and RecoverPoint also have VSS-based integration either with Windows, Exchange, or both.
  These products will not be covered in depth in this version of the guide; however, more information can be found in their respective product guides.

Managing backup integration

The two most important aspects to deal with when implementing any Exchange backup product in a storage area network (SAN) are managing LUNs and coordinating timings for backup operations. Managing LUNs in a SAN environment for Exchange can potentially require a large number of LUNs and the need to dynamically make LUNs visible and invisible to backup hosts. this now more apparent with the ability to have 100 databases per server. If 100 databases are deployed it would be best practice to configure 100 LUNs to host the
Integrating EMC technologies with Exchange Server backups

Backing Up Exchange Server 2010 databases and 100 LUNs to host the log files making a total of 200 LUNs. All paths for databases that participate in a DAG have to be identical in every Exchange server.

The potential number of LUNs is the simplest issue to address. Using a logical mount directory structure and carefully naming all volumes related to each LUN can remove much of the complexities involved with managing a large number of LUNs. For example, this could be achieved by using the server name and database number in the label of the volume (EXCH1-DB2-1). With a descriptive label it will be easy to identify where a particular volume should be mounted. Additionally, if the data (log or databases) on this volume is also in a directory with the same name, viewing the directory structure through a command shell or Windows Explorer will provide the same information. Mounting this volume to a location on disk, such as G:\Exc_Mounts\Exch1\DB2-1, will ensure that mistaken mounts can be resolved easily. This is important in Exchange Server 2010 with a potential of 200 LUNs for Exchange storage—one LUN for each database file (100 in total) and one LUN for each transaction log area (100 in total).

Dynamically making LUNs visible (unmasking) and invisible (masking) to the host is another powerful feature of SAN environments. When incorrectly used, this functionality can cause problems for Windows hosts due to the methods Windows uses to mount and lock volumes, and typically requires tools that can coordinate this activity in a structured and consistent manner. EMC provides the Symmetrix Integration Utility product to appropriately manage this functionality.

The Symmetrix Integration Utility (SYMNTCTL.EXE) allows administrators to mount and unmount volumes, flush volume buffers, report on processes with open handles on disks, and rescan the SCSI bus for volumes.

Within Windows Server 2008, Microsoft provides the DISKPART.EXE, tool to allow administrators to perform disk management tasks from the command line or through a script. Most importantly, it has the capability to scan the SCSI bus, control volume mounts, and display specific volume attributes.

Coordinating the time it takes to perform a backup operation in conjunction with the other tasks that must occur on a system, such that they do not conflict, is an important aspect. All mailbox servers have several tasks that must be completed to maintain an operationally efficient state.
**TimeFinder Exchange Integration Module**

The TF/EIM is one of EMC’s tools that can be utilized within an Exchange backup, restore, and operational continuity plan to provide backup images of the Exchange databases. TF/EIM is a command line tool that easily integrates with other subsystems such as the VSS framework, and a variety of other backup applications. TF/EIM solves the problem of running VSS-based backups for Exchange Server against an EMC Symmetrix storage platform.

In general, TF/EIM does this by matching the disk replication technology built into the Symmetrix with the needs of the Exchange application and the requirements of the VSS framework. This section will discuss how TF/EIM matches the TimeFinder replication, including TimeFinder/Mirror, TimeFinder/Clone, and TimeFinder/Snap with both the Symmetrix Remote Data Facility (SRDF) capabilities and the Exchange 2xxx VSS backup capabilities.

**Planning to use TF/EIM**

Prior to using TF/EIM as a part of a backup, restore, or continuity plan, certain considerations and planning must take place.

The most important decision to make for planning a backup strategy is how it will coincide with the overall recovery strategy (both local recovery and operational continuity recovery). Determining these requirements necessitates an understanding of which local replication technology will be used to make the copies, what backup type will be used, how many copies will be kept (on disk and on tape), and the timing of those components. The Replication and recovery decisions will be covered in more depth in their associated chapters.

The types of local replication available to the Symmetrix platform, TF/EIM, and the VSS framework are quite robust. Together they support the use of Disk Mirrors, Disk Clones, and Snapshots. These technologies are described. It is important to note that these terms refer only to the type of local data replication between disks inside the Symmetrix.

It is generally recommended to utilize TimeFinder/Mirror or TimeFinder/Clone devices as these technologies provide the most flexibility, highest performance, and best recovery time. Both TimeFinder/Mirror and TimeFinder/Clone result in fully independent LUNs, while TimeFinder/Snap devices require an ongoing dependency on the source LUNs. Specifically when used for
Backing Up Exchange Server 2010

backup purposes, TimeFinder/Snap devices can require significant I/O bandwidth, which is ultimately directed to the source LUNs and thereby possibly impacting production performance.

The types of Exchange VSS backups available to TF/EIM are full backups, copy backups, and differential backups. Choosing which type of backup to use will drastically affect the number of copies that must be kept and the length of time those copies must be maintained. Since VSS is the only supported means of backup for an Exchange Server 2010 server, the -vss is a required option.

The time required to execute these backup operations differs depending on the amount of data to be backed up and the state the disks are in prior to executing the backup command. If the disks that are to be used as targets for the backup are not already in use for another function (for example, being used as a source for tape or other disk copy, being used as a gold copy, being used in recovery), the disks can be synchronized with the production disks ahead of time. This is called preestablishing the disks. The use of preestablishing the disks will cut down on the time required to execute a backup during the backup window as the preestablish can happen at any time with minimal impact to the production disks.

It is highly recommended that the EMC TimeFinder/Integration Module Product Guide is consulted and the “Setting up a production server for TF/EIM” section is thoroughly reviewed prior to any TF/EIM deployment.

Implementing backups with TF/EIM

To initiate a backup using TF/EIM, there are many ways to compose the command line options. The options used are primarily dictated by the variables decided on during implementation.

The ExBackup2010 command is for Windows Server 2008 and above. This TF/EIM command backs up mailbox databases for a single or Exchange Server 2010 DAG Server.

Exbackup2010 –s <servername>

If the plan is to back up a mailbox store with different commands the –mb option should be specified next.

It is recommended that an output file is used to capture the output of the command. This is done with the –o option which follows the servername and storage group options. The output will be appended to the filename specified in this option when the command is run.
Exbackup2010 -s <servername> -o C:\TEIM\<servername>.log

Additionally, it is recommended that a single backup server be used to backup more than one production host. Testing has shown that on average, a single backup server can handle the backups of three fully configured production hosts (four storage groups with five databases each). With smaller Exchange Server configurations, a single TF/EIM backup server may be able to process more Exchange servers. To accomplish these backups, each command can be run separately or the use of an initialization file can be implemented. Usually integration with the overall backup strategy will determine which of these two options provides the best alternative.

The usage of the initialization file may be the most efficient method as this will result in a single command execution and the outputs of all backups can be placed in individual files, while the logging of the overall command is done separately. The limitation in this environment is that each backup process cannot be initiated at separate times and there are not separate backup processes to output status from.

To use the initialization file, the –f option replaces all of the options previously discussed. Additional information regarding available options is provided in the EMC TimeFinder Exchange Integration Module Product Guide.

Implementing TF/EIM

There are several other things to consider when implementing backups with TF/EIM. These requirements relate to both how TF/EIM interacts with Exchange and how TF/EIM interacts with the storage.

TF/EIM interacts with Exchange through the VSS framework. It does this by specifying the –vss switch when the command is issued. By default, the VSS framework creates full backups, meaning the database and log devices are incrementally synchronized on the back end and then split during the 10 second freeze time allowed by VSS. At this point the databases are checked for consistency and following a successful consistency check the transaction logs on production are truncated.

The framework can be further leveraged with the use of the –copyonly switch, the –vssdiff switch, and the –scc switch.
The –copyonly switch provides a copy backup of the database and logs. This backup is the same as a full backup with the exception of the log truncation portion. This is useful if it is desired to maintain the logs until a future backup cycle or if log truncation is not possible:

```
Exbackup2010 -s ExSolG -vss -copyonly
```

The –vssdiff switch provides a differential backup of the logs. This is done by incrementally synchronizing the log BCV with the production log volume, and then splitting the log volume during the 10 second freeze time. Once this is done, the logs are truncated from the production log volume.

The –scc switch enables Sequential Consistency Check, which is used to run the consistency check sequentially across multiple storage groups’ log files. This is a necessary process if there are log or database files from more than one storage group share a single LUN.

Should there be any issues during this operation it is possible that the disks will be in a temporarily unusable state. This is due to the masking and Hidden flags that are set during the VSS operations.

TF/EIM provides additional command line options that can be used to revert devices to a clean, usable state. The first is the –cleanup switch. This switch will attempt to clean up the environment after a failed backup attempt. It will ensure that the BCVs are split, unmask the devices, and clear any read-only or hidden flags.

In the event you need to work with EMC for support, there is an additional command to assist in obtaining a deeper set of information. The -debugmode switch will provide debug level information for TF/EIM and should be reserved for special scenarios:

```
Exbackup2010 -s ExSolG -vss -debugmode
```

TimeFinder/Mirror is an EMC software product that allows an additional hardware mirror to be attached to a source volume. The additional mirror is a specially designated volume in the Symmetrix configuration called a business continuance volume, or BCV. The BCV is synchronized to the source volume through a process called an establish. While the BCV is established, it is not ready to all hosts. At an appropriate time, the BCV can be split from the source volume to create a complete point-in-time copy of the source data that can be used for multiple purposes, including backup, decision support, and regression testing.
The use of BCVs and TimeFinder/Mirror is generally managed with device or consistency groups. These groups are a method of associating the Standard volumes with the BCVs and controlling the established and split state. This method of management is a good fit for Exchange environments where the VSS framework is used for backup and restore operations. To put the two processes together, TF/EIM manages both processes to create a backup. Further information about device consistency groups can be found in Chapter 2, “EMC Foundation Products.”

The process demonstrated in Figure 42 on page 182 shows the necessary steps required to create a backup of an Exchange Server 2010 using TimeFinder/Mirror and TF/EIM being executed in local mode on the production server:

1. Establish BCVs to Standard devices
2. Execute TF/EIM backup command
   2.1 Validate files and devices for use
   2.2 Create Shadow copy
   2.3 Unmask BCVs to Backup host
   2.4 Import Shadow copy and run ESUTIL

![Figure 42 TimeFinder Exchange Integration Module with TF/Mirror](image)

1. The first action to manage TimeFinder/Mirror pairs is to create a device or consistency group that can be used to manage the devices and add the desired devices to that group

   ```bash
   symdg create <device_group> -type regular
   symld -g <device_group> addall dev -range <devices>
   ```

2. It is then necessary to add the necessary BCV volumes to the devices group

   ```bash
   symbcv -g <device_group> associateall -range <devices>
   ```
3. The next action is to establish the BCVs to the standard devices. This operation occurs in the background and should be executed before the generation of the BCV copy:

```
symmir -g <device_group> establish -full -noprompt
```

**Note:** The first iteration of the establish needs to be a full synchronization and may require the additional usage of the `-exact` option. Subsequent iterations are incremental and do not need the `-full` or `-exact` keywords.

Once the command is issued, the array begins the synchronization process using only Symmetrix resources. Since this is asynchronous to the host, the process must be interrogated to see when it is finished. The command to interrogate the synchronization process is:

```
symmir -g <device_group> -synched verify
```

This command will return a 0 return code when the synchronization operation is complete.

4. When the volumes are synchronized, the TF/EIM backup operation may be executed:

```
Exbackup2010 -s <servername> -vss -o <logfile> -v -preestablish
```

**Note:** It is important to note that TimeFinder/Mirror is the default product used by TF/EIM. More information can be found in the product guide.

---

**Using TimeFinder/Clone with TF/EIM**

TimeFinder/Clone is an EMC software product that copies data internally in the Symmetrix array. A TimeFinder/Clone session is created between a source data volume and a target volume. The target volume needs to be equal to or greater in size than the source volume. The source and target for TimeFinder/Clone sessions can be any hypervolumes in the Symmetrix configuration. This includes business continuance volumes, standard volumes, and volumes where the RAID protection is mismatched (for example, cloning a mirrored Standard volume to a RAID 5 volume). The process for TimeFinder/Clone operation is detailed in Figure 43 on page 184.
The TimeFinder Exchange Integration Module supports clone relationships created with either native clone commands, or in the case of RAID 5 BCV’s, TEIM will utilize the Symmetrix Clone Emulation mode to facilitate backup operations for an Exchange Server database.

Figure 43  TimeFinderExchange Integration Module with TF/Clone

1. The first action to manage TimeFinder/Clone pairs is to create a device or consistency group that can be used to manage the devices and add the desired devices to that group:

   *symdg create <device_group> -type regular*

   *symld -g <device_group> addall dev -range <devices>*

2. The next action is to create the TimeFinder/Clone pairs. It is critical to use the -precopy command and the -diff command when creating the pairs. The -precopy command will begin a background copy of the data prior to the activate command and the -diff command will enable subsequent sessions to be incremental.

   *symclone -g <device_group> create -precopy -diff -noprompt DEV001 sym 1d BCV001*

   TF/EIM requires that the clone is in the created state before it can execute.
Note: Differential Clone functionality is available only for 5671 Symmetrix DMX microcode or later.

3. Once the session is created, the TF/EIM command can be executed:

   Exbackup2010 -s <servername> -vss -clone -o <logfile> -v

Using TimeFinder/Snap with TF/EIM

TimeFinder/Snap enables users to create complete copies of their data while consuming only a fraction of the disk space required by the original copy.

TimeFinder/Snap is an EMC software product that maintains space-saving pointer-based copies of disk volumes using VDEVs and SAVDEVs. The VDEVs contain pointers either to the source data (when it is unchanged) or to the SAVDEVs (when the data has been changed).

TF/EIM provides a way to integrate with the TimeFinder/Snap functionality through the use of the -snap option at the command line.

Note: Use of TimeFinder Snap with Exchange is not generally recommended due to the heavy random activity of Exchange and the sequential read activity of ESEUTIL.

When using this option, the TimeFinder/Snap technology is used to capture the image of the Exchange database.

Unlike TimeFinder/Mirror or TimeFinder/Clone, the snap relationship is created and activated when it is needed. No prior copying of data is possible. The create operation establishes the relationship between the standard devices and the VDEVs, and it also creates the protection metadata.

An important thing to keep in mind with snap technology is that the databases copied using TimeFinder/Snap are subject to a COFW penalty while the snap is activated. The COFW penalty means that if a track is written to the source volume and it has not yet been copied to the snap save area, it must first be copied to the save area before the write from the host is acknowledged.
Figure 44 on page 186 depicts the necessary steps to make a backup of an Exchange Server using TimeFinder/Snap.

**Figure 44 TimeFinder Exchange Integration Module with TF/Snap**

1. The first action is to create the TimeFinder/Snap pairs. The following command creates the TimeFinder/Snap pairings and protection bitmaps. No data is copied or moved at this time:

   ```
symsnap –g <device_group> create -noprompt
   ```

   Unlike TimeFinder/Mirror, the snap relationship is created and activated when it is needed. No prior copying of data is necessary. The create operation establishes the relationship between the standard devices and the VDEVs, and it also creates the protection metadata.

2. After the snap is created, the TF/EIM backup operation may be executed:

   ```
   Exbackup2010 –s <server name> –snap –vss
   ```

**Using Remote Replication with TF/EIM**

There are two types of remote replication scenarios that work with TF/EIM. These are Symmetrix Remote Data Facility (SRDF) and Open Replicator (OR). Both replication technologies are integrated with the product and are easy to use once they have been configured. Due to limitations in the VSS framework, only some scenarios are supported for backup/restore. These are the only scenarios that will
be covered in this chapter. The scenarios that are not supported by VSS are covered in greater detail in Chapter 4, "Backing Up Exchange Server 2010," and Chapter 6, "Understanding Exchange Server 2010 Continuity of Operations."

TF/EIM provides support for a number of SRDF backup scenarios. The supported scenarios for a given version of the TF/EIM product are documented in the *EMC TimeFinder Integration Module Product Guide*. As of the latest version of TF/EIM at the time of this document’s publication, support is provided for BCV Mirroring (STD → BCV/R1 → R2 → BRBCV), Remote BCV (R1 → R2 → RBCV) and Remote Clone (R1 → R2 → RClone) where each configuration is invoked by using the appropriate command line option of –brbcv, -rbcv and –rclone, respectively. Remote BCV and Remote Clone scenarios require the implementation of SRDF/Synchronous.

As an example, the BCV Mirroring is supported when the –brbcv option is used in conjunction with a local R1 BCV. The local R1 BCV is a normal BCV device that can act as a target to back up the standard disk, it can also act as a source in a SRDF relationship. This means that the R1 BCV would first need to synchronize with the production LUN, then split and begin a synchronization with the remote LUN. The path that the backup copy takes in this situation is Standard (STD) → BCV/R1 → Std/R2 → BRBCV as shown in Figure 45 on page 188.
Using this scenario, the backup is able to meet the 10 second split time required by VSS. The backup also maintains a copy at the local site which can be used for immediate local recoveries.

Figure 45  TimeFinder Exchange Integration Module with SRDF

The full command to issue for a BRBCV backup is:

\texttt{Exbackup2010 -s <Server Name> -vss -brbcv}
Replication Manager

RM uses a LAN and SAN to communicate and control storage-based functions. Figure 46 on page 189 shows the RM architecture and the components that reside in various parts of the system. An introduction to each component is provided.

![RM software architecture](image)

**Figure 46** RM software architecture

In RM, an application set is the user-defined set of production data to be replicated. A job is the set of actions that creates a replica of a given application set.

**Inside Replication Manager’s Exchange Agent**

RM uses the VSS Framework to create and restore application consistent replicas of Exchange Server 2010 mailbox databases. The VSS Framework consists of four main components.

- The Volume Shadow Copy Service (VSS)
- Hardware or software providers
- Requestors
VSS writers

The VSS service is included with operating systems, starting with Windows 2003. Exchange Server 2010 includes two VSS writers—the Information Store writer for the active databases and the Replica Writer for the passive databases. The RM Exchange agent performs the requestor and hardware provider roles. It works with VSS and Exchange Server 2010 VSS Writers to create replicas. VSS refers to these replicas as hardware shadow copies.

The RM Exchange agent for Exchange Server 2010 can create two types of VSS shadow copies—full or copy. A full shadow copy creates a copy of the database and log volumes and then truncates the transaction logs, usually after a consistency check has been run. A copy shadow copy creates a copy of the database and log volumes, but does not truncate the logs.

The RM Exchange agent consists of several components. The main component is part of the ermanywriter.dll, which runs as part of the RM client (IRCCD) service. This is the component that works with the Exchange VSS Writers to discover online mailbox databases, create replicas, and restore and recover mailbox databases.

When you install the Exchange agent for Exchange 2007 or 2010, you are asked to provide credentials for the agent. These credentials are needed to install the second component, called the RM Exchange Interface service. This service uses PowerShell APIs to run Exchange cmdlets. The cmdlets are used to gather configuration information about Exchange and to control mailbox databases during restore. This service is also used to run consistency checks using the Exchange Store Consistency Check API. The RM Exchange Interface service uses APIs that are compatible with .NET Framework 2.0. IRCCD and the Exchange Interface service communicate through DCOM. The Exchange Interface service supports both Exchange 2007 and Exchange Server 2010.

A third component was added in RM 5.3.0. It is used to run consistency checks of Exchange 2007 databases and logs when the Exchange Server 2010 Management Tools are installed. This component is an executable called RM_ESM_Util.exe. It is run as needed.
Preparing the Exchange Server 2010 environment

The RM Product Guide has the latest information about how to prepare your Exchange environment for RM. In addition to providing information about the placement of databases and logs on volumes, it discusses security settings and requirements.

The architecture of the Exchange environment has an impact on the capabilities of RM within that environment. Consider the impact of the following decisions when planning how to deploy RM in an Exchange Server 2010 environment.

RM restores at the LUN level, not the file level. It is important to configure database and log files on volumes in a manner that supports the intended restore granularity. If the database and logs are on the same volume, restoring the database file would also cause the logs to be restored. Logs that are not part of the replica would be deleted. If two databases share a volume, RM cannot restore just one database file. Both would have to be restored and recovered.

Another consideration is the use of mount points. VSS has trouble importing replicas that contain nested mount points. If the logs are on a volume like L: and a mount point is created on that volume called “DB_MP” with the .edb file there, it is a nested mount point. The volume L: and the volume “DB_MP”, which is mounted on L:, would be in the same replica. RM would experience VSS errors when importing the replica.

Example of nested mount points

L:\
(one mount point)
L:\DB_MP
(a nested mount point DB_MP mounted to L:\)

There are two ways to mitigate this issue. One is to create a “tiny” LUN and create the L: volume on it. Create two mount points on the volume, one for the database file and one for the transaction logs. The other option is to create the mount points on one of the volumes that are local to the server.

Installing Replication Manager

RM client can be installed in any of the following environments:

- Stand-alone mailbox servers
- Member servers in a native DAG
Member servers in a DAG with third-party synchronous replication enabled (REE with MV/S is supported)

Servers that have the Exchange Server 2010 management tools installed (mount host)

**Prerequisites**

Before installing the RM Exchange agent on an Exchange Server 2010 mailbox server or a mount host, make sure that the following are carried out:

- A user account is created with the required permissions as outlined in the Product Guide.
- Windows Authentication for the PowerShell IIS application as outlined in the EMC RM Product Guide. (Not required on the mount host if just the management tools are installed) is enabled.

The Exchange agent will not install or work properly if these steps are not completed correctly. The *EMC Replication Manager Product Guide* provides more information.

RM installs native agents on Windows 2008 servers. Since Exchange Server 2010 is only supported on x64 editions of Windows 2008, Replication Manager will install the native x64 version of the Exchange agent. This means that the x64 version of Solutions Enabler needs to be installed. The ESM provides more information.

**Native DAG**

Install RM on the servers that are planned to be used for replicating Exchange databases. It is not currently required to install it on all member servers.

When adding hosts to the RM, the virtual network name for the DAG in addition to all the servers in a DAG, needed to be added, if the jobs are to run on different servers. RM can use this DAG virtual network name to locate the active copy of the database if a member server is down or unreachable for some reason.

**Protecting mailbox databases**

RM can be used to create replicas of the Exchange Server 2010 mailbox databases. RM can replicate:

- Databases on stand-alone server
- Active and passive databases on native DAG member servers

**Creating application sets**

RM uses application sets to define which Exchange Server 2010 databases to replicate. When creating an application set, a mailbox server is selected. RM displays a list of databases available on that
server. RM uses a combination of information retrieved from the Exchange VSS Writers and from Exchange PowerShell cmdlets to create the list of databases

- For stand-alone servers - RM lists all databases mounted on the selected server. A warning message is displayed for any databases that are not mounted.
- For native DAG member servers - RM lists all of the databases on the selected server. It does not distinguish between active and passive databases. A warning is displayed for any database that is not mounted somewhere in the DAG or if the passive copy is not healthy.

**Creating jobs**

Jobs define the type of replica that RM should create, whether or not to mount it, and when to delete it. A job is associated with an application set and it runs on the server that was selected in the application set. Adding a mount action to a job is not required, but is considered best practice for Exchange Server 2010 jobs, so a consistency check of the database and log files can be run.

When a job is created, the first thing RM does is contact the production server to discover the location of the database and log files. It determines the type of storage array that the files reside on—CLARiiON, Symmetrix, Celerra, RecoverPoint, etc. Then the job wizard is tailored for that type of storage and for the version of Exchange.

The *EMC Replication Manager Product Guide* provides online help for details on creating jobs for Exchange Server 2010. These sources explain the various options for replication and mount as well as how to link jobs, create copy jobs, and schedule jobs.

**Running jobs**

When RM runs a job, it first verifies that the Exchange database is in a state that it can replicate. If any of the checks fail, the job fails. RM:

- Verifies that circular logging is disabled.
- Checks if the database and logs are on the same volume. If the option to enable support for database and logs on the same volume is selected, RM will log an informational message. Otherwise it will log an error and fail the job.
- Verifies that the database copy is healthy if the copy is a passive database.
Verifies that the database is mounted. VSS requires that the database be mounted. In a native DAG, the database has to be mounted on one of the servers in the DAG, not necessarily on the server RM is using to create a replica.

Once verification passes, RM discovers the current location of the database and log files and checks the application event log for JET database errors (-1018, -1019, -1022, and so on), which would indicate damage to a database or log file. If there are no errors, RM creates the replica. The steps for creating a replica are as follows:

1. RM selects the target storage.
2. It prepares storage to create a VSS hardware shadow copy. For example, if the target storage is a clone, it synchronizes the clones.
3. It issues a VSS command to create the shadow copy.
4. VSS instructs the Exchange Writer to prepare for the shadow copy.
5. A 60-second timer starts for the Exchange Writer. Within that window, the following activities must complete:
   6. Database I/O freeze
   7. Shadow copy created
   8. I/O has to thaw
   9. The Exchange Writer freezes the write operations to the database and transaction logs. (Read access is still allowed.)
10. VSS instructs the hardware provider (ERM VSS Provider) to prepare to create the shadow copy.
11. A 10-second timer starts. Within that window, the following activities must complete:
   12. I/O to the disk must be flushed and suspended
   13. The shadow copy must be created
   14. I/O must resume
   15. VSS flushes data to the disk and holds writes.
   16. The ERM VSS Provider creates the shadow copy by splitting the LUNs or creating the snap.
   17. VSS resumes writes to the disk.
18. Exchange resumes writes to the database and transaction logs.

19. RM mounts the replica using VSS to import it.

20. The Exchange agent runs the consistency check against the database and logs.

21. If the mount and consistency check are successful, the Exchange agent tells Exchange that the shadow copy was successful. Otherwise the agent tells Exchange that the shadow copy failed.

22. If the Full consistency method is selected and the shadow copy was successful, Exchange truncates the transaction logs. In a native DAG, Exchange does not truncate logs until they have been replayed into the database copies and the log truncation delay time has passed.

---

**Considerations for DAGs**

When working with RM in a DAG, there are a couple of things to consider. One is the placement of database and log files and the other is the creation of application sets.

RM supports configurations that place the database and logs on the same volume, but this should only be considered when a database has more than one copy. There is an option when you create an application set that enables this support. Because restore is at the LUN level, when the database and log files are placed on the same volume, restore options are limited.

Because failover is at the database level, consider configuring application sets with only one database each. If there is more than one database in the application set and one of the databases fails and moves to another server, jobs will fail. In native DAGs, all of the database copies in an application set have to be healthy on the same server or jobs will fail.

**Considerations for native DAGs**

Jobs run on the server selected in the application set. On servers in a native DAG, jobs replicate active and passive databases, as long as the status is “mounted” or “healthy”. If the status of the copy on the selected server becomes “failed,” jobs will fail. Consider creating application sets and jobs for a second server in case this situation occurs. Create a schedule and disable it for this second set of jobs. If a database fails on the first server, disable the schedule for the failing job and enable the schedule for the job on the second server.
A replica can be mounted as part of the job or it can be mounted on demand. It is considered best practice to mount an Exchange replica as part of the job to run consistency checks against the database and logs. It is also common practice to back up the database, log, and VSS metadata files to long-term storage after the consistency check succeeds.

An Exchange replica can be mounted at any time to recover an individual mailbox or to import data from a replica that was backed up to long-term storage.

In general, RM can mount an Exchange replica to any of the following environments:


In stand-alone configurations, RM can also mount to:

- The same production stand-alone mailbox server

In native DAG configurations, RM can mount to a stand-alone server or:

- A server in another native DAG
- A separate server in the same native DAG
This chapter includes the following topics:

- Introduction ................................................................. 198
- EMC Consistency technology and restore ......................... 199
- Exchange Server restore functionality ............................. 200
- Integrating EMC technologies with Exchange Server recovery 203
- TimeFinder Exchange Integration Module .......................... 204
- Replication Manager ....................................................... 213
**Introduction**

Recovering a production database is an event that all Exchange administrators hope is never required. Nevertheless, they must be prepared for unforeseen events such as hardware failures or user errors, which require database recovery operations. The keys to a successful database recovery include:

- Identifying database recovery time objectives
- Planning the appropriate recovery strategy based upon the backup type (full, copy)
- Documenting the recovery procedures
- Validating the recovery process

Microsoft Exchange Server database recovery depends upon the backup methodology used. With the appropriate backup procedures in place, an Exchange Server database can be recovered to any point in time between the end of the full backup and the point of failure using a combination of a full backup, any incremental transaction log backups, and any other log files that may be available sequentially past the last backup.
EMC Consistency technology and restore

Recovery typically involves copying the previously backed up files into their appropriate locations and, if necessary, performing recovery operations to ensure that the database is consistent. To mitigate the amount of data loss sustained to a production database, the first step is to ensure that a complete chain of database backups and incremental log backups are maintained.

This section assumes that EMC technology has been used in the backup process as described in Chapter 4, “Backing Up Exchange Server 2010.”

EMC provides a number of solutions built around a foundation of Consistency technology. Using this technology, it is possible to create valid restartable images of an Exchange Server database environment using dependent-write principles. These styles of solutions are described in Chapter 3, “Creating Exchange Server 2010 Clones.”

It is entirely possible to utilize the images created through Consistency technology to facilitate the ability to create valid restart points. It is also possible to stream these images to tape or other media for longer term storage and subsequent restoration. However, while these images represent valid restart points, they do not facilitate any of the standard Exchange Server recovery methods provided by the streaming API and the VSS Framework that are described in the remainder of this chapter.

The Recovery Point Objective (RPO) of a restartable image is based on a static point in time, and will increase as changes to the source production environment proceed. It is only when the next point of consistency is created that the RPO can be reset to its initial state. During intervening times between cycles, the RPO point will vary, it is important to control the cycle of such a process to ensure that the solution falls within the maximum RPO.
Exchange Server restore functionality

First it is important to understand the difference between a restoration of service and a restoration of data. Restoration of service means that the service, in this case the ability to send and receive e-mail is restored and users are able to access that service. Restoration of data means that the data, in this case the e-mail stored on Exchange is restored and users are able to open previously received e-mails, access stored attachments, and so on. EMC has several options that work with Exchange to help administrators restore the e-mail data.

Microsoft Exchange Server provides a number of standard interfaces to facilitate database restore processing. The VSS Writer is the only method in use today for Exchange Server 2010 and is where the most development is occurring. Integration with VSS restoration will be discussed in depth.

Types of Exchange Server restores

Exchange Server 2010 supports two basic types of restores: point-in-time and roll forward. Depending on which of these two options is chosen, the amount of data and work to complete the restore process will vary.

Point-in-time restores

A point-in-time restore is a restore of data to a specific point in time. Generally this is associated with a backup image that was taken at a specific date and time through either the Exchange backup streaming API or the VSS Framework.

Roll forward restores

A roll forward restore is a restore that starts at a point in time and is subsequently moved forward to a later point in time. Generally, this is done to revert a database copy that was restored, back to the most current point in time available, and is achieved through applying transaction logs.

As an example, at 7 A.M. a backup of the Exchange logs and database is taken and stored on disk. At 8 A.M. the production copy of the database suffers a critical failure rendering the database unusable.

In, the default process of restoring the backup is depicted. This restore operation puts the database that was backed up at 7 A.M. back into place. In this scenario, all of the e-mail items that were delivered between 7 A.M. and 8 A.M. are lost and are not retrievable.
In some scenarios this is necessary; for example, a scenario where there was a large outburst in SPAM or virus infected e-mail messages that consumed the sum of all mail delivered.

Figure 47 Exchange Server point in time restore

In Figure 48 on page 202, the default process of restoring the backup is depicted, this time the logs that were created between 7 A.M. and 8 A.M. are copied to the location of the restored files and all mail that was successfully delivered (recorded as transactions within the logs) during that time will be replayed out of the log files and into the database. In most scenarios this is the desirable restoration procedure as it ensures all mail that was intended to be delivered is still in a delivered state.
Figure 48  Exchange Server roll forward recovery
Integrating EMC technologies with Exchange Server recovery

Disk mirror-based functionality of storage arrays can significantly decrease the amount of time required to execute restore operations. As documented within Chapter 2, EMC provides a number of products that can be used to facilitate restore operations.

Because disk mirror restore operations will necessitate the reverse synchronization of the mirror device back to the standard devices, a number of issues need to be addressed:

- Restoration of this type at the LUN level cannot be an online operation for the Exchange objects being restored (database or storage group, depending on granularity).
- The LUN must be dismounted before the reverse synchronization (restore).

This ensures that Windows does not retain a pre-restored view of the specific LUNs. Should a restore be executed with the LUN mounted, Windows will mark the volume as questionable, and possible data corruption may result. In all circumstances, it is necessary to follow the guidelines provide by the specific EMC product being used to execute the restore to avoid issues during the recovery process.

VSS integration

TF/EIM and RM are the two main applications that EMC provides to facilitate backup and restore using the VSS framework. Additionally, EMC provides applications that use the VSS framework as a part of their functionality but are not strictly backup/restore applications.

TF/EIM and RM are covered in-depth in this version of the TechBook.
TimeFinder Exchange Integration Module

TF/EIM is one of EMC’s tools that can be utilized within an Exchange backup, restore, and operational continuity plan to provide backup images of the Exchange databases. TF/EIM is a command line tool that easily integrates with other applications like the VSS framework, Solutions Enabler, and a variety of other backup applications. TF/EIM solves the simple problem of running VSS-based backups/restore for Exchange Servers against an EMC Symmetrix storage platform. This is a problem that is also addressed by RM, which is discussed in the RM product guide.

In general, TF/EIM does this by matching the disk replication technology built into the Symmetrix with the needs of the Exchange application and the requirements of the VSS framework. This section will discuss how TF/EIM matches the TimeFinder replication, which is the internal replication family that includes TimeFinder/Mirror, TimeFinder/Clone, and TimeFinder/Snap with both the Symmetrix Remote Data Facility (SRDF) capabilities and the Exchange 2xxx VSS restore capabilities.

Planning to use TF/EIM

Prior to using TF/EIM as a part of a backup, restore, or continuity plan, certain considerations and planning must take place. “Planning to use TF/EIM” on page 178 provides more information on planning the environment.

It is also recommended that the EMC TimeFinder Integration Module Product Guide is consulted and the “Setting up a production server for TF/EIM” section is thoroughly reviewed prior to any TF/EIM deployment.

Implementing restores with TF/EIM

To initiate a restore using TF/EIM there are a number of ways to compose the command line options. The options used are primarily dictated by the variables decided on as shown in Chapter 4, “Backing Up Exchange Server 2010.” The most important of these is using the proper version of the command. Exchange Server 2010 should be restored using the command ExBackup2010.

The technology required to perform a restore is dependent on that technology used to backup the Exchange Server. Refer to Chapter 5, “Restoring and Recovering Microsoft Exchange Databases,” for more information on each of these technologies.
When using TF/EIM to perform the restore, it is important to keep in mind that TF/EIM walks through the steps with some portions done in an automated fashion and some portions done in a manual fashion.

To perform a restore using VSS and TimeFinder for Exchange Integration Module, it is required that the restore take place at the Exchange storage group or database file level. The complete command to run a server restore is:

```
ExBackup2010 -s <servername> -mb <store> -vss restore -bcd <BCDXMLFile> -wmd <WMLXMLFile> –v
```

This command specifies a backup of the entire server and will be a point-in-time restore of the databases. The bcd and wmd files should be the files that were created along with the backup data.

If it is required to execute a point-in-time restore, add the `-logs` option to the command. Use of this option indicates restoration of both the databases and transaction log volumes created during the backup. Normally, the `-logs` option should not be included, since without it, the existing transaction logs will be used for the storage group to roll forward all databases from their restored state to the end of the last valid log.

When restoring individual databases, the logs cannot also be restored. This is due to the fact that logs are per storage group and cannot be separated. To restore individual databases, the command needs to be run per storage group and each database should be specified individually:

```
ExBackup2010 -s <Server Name> -mb <Store> -vss restore -bcd <BCDXMLFile> -wmd <WMLXMLFile> –v
```

As discussed in “Using TimeFinder/Mirror with TF/EIM” on page 181, TimeFinder/Mirror is an EMC software product that allows an additional hardware mirror to be attached to a source volume. The additional mirror is a specially designated volume in the Symmetrix configuration called a business continue volume, or BCV. The BCV is synchronized to the source volume through a process called an establish. While the BCV is established, it is not ready to all hosts. At an appropriate time, the BCV can be split from the source volume to create a complete point-in-time copy of the source data that can be used for multiple purposes, including backup, decision support, and regression testing.
This point in time copy can be restored to the standard volume by following the same process in reverse as depicted in Figure 49 on page 206:

**Figure 49 TimeFinder/Mirror restore using TF/EIM**

1. The first step in the restore process will be to run the ExBackup2010 restore command:

   ```bash
   exbackup2010 restore -vss -v -s <ServerName> -bcd <BCDXMLFile> -wmd <WMLXMLFile> -mb <Store>
   ```

2. Next, the ExBackup2010 command will begin a series of active prompts at the command line. The first prompt will be to dismount the corrupt Exchange database(s). To do this, use Exchange System Manager and right-click the database that will be restored and select the “dismount” option.

3. Continue to follow the prompts to dismount all the STDs and BCVs on the Production and BCV hosts. The SIU command to umount when using mount points is:

   ```bash
   symntctl umount -path <Mount_point>
   ```

4. Initiate restoration of the required volumes for the databases. It is possible to execute a protected restore operation, or a standard restore. In the event that a standard restore is initiated, it will be necessary to wait until the restore is complete and then split the
Restoring and Recovering Microsoft Exchange Databases

devices prior to using them, as in this case, updates to the STD volumes will be propagated to the BCV. In the case of protected restore, the NCV devices are not updated, and the STD volumes are available immediately. Although it is necessary to eventually split the BCV from the STD device once the entire protected restore is complete.

Standard restore command:

```
symmir -g <devicegroup> restore –noprompt
```

Protected restore command:

```
symmir -g <devicegroup> restore –protect –noprompt
```

5. If using the standard BCV restore operation, the BCVs need to be split before moving on. To check the restore state it is possible to use the following command. It is also possible to use this command to verify the state of a protected restore, although it is not necessary to wait for a protected restore to complete before proceeding (it is necessary to ultimately split:

```
symmir –g <devicegroup> verify –restored
```

Once the restore process has been completed, it is necessary to split the BCV devices from the standards by using the following command:

```
symmir -g <devicegroup> split
```

6. TF/EIM will finish the restore process and clear the Read-Only flags on the STD devices. TF/EIM will then prompt again to have the BCVs remounted to the BCV mount host.

7. Finally, TF/EIM will prompt you to remount the Exchange databases before completing.

Using TimeFinder/Clone with TF/EIM

As stated in “Using TimeFinder/Clone with TF/EIM” on page 183, TimeFinder/Clone is an EMC software product that copies data internally in the Symmetrix array. A TimeFinder/Clone session is created between a source data volume and a target volume. The target volume needs to be equal to or greater in size than the source volume. The source and target for TimeFinder/Clone sessions can be any hypervolumes in the Symmetrix configuration. This includes business continuance volumes, standard volumes, and volumes where the RAID protection is mismatched (for example, cloning a mirrored Standard volume to a RAID 5 volume). The process for
TimeFinder/Clone operation is detailed in Figure 43 on page 184. The TimeFinder Exchange Integration Module utilizes the Symmetrix Clone Emulation mode to facilitate backup operations for an Exchange Server database. Thus, all TimeFinder/Mirror operations are mapped transparently to TimeFinder/Clone operations. If the –differential switch was used during the backup process, no specific switch will be required to specify a differential or a full restore. If the –differential switch was not used, the –full switch will be required to perform a full restore. In both cases the original session will be terminated and a new session will be created for the restore. This means that if you want to perform a new backup afterwards, a new session will need to be created. The process is straightforward as depicted in Figure 50 on page 208:

Figure 50  TimeFinder/Clone restore using TF/EIM

1. The first step in the restore process will be to run the ExBackup restore command:

   exbackup2010 restore -vss -v -s <ServerName>
   -bcd <BCDXMLFile> -wmd <WMLXMLFile> -mb
   <Store>
2. Next, the ExBackup2010 command will begin a series of active prompts at the command line. The first prompt will be to dismount the corrupt Exchange database(s). To do this, use Exchange System Manager and right-click the database that will be restored and select the “dismount” option.

3. Continue to follow the prompts to dismount all the STDs and BCVs on the Production and BCV hosts. The SIU command to unmount when using mount points is:

   `symntctl umount -path <MountPnt>`

4. Restore all of the required clone devices, using the command:

   `symclone -g <Device Group> restore <Restore Target> sym ld <Clone Device>`

5. Once the data has completely been restored, the clone devices need to be split before moving on:

   `symclone -g <Device Group> split <Restore Target> sym ld <Clone Device>`

6. TF/EIM will finish the restore process and clear the READONLY flags on the STD devices. TF/EIM will then prompt again to have the BCVs remounted to the BCV mount host.

7. Finally, TF/EIM will prompt you to remount the Exchange databases before completing.

**Using TimeFinder/Snap with TF/EIM**

As described in “Using TimeFinder/Snap with TF/EIM” on page 185, TimeFinder/Snap enables users to create complete copies of their data while consuming only a fraction of the disk space required by the original copy. When using this option, the TimeFinder/Snap technology is used to capture the image of the Exchange database. Unlike TimeFinder/Mirror or TimeFinder/Clone, the snap relationship is created and activated when it is needed.

The restore process is still easy to accomplish as shown in Figure 51 on page 210 and described next:
1. The first step in the restore process will be to run the Exbackup2010 restore command:

   Exbackup2010 restore -vss -v -s <ServerName> -bcd <BCDXMLFile> -wmd <WMLXMLFile> -sg <Store>

2. Next, the Exbackup2010.EXE command will begin a series of active prompts at the command line. The first prompt will be to dismount the corrupt Exchange database(s). To do this, use Exchange System Manager and right-click the database that will be restored and select the “dismount” option.

3. Continue to follow the prompts to dismount all the STDs and BCVs on the Production and BCV hosts. The SIU command to unmount when using mount points is:

   symntctl umount -path <MountPnt>

4. Restore all of the snap devices, using the command:

   symsnap restore <Restore Target> vdev 1d <Virtual Device>

5. Once the data has been restored, the snap devices need to be split before moving on:
6. TF/EIM will finish the restore process and clear the READONLY flags on the STD devices. TF/EIM will then prompt again to have the BCVs remounted to the BCV mount host.

7. Finally, TF/EIM will prompt you to remount the Exchange databases before completing.

Using remote replication with TF/EIM

There are two types of replication scenarios that work with TF/EIM. These are Symmetrix Remote Data Facility (SRDF) and Open Replicator (OR). The scenarios that are not specifically supported by VSS are covered in greater detail in Chapter 3, “Creating Exchange Server 2010 Clones,” and Chapter 6, “Understanding Exchange Server 2010 Continuity of Operations.”

To facilitate a restore of a remotely replicated backup using Symmetrix Remote Data Facility (SRDF) with TF/EIM, it is necessary to follow the guidance provided in the EMC TimeFinder Integration Module Product Guide. The following example details the steps required to facilitate a restore operation for a BCV Mirroring scenario.

In the BCV Mirroring scenario, the disks must first be restored across their links using TimeFinder and SRDF commands. Once the data has been restored to the local BCV, the standard restoration techniques can be used:

1. Dismount the BRBCV, R2, R1-BCV, and the standard volume.
2. Initiate the restore of the BRBCV by issuing the following command:
   \`
   symmir -g <device group> restore -rdf -bcv
   \`
3. Once the BRBCV has been fully restored, the BRBCV should be split by using the following command:
   \`
   symmir -g <device group> split -rdf -bcv
   \`
4. Next, the R1/BCV will need to be restored from the R2 with the following command:
   \`
   symrdf -g <device group> restore -bcv
   \`
5. Once the restore of the R2 has been completed, the R2 will need to be split with the following command:
   \`
   symrdf -g <device group> split -bcv
   \`
6. In this state is it possible to execute the appropriate TF/EIM to execute the restore:

```bash
Exbackup2010 restore -vss -v -s <ServerName> -bcd <BCDXMLFile> -wmd <WMLXMLFile> -mb <store>
```

Complete the steps as detailed in “Using TimeFinder/Mirror with TF/EIM” on page 205 as this now becomes a TF/EIM restore from TimerFinder/Mirror devices.
Replication Manager

It is critical that RM users are able to recover their Exchange Server 2010 data from a RM replica. This section describes how to recover Exchange Server 2010 data from a disk replica.

RM supports restoring and recovering Exchange Server 2010 stores and transaction logs. In version 5.3.2 and later, RM supports creating, mounting, and restoring replicas of the Exchange Server 2010 transaction log and system file volumes. This type of replica is a Differential replica and it does not truncate logs. Restoring just the transaction log and system file volume(s) from a complete replica (Full or Copy) was also added in an earlier version. Having the flexibility of restoring databases from one replica and logs from another is now possible with RM.

The high-level steps that occur during restore and recovery are:

1. RM dismounts all databases in the selected stores.
2. RM sets the databases for override by restore.
3. RM sends a list of components (stores or log directory) to the Exchange Store Writer.
4. The Exchange Store Writer initiates its pre-restore checks.
5. RM restores the volumes on which the selected components reside and communicates with the Exchange Store Writer when the restore is complete.
6. The Exchange Store Writer creates or updates the EnnRestore.env file, checks for gaps in the log sequence, plays the log files back, and then deletes the EnnRestore.env file.
7. RM mounts the databases.

When using replicas created by RM to recover Exchange data, the most common recovery scenarios are:

- Recovering a store.
- Recovering a database up to the failure point.

The recovery scenario that is right for your situation depends on the nature of the failure. Some possible scenarios are discussed next.
Restoring and Recovering Microsoft Exchange Databases

Recovering mailbox databases

Replication Manager can restore Exchange Server 2010 replicas created from the following configurations:

- Databases on a stand-alone server
- Active and passive databases on native DAG member servers

Two types of restores can be performed with RM:

- Point-in-time of the replica
- Roll-forward

By restoring the database and log volumes, an Exchange Server 2010 database can be restored to the point-in-time that a replica was created. Or if just the database volume can be restored while preserving the logs on the log volume. The logs are played into the restored database when the database is recovered. This is referred to as a roll-forward restore.

It is important to remember that when RM performs a restore, it is at the LUN level. This means that the data on the production LUN will be overwritten with the data from the replica. Any newer data will be lost.

Restoring a database to the point-in-time of a replica

This section describes how to restore and recover a database on a stand-alone mailbox server. This type of restore is used when the data on the production database and log volumes becomes corrupt or when there is not a contiguous set of transaction log files (which can occur when restoring from an old replica).

Use the RM Restore Wizard to select the restore of the entire database (the database file and the logs). This causes the database and transaction log volume to be completely overwritten. Both the database and log volumes can be restored and any logs that are newer than the replica creation time are lost. After the database is recovered, it will be at the point in time of when the replica was created.

Note: This type of restore essentially deletes newer log files. It results in data loss.

To perform a restore of the database and log volumes:

1. An Application Set in the Successful state and select the Restore option.
2. RM’s Restore Wizard walks you through the steps to restore the selected replica. Figure 52 on page 215 shows the Replica to Restore dialog where the specific replica is selected.

![Replica to Restore](image)

**Figure 52** Restore Wizard—Replica to restore

3. Select the replica for restore as shown in Figure 53 on page 216. Normally this is the most recent replica.

4. Click Next to access the Objects to be Restored dialog.
5. Select the database for restore. When the database is selected, the database file and logs will also be selected indicating that all components of the database will be restored. If there are two databases on the same volume, Replication Manager automatically selects them both.

6. Click Next to access the Restore Options dialog.
7. There are two options, Recover and mount databases and Activate databases before restore. In an Exchange Server 2010 DAG environment, replicas can be restored only to the same server on which the replica was originally created. Microsoft restricts restores of database copies to the active copy only. If the server is not hosting the active copy of the database. To do this, select the Activate databases before restore checkbox in the Restore Options screen of the Restore Wizard.

This option indicates that VSS should replay logs and recover the database and that RM should mount the database.

If the Recover and mount databases option is not selected, the database must manually mounted and recovered. The EMC Replication Manager Product Guide provides details of how to do this.
8. Click **Finish** in the Restore Options dialog box. A warning about overwriting existing data is displayed. Note the suggestion to restore only the databases and replay the logs separately, which is the Roll-Forward restore scenario.

9. Click **Yes** to start the restore operation.

10. RM begins the restore process. From the job progress dialog RM can be monitored as to what is happening.

**Overview of restore and recovery**

An overview of the steps performed during an Exchange Server 2010 database restore and recovery are as follows:

1. RM uses an Exchange PowerShell cmdlet to dismount all of the databases selected for the restore operation.

2. Using another cmdlet, RM marks the database for overwrite by restore.
3. RM initializes the VSS interface for restore by providing the information in the Backup Components Document and the VSS Writer metadata file that were saved when the replica was created.

4. The VSS Exchange Store Writer performs pre-restore checks, including the following checks:
   - Confirms that none of the databases selected for restore are mounted
   - Confirms that circular logging is disabled
   - Confirms that all selected databases are marked for overwrite
   - Confirms that the database GUIDs selected for restore exist in Active Directory. This is important when a database has been deleted and recreated since the replica was created. A new GUID is given to the recreated database and RM has to direct the restore to the new GUID.

5. RM restores the volumes for the selected databases and logs and then notifies the VSS Exchange Store Writer that the restore is complete.

   If the Recover and Mount option was selected, the VSS Exchange Store Writer performs these additional steps:

6. Creates the EnnRestore.env file in the log directory. The EnnRestore.env file contains information about the database being restored. Enn is the log file base name.

7. Checks for gaps in the log sequence.

8. Plays any required logs into the database and brings the database to a clean shutdown state.


   After that, RM mounts the databases

Restoring a mailbox database to a native DAG is more complex. In a native DAG environment there are multiple copies of a database, one active copy and one or more passive copies. RM can replicate either active or passive database copies; however, Exchange Server 2010 can only support restores to the active copy. RM can only restore to the server that created the replica. So, if the passive copies of database are being replicated, that copy of the database has to be activated before a restore can be performed.
RM has an option to automatically activate the database copy before the restore, if it is currently a passive copy. This is the Activate database before restore option that is part of the Restore Options dialog box. This is present only when RM is performing a restore where the current server is part of a DAG environment. The option details will be shown in the restore dialog boxes presented later on.

When using the RM Restore Wizard to perform a restore of a database in DAG environment, the operation is initiated in the same manner as a restore for an Exchange Server 2010 stand-alone server, select a replica and the restore operation.

1. The Restore Wizard dialog boxes are shown in the examples below

   - The **Replica to Restore** dialog box allows selection of which replica to restore.
   - The **Objects to be Restored** dialog box allows selection of the databases required for restore.
   - The dialog shows which databases are active and which are passive.
   - The **Restore Options** dialog box contains the additional Exchange Server 2010 DAG related option, Activate databases before restore checkbox. The dialog is shown in the Restore options dialog box in Figure 56 on page 221.
   - As with an Exchange Server 2010 stand-alone restore, when the restore logs option is selected, RM displays the same warning message, indicating that the restore operation will erase all current files on the log volume.
2. Once the DAG restore is started, RM performs the following verification steps:

   - Verifies that the current server is the same server that created the replica.
   - Verifies that the database copy is active on the server where the restore is running.
   - If the database is not active RM checks the Activate database before restore flag. If the flag is selected, RM attempts to activate the database. If the activate fails for any reason, or the flag is not selected, the restore fails.
   - Verifies that all of the passive database copies are in a Healthy state.

3. RM uses PowerShell cmdlets to perform the following operations:

   - Dismounts all of the databases selected for restore.
   - Suspends replication to all of the passive database copies for the database being restored.
   - Marks the database for overwrite by restore.
Restoring and Recovering Microsoft Exchange Databases

- Initializes the VSS interfaces for restore by providing the information in the Backup Components Document and the VSS Writer metadata file that were saved when the replica was created.

4. The VSS Exchange Store Writer performs pre-restore checks. Some of the things the writer verifies include:
   - Ensures that none of the databases selected for restore are mounted.
   - Ensures that circular logging is disabled.
   - Ensures that all of the selected databases are marked for overwrite.
   - Ensures that the database GUIDs selected for restore exist in Active Directory.
   - This is important when a database has been deleted and re-created since the replica was created. A new GUID is given to the re-created database and RM has to direct the restore to the new GUID.

5. RM restores the volumes for the selected databases and logs and then notifies the VSS Exchange Writer that the restore is complete.

6. If the **Recover and Mount** option was selected, the VSS Exchange Store Writer performs these operations:
   - Creates the EnnRestore.env file in the log directory. The EnnRestore.env file contains information about the database being restored. Enn is the log file base name.
   - Checks for gaps in the log sequence
   - Plays any required logs into the database and brings the database to a clean shutdown state.
   - Deletes the EnnRestore.env file.

7. RM then mounts the Exchange databases.

When the RM restore is complete, the replication of the DAG databases is always left in the suspended state. This is to allow the user to determine how to best proceed in restoring replication to the passive copies of the DAG database. Normally, when the database and log volumes are restored the database copies will require reseeding.
Restoring a database using roll-forward

In addition to point-in-time restores, RM can restore just the database volume. This is considered a roll-forward restore, since the Exchange log volumes are not restored. After the restore of the database volume, Exchange replays the newer logs into the restored database. This type of restore can be run on a stand-alone server or a native DAG member server. Roll-forward restores are used to restore from the most recent replica and then recover the data in the current transaction log files.

The steps involved in a roll-forward restore are similar to a point-in-time restore; however, in the Restore Wizard Objects to Restore dialog box, only the database files are selected.

Recovering individual mailboxes

Recovery of an Exchange Server 2010 mailbox is done using a recovery database (RDB). A recovery database is a non-standard database that allows an administrator to mount a restored mailbox database and extract data from the restored database as part of a recovery operation. Mounting the recovered data as a recovery database allows an administrator to restore individual mailboxes or specific items from a mailbox.

A recovery database is different from a standard mailbox database. These differences introduce limitations to their use:

- An RDB is created using the Exchange Management Shell.
- Mailboxes in a RDB cannot be connected directly to user accounts. To allow user access data in an RDB mailbox, the mailbox must be merged into an existing mailbox or exported to a folder.
- An RDB is used for recovering only mailbox database data—Public Folder data cannot be recovered from an RDB.
- Only one RDB can be mounted at any time on a Mailbox server. The use of an RDB does not count against the 100 database limit on a Mailbox server.
- Mailbox database copies of an RDB cannot be created.
- An RDB can be used as a target for restore operations, but not backup operations. However, RM does not support restores to an RDB.
- A recovered database mounted as an RDB is not tied to the original database in any way.
- The Target mailbox database must reside in the same Active Directory forest as the database mounted in the RDB.
Scenarios for using recovery database to recover data

- The logical information about the original database and its databases must be intact and unchanged in Active Directory.

The following scenarios describe typical ways that users recover data using recovery databases:
- The recovery of a single mailbox or mailbox database.
- The recovery of a database on a server other than the original server for that database. If necessary the recovered database can be merged back to the original database.
- Recovering deleted items that users previously deleted from their mailbox, after the deletion retention period for that item has expired.

Mailbox recovery steps

Follow these steps to use RM to mount a replica containing an Exchange Server 2010 database and then recover that database as a recovery database.

1. Create a recovery database.

The recovery database is created with the following Exchange Management Shell command:

```powershell
New-MailboxDatabase -Recovery -Name <RecoveryDatabaseName> -Server <ExchangeServerHostName> -EdbFilePath <DatabaseFilePath> -LogFilePath <LogFilePath>
```

where:

- **RecoveryDatabaseName** - A unique name for the recovery database
- **ExchangeServerHostName** - The name of the Mailbox server where the recovery database is to be created
- **DatabaseFilePath** - The full path to the database file location for the recovery database. Use the path where RM will mount the database volume. This file path should include the database name and .edb extension.
- **LogFilePath** - The full path to the log file location for the recovery database. Use the path where RM mounts the log volume.

For example:

If the production logs are mounted to \Logs and RM is mounting the replica to \RDB, the LogFilePath should be \RDB\NDrive\Logs.
The cmdlet to create the recovery database would be:

```
[PS] C:\>New-MailboxDatabase -Recovery -Name RDBTest -Server lrmf158 -EdbFilePath "C:\RDB\NDrive\db\LRMF158 CLAR DB on N and O.edb" -LogFolderPath "C:\RDB\ODrive\Logs"
```

2. Delete the folders created under the mount point.

When the Exchange recovery database is created, database and log folders are created under the mount point. These must be deleted for the RM mount to succeed. So, expanding on the previous example, you would delete the N Drive and O Drive folders.

3. Mount the RM replica as shown in Figure 57 on page 226.

Select and mount the RM replica containing the mailbox database that requires recovering:

- Specify the Exchange Server where the RDB was created as the mount host.
- Use the same mount path used to create the Exchange recovery database.
- Clear the **Mount the replica volumes read only** checkbox.
4. After the mount, the recovery database is in a dirty shutdown state because we are using a replica created with the VSS framework.

5. Use Exchange ESEUTIL to recover the database to a clean shutdown state by performing a soft recovery of the recovery database. Open a command window and type the following command:

```bash
eseutil /r E<nn> /d <PathToEdbFile> /l <PathToLogFiles> /s <PathToLogFiles>
```

where:

- `<nn>` - The Log file prefix
- `<PathToEdbFile>` - The path to the Exchange recovery database file. Do not include the .edb file extension.
<PathToLogFile> - The path to the log files and checkpoint file that was specified when the Exchange recovery database was created.

6. Allow the database to be overwritten by a restore. Use the following Exchange Management Shell cmdlet:

   Set-MailboxDatabase -Identity <recoverydatabase> -AllowFileRestore $true

7. Mount the recovery database into the Exchange Server by running the following Exchange Management Shell cmdlet:

   mount-database -identity <recoverydatabase>

8. Confirm that the recovery database is properly mounted by running the following Exchange Management Shell cmdlet, as shown in Figure 58 on page 227:

   Get-mailboxDatabase -Identity <recoverydatabase> -status | fl Name, Server, Mounted, Recovery

Figure 58 Get-MailboxDatabase example

Verify that the mailbox is in the recovery database by using the following Exchange Management Shell cmdlet to list the mailboxes and their item count:

   Get-MailboxStatistics -Database <recoverydatabase>

   For example, the contents of the RDB1 recovery database are shown in the output of Figure 59 on page 228.
9. Recover the mailbox data. In order to do this, you need Organization Management rights in Exchange Server 2010. The contents of an individual mailbox can be recovered by running the Restore-Mailbox Exchange Management Shell cmdlet.

- To recover an entire mailbox from the recovery database, run the Restore-Mailbox cmdlet as follows:

  ```powershell
  Restore-Mailbox -Identity <target mailbox> -RecoverDatabase <recovery database>
  ```

- To recover a mailbox from the recovery database back to a folder in the same user's mailbox, use this alternate format of the Restore-Mailbox cmdlet:

  ```powershell
  Restore-Mailbox -Identity <target mailbox> -RecoverDatabase <recovery database> -RecoveryMailbox <source mailbox> -TargetFolder <target mailbox folder>
  ```

  For example:

  ```powershell
  Restore-Mailbox -Identity "Tom Jones" -RecoveryDatabase RDBTest -RecoveryMailbox "Tom Jones" -TargetFolder RestoredMail
  ```

  The example shown in Figure 60 on page 229 illustrates how to restore Tom Jones’ mailbox from the RDB to his current mailbox in the RestoredMail folder.
Starting with Exchange Server 2010 SP1, an individual mailbox can be recovered by running the New-MailboxRestoreRequest Exchange Management Shell cmdlet.

- To recover an entire mailbox from the recovery database, run the New-MailboxRestoreRequest cmdlet as follows:
New-MailboxRestoreRequest -SourceDatabase <source database> -TargetMailbox <target mailbox>

• To recover a mailbox from the recovery database back to a folder in the same user’s mailbox, use this alternate format of the New-MailboxRestoreRequest cmdlet:

New-MailboxRestoreRequest -SourceDatabase <source database> -SourceStoreMailbox <source mailbox> -TargetMailbox <target mailbox> -TargetRootFolder <target mailbox folder>

For example:

New-MailboxRestoreRequest -SourceDatabase RDBTest -SourceStoreMailbox "Tom Jones" -TargetMailbox "Tom Jones" -TargetRootFolder RestoredMail

• To recover a mailbox from the recovery database to the archive mailbox, use this format of the New-MailboxRestoreRequest cmdlet.

New-MailboxRestoreRequest -SourceDatabase <source database> -SourceStoreMailbox <source mailbox> -TargetMailbox <target mailbox> -TargetIsArchive

10. Recover specific e-mails.

In addition to recovering all e-mails for a mailbox, filters can be applied to the Restore-Mailbox cmdlet. The following additional qualifiers can be applied:

• AttachmentFileNames <string> - Filter attachment filenames. For example, a string of "*.txt" would include all e-mails with a text file attached.

• ContentKeyWords - Filter keywords contained in the message body and attachments.

• SubjectKeyWords - Filter keywords from the subject title.

• SenderKeyWords - Filter keywords from the senders name.

Recovery database cleanup

When the data has been recovered, the following steps are required to clean up the recovery database:

1. Dismount the Recovery Database. Using the Exchange Management Shell command:

   dismount-database <recoveryDatabase>

2. Unmount the replica using RM.
3. Remove the Recovery Mailbox Database, using the Exchange Management Shell command:

   remove-MailboxDatabase <recoveryDatabase>
This chapter includes the following topics:

- Introduction ................................................................. 234
- Definitions ........................................................................... 236
- Design considerations for continuity of operation plans ...... 238
- Remote replication considerations ........................................ 243
- Continuity and Replication Technologies ............................. 248
- Array-based redundancy features ........................................ 249
- Summary of site resiliency technologies .............................. 264
- Alternative replication solutions ......................................... 265
- AutoStart integration with Exchange Server 2010 .............. 268
Understanding Exchange Server 2010 Continuity of Operations

Introduction

A critical part of managing a messaging environment is planning for unexpected loss of service or data. The loss can occur from a disaster like fire or flood, or it can come from hardware or software failures. It can even be caused by human error or malicious intent. In any instance, the messaging environment must remain at or be restored to an accepted level of service for business operations to continue.

The effectiveness of any plan for continuity of operations involves answering the following questions:

- How much downtime is acceptable to the business?
- How much data loss is acceptable to the business?
- How complex is the solution?
- Does the solution accommodate the messaging architecture?
- How much does the solution cost?
- What disasters does the solution protect against?
- Is there protection against logical corruption?
- Is there protection against physical corruption?
- Is service resumption without data an option?
- Will the solution allow for ease of operation?
- Will the environment be restartable or recoverable?
- Can the solution be tested regularly?
- If failover happens, will failback work?

All continuity of operation plans include a replication component. In its simplest form, the replication process may be as easy as making a tape copy of the database and application. In a more sophisticated form, it could be real-time replication of all changed data to some remote location. Remote replication of data has its own challenges centered on:

- Distance
- Propagation delay (latency)
- Network infrastructure
- Data loss
This chapter provides an introduction to the spectrum of continuity of operations solutions for Exchange Server databases on EMC Symmetrix arrays.
In the next sections, the terms continuity of operations, database restart, database recovery, dependent-write consistency, and roll forward recovery are used. A clear definition of these terms is required to understand the context of this section.

**Continuity of Operations**

Continuity of Operations is a term that is used to encompass all solutions that enable the continuance of a service’s operational states. This includes the following terminology. High-availability solutions, where the operational state of the service has the ability to remain intact during problems that could cause service unavailability. Disaster recovery solutions, where the operational state of the service is restarted or recovered. Functional service continuity solutions, where the service and not the data is available immediately for use.

**Exchange database restart**

Database restart is the implicit application of Exchange logs during its normal initialization process to ensure a transactionally consistent data state.

If an Exchange database is shut down normally, the process of getting to a point of consistency during restart requires minimal work. If the database abnormally terminates, then the restart process will take longer, depending on the number of transaction logs needed to be applied at the time of termination—that is, all transaction logs past the last consistency checkpoint.

**Exchange database recovery**

Database recovery is the process of rebuilding an Exchange database from a backup image, and then explicitly applying subsequent logs to roll the data state forward to a designated point of consistency. Database recovery is only possible with databases configured with the appropriate level of database logging, that is, not using circular logging. In the context of Exchange server, recovery of this style is at the storage group level, and will necessitate taking all databases within the storage group offline.

A recoverable database copy can be taken in one of three ways:

1. With the database shut down and copying the database components using external tools.
2. With the database running using the Exchange Server VSS framework.
**Dependent-write consistency**

A dependent-write I/O is one that cannot be issued until a related predecessor I/O has completed. Dependent-write consistency is a data state where data integrity is guaranteed by dependent-write I/Os embedded in application logic. Database management systems are good examples of the practice of dependent-write consistency.

Database management systems must devise protection against abnormal termination to successfully recover from one. The most common technique used is to guarantee that a dependent-write cannot be issued until a predecessor write has completed. Typically, the dependent-write is a data or index write, while the predecessor write is a write to the log. Because the write to the log must be completed before issuing the dependent-write, the application thread is synchronous to the log write, that is, it waits for that write to complete before continuing. The result of this kind of strategy is a dependent-write consistent database.

**Roll forward recovery**

With some databases, it may be possible to take a restartable image of the database, and apply subsequent incremental transaction log backups, to roll forward the database to a point in time after the image was created. This means that the image created can be used in a backup strategy in combination with transaction log backups.
Design considerations for continuity of operation plans

E-mail data loss or e-mail service loss has a varying impact from one business type to another. In banking environments that process rate lock-in agreements through an e-mail system, service unavailability may result in the loss of millions by missing a point or two for thousands of customers. On the other side of the spectrum, an auto-parts reseller who processes orders requests through e-mail may lose millions due to loss of data containing pending orders. The two factors, loss of data and loss of uptime, are the business drivers that are baseline requirements for a continuity solution. When quantified, these two factors are more formally known as Recovery Time Objective (RTO) and Recovery Point Objective (RPO), respectively.

When evaluating a solution, the RTO and RPO requirements of the business need to be met. In addition, the solution needs to consider operational complexity, cost, and the ability to return the whole business to a point of consistency. Each of these aspects is discussed in the following sections.

Recovery Time Objective (RTO)

The RTO is the maximum amount of time allowed for recovery or restart to a specified point of consistency. This time involves many factors including the time it takes to:

- Provision power, utilities, and so on
- Provision servers with the application and database software
- Configure the network
- Restore the data at the new site
- Roll forward the data to a known point of consistency
- Validate the data

Some delays can be reduced or eliminated by choosing certain continuity options like having a hot site where servers are preconfigured and on standby. If storage-based replication is used, the time it takes to restore the data to a usable state is completely eliminated. Using a high-availability architecture like geographically dispersed clustering can eliminate the recovery time entirely.

Each solution for RTO will have a different cost profile. Defining the RTO is usually a compromise between the cost of the solution and the cost to the business when the messaging environment and the associated business processes are unavailable.
The RPO is a point of consistency to which a user wants to recover or restart. It is measured in the amount of time from when the point of consistency was created or captured to the time the disaster occurred. This time equates to the acceptable amount of data loss. Zero data loss (no loss of committed transactions from the time of the disaster) is the ideal goal, but the high cost of implementing such a solution must be weighed against the business impact and cost of a controlled data loss.

Some organizations have zero data loss requirements. The e-mail messages received at one location must be replicated immediately to another location. This can have an impact on application performance when the two locations are far apart. On the other hand, keeping the two locations close together might not protect against regional disasters like a power outage, a tsunami, an earthquake, or hurricanes.

Defining the required RPO, like the RTO, is usually a compromise between the needs of the business, the cost of the solution, and the risk of a particular event happening.

RPO and RTO go hand in hand with one another. Adjusting these two parameters is one of the most common ways to adjust the cost, bandwidth required, number of copies, and operational complexity. The RPO and RTO can either be adjusted together (having a low RPO and RTO is commonly the most desirable), or they can be adjusted independently (having a low RTO and a high RPO is perfectly acceptable for some businesses). It is important to understand these two components and the potential effect the different solutions have on them.

The operational complexity of a continuity solution may be the most critical factor in determining the success or failure of a continuity activity. The complexity of a continuity solution can be considered as three separate phases:

1. Initial setup of the implementation.
2. Maintenance and management of the running solution.
3. Ongoing testing and verification of the solution.
4. Execution of the continuity plan in the event of a disaster.
While initial configuration complexity and running complexity can be a demand on people resources, the fourth phase, execution of the plan, is where automation and simplicity must be the focus. When a disaster is declared, key personnel may not be available in addition to the loss of servers, storage, networks, buildings, and so forth. If the complexity of the continuity solution is such that skilled personnel with an intimate knowledge of all systems involved are required to restore, recover, and validate application and database services, the solution has a high probability of failure.

Some things that are commonly overlooked are:

- How will Online Maintenance be scheduled in the new site?
- How will e-mail archiving be configured?

**Source server activity**

Continuity solutions may or may not require additional processing activity on the source servers. The extent of that activity can impact both response time and throughput of the production application. This effect should be understood and quantified for any given solution to ensure that the impact to the business is minimized. Some solutions do not require additional processing and may be preferred in certain environments for that reason.

**Production impact**

Some continuity solutions delay the host activity while taking actions to propagate the changed data to another location. This action only affects write activity and although the introduced delay may only be of the order of a few milliseconds, it can impact response time in a high-write environment. Synchronous solutions introduce delay into write transactions at the source site, asynchronous solutions generally do not.

**Target server activity**

Some continuity solutions require a target server at the remote location to perform replication operations. The server has both software and hardware costs and needs personnel with physical access to it for basic operational functions like component replacement. Some continuity solutions require more target server activity and some require none. When using target servers it is a general rule of thumb to use servers that are of similar types and capacities to reduce strain in the event those systems are used.
Design considerations for continuity of operation plans

**Number of copies**

Continuity solutions require replication of data in one form or another. Replication of a database and associated files can be as simple as making a tape backup and shipping the tapes to a recovery site or as sophisticated as asynchronous array-based replication. Some solutions require multiple copies of the data to support recovery functions. More copies of the data may be required to perform testing of the continuity solution in addition to those that support the production continuity processes. The number of copies will play heavily into the overall cost and manageability of the solution.

**Distance for solution**

Disasters, when they occur, have differing ranges of impact. For instance, a fire may take out single server room or an entire building, an earthquake may destroy a city, or a tidal wave may devastate a region. The level of protection for a continuity solution should address all scales of probable disasters for a given location. For example, when protecting against an earthquake, the recovery site should not be in the same locale as the production site. For regional protection, the two sites need to be in two different regions. The distance associated with the sites involved affects the kind of continuity solution that can be implemented.

**Bandwidth requirements**

One of the largest costs for continuity is in provisioning bandwidth for the solution. Bandwidth costs are an operational expense of the solution; this makes solutions that have reduced bandwidth requirements very attractive to customers.

Incorrect provisioning of bandwidth for continuity solutions can adversely affect production performance and can invalidate the overall solution.

**Testing the solution**

Tested, proven, and documented procedures are required for a continuity solution. Many times, the continuity test procedures are operationally different from the set of disaster procedures. Operational procedures need to be clearly documented and consistent between operational testing and execution within a disaster scenario. In the best-case scenario, companies should periodically execute the actual set of procedures. This could be costly to the business because of the application downtime required to perform such a test, but is necessary to ensure validity of the procedures and meet regulatory requirements.
Cost

The cost of implementing a continuity plan can be justified by comparing it to the cost of not doing it. What does it cost the business when the messaging environment and associated business processes are unavailable to users? For some companies this is easily measurable, and revenue loss can be calculated per hour of downtime or per hour of data loss.

Whatever the business, the continuity cost is going to be an extra expense item and, in many cases, with little in return. The costs include, but are not limited to:

- Hardware (storage, servers, and maintenance)
- Software licenses and maintenance
- Facility leasing/purchase
- Utilities
- Network infrastructure
- Personnel
Remote replication considerations

Replicating database information over long distances for the purpose of disaster recovery is challenging. Synchronous replication over distances greater than 200 km will likely result in poorly performing write operations. This is due to propagation delays inherent in networks over large distances. To work around these issues, some form of asynchronous replication must be adopted. Considerations in this section apply to all forms of remote replication technology, whether they are array-based, host-based, or managed by the e-mail system.

Remote replication solutions usually start with initially copying a full database image to the remote location. This is called instantiation of the database. There are a variety of ways to perform this. After instantiation, only the changes from the source site are replicated to the target site in an effort to keep the target up to date.

These methodologies may require periodic reinstantiation of the database at the remote site.

The following considerations apply to remote replication of databases:

- Propagation delay (latency because of distance)
- Bandwidth requirements
- Network infrastructure
- Method of instantiation
- Method of reinstantiation
- Change rate at the source site
- Locality of reference
- Expected data loss
- Failback operations

**Propagation delay**

Electronic operations execute at the speed of light. The speed of light in vacuum is 186,000 miles per second. The speed of light through glass (in the case of fiber optic media) is less, approximately 115,000 miles per second. In other words, in an optical network like SONET for instance, it takes 1 millisecond to send a data packet 125 miles.
(201.168 km) or 8 milliseconds for 1000 miles (1609.344 km). All remote replication solutions need to be designed with a clear understanding of the propagation delay impact.

**Bandwidth requirements**

All remote replication solutions have some bandwidth requirements because the changes from the source site must be propagated to the target site. As the amount of changes increase, the amount of bandwidth increases. It is the change rate and replication methodology that determine the bandwidth requirement, not necessarily the size of the e-mail databases.

Data compression can help reduce the quantity of data transmitted and therefore the size of the “pipe” required. Certain network devices, like switches and routers, provide native compression, some by software and others by hardware. Gig-E directors provide native compression in a Symmetrix to Symmetrix SRDF pairing. The amount of compression achieved is dependent on the type of data that is being compressed. Typical character and numeric database data compresses at about a 2-to-1 ratio. A good way to estimate how the data will compress is to assess how much tape space is required to store the database during a full backup process. Tape drives perform hardware compression on the data before writing it. For instance, if a 300 GB database takes 200 GB of space on tape, the compression ratio is 1.5 to 1.

For most customers, a major consideration in the disaster recovery design is cost. It is important to recognize that some components of the end solution represent a capital expenditure and some, an operational expenditure. Bandwidth costs are operational expenses and thus any reduction in this area, even at the cost of some capital expense, is highly desirable.

**Network infrastructure**

The choice of channel extension equipment, network protocols, switches, and routers ultimately determines the operational characteristics of the solution. EMC has a proprietary BC Design Tool to assist customers in their analysis of the source systems and to determine the required network infrastructure to support a remote replication solution. This tool uses a variety of factors like change rate, I/O patterns, and compression rates to help derive the necessary bandwidth on different network technologies.
Remote replication considerations

Method of instantiation

In all remote replication solutions, a common requirement is for an initial, consistent copy of the complete database to be replicated to the remote site. The initial copy from source to target is called instantiation of the database at the remote site. Following instantiation, only the changes made at the source site are replicated. For large databases, sending only the changes after the initial copy is the only practical and cost-effective solution for remote database replication.

In some solutions, instantiation of the database at the remote site uses a process that is similar to the one that replicates the changes. Some solutions do not even provide for instantiation at the remote site (for instance, log shipping). In all cases, it is critical to understand the pros and cons of the complete solution.

Method of reinstatination

Some methods of remote replication require periodic refreshing of the remote system with a full copy of the database. This is called reinstatiation. Technologies such as log shipping frequently require this, because not all activity on the production database may be represented in the log. In these cases, the disaster recovery plan must account for reinstatiation and also for the fact that there may be a disaster during the refresh. The business objectives of RPO and RTO must likewise be met under those circumstances.

Change rate at the source site

After instantiation of the database at the remote site, only changes to the database are replicated remotely. There are many methods of replication to the remote site and each has differing operational characteristics. The changes can be replicated using, for example, logging technology, hardware and software mirroring. Before designing a solution with remote replication, it is important to quantify the average change rate. It is also important to quantify the change rate during periods of burst write activity. These periods might correspond to end-of-month/quarter/year processing, billing, or payroll cycles. The solution needs to be designed to allow for peak write workloads.

Locality of reference

Locality of reference is a factor that needs to be measured to understand if there will be a reduction of bandwidth consumption when any form of asynchronous transmission is used. Locality of reference is a measurement of how much write activity on the source is skewed. For instance, a high locality of reference application may
make many updates to a few tables in the database, whereas a low locality of reference application rarely updates the same rows in the same tables during a given period of time.

It is important to understand that while the activity on the tables may have a low locality of reference, the write activity into an index might be clustered when inserted rows have the same or similar index column values, rendering a high locality of reference on the index components.

In some asynchronous replication solutions, updates are “batched” up into periods of time and sent to the remote site to be applied. In a given batch, only the last image of a given row/block is replicated to the remote site. So, for highly skewed application writes, this results in bandwidth savings. Generally, the greater the time period of batched updates, the greater the savings on bandwidth.

Log shipping technologies do not take into account locality of reference. For example, a row updated 100 times, is transmitted 100 times to the remote site, whether the solution is synchronous or asynchronous.

**Expected data loss**

Synchronous continuity solutions are zero data loss solutions, that is, there is no loss of committed transactions from the time of the disaster. Synchronous solutions may also be impacted by a rolling disaster in which case, work completed at the source site after the rolling disaster started, may be lost. Rolling disasters are also discussed in detail in this chapter.

Non-synchronous continuity solutions have the potential for data loss. How much data is lost depends on many factors, most of which have been previously defined. For asynchronous replication, where updates are batched and sent to the remote site, the maximum amount of data lost will be two cycles or two batches worth. The two cycles that may be lost include the cycle currently being captured on the source site and the one currently being transmitted to the remote site. With inadequate network bandwidth, data loss could be increased because of the increased transmission time.

**Failback operations**

If there is the slightest chance that fail over to the continuity site may be required, then there is a 100 percent chance that failback to the primary site will also be required, unless the primary site is lost permanently. The continuity architecture should be designed in such a way as to make failback simple, efficient, and low-risk. If failback is
not planned for, there may be no reasonable or acceptable way to move the processing from the continuity site, where the applications may be running on tier 2 servers and tier 2 networks, back to the production site.

Ideally, the continuity process should be tested once a quarter, with database and application services fully failed over to the continuity site. The integrity of the application and database needs to be verified at the remote site to ensure that all required data was copied successfully. Ideally, production services are brought up at the continuity site as the ultimate test. This means that production data would be maintained on the continuity site, requiring a failback when the continuity test completed. While this is not always possible, it is the ultimate test of a continuity solution. It not only validates the continuity process, but also trains the staff on managing the continuity process, should a catastrophic failure ever occur. The downside for this approach is that duplicate sets of servers and storage need to be present to make an effective and meaningful test. This tends to be an expensive proposition.
Continuity and Replication Technologies

Continuity and Replication Technologies refer to technologies that enable operational continuity in the event of a partial or total site failure. This can include a single server losing power, a single storage array losing power, half of the data center losing power, or the entire site disappearing from existence. Site Resiliency should be considered a top priority when designing any kind of Continuity solution and is therefore discussed here.

There are three main technologies that can be used to provide CooP during site failures. These technologies are: application level redundancy features, operating system level redundancy features, and hardware level redundancy features. All of these three technologies have varying degrees of support for different failures and have different levels of integration with each other. The technologies will be briefly introduced before delving into how they are used in site failure scenarios.

Application-level redundancy features refer to the types of redundancy that is built into the application itself. For Exchange Server 2010, there is a built-in redundancy feature, Extended DAG as described in Chapter 1, “Microsoft Exchange Server.” It relies on the IP network to replicate data between data centers.

Operating system level redundancy only makes the operating system redundant on multiple nodes. In these scenarios the operating system is installed on more than one node and an application such as Exchange Server is installed on the cluster of operating systems. In the event the active operating system node suffers a physical servers failure, the single instance of the application is moved to a passive operating system node. This move can be automated or manual and usually comes with a varying set of other requirements (such as server hardware, network configuration, and so on). For this move to happen, the data that the application uses must already be accessible on the passive node. This can be accomplished through the use of shared storage, an application level redundancy feature, or an array replication feature. As Exchange Server 2010 does not support a single copy cluster a new technology was introduced called the Synchronous API in order to allow the replication of an Exchange Server 2010 database using third party technologies.
Array-based redundancy features

Customers can use the capabilities of a Symmetrix storage array to replicate the database from the production location to a secondary location. No host CPU cycles are used for this, leaving the host dedicated to running the production application and database. In addition, no host I/O is required to facilitate this, the array takes care of all replication and no hosts are required at the target location to manage the target array.

EMC provides multiple solutions for remote replication of databases:

- SRDF/S: SRDF/Synchronous
- SRDF/A: SRDF/Asynchronous
- SRDF/AR: SRDF/Automated Replication

Each of these solutions is discussed in detail in the next sections. In order to use any of the array-based solutions, it is necessary to coordinate the disk layout of the databases with this kind of replication in mind.

Planning for array-based replication

All Symmetrix solutions replicating data from one array to another are disk based. This allows the Symmetrix to be agnostic to the volume manager, file system, database system, and so on. However, this does not mean that file system and volume manager concerns can be ignored. It is important to understand the relationship of Windows NT File System (NTFS) volumes and LUNs. On Windows, the smallest unit of granularity for storage-based replication is the LUN, a volume set, or disk group, depending on how the disks are set up in disk manager.

In addition, if a database is to be replicated independently of other databases, it should have its own dedicated disks (LUNs). That is, the disks used by a database should not be shared with other applications or databases.

When a set of volumes has been defined for a database for remote replication, care must be taken to ensure that the disks contain everything that is needed to restart the database at the remote site. For Exchange Server databases, this must be the complete set of data and transaction log files comprising the database.
SRDF/Synchronous, or SRDF/S, is a method of replicating production data changes from locations that are no greater than 200 km apart. Synchronous replication takes writes that are inbound to the source Symmetrix and copies them to the target Symmetrix. The write operation is not acknowledged as complete to the host until both Symmetrix arrays have the data in cache. It is important to realize that while the following examples involve Symmetrix, the fundamentals of synchronous replication described here are true for all synchronous replication solutions. Figure 61 on page 250 depicts the process:

1. Write issued from host/server into cache of Source Array
2. Write is transmitted to the cache of Target Array
3. Receipt of Write is transmitted to Source Array
4. Acknowledge of write is returned to host/server

Figure 61 SRDF/S replication process

1. A write is received into the source Symmetrix cache. At this time, the host has not received acknowledgement that the write is complete.
2. The source Symmetrix uses SRDF/S to push the write to the target Symmetrix.
3. The target Symmetrix sends an acknowledgement back to the source that the write was received.
4. Ending status of the write is presented to the host.

These four steps cause a delay in the processing of writes as perceived by the database on the source server. The amount of delay depends on the exact configuration of the network, the storage, the
write block size, and the distance between the two locations. Note that reads from the source Symmetrix are not affected by the replication.

The following steps outline the process of setting up synchronous replication using Solutions Enabler (SYMCLI) commands.

Before the synchronous mode of SRDF can be established, initial instantiation of the database has to take place. In other words, a baseline full copy of all the volumes that are going to participate in the synchronous replication must be executed first. This is usually accomplished using the adaptive copy mode of SRDF. The following command creates a group where `<device_group>` is a user-specified name:

```
symdg create <device_group> -type rdf1
```

The type of the device group is dependent on the location of the system being used to define the device group. The RDF1 type is used if the host is connected locally to the Source/R1 array:

1. Add devices to the group.
2. The following command puts the group into adaptive copy mode:

```
symrdf -g <device_group> set mode acp_disk -noprompt
```

3. The following command causes the source Symmetrix to send all the tracks on the source site to the target site using the current mode:

```
symrdf -g <device_group> establish -full -noprompt
```

The adaptive copy mode of SRDF has no impact to host application performance. It transmits tracks to the remote site that have never been sent before or that have changed since the last time the track was sent. It does not preserve write order or dependent-write consistency.

4. When both sides are synchronized, SRDF can then be put into synchronous mode. In the following command, the device group is put into synchronous mode:

```
symrdf -g <device_group> set mode sync -noprompt
```
There is no requirement for host availability at the remote site during the synchronous replication. The target Symmetrix itself manages the in-bound writes and updates the appropriate volumes in the array.

Dependent-write consistency is inherent in a synchronous relationship as the target R2 volumes are at all times equal to the source, provided that a single RA group is used. If multiple RA groups are used or if multiple Symmetrix arrays are used on the source site, SRDF/Consistency Groups (SRDF/CG) must be used to guarantee consistency. SRDF/CG is described next.

In the event of a disaster where the primary source Symmetrix is lost, it becomes necessary to run database and application services from the continuity site. A host at the continuity site is required for this. The first requirement is to write-enable the R2 devices. If the device group is not yet built on the remote host, it must be created using the R2 devices that were mirrors of the R1 devices on the source Symmetrix. Group Named Services (GNS) can be used to propagate the device group to the remote site if there is a host being utilized there. The Solutions Enabler Symmetrix Base Management CLI Product Guide provides more details on GNS.

The following command write-enables the R2s in a group:

```
symld -g <device_group> rw_enable -noprompt
```

At this point, the host can issue the necessary commands to access the disks.

Once the data is available to the host, the database can be restarted. The database will perform an implicit recovery when restarted. Transactions that were committed but not completed are rolled forward and completed using the information in the transaction log. Transactions that have updates applied to the database but were not committed are rolled back. The result is a transactionally consistent database.

SRDF/A, or SRDF/Asynchronous, is a method of replicating production data changes from one Symmetrix to another using delta set technology. Delta sets are the collection of changed blocks grouped together by a time interval that can be configured at the source site. The default time interval is 30 seconds. The delta sets are then transmitted from the source site to the target site in the order they were created. SRDF/A preserves dependent-write consistency of the database at all times at the remote site.
The distance between the source and target Symmetrix is unlimited and there is no host impact. Writes are acknowledged immediately when they hit the cache of the source Symmetrix. SRDF/A is only available on the Symmetrix family. Figure 62 on page 253 depicts the process:

1. **CAPTURE** delta set collects application write I/Os
2. Delta-set switch: dependent write consistent state
3. **TRANSMIT** delta-set sends N-1 set of I/Os to RECEIVE delta-set on Target
4. **APPLY** delta-set: After TRANSMIT is complete, data applied to disk
5. Cycle repeats

**Figure 62 SRDF/Asynchronous replication process**

1. Writes are received into the source Symmetrix cache. The host receives immediate acknowledgement that the write is complete. Writes are gathered into the capture delta set for 30 seconds.
2. A delta set switch occurs and the current capture delta set becomes the transmit delta set by changing a pointer in cache. A new empty capture delta set is created.
3. SRDF/A sends the changed blocks that are in the transmit delta set to the remote Symmetrix. The changes collect in the receive delta set at the target site. When the replication of the transmit delta set is complete, another delta set switch occurs and a new empty capture delta set is created with the current capture delta set becoming the new transmit delta set. The receive delta set becomes the apply delta set.
4. The apply delta set marks all the changes in the delta set against the appropriate volumes as invalid tracks and begins destaging the blocks to disk.
5. The cycle repeats continuously.
With sufficient bandwidth for the source database write activity, SRDF/A will transmit all changed data within the default 30 seconds. This means that the maximum time the target data will be behind the source is 60 seconds (two replication cycles). At times of high write activity, it may not be possible to transmit all the changes that occur during a 30 second interval. This means that the target Symmetrix will fall behind the source Symmetrix by more than 60 seconds. Careful design of the SRDF/A infrastructure and a thorough understanding of write activity at the source site are necessary to design a solution that meets the RPO requirements of the business at all times.

Consistency is maintained throughout the replication process on a delta set boundary. The Symmetrix will not apply a partial delta set which would invalidate consistency. Dependent-write consistency is preserved by placing a dependent-write in either the same delta set as the write it depends on or a subsequent delta set.

**Note:** There is no requirement for a host at the remote site during asynchronous replication. The target Symmetrix manages in-bound writes and updates the appropriate disks in the array.

Different command sets are used to enable SRDF/A depending on whether the SRDF/A group of devices is contained within a single Symmetrix or is spread across multiple Symmetrix arrays.

When using SRDF/A, it is recommended that Generic Safe Write is enabled on the volumes that will house the Exchange databases. This will ensure that all writes are validated to local cache prior to being committed locally and remotely.

To enable Generic Safe Write run the following command on each database device:

```
symchksum -sid <symmetrixID> -type generic enable dev <DeviceID>
```

Before the asynchronous mode of SRDF can be established, initial instantiation of the database has to take place. In other words, a baseline full copy of all the volumes that are going to participate in the asynchronous replication must be executed first. This is usually accomplished using the adaptive copy mode of SRDF.

The following steps outline the process of setting up asynchronous replication using Solutions Enabler (SYMCLI) commands:
1. To create an SRDF disk group for the source side of the synchronous relationship, that is, the R1 side:
   
   ```bash
   symdg create <device_group> -type rdf1
   ```

2. Add devices to the group.

3. The following command puts the device group into adaptive copy mode:
   
   ```bash
   symrdf -g <device_group> set mode acp_disk -noprompt
   ```

4. The following command causes the source Symmetrix to send all the tracks at the source site to the target site using the current mode:
   
   ```bash
   symrdf -g <device_group> establish -full -noprompt
   ```

The adaptive copy mode of SRDF has no impact to host application performance. It transmits tracks to the remote site that have never been sent before or that have changed since the last time the track was sent. It does not preserve write order or consistency. When both sides are synchronized, SRDF can be put into asynchronous mode. In the following command, the device group is put into asynchronous mode:

```bash
symrdf -g <device_group> set mode async -noprompt
```

**Note:** There is no requirement for a host at the remote site during the asynchronous replication. The target Symmetrix manages the in-bound writes and updates the appropriate disks in the array.

---

**Moving between SRDF/A and SRDF/S**

In certain situations it is necessary to change between Asynchronous and Synchronous SRDF replication. Care should be taken to ensure that dependant write consistency is maintained during the mode change. The write consistency currently can only be maintained for device groups and device files and not composite groups or SRDF/A MSC. The product documentation provides further information.

**Restart processing**

In the event of a disaster, when the primary source Symmetrix is lost, database and application services must be run from the continuity site. A host at the continuity site is required for this. If the device or
composite group is not defined yet on the remote host, it must first be created using the R2 devices that were mirrors of the R1 devices on the source Symmetrix. The first step is to write-enable the R2 devices.

R2s on a single Symmetrix:

```
symld -g <device_group> rw_enable -noprompt
```

R2s on multiple Symmetrix:

```
symcg -cg <composite_group> rw_enable -noprompt
```

At this point, the host can issue the necessary commands to access the disks. This includes the steps required for mounting the volumes appropriately as mount points or drive letters.

Once the data is available to the host, the database can be restarted. The database will perform crash recovery when the database is attached. Transactions that were committed, but not completed, are rolled forward and completed using the information in the active logs. Transactions that have updates applied to the database but not committed are rolled back. The result is a transactionally consistent database.

---

**SRDF/AR single hop**

SRDF/Automated Replication, or SRDF/AR, is a continuous movement of dependent-write consistent data to a remote site using SRDF adaptive copy mode and TimeFinder consistent split technology. TimeFinder BCVs are used to create a dependent-write consistent point-in-time image of the data to be replicated. The BCVs also have an R1 personality, which means that SRDF in adaptive copy mode can be used to replicate the data from the BCVs to the target site. Since the BCVs are not changing, replication is completed in a finite length of time. The length of time for replication depends on the size of the network “pipe” between the two locations, the distance between the two locations, the quantity of changed data tracks, and the locality of reference of the changed tracks. On the remote Symmetrix, another BCV copy of the data is made using data on the R2s. This is necessary because the next SRDF/AR iteration replaces the R2 image in a nonordered fashion, and if a disaster were to occur while the R2s were synchronizing, there would not be a valid copy of the data at the continuity site.

The BCV copy of the data in the remote Symmetrix is commonly called the “gold” copy of the data. The whole process then repeats.
With SRDF/AR, there is no host impact. Writes are acknowledged immediately when they hit the cache of the source Symmetrix. Figure 63 on page 257 depicts the process:

1. Consistent split in Production
2. SRDF Mirroring resumed
3. Incremental Establish initiated on both Source and Target BCVs
4. BCV split on Target
5. Cycle repeats

Figure 63  SRDF/AR single-hop replication process

1. Writes are received into the source Symmetrix cache and are acknowledged immediately. The BCVs are already synchronized with the STDs at this point. A consistent split is executed against the STD-BCV pairing to create a point-in-time image of the data on the BCVs.

2. SRDF transmits the data on the BCV/R1s to the R2s in the remote Symmetrix.

3. When the BCV/R1 volumes are synchronized with the R2 volumes, they are reestablished with the standards in the source Symmetrix. This causes the SRDF links to be suspended. At the same time, an incremental establish is performed on the target Symmetrix to create a gold copy on the BCVs in that frame.

4. When the BCVs in the remote Symmetrix are fully synchronized with the R2s, they are split and the configuration is ready to begin another cycle.

5. The cycle repeats based on configuration parameters. The parameters can specify the cycles to begin at specific times, specific intervals or to run back to back.
Cycle times for SRDF/AR are usually in the minutes-to-hours range. The RPO is double the cycle time in a worst-case scenario. This may be a good fit for customers with relaxed RPOs.

The added benefit of having a longer cycle time is that the locality of reference will likely increase. This is because there is a much greater chance of a track being updated more than once in a one-hour interval than in, say, a 30 second interval. The increase in locality of reference shows up as reduced bandwidth requirements for the final solution.

Before SRDF/AR can be started, instantiation of the database has to take place. In other words, a baseline full copy of all the volumes that are going to participate in the SRDF/AR replication must be executed first. This means a full establish to the BCVs in the source array, a full SRDF establish of the BCV/R1s to the R2s, and a full establish of the R2s to the BCVs in the target array is required. There is an option to automate the initial setup of the relationship.

As with other SRDF solutions, SRDF/AR does not require a host at the continuity site. The commands to update the R2s and manage the synchronization of the BCVs in the remote site are all managed in-band from the production site.

**Restart processing**

In the event of a disaster, it is necessary to determine if the most current copy of the data is located on the remote site BCVs or R2s at the remote site. Depending on when in the replication cycle the disaster occurs, the most current version could be on either set of disks. This determination is simple and is described in the SRDF/AR solutions guide.

**SRDF/AR multihop**

SRDF/Automated Replication multihop, or SRDF/AR multihop, is an architecture that allows long-distance replication with zero seconds of data loss through use of a bunker Symmetrix. Production data is replicated synchronously to the bunker Symmetrix, which is within 200 km of the production Symmetrix, allowing synchronous replication but also far enough away that potential disasters at the primary site may not affect it. Typically, the bunker Symmetrix is placed in a hardened computing facility.

BCVs in the bunker frame are periodically synchronized to the R2s and consistent split in the bunker frame to provide a dependent-write consistent point-in-time image of the data. These bunker BCVs also have an R1 personality, which means that SRDF in adaptive copy mode can be used to replicate the data from the bunker array to the
target site. Since the BCVs are not changing, the replication can be completed in a finite length of time. The length of time for the replication depends on the size of the “pipe” between the bunker location and the DR location, the distance between the two locations, the quantity of changed data, and the locality of reference of the changed data. On the remote Symmetrix, another BCV copy of the data is made using the R2s. This is because the next SRDF/AR iteration replaces the R2 image, in a non-ordered fashion. If a disaster were to occur while the R2s were synchronizing, there would not be a valid copy of the data at the DR site. The BCV copy of the data in the remote Symmetrix is commonly called the gold copy of the data. The whole process then repeats.
With SRDF/AR multihop, there is minimal host impact. Writes are only acknowledged when they hit the cache of the bunker Symmetrix and a positive acknowledgment is returned to the source Symmetrix. Figure 64 on page 260 depicts the process:

1. Consistent split in Bunker Array
2. SRDF Mirroring resumed
3. Incremental Establish initiated on both Source and Target BCVs
4. BCV split on Target
5. Cycle repeats

**Figure 64** SRDF/AR multihop replication process

1. BCVs are synchronized and consistently split against the R2s in the bunker Symmetrix. The write activity is momentarily suspended on the source Symmetrix to get a dependent-write consistent point-in-time image on the R2s in the bunker Symmetrix, which creates a dependent-write consistent point-in-time copy of the data on the BCVs.

2. SRDF transmits the data on the bunker BCV/R1s to the R2s in the DR Symmetrix.

3. When the BCV/R1 volumes are synchronized with the R2 volumes in the target Symmetrix, the bunker BCV/R1s are established again with the R2s in the bunker Symmetrix. This causes the SRDF links to be suspended between the bunker Symmetrix and the DR Symmetrix. At the same time, an incremental establish is performed on the DR Symmetrix to create a gold copy on the BCVs in that frame.

4. When the BCVs in the DR Symmetrix are fully synchronized with the R2s, they are split and the configuration is ready to begin another cycle.
5. The cycle repeats based on configuration parameters. The parameters can specify the cycles to begin at specific times, specific intervals, or to run immediately after the previous cycle completes.

It should be noted that even though cycle times for SRDF/AR multihop are usually in the minutes-to-hours range, the most current data is always in the bunker Symmetrix. Unless there is a regional disaster that destroys both the primary site and the bunker site, the bunker Symmetrix will transmit all data to the remote DR site. This means zero data loss at the point of the beginning of the rolling disaster or an RPO of 0 seconds. This solution is a good fit for customers with a requirement of zero data loss and long-distance DR.

An added benefit of having a longer cycle time means that the locality of reference will likely increase. This is because there is a much greater chance of a track being updated more than once in a one-hour interval than in, say, a 30-second interval. The increase in locality of reference shows up as reduced bandwidth requirements for the network segment between the bunker Symmetrix and the DR Symmetrix.

Before SRDF/AR can be initiated, initial instantiation of the database has to take place. In other words, a baseline full copy of all the volumes that are going to participate in the SRDF/AR replication must be executed first. This means a full establish of the R1s in the source location to the R2s in the bunker Symmetrix. The R1s and R2s need to be synchronized continuously. Then a full establish from the R2s to the BCVs in the bunker Symmetrix, a full SRDF establish of the BCV/R1s to the R2s in the DR Symmetrix and a full establish of the R2s to the BCVs in the DR Symmetrix is performed. There is an option to automate this process of instantiation.

**Restart processing**

In the event of a disaster, it is necessary to determine if the most current copy of the data is on the R2s on the remote site or on the BCV/R1s in the bunker Symmetrix. Depending on when the disaster occurs, the most current version could be on either set of disks. This determination is simple and is outlined in the SRDF/AR solutions guide.

**Integrating Exchange with array-based replication**

Write order consistency is important to the reliability of Exchange Server’s ESE database. For any software or array to integrate with this database, it is important that the integration honors the write
order dependency of the database. Not honoring the write order can result in a number of serious problems, from simple torn pages to complete database corruption.

EMC’s Symmetrix arrays honor write order consistency in all native operations. This means that all mirroring activities, including but not limited to production mirrors, business continuance mirrors, clone mirrors, and SRDF R2 mirrors, all honor write ordering. This is important as it is an inherent technology in the array that protects against data corruption. That inherent technology protects data whether it is replicated locally or remotely.

To add to the write order consistency mechanisms, EMC has added a generic checksum feature to further enhance the write durability of Exchange I/O to the disk. Generic Checksum should be enabled on all database volumes especially if those volumes are to be replicated to a remote array. Generic checksum is a feature that will hold each write in cache until the entire I/O has been delivered from the host. Once the entire write has been received, the array will checksum it and send it to disk where it is validated against the checksum to ensure integrity of the individual write. To Exchange, this means each write is guaranteed to be complete on disk and the chance of torn pages locally and remotely is removed.


---

**Symmetrix Remote Data Replication Cluster Enabled (SRDF/CE)**

To extend the functionality of single site Windows Failover Cluster (WFC) configurations and provide additional multisite protection, EMC provides the SRDF/CE for WFC geographically dispersed clustering product. This solution extends the single array WFC configuration to allow for multiple sites with array-based replication between the two sites. Certification of the solution is provided by Microsoft in a similar manner to single WFC configurations. Certified configurations may be found on the Windows Catalog available from the Microsoft home page.

SRDF/CE for WFC provides a level of abstraction at the storage level such that WFC believes it is executing in a standard mode. All typical WFC functions, procedures, and restrictions remain in place. The benefit provided by SRDF/CE for WFC is that it can provide for failures within a site and entire site failures. This ability drastically enhances the RTO for applications by automatically managing recovery of resources groups.
Geographically dispersed clustering solutions such as that of SRDF/CE for WFC based on SRDF/S provide zero data loss solutions with extremely small RTO, since most processes are automated. SRDF/CE can also use SRDF/A to provide near zero data loss on lower bandwidth connections. This enables a lower cost solution and maintains the small RPO/RTO.
Summary of site resiliency technologies

Table 12 details the basic components about the technologies and their supportability.

<table>
<thead>
<tr>
<th>Technology</th>
<th>What is replicated</th>
<th>Data loss</th>
<th>RPO</th>
<th>RTO</th>
<th>Type of replication</th>
<th>Primary support</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAG</td>
<td>Closed logs</td>
<td>Possible</td>
<td>Low-Med</td>
<td>Med-High</td>
<td>Asynch</td>
<td>Microsoft</td>
</tr>
<tr>
<td>SRDF/A</td>
<td>Write-order data sets</td>
<td>Possible</td>
<td>High</td>
<td>Med-High</td>
<td>Asynch</td>
<td>EMC</td>
</tr>
<tr>
<td>SRDF/S</td>
<td>Every I/O</td>
<td>Zero</td>
<td>High</td>
<td>Med-High</td>
<td>Synch</td>
<td>EMC</td>
</tr>
<tr>
<td>SRDF/CE</td>
<td>Every I/O</td>
<td>None</td>
<td>High</td>
<td>High</td>
<td>Synch</td>
<td>EMC</td>
</tr>
</tbody>
</table>
Alternative replication solutions

There are several alternative replication features that are both entirely storage agnostic and partially storage agnostic.

Entirely storage agnostic replication solutions are a requirement for many organizations and are a valid option for continuity planning. For most database products, storage agnostic replication is accomplished through a log shipping or data packet synchronization mechanism.

Microsoft supports file system filter log shipping for Exchange 2003. These solutions generally provide fewer resiliencies or are less performant than array-based replication. This should be kept at the forefront while evaluating these solutions along with the cost of the solution being evaluated.

At a basic level, log shipping solutions function by making a consistent copy of the mailbox databases and sending them to a remote server (this is true regardless of the storage the two servers are using). Once the database copies are on the remote server, the log shipping functionality takes over and the remote server begins to receive all log files as they are closed on the production system.

There are two supported methods to create a consistent copy of the mailbox databases. A log shipping application can either create a VSS snapshot or use the Exchange streaming API. Microsoft’s application integrated solutions use the latter while most supported vendors use the former. Once the database copy is obtained, it must be quickly delivered to the remote server. This process can be accomplished a number of ways including, network file transfers, storage replication transfers, or manual transfers with external media.

There are also two methods to transfer closed log files to the remote server, using a file system filter or integrating with Exchange. Using a file system filter is the only supported method for third-party applications to perform this operation.

The most common form of data packet synchronization is storage area network devices that sit either in the data path or near the data path. Devices in the data path duplicate each I/O packet before passing it along to its original destination. Devices that are near the data path are sent duplicated packets by either an operating system interface or a storage device. In both cases, the duplicated packet is then sent to the remote site where it can be inserted into the remote
storage array. It is important to note that Microsoft officially supports these methods when they are integrated with the Volume Shadow Copy Service.

When considering a log shipping or I/O synchronization strategy, it is important to understand:

- What log shipping covers?
- What log shipping does not cover?
- What is the exposure to data loss during periods of change in recovery modes?
- What are the server requirements?
- How to instantiate and reinstantiate the target database?
- How failback works?
- What is the amount of data loss in event of site disaster?
- What is the manageability of the solution?
- What is the scalability of the solution?

Partially, storage agnostic solutions such as Open Replicator require a Symmetrix system to be involved; however, the remote array can be of any other type. Open Replicator can be used to push data from a Symmetrix to another array or pull data from another array to a Symmetrix. Both techniques are useful for continuity solutions and are fully supported by EMC’s consistency technologies.

Open Replicator works by having a control device that data is pushed from or pulled to. This device can be the production device, or can be copy of the production device. During push operations, the live data or copy of the live data is pushed to a remote array from the Symmetrix array’s control device. During a pull operation, the live data or a copy of the live data is pulled to the Symmetrix array’s control device. Both of these techniques use a standard Front End Adapter to transfer the data to a World Wide Name of the remote device. This traffic is treated the same as host traffic and follows the same internal rules.

There are a few things that should be considered when using Open Replicator in conjunction with a continuity strategy:

- Only standard devices, clones, and Business Continuance Volumes (BCVs) can be used as control devices which leaves TimeFinder/Snap as the only type of device that cannot be used.
Donor Update should be used when pulling to a Symmetrix. This allows any changes made to the data on the Symmetrix to be replicated back to the original donor of the data. This will allow a migration or transfer to be rolled back to the donor system in the event there are any issues with the application portion of the migration.

Tape-based solutions

Tape-based disaster recovery

Traditionally, the most common form of continuity of operations was to make a copy of the e-mail databases onto tape using the streaming API and taking the tapes offsite to a hardened tape storage facility. In most cases, the e-mail system needed to be available to users during the backup process. Recovery usually involved recalling the tape, restoring the data from the tape to an offline e-mail server, and finally restarting the e-mail database to restore service to users.

This process is still in effect today by some customers and is still a supported methodology. The rapid proliferation of e-mail over the last several years has caused this method to become unmanageable. Due to this management issue, this technology is becoming less appealing to customers and is on its way to becoming an unsupported technology.

Tape-based disaster recovery

Tape-based disaster restart is a recent development in disaster recovery strategies due to the recent addition of consistent restart technologies. A “restart” copy of the e-mail database is created by locally mirroring the disks that contain the production data and logs, then splitting off the mirrors to create a dependent-write consistent point-in-time image on the disks. This is a restartable image as previously described. Thus, if this image was restored and the database brought up, the database would perform an implicit recovery to attain transactional consistency. Roll-forward recovery using incremental transaction logs from this database image requires manual modification of the checkpoint. The restartable image on the disks can be backed up to tape and moved offsite to a secondary facility. Alternatively, the data can be replicated to an offsite facility and the restartable copy can be taken and stored there for future use.

Writing the image to tape and reading the image back during a restore can be a lengthier process than using a disk to store the image. This affects the overall RPO and should be considered when planning a continuity solution.
AutoStart integration with Exchange Server 2010

Exchange Server 2010 offers an API to allow synchronous replication vendors to integrate with the DAG framework. AutoStart 5.3 SP5 has been updated to include an Exchange Server 2010-specific module to integrate with this API. AutoStart provides support for specific storage-based technologies based on components called data sources. In combination with the AutoStart Exchange Server 2010 module, the following AutoStart data sources are supported:

Shared disk device for Windows data source

A shared disk device for Windows data source is a common LUN that can be shared by multiple Windows hosts. Only one Windows host will have read/write access (defined as “attached” within AutoStart) to the disk at a given time. When a node does not have access to a disk, the device is marked as not ready by the AutoStart filter driver. The not ready state will prevent unwanted access to the LUN for the nodes where the resource is detached. To use the shared disk device for the Windows data source, the disk must be created as a Basic disk. Either MBR or GPT partitioning can be used.

Mount point data source

A mount point data source is simply a LUN that is mounted as a folder within a directory structure. AutoStart supports the use of mount points in combination with the shared disk device data source. The shared disk data source will represent the root of the mount point, while the mount point data source represents the LUN and folder mounted within the root. The startup sequence will be such that the shared disk data source comes online first, followed by the mount point data source.

EMC SRDF/S mirroring data source

SRDF/S provides for synchronous replication between LUNs between multiple Symmetrix® storage arrays. Microsoft support for third-party replication requires that the replication mechanism be synchronous. This limits Microsoft support to SRDF/S. A mount point or shared disk data source should be configured above the SRDF/S data source to provide access control.

Much of the DAG framework remains the same for when third-party replication is enabled. The DAG will continue to operate as a node majority cluster. AutoStart will support the maximum number of nodes within a DAG of 16 and the maximum number of 100 databases per mailbox server.
DAG Creation

A DAG is created in much the same way, with the exception of marking the DAG as having “thirdpartyreplication” enabled. Database copies are created within the DAG like with native Exchange replication; however, the term “copy” takes on a new meaning when AutoStart is used.

When a database copy is added to a DAG configured for third-party replication, no database seeding process will occur and the native log shipping mechanism otherwise used by Exchange will be disabled. A database copy under AutoStart will represent either a shared disk or a copy as replicated with SRDF/S. A copy under shared disk functionality will represent the same physical copy (as protected by RAID 1, RAID 5, or RAID 6 within the storage array) of the database across multiple nodes. The only true “copy” in a physical sense will be the copy synchronously replicated by SRDF/S to an alternate Symmetrix storage array, respectively. Figure 65 on page 270 provides a logical overview for how AutoStart can allow for shared storage as well as remote synchronous storage management within a DAG.

The active managers will also continue to provide failure detection for databases and Exchange-specific processes within a DAG where third-party replication is enabled. While the active managers continue to monitor the environment, they no longer control which database copy should be active. When a failure is detected, the active manager will alert AutoStart, which essentially becomes an extension to the PAM, and will then control where databases will become active. By default AutoStart will first move active database copies to nodes with shared disk device access before moving resources to remote nodes that maintain copies as replicated by SRDF/S. AutoStart will also continue to monitor the environment and can initiate movement of active databases between nodes independent of the active managers.
Prior to configuring the Exchange Server 2010 DAG and AutoStart environment, several requirements should be met. The following headings outline these requirements:

Log and database file placement

Exchange Server 2010 has removed the concept of a storage group as existed for previous Exchange versions and now supports up to 100 discrete database instances (database and log pairs) per server. For support with AutoStart, each database instance must reside on dedicated LUNs as presented to the Windows operating system. AutoStart performs the movement of active databases at the LUN level, therefore to maintain database level granularity for moves between servers, a database must reside on its own LUNs. A database file and log directory combination for a given database can share a LUN or be on separate LUNs.
AutoStart requires that all LUNs to be used for either shared disk access or replication be zoned, mapped, and masked to the nodes participating in the DAG. As stated previously, shared disk access requires that the LUN be configured as a basic disk with either MBR or GPT partitioning. For the initial configuration of the shared disk resources, each node must mount its respective LUNs to the same mount points on all servers in the DAG.

For SRDF/S data sources to be configured, the replication pairs must first be defined prior to adding the data source within AutoStart. With SRDF/S this entails creating the SRDF/S R1 and R2 relationship. Additionally for SRDF/S, a device group (DG) or composite group (CG) must be created either locally on each appropriate node within the DAG, or configured within Group Name Services (GNS) as maintained globally in the Symmetrix. The requirement to manually create a DG or CG may be changed in a future release of AutoStart; please check release notes for details. Also for SRDF/S, it is recommended to present two unique gatekeepers on all nodes for each configured data source.

As previously noted, for Microsoft support of the third-party replication environment, it must be synchronous replication, thus SRDF/S for Symmetrix.

Further information can be found on Power link in the White Paper: *EMC AutoStart with Microsoft Exchange Server 2010—Applied Technology.*
This chapter includes the following topics:

- Introduction ........................................................................................ 274
- The Exchange Server 2010 challenge in designing storage ....... 277
- Exchange Server 2010 disk I/O patterns and behavior......... 279
- Non-user-transactional workloads and patterns.................... 288
- Exchange 2010 performance tuning ................................................. 291
- Considerations and planning for storage design............... 295
- Storage architecture and I/O principles for EMC Symmetrix ... 307
- Storage provisioning for Exchange Server 2010...................... 314
- Summary ............................................................................................. 327
Introduction

Messaging in general has evolved for many organizations to the point that today, it is considered to be one of, if not the most mission-critical application within the enterprise. As a result, it should be no surprise that the messaging data that these organizations create and maintain on a routine basis has exploded exponentially in terms of physical size, scope, and context. The requirement to size, configure, and protect this burgeoning mission-critical messaging data has similarly increased in recent years, forcing many organizations to struggle with balancing the pressures of maintaining optimally performing systems for routine messaging operations (daily e-mail), while providing protection and availability in the event of a disaster. These pressures can constrain an organization’s ability to maintain these messaging infrastructures. Consider the following:

- **Availability pressures**—The need to protect the data center from disaster. Many large organizations require 24/7 access to their messaging infrastructure. Today, many enterprise-class messaging deployments need to continue business operations even in the event of a total loss of the data center from some form of rolling disaster while maintaining zero data loss.

- **Regulatory pressures**—Legislation (Sarbanes-Oxley Act) has necessitated the need for e-mail archiving and almost instantaneous message retrieval.

- **Environmental pressures**—E-mail virus activity has been insidious and continues to plague corporate e-mail environments, costing millions of dollars in downtime and lost productivity. Additionally, today’s environments are plagued with spam and e-mail phishing scams. Policy-based spam filters can put additional front-end pressures on messaging infrastructure.

- **New technological pressures—the e-mail everywhere effect**.—Wireless-enabled Exchange Server 2010 mailboxes require constant and persistent connections that often translate to more disk activity.

All of these conditions are placing significant performance pressures on today’s enterprise-class messaging infrastructure. Exchange Server performance can generally be characterized or quantified in many different ways depending on the role or function that the
specific Exchange Server performs. Exchange mailbox servers, for example, perform different functions and tasks than the Exchange Hub Transport or Client Access servers. Similarly, the performance metrics used to evaluate each server type vary depending on their respective roles. Ultimately, however, it is the end user’s perception of performance that will dictate whether the Exchange mailbox server is performing “well.”

End-user client performance may be characterized by how fast the client/user can write to the Exchange Server transaction log (submits/transaction “write” activity) and how fast the client can read from, and update (write to), the Exchange Server database files (read/write). These physical disk-related operations are processed on Exchange mailbox servers. The Exchange mailbox server must be able to process all database-related I/O operations that include all end-user related transactions as well as database backup and maintenance operations. It is critical that the storage subsystem used to deploy the Exchange mailbox server is properly sized and configured in order to meet these physical disk-related workloads in accordance with Microsoft recommendations with respect to physical disk I/O response times.

A properly sized and configured Exchange server from a storage subsystem perspective must be able to satisfy all of the transactional demands of its end users during peak business hours, while meeting the demands of Exchange-related backup and maintenance operations during nonbusiness hours. For Exchange Server, satisfying both workloads is fundamental to any successful deployment of the application within the Enterprise. Not being able to satisfy both of these unique physical disk workloads can have serious consequences for an enterprise messaging deployment.

Sizing and configuring storage for Exchange Server 2010, therefore, generally revolves around understanding or quantifying the application’s requirements from two physical disk workload perspectives:

- Microsoft Exchange End-User transactional requirements—often expressed as disk input/output operations per second (IOPS) requirements.
- Microsoft Exchange Non-transactional requirements—often expressed as megabytes or gigabytes per second (data rate) for workloads such as backup operations or Microsoft Exchange checksum verification operations (eseutil.exe /k/i).
Therefore, the focus of this discussion is the sizing and configuring of the EMC Symmetrix storage subsystem when used with Exchange Server 2010, in order to meet these demands. Finally, as new versions of Exchange Server or service packs for Exchange Server are released, and as EMC Symmetrix array technology advances, best practice information for Exchange storage design will evolve.
The Exchange Server 2010 challenge in designing storage

The Exchange Server 20xx disk I/O workload has been historically characterized as a very random bursty (mostly 4 KB random access) read and asynchronous write operation to the database files with a sequential, synchronous (mostly 512 byte) write operation to the transaction logs. Access to the Microsoft Exchange database files has generally been observed to maintain a 70:30 or 60:40 read-to-write ratio, while the access pattern to the Microsoft Exchange transaction logs is almost always 100 percent writes.

Whilst the random nature has largely remained the same for Exchange Server 2010 the I/O characteristics changed in that the page size for the database was increased to 32KB. Also due to the 64-bit server architecture and the subsequent increase in server RAM available to the database cache (detailed in chapter 1) the overall I/O to the disk has been reduced. The read-to-write ratio for database access is 60:40. For single Exchange server configurations the log files are 100 percent write, while for configurations using DAG there is a small amount of read operations on log files as they are read for the log transmission to update the data replica(s).

For Exchange Server 2010 the random nature of accessing the database has remained, however the new structure of the database leads to more sequential writes under certain circumstances, typically when background database scanning and online defragmentation is configured to run 24x7. As already stated in "Exchange Server Store Instances" on page 42 the page size for Exchange Server 2010 has been increased to 32 KB. The page size is the minimum size for reading and writing to the database; it’s also the unit size used for database caching. Reading from the disk is slower than performing operations in memory; therefore, by increasing the page size to 32 KB, ESE reduces IOPS, which increases performance by caching the larger page size in memory. The typical read/write ratio for Exchange Server 2010 databases in a DAG configuration is 60:40.

The one exception to the transaction log access pattern of 100 percent writes occurs during a Exchange Server dirty shutdown, or crash, condition. In this case, transaction log files must be replayed into the database for database recovery operations. This recovery operation is often overlooked by storage administrators and can result in longer recovery times during Exchange Server recovery operations.
Physical disk read operations to the Exchange Server database volumes can be 60 to 70 percent read miss, while write operations to the database volumes and transaction log volumes are 100 percent EMC Symmetrix cache hit. This general workload is made more challenging as there are often large discrepancies between peak and nonpeak workloads.

It is this random, and at times, bursty workload with periods of high peaks that makes designing a well-performing storage solution with Exchange Server a challenge. The storage administrator must also take into account backup operations, as this routine represents different I/O access patterns than normal transactional operations. Finally, while the observations mentioned previously are general in nature, every enterprise-class deployment of Exchange Server is unique. Different corporate environments have different user and storage requirements, making the use of generalizations for storage design a risky proposition.

EMC Symmetrix storage systems provide a great deal of flexibility and functionality in designing storage solutions for use with Exchange Server. EMC Symmetrix RAID types, disk types and replication technologies, as well as Symmetrix front-end (FAs) and back-end (DAs) configurations, must all be considered when designing a solution because of the heavy demands Exchange Server can place on the storage subsystem. Hardware component failure (disk or board failure), while generally not common, is often overlooked and should be factored into any design, as this impairs performance until the failure has been isolated and repaired.
Exchange Server 2010 disk I/O patterns and behavior

**Transaction processing**

When a user attempts to make modifications to any of the databases contained in the specific ESE98 instance, the calling thread will start the transaction and proceed to perform the necessary updates or page modifications. These page modifications are first stored in the transaction log buffers, and the corresponding database pages are then modified in main memory, database cache (ESE cache). Modified or updated database pages in memory should then be considered “dirty” as these pages now contain user-generated updates that are considered “owed” to the physical database file on disk (the Exchange Property Store). When all of the appropriate page modifications have been made, the calling thread will commit the transaction to the database. This process may be described as an “all or none” process in that either all of the corresponding page modifications are made or none are made at all.

The Exchange Server Version Store described in “ESE functionality improvements for Exchange Server 2010” on page 38 ensures that while a given transaction is in progress, the relative database pages that the transaction will eventually modify, will be seen in their original content by all other working threads of the Store. Read and write coalescing to the Exchange Server 2007 and Exchange Server 2010 databases have increased from 64 KB to 1 MB, which is another aid to reducing the overall I/O going to the disk by increasing the opportunity to read and write larger I/O.

The process of committing a transaction consists of writing sequentially the changes made in the transaction log buffers to the transaction log files on physical disk. When the commit transaction buffer has been written to the physical transaction log volume, the calling thread can assume that the database modification has been completed and may resume execution. Therefore, if the process of writing to the transaction log drive is slow, the user may experience slow response times in addition to the Exchange Server experiencing lower overall throughput.

**Transaction logging for Microsoft Exchange**

Transaction logging for Exchange Server 2010 is defined as the process of writing the data that has been previously written to the log buffers and subsequently writing this data sequentially to the physical volume containing the transaction log files. The transaction log update should be generally understood to be directly in the
critical path of a user-generated update and therefore is directly related to that user’s perception of a well-performing system, as well as a key metric in determining the overall health of an Exchange Server. The biggest change to logging for Exchange Server 2007 and Exchange Server 2010 was the reduction of the log file size to a default of 1 MB, from a default of 5 MB in Exchange Server 2003. The size of the log files is configurable between 1 MB and 5 MB.

Microsoft Exchange 2000 first introduced the concept of log buffer gathering, which is the process of grouping separate log writes into a single autonomous physical disk write operation to make these operations more efficient. As an example, assume the case where Exchange was obligated to write five records each at 1 KB in size. Without write gathering, Exchange would be required to process five autonomous physical disk write operations as opposed to the same operation leveraging write gathering where 1 x 5 KB disk write operation is performed. This translates to different transaction log behavior for Exchange 2000 as compared to previous versions of Exchange in that the actual disk I/O size could vary in size, and in fact could be as large as 64 KB and greater if the workload requires.

For normal user-generated transactional processing, the physical I/O access patterns to the Exchange Server 2010 transaction logs have been typically observed to be 512-byte synchronous and sequential write operations to the physical disk. An exception to the typical transaction log access pattern of writes occurs during a Exchange Server 2010 dirty shutdown or crash condition. In this case, transaction log files must be replayed into the database for database recovery operations and results in a significant disk read workload for the transaction log volume. This recovery operation is often overlooked by storage administrators and can result in longer recovery times during Exchange Server 2010 recovery operations.

The default size of each Exchange Server 2010 transaction log file generation is approximately 1 MB. As Exchange Server 2010 transactions are processed, they are written to the current active transaction log generation (E000000000x.log) until the 1 MB transaction log is filled and a subsequent log file is generated. Assuming that Microsoft Exchange Circular Logging is disabled (and it is by default on Exchange Server 2010 for the purposes of disaster recovery), these transaction log files accrue on the transaction log volume until a Microsoft Exchange “full” or “incremental” online backup is performed. Additionally, EMC hardware-assisted VSS
backup processes will also truncate the already committed transaction log files; that is, log files that have previously been committed to the Exchange Server 2010 database files.

### Transaction log replay

The synchronous write disk I/O patterns observed as part of normal transactional processing for the transaction log volume will differ greatly in behavior in the event that the transaction log files are required to be replayed into a database during recovery operations when a recovery situation occurs. During Microsoft Exchange recovery operations it is very likely that some form of transaction log replay will have to occur in order for the Exchange database in question to be mounted. As a result, the disk I/O pattern normally observed on the physical volume changes from a sequential write workload to a sequential read operation as transaction log files are first read in order and subsequently replayed into the database.

The speed in which these transaction logs can be replayed into a recovering database has been much improved with the release of Exchange Server 2010. Some transaction log files may contain transaction data that is “nested” or referentially related to data in other log files that can increase the replay time. An example of this would be a “commit” operation for a long running transaction, one that spans more than one transaction log file. This process is also reliant on the ability of the volumes that contain the transaction logs and database files to process this data as quickly as possible as it is replayed.

As noted earlier, a crucial determining factor for Microsoft Exchange database performance is transactional throughput to the Microsoft Exchange transaction log volume. The ability to scale users for a given Exchange the databases store is determined by the transaction log volumes that support the I/O workload of the active users. In fact, the most often diagnosed cause of poor database performance has been the inability of the transaction log to support a given number of active users and subsequently the IOPS these users generate. Microsoft Exchange and Storage Administrators should view a well-performing transaction log volume not for its ability to sustain a high MB throughput, but rather by its ability to sustain high I/O rates at very fast speeds.

Microsoft Corporation has published some key metrics for understanding Exchange Server 2010 performance. These Microsoft Windows Server Perfmon counters are important determinants in isolating Exchange Server 2010 performance problems and should be considered guidelines for acceptable performance.
Table 13 and Table 14 on page 282, and Table 15 on page 283, summarize these key Microsoft performance counters for the Exchange Server 2010 active, passive and log latency requirements for Exchange Server 2010.

### Table 13  Recommended Mailbox server performance counters—Active database copy I/O

<table>
<thead>
<tr>
<th>Latency Requirements Counters</th>
<th>Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSEExchange Database\I/O Database Reads (Attached) Average Latency</td>
<td>The average value should be below 20 ms. Spikes (maximum values) shouldn't be higher than 100 ms</td>
</tr>
<tr>
<td>Indicates the average time, in ms, to read from the database file.</td>
<td></td>
</tr>
<tr>
<td>MSEExchange Database\I/O Database Writes (Attached) Average Latency</td>
<td>In general, this latency should be less than the MSEExchange Database\I/O Database Reads (Attached) Average Latency.</td>
</tr>
<tr>
<td>Indicates the average time, in ms, to write to the database file.</td>
<td></td>
</tr>
<tr>
<td>Database\Database Page Fault Stalls/sec</td>
<td>This counter should be 0 on production servers.</td>
</tr>
<tr>
<td>Indicates the rate of page faults that can't be serviced because there are no pages available for allocation from the database cache.</td>
<td></td>
</tr>
</tbody>
</table>

### Table 14  Recommended Mailbox server performance counters—Active Log I/O

<table>
<thead>
<tr>
<th>Latency Requirements Counters</th>
<th>Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSEExchange Database\I/O Log Writes Average Latency</td>
<td>This counter should be 10 on production servers</td>
</tr>
<tr>
<td>Indicates the average time, in ms, to write a log buffer to the active log file</td>
<td></td>
</tr>
<tr>
<td>Database\Log Record Stalls/sec</td>
<td>The average value should be below 10 per second.</td>
</tr>
<tr>
<td>Indicates the number of log records that can't be added to the log buffers per second because the log buffers are full.</td>
<td>Spikes (maximum values) shouldn't be higher than 100 per second.</td>
</tr>
<tr>
<td>Database\Log Threads Waiting</td>
<td>The average value should be less than 10 threads waiting.</td>
</tr>
<tr>
<td>Indicates the number of threads waiting to complete an update of the database by writing their data to the log.</td>
<td></td>
</tr>
</tbody>
</table>
After all of the appropriate and corresponding database pages have been modified in database cache (now marked as "dirty", requiring flushing), ESE98 will attempt to asynchronously write these dirty database pages now residing in database cache to the physical database files on disk. ESE98 will flush these “dirty” database pages to disk as soon as possible (or as soon as ESE98 gets around to it). The mechanism used to accomplish this process is referred to as the lazy writer. As the name implies, the lazy writer manages the process of destaging dirty pages to the database volumes when ESE98 is able or required to, or, put more appropriately, when sufficient host server I/O bandwidth is available or when the database checkpoint depth has been reached. The process of periodically flushing these dirty pages to the database disks using the lazy writer is referred to as database checkpointing. ESE98 attempts to balance ESE-generated I/O through a leveling algorithm that attempts to flush dirty pages to disk or checkpoint without overloading the disk subsystem.

This database checkpointing process has incrementally evolved with every release of Exchange Server. For example, with Exchange Server 2010, the Exchange Lazy Writer was observed to be running on a timer basis, which was approximately every 30 seconds, potentially issuing in parallel up to 64 x 4 KB disk, individual write I/Os to the database volumes. As a result, it was not uncommon to observe fairly frequent, large bursts of asynchronous disk write activity that could potentially

---

<table>
<thead>
<tr>
<th>Latency Requirements Counters</th>
<th>Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSEExchange Database I/O Database Reads (Recovery) Average Latency</td>
<td>The average value should be below 20 ms. Spikes (maximum values) shouldn't be higher than 100 ms.</td>
</tr>
<tr>
<td>Indicates the average time, in ms, to read from the database file</td>
<td></td>
</tr>
<tr>
<td>MSEExchange Database I/O Database Writes (Recovery) Average Latency</td>
<td>In general, this latency should be less than the MSEExchange Database I/O Database Reads (Recovery) Average Latency.</td>
</tr>
<tr>
<td>Indicates the average time, in ms, to write to the database file</td>
<td></td>
</tr>
<tr>
<td>Database/Database Page Fault Stalls/sec</td>
<td>This counter should be 0 on production servers.</td>
</tr>
<tr>
<td>Indicates the rate of page faults that can't be serviced because there are no pages available for allocation from the database cache.</td>
<td></td>
</tr>
</tbody>
</table>
impact the physical disks ability to process synchronous user-related
disk read operations. Beginning with Exchange Server 2000, the
Exchange Lazy Writer will flush these “dirty” database pages using a
threshold approach, referred to as the checkpoint depth instead of a
timer-based operation. This implies that for Exchange Server 20xx
these database pages will be flushed or written to the physical
database volumes when the checkpoint advances, which largely
depends on the number of outstanding transactions (owed to the
database files on disk), and the checkpoint depth itself. A large
checkpoint depth ultimately attempts to limit ESE98 write activity by
avoiding frequent access to the database physical disks.

The second improvement to the checkpointing process for Microsoft
Exchange is in functionality commonly referred to as “write
coalescing.” Write coalescing is the process by which ESE will
attempt to write together dirty database pages that are similar in
nature to the write gathering that takes place for transaction log
activity. Write coalescing attempts to reduce the number of physical
disk I/O operations and increase the overall throughput to the
physical database volumes.

One of the most significant changes is increasing the log checkpoint
depth target. The log checkpoint depth target is used to ensure that
changes made to the log/database cache are written to the database
file in a reasonable amount of time. It has been increased from 20 MB
per database in a stand-alone mailbox server to 100 MB per database
when a database has more than one copy (in a database availability
group, or DAG). Table 16 provides the default log checkpoint depth
targets for Exchange 2010.

<table>
<thead>
<tr>
<th>Database Configuration</th>
<th>Log checkpoint depth target (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone (one database copy)</td>
<td>20</td>
</tr>
<tr>
<td>Mailbox in DAG</td>
<td>100</td>
</tr>
<tr>
<td>Passive database copy</td>
<td>5</td>
</tr>
</tbody>
</table>

Due to this change, the database write I/O for an active database
with two or more copies can be up to 40 percent less than the
database write I/O for a stand-alone database. When the database
has a higher checkpoint depth target, it’s able to retain database file
changes in memory for a longer period; thus, improving its ability to
combine I/Os (coalescing) and by reducing repeated write I/Os (I/Os that can be saved by delaying the write long enough so multiple database changes can be made in memory prior to writing the change to the database file).

This change was only made for mailbox resiliency solutions because a deeper checkpoint depth target can significantly increase the time it takes the active database to recover the failed database after a failure. This problem has been addressed in mailbox resiliency configurations because if the active database fails, a failover is automatically triggered to another healthy copy. Log replay operations will resume when the failed database is recovered.

The checkpoint depth target has subsequently been reduced on passive database copies to reduce the time a database switchover/failover takes. A passive database copy with a 5 MB checkpoint depth target can be activated much faster than a copy with a higher checkpoint depth target (there are fewer logs to play through when transitioning between the passive and active state). A passive database copy does not have transactional I/O so there is spare capacity to handle the additional write I/O with the lower Checkpoint Depth Target.

The Exchange ESE arranges database pages in a balanced binary tree format referred to as B+Trees. The manner in which these database pages are stored on the physical disk inside the Exchange EDB file is quite different from their organization within the “balanced tree.” That is, when an end user attempts to write a new e-mail to an Exchange database of 100 GB in size, for example, the Store could potentially update any single database page in the physical database file, depending on the page allocation scheme.

The behavior is often described as “random disk I/O across the entire database seek range”. This is an important distinction for Exchange Server 2010 database behavior over other relation database applications in that, for Exchange Server 2010, there is no concept of a separate index area and a separate data area that would favor cache-centric storage arrays such as the EMC Symmetrix.

The physical disk I/O patterns for transactional operations such as reads and writes are typically:

- Database disk writes—Generally observed to be ~34 KB for writes.
Database disk reads—Generally observed to be ~80 KB for the majority of the I/O.

Exchange Server 2010 database files are created with and grow using 16 MB physical extents.

The Exchange Server 2010 database I/O read/write ratio differs depending on the number messages sent and received and whether the server is a stand-alone mailbox server or is participating in a DAG environment. This does not include log volume I/O.

With Exchange Server 2010, the larger database the cache will decrease the number of reads to the database on disk causing the reads to shrink as a percentage of total I/O.

Exchange Server 2010 disk read activity is further characterized as a fairly high read-miss activity approaching 70 percent in some environments. The read-miss workload is a critical consideration for EMC Symmetrix deployments given that disk read operations not already contained in EMC Symmetrix cache must be serviced from disk, and subsequently read from EMC Symmetrix cache once the physical disks have provided the data for the end-user read operation. Read-miss operations should be considered physical-disk-intensive operations.

### Table 17  Mailbox database I/O read/write ratios

<table>
<thead>
<tr>
<th>Messages sent/received per mailbox per day</th>
<th>Stand-alone databases</th>
<th>Databases with mailbox resiliency</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>1:1</td>
<td>3:2</td>
</tr>
<tr>
<td>100</td>
<td>1:1</td>
<td>3:2</td>
</tr>
<tr>
<td>150</td>
<td>1:1</td>
<td>3:2</td>
</tr>
<tr>
<td>200</td>
<td>1:1</td>
<td>3:2</td>
</tr>
<tr>
<td>250</td>
<td>1:1</td>
<td>3:2</td>
</tr>
<tr>
<td>300</td>
<td>2:3</td>
<td>1:1</td>
</tr>
<tr>
<td>350</td>
<td>2:3</td>
<td>1:1</td>
</tr>
<tr>
<td>400</td>
<td>2:3</td>
<td>1:1</td>
</tr>
<tr>
<td>450</td>
<td>2:3</td>
<td>1:1</td>
</tr>
<tr>
<td>500</td>
<td>2:3</td>
<td>1:1</td>
</tr>
</tbody>
</table>
The Exchange Server 2010 database may also be characterized as having a very random, bursty (mostly 32 KB random access) asynchronous, nonblocking write operation to the database files. Additionally, disk read operations although random over the physical database files may be described as synchronous from the end-user experience, in that performance is largely driven from a user's ability to “read” e-mail from the Exchange Server database files. Thus, while Exchange end users are not necessarily waiting from a client perspective for these nonblocking asynchronous writes to be processed, when a user attempts to read from the database, the e-mail client will wait for the operation to complete. The term nonblocking is used here to imply that these database write operations can be flushed to disk in bulk format rather than the sequential synchronous nature inherent in transaction log writes (users are not waiting for these writes to complete). However, because these asynchronous writes must be processed in bulk during database checkpoints, these bursts of asynchronous disk write operations can impact the physical disks ability to process user-related disk read operations, therefore impacting end-user client latencies.

The transactional processing summary is as follows:

- The calling thread starts the transaction that is comprised of one or more database page modifications.
- The first page to be modified is processed in main memory (original page is saved in the version store).
- The first modification gets written to the transaction log buffer.
- Additional database pages are modified if necessary, for example, in the case of a large message with attachments.
- The specific calling thread commits (wait).
- The transaction log buffers are sequentially written to the transaction log file on physical disk one after another in order.
- The specific calling thread resumes execution (the commit transaction is accepted).
- The version store is purged from the original database pages which were initially stored at the start of the transaction or update.
- The corresponding modified database pages are subsequently flushed at a later point in time by the Exchange Lazy Writer at Database Checkpoint time.
Non-user-transactional workloads and patterns

Exchange Server backup operations

The additional workloads caused by backup operations occurs during an EMC Hardware Assisted VSS Backup Operation. These incorporate a physical checksum of the database and log files as required in complying with the Microsoft VSS framework. This checksum process, usually performed on EMC Symmetrix TimeFinder BCV mirrored devices and clones can be a disk-intensive process as this operation represents a 100 percent large sequential disk read activity performed in 64 KB disk I/Os.

Chapter 4, “Backing Up Exchange Server 2010,” provides further guidance; however, some general recommendations for BCV or clone volumes and layout is discussed in the following sections.

TimeFinder and sharing spindles with Exchange

Storage group cloning is useful when Exchange Administrators wish to create backup or other business continuance images of a database. A common question when laying out Exchange databases is whether BCVs or clones should share the same physical spindles as the production volumes or whether the BCVs or clones should be isolated on separate physical disks. There are pros and cons to each of the solutions; the optimal solution generally depends on the anticipated workload.

The primary benefit of spreading BCVs or clones across all physical spindles is performance. By spreading I/Os across more spindles, there is a reduced chance of developing bottlenecks on the physical disks. Workloads that utilize BCVs or clones, such as backups, may generate high I/O rates. Spreading this workload across more physical spindles may significantly improve performance in these environments.

The main drawbacks to spreading BCVs or clones across all spindles in the Symmetrix are that synchronization may cause spindle contention during resynchronization, and that BCV or clone workloads may negatively impact production system performance. When resynchronizing the BCVs or clones, data is read from the production hypervolumes and copied into Symmetrix cache. From cache, it is then destaged to the BCV or clone. When the physical disks share both production and BCVs or clones, the synchronization rates can be greatly reduced because of increased seek times due to the conflict between reading from one part of the disk and writing to another. Another drawback of sharing physical disks is the increased
workload on the spindles that may impact performance on the production volumes. Sharing the spindles increases the chance that contention may arise, decreasing overall system performance.

Determining the appropriate location for BCVs or clones, whether sharing the same physical spindles or isolated on their own disks, depends on customer preference and workload. In general, it is recommended that the BCVs or clones not share the same physical spindles for high-end Exchange Server 2010 deployments especially for those leveraging VSS hardware-assisted backups with TF/EIM and RM as a result of the additional sequential disk read I/O activity incurred with the checksum process executed with ESEUTIL.

**Exchange Server 2010 page zeroing**

By default, most storage systems (file systems and databases) don’t write over the actual data when it’s deleted. They delete the pointer to the data and add the pages and blocks backing the data to a free or available list. The data is eventually deleted when the pages and blocks are re-used. Data zeroing is a mechanism that writes either zeros or a binary pattern over deleted data in an attempt to make the data much more difficult to recover. This action is taken for security reasons. Data zeroing occurs prior to the pages and blocks being re-used by the storage system. Only the Mailbox database file (.edb) has provisions for page zeroing. Neither the transaction logs or the Context index catalog files have provisions for page zeroing.

In previous versions of Exchange Server 2010, page zeroing occurred only during a backup or scheduled maintenance process (when configured), and caused significant database disk I/O, however in Service Pack 1 for Exchange Server 2010, page zeroing is on by default. There is no mechanism to disable it. Page zeroing operations are recorded in the transaction log files so that all copies of a database are page-zeroed in a similar manner. That is, zeroing a page on the active database causes the page to get zeroed on a passive database after the passive database replays the transaction log with the page zeroing log record. There is no mechanism for the ESE to prioritize the reutilization of zeroed pages over allocating new space. Tables which have sequential space allocation assigned will intentionally skip fragmented or zeroed pages in favor of using new or free sequential pages. This approach reduces the database I/O footprint of the server.

There are 2 improvements that page zeroing in SP1 of Exchange Server 2010 gives over the RTM version of Exchange Server 2010. These are:
Optimized storage and network capacity—The Extensible Storage Engine (ESE) writes a page-zeroing record to the transaction log file instead of logging the entire page image. This approach reduces log write I/O, keeps the capacity footprint of the logs as low as possible, and reduces the bandwidth requirements to ship the logs from active to passive copies.

Optimized database disk I/O—Page zeroing occurs by default and happens primarily at transaction time. For the majority of cases, the zeroing occurs immediately after the hard delete. This design allows the database to utilize the checkpoint depth capability of the engine, which ensures dirty pages stay in cache for a certain amount of time so additional page updates that occur in close time proximity don't cause additional database write I/O. Because of this design, page zeroing has no significant database I/O impact on a production server.

Background Database Maintenance

Configured by default, background database maintenance is a process which continuously checksums and scans the database in the background. Its primary function is to checksum the database pages, but it also handles cleaning up after Exchange Server 2010 Store crashes (cleaning up space and zeroing out records and pages which did not occur due to the crash). Background database maintenance processes approximately 5 MB per second per database. If timely page zeroing is a priority database sizes can be reduced to ensure page zeroing occurs for the crash recovery cases in a shorter time period. Background database maintenance runs continuously so unlike previous versions of Exchange there is no start or stop events registered in the event log. The process of background database maintenance can be tracked by using the MSExchange Database =>Instances=>Database Maintenance Duration: counter in Windows performance monitor.
Exchange 2010 performance tuning

As previously stated Exchange Server 2010 is a true 64-bit platform and should only be implemented into production environments as such. As a result of having a 64-bit architecture, Exchange Server 2010 enables much better memory utilization than previous versions of Exchange Server. For example, because of the virtual address space limitations of a 32-bit platform, Exchange Server 2003 is limited to using 4 gigabytes (GB) or less of physical memory. In contrast, Exchange 2010 can use many GB of memory. It is not uncommon to see Exchange Server 2010 configurations of more than 32 GB of RAM.

Table 18  Memory configurations for Exchange 2010 servers based on installed server roles

<table>
<thead>
<tr>
<th>Exchange Server 2010 Server Role</th>
<th>Minimum per Server</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge Transport</td>
<td>4 GB</td>
<td>1 GB per processor core (4 GB min)</td>
</tr>
<tr>
<td>Hub Transport</td>
<td>4 GB</td>
<td>1 GB per processor core (4 GB min)</td>
</tr>
<tr>
<td>Client Access</td>
<td>4 GB</td>
<td>2 GB per processor core (8 GB min)</td>
</tr>
<tr>
<td>Unified Messaging</td>
<td>4 GB</td>
<td>2 GB per processor core (4 GB min)</td>
</tr>
<tr>
<td>Mailbox</td>
<td>4 GB</td>
<td>2 GB per processor core plus between 3 MB and 30MB per mailbox based upon user profile and database cache size</td>
</tr>
<tr>
<td>Client Access/Hub Transport</td>
<td>4 GB</td>
<td>2 GB per processor core (8 GB min)</td>
</tr>
<tr>
<td>Multi-Role (combinations of HTS, CAS and Mailbox)</td>
<td>8 GB</td>
<td>8 GB per processor core plus between 3 MB and 30MB per mailbox based upon user profile and database cache size</td>
</tr>
</tbody>
</table>

Mailbox server role memory requirements

The memory configuration process for the Mailbox server role is more complex than the other roles because the optimal memory configuration depends upon the mailbox count and the client profile (similar to estimating processor core requirements). Memory sizing for the Mailbox server role is critical to reducing disk input/output (I/O) on the server. The more memory is added to the Mailbox server, the less disk I/O that will be generated by Exchange. There is, however, a point of diminishing returns at which adding memory to the server may not be justifiable based on price and performance.
Defining the memory configuration of a Mailbox server is required prior to defining the storage requirements and configuration for that server. Table 19 can be used to estimate the user IOPS requirements of a specific Mailbox server with a specific number messages sent and received, whether it participates in a DAG or is stand-alone.

Table 19 Estimated IOPS per mailbox based on message activity and mailbox database cache

<table>
<thead>
<tr>
<th>Messages sent/received per mailbox per day (~75KB average message size)</th>
<th>Database cache per user (MB) for Single database copy (Stand-alone):</th>
<th>Estimated IOPS per mailbox for Multiple database copies (DAG):</th>
<th>Estimated IOPS per mailbox Stand-alone server</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>3</td>
<td>0.06</td>
<td>0.05</td>
</tr>
<tr>
<td>100</td>
<td>6</td>
<td>0.12</td>
<td>0.10</td>
</tr>
<tr>
<td>150</td>
<td>9</td>
<td>0.18</td>
<td>0.15</td>
</tr>
<tr>
<td>200</td>
<td>12</td>
<td>0.24</td>
<td>0.20</td>
</tr>
<tr>
<td>250</td>
<td>15</td>
<td>0.30</td>
<td>0.25</td>
</tr>
<tr>
<td>300</td>
<td>18</td>
<td>0.36</td>
<td>0.30</td>
</tr>
<tr>
<td>350</td>
<td>21</td>
<td>0.42</td>
<td>0.35</td>
</tr>
<tr>
<td>400</td>
<td>24</td>
<td>0.48</td>
<td>0.40</td>
</tr>
<tr>
<td>450</td>
<td>27</td>
<td>0.54</td>
<td>0.45</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>0.60</td>
<td>0.50</td>
</tr>
</tbody>
</table>

Servers now on the market have the ability to scale their memory configurations to 32 GB to 64 GB and even to 128 GB and beyond. There are many reasons why Exchange Server 2010 is not recommended to run on server with limitless amounts of RAM. These are:

- **Cost** - Although the cost of server RAM is now relatively low it is still a significant investment to install more than is required.
- **Nontransactional I/O** - The Mailbox server utilizes additional physical RAM by caching more data, which reduces the database disk I/O footprint for transactional disk I/O (I/O that is generated by client activity). There are several sources of nontransactional disk I/O on the Mailbox server. These include online maintenance, offline maintenance (for example, offline
database defragmentation or database repair operations), backup, restore, or recovery operations, and mailbox management operations. All of these operations require disk I/O to properly maintain the server. Although Exchange Server 2010 has reduced transactional I/O significantly, adequate storage performance is still required for proper maintenance of the Mailbox server. For this reason, there is a point of diminishing returns when adding memory to the server. In general, the purpose of adding memory to the Mailbox server is to reduce the disk I/O requirements.

- **Cold state operation** - Cold state is defined as the state of the Mailbox server immediately following a server restart or a restart of the Microsoft Exchange Information Store service. The database cache, which is used to cache read/write operations, is small in size (cold) during this period, so it has a significantly diminished ability to reduce read I/O operations. As the Mailbox server processes messages, the database cache size grows, increasing the effectiveness of the cache and subsequently reducing disk I/O on the server. The more physical memory in the server, the longer it takes the database cache to reach its optimal size. If the storage solution is designed and sized for a server with a large amount of physical RAM (greater than 32 GB), and the disk I/O profile of the users assumes an optimal database cache state (for example, a large, warm cache), the client experience may be compromised due to insufficient disk performance during the cold state periods. Similar to the issue of nontransactional I/O, the storage requirements may be the same for a server with 32 GB of memory as a server with more than 32 GB of RAM. On a properly configured Mailbox server, it should take about 15 minutes to reach the optimal cache state after a cold operation has occurred.

As previously stated, the maximum number of stores per server in Exchange Server 2010 has been increased to 100. This increase provides much greater flexibility in server and storage architecture, but this increase also has a significant effect on the memory utilization of the Mailbox server. Increasing the number of stores primarily affects the utilization of the database cache. The database cache is used for both reads and writes. Due to the way checkpointing works, adding a store increases the amount of the database cache used for write activity. This has a positive impact of reducing database write I/O, but if too many storage groups are configured on a server with insufficient physical memory, the
effectiveness of the database read cache may be reduced. This can have an overall negative effect on the performance of the server. For this reason, it is important to maintain a ratio between the number of storage groups and the amount of physical memory in the server.

Table 20 on page 294 identifies the specific minimum memory requirements per database, based on the number of mailbox stores.

Table 20  Required minimum memory per mailbox database

<table>
<thead>
<tr>
<th>Database count</th>
<th>Minimum required physical memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>2 GB</td>
</tr>
<tr>
<td>11-20</td>
<td>4 GB</td>
</tr>
<tr>
<td>21-30</td>
<td>6 GB</td>
</tr>
<tr>
<td>31-40</td>
<td>8 GB</td>
</tr>
<tr>
<td>41-50</td>
<td>10 GB</td>
</tr>
<tr>
<td>51-60</td>
<td>12 GB</td>
</tr>
<tr>
<td>61-70</td>
<td>14 GB</td>
</tr>
<tr>
<td>71-80</td>
<td>16 GB</td>
</tr>
<tr>
<td>81-90</td>
<td>18 GB</td>
</tr>
<tr>
<td>91-100</td>
<td>20 GB</td>
</tr>
</tbody>
</table>
Considerations and planning for storage design

Numerous considerations should factor into every storage design for the Exchange Server 2010 environment. While each of these considerations may impact the overall new design at some point, it is knowledge of the preexisting environment that should form the basis for building and deploying the design. The more information known about an organization’s current use of its messaging system, the better defined are the requirements for the new implementation. It is this information that determines and drives the overall storage design and defines some of the architectural design elements for building the mailbox stores. As a result, the more accurate and detailed this information, the better the chances that the deployed design meets the demands of its end users, the storage and Exchange administrators.

The following section outlines some important guidelines, considerations, and concepts that should be taken into account as the organization begins to plan for sizing and configuring the EMC Symmetrix for deployment. Whenever possible, it is desirable to have this data supported by empirical measurements (concurrent users, IOPS, read/write ratio, log files per day, average message size and so on) from the current environment.

Table 21 on page 295 describes the Mailbox server role activities, and how these activities affect disk I/O:

<table>
<thead>
<tr>
<th>Activity</th>
<th>How activity affects disk I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ese database (.edb file)</td>
<td>The Mailbox server stores all mail in an ESE database. The ESE database is randomly accessed and uses an 32 KB page size, although I/O coalescing can result in larger I/Os. For reliability, and in some cases for performance reasons, the database should be on disks that do not contain transaction logs.</td>
</tr>
<tr>
<td>Transaction Log</td>
<td>All changes made to the database are first committed to the transaction log, which is a sequential write to the disk. The writes vary in size from 512 bytes to the log buffer size.</td>
</tr>
<tr>
<td>Content Indexing</td>
<td>Content indexing is a random workload that should be placed on the same LUN as the database and will typically be about 5 percent of the database size. Because content indexing runs in the background, indexing messages as they arrive, the disk I/O impact is minimal.</td>
</tr>
</tbody>
</table>
Microsoft Exchange Server on EMC Symmetrix Storage Systems

Table 21  Causes of disk I/O by the Mailbox server (page 2 of 3)

<table>
<thead>
<tr>
<th>Activity</th>
<th>How activity affects disk I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paging</td>
<td>If a process requests a page in memory and the system cannot find the page at the requested location, a page fault occurs. If the page is elsewhere in memory, the fault is a soft page fault. If the page must be retrieved from disk, the fault is a hard page fault. Most processors can handle large numbers of soft page faults without consequence. However, hard page faults can cause significant delays. Continuous high rates of disk paging indicate a memory shortage.</td>
</tr>
<tr>
<td>Content conversion</td>
<td>The native method to store data in Exchange uses MAPI messages that are encapsulated in Transport Neutral Encapsulating Format (TNEF). This allows MAPI messages to be transported over SMTP and provides MAPI messages for MAPI clients, such as Microsoft Outlook. Non-MAPI clients require messages in MIME format. This format requires Exchange to perform a content conversion process from TNEF / MAPI to MIME format. All content conversion is performed on the Client Access servers and Hub Transport servers. However, legacy Web Distributed Authoring and Versioning (WebDAV) applications, such as Microsoft Entourage, access the Mailbox server directly. In this scenario, the content conversion process occurs directly on the Exchange Server 2010 Mailbox server. When a legacy WebDAV client requests data that must be converted on a Client Access server, the data is accessed from the Exchange Server 2010 Mailbox server by accessing the /Exchange virtual directory. (Some tools access the /ExAdmin virtual directory to access data.) The data is converted in the Tmp directory on the Mailbox server, and then sent to the Client Access server. The greatest usage of the Tmp directory often occurs after non-MAPI client users are moved to a new server. This behavior occurs because there may be a large amount of content conversion when users first connect to their mailboxes. To improve performance, the Tmp directory should not be on the same LUN as the page file or the operating system.</td>
</tr>
<tr>
<td>Backup and restore</td>
<td>The process of verifying a VSS backup requires that data be read from database and transaction log file volumes as part of the ESEUTIL process. This additional I/O can affect user response times and should be avoided during business hours. The process of soft recovery requires that the ESE play back all of the transaction log files. This causes the I/O profile to be a sequential read stream. As a result, the recovery performance improves if the transaction log files are on a disk with fast sequential disk access. One way to avoid this is to use continuous replication, which enables you to offload Volume Shadow Copy Service (VSS)-based backups from the active copy of the database to a passive copy of the database.</td>
</tr>
</tbody>
</table>
Considerations and planning for storage design

Anticipating messaging growth beyond deployment time

Exchange, in recent history, has been demonstrated to be an aggressive growth application, both in terms of transactional throughput and pure storage capacity requirements.

**Note:** Best practice dictates to plan for 20-50 percent growth for the enterprise class messaging deployments depending on business needs. It is important to consider all of the different manners in which growth can occur for large-scale deployment of Exchange.

Messaging growth within the Enterprise can be thought of growing organically by adding third-party applications such as Antivirus, Anti-Spam-Phishing tools, or perhaps custom MAPI, or “Inbox” search tools that add additional overhead. Perhaps the organization has decided to add or scale more wireless-enabled mailboxes with RIM’s Black Enterprise Server as an example. The important concern here for storage growth may not necessarily be related to pure
storage capacity concerns but rather the requirement to maintain the same levels of performance and availability when layering in or adding these types of third-party applications.

It is also very possible to grow a messaging deployment by corporate acquisitions as has been the case with many large organizations. This style of growth can substantially increase the growth of Exchange environments over relatively short periods of time.

When analyzing sizing needs, the determination should first be made as to whether 100 percent of the workload should be addressed, or some amount slightly less than that. Addressing the full 100 percent would imply that the EMC Symmetrix would be configured to process the full transactional workload, regardless of the day or hour, which in some cases could cause the environment to be over-configured or provisioned from a storage perspective.

Determining the existing transaction workload from an existing production system requires the use of Windows Performance Monitor objects. The goal is to bring the physical disk utilization and application level workload together and into context, such that the specific server workload can be understood and better accounted for during the sizing exercise for better deployments.

A variety of counters may be used to measure the application workload. This is dependent upon how the information is to be reused. For example, measuring the number of mailboxes defined on the server does not take into account whether or not the mailboxes are “logged-in.”

Measuring the number of users connected to the server more accurately represents the actual ratio between server activity and disk activity. It does not, however, account for the following:

- Connected, but inactive users.
- A connected user using more than one mailbox. For example, a manager may provide delegate access to his assistant where the assistant will have access to multiple mailboxes as a result.
- The background synchronization process of Outlook 2003/2007/2010 in cached mode may be issuing more than one connection to the server.

A single user may have more than one connection. However, this will not account for users who are connected but who are not “active” users. The number of active users may be significantly different from
Considerations and planning for storage design

Microsoft Exchange Layouts on EMC Symmetrix

the number of mailboxes defined on the server which, in turn, may be different from the number of mailboxes defined on the server. Consider also that the activity factor depends greatly on the client protocol. For example, POP and IMAP users establish a connection to the server, synchronize e-mail, and then terminate their connection. The concept of an idle connection does not apply in such cases.

Mailbox quota limits

Perhaps no other sizing consideration for Exchange is discussed more often than mailbox size, and for good reason. Growing mailbox size and quotas has a dramatic effect of the overall workload for the back-end Exchange server primarily due to the fact that larger mailboxes generally produce greater disk workloads.

Generally speaking there are no inherent size limits on individual mailboxes for Exchange Server 2010. The primary considerations that limit mailbox size, practically speaking, are available disk space, backup/restore times, service level agreements, and Outlook performance. In the latter case, client performance may be impacted by the size of the user mailbox. More specifically, this performance consideration is not necessarily related to the overall physical size or quota of the mailbox, but rather by the number of mail items in the folder or folders that the Exchange server regularly interacts with as a matter of normal and routine operation. These unique folders are commonly referred to as Exchange critical folders. These Exchange critical folders are outlined as follows:

- **“Inbox”:** The Exchange back-end server always requires access to the user’s Inbox for message delivery and for executing server-side rules processing even when the user is not actively “logged” on to the server.
- **“Calendar”:** Calendar entries may be made by other users when checking free/busy details, and so on.
- **“Contacts”:** This folder contains a user’s private e-mail addresses and contacts that are used for address lookups when a user composes e-mail.
- **“Sent Items”:** The folder contains items that have been sent by the user when the user has chosen to save copies.
- **“Deleted Items”:** The items that have been deleted form a user’s mailbox folder but have not been purged from the mailbox.

From the client perspective, folder-related operations that are influenced by the number of items in a folder include adding a new column to a user created “view,” for example, or sorting on a new
column, as well as user “find” and “search” operations. From an Exchange database perspective, these user-generated folder operations represent the creation and maintenance of ESE secondary indexes. For Exchange Server 2010, these indexes, once created, will be maintained and continue to “live” for up to 40 days, at which time online maintenance will expire them if they are regularly used. The physical disk I/Os required to create and maintain these secondary ESE indexes can be significantly greater when these folders contain a large number of items. This additional disk I/O workload may be further qualified as a heavier disk “read” workload. Even the simple act of delivering a message to an “Inbox” that contains greater than 2,500 items will require greater than normal database page access.

The incremental disk read I/O associated with critical folders containing a large number of items originates from the manner in which ESE will satisfy the user or MAPI request for data when a new view of that folder is requested. ESE will satisfy these requests for data in individual, serialized requests from the disk, not necessarily in bulk operations.

If the current physical disk read response times for the Exchange Server are averaging 10 ms, then this view request would translate to a 2 s request. If the disks were averaging 20 ms for read operations then the request could take up to 4 s. This effect is variable in nature, and largely driven by user behavior, in that, it is entirely possible that users with 4,000 items in their inbox who do not perform a significant number of sorts or do not create many views in general, may not be generating any more or less disk I/O other than their normal activity.

Table 22 on page 300 lists Microsoft guidance with respect to the maximum number of items for critical folders that should be used as guidelines or points at which user activity could begin to impact disk read activity beyond normal levels.

<table>
<thead>
<tr>
<th>Exchange Version</th>
<th>Folder Item Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchange Server 2000/2003</td>
<td>5,000</td>
</tr>
<tr>
<td>Exchange Server 2007</td>
<td>20,000</td>
</tr>
<tr>
<td>Exchange Server 2010</td>
<td>100,000</td>
</tr>
</tbody>
</table>
There are a number of ways to mitigate the heavier workloads associated with larger size mailboxes. The goal here is to not necessarily decrease the size of the individual mail items, but rather manage the number of items in the Exchange critical folders.

Third-party archiving solutions like EMC SourceOne™ assist and manage the physical size of mailboxes by extracting users’ mailbox data and ultimately storing this data in a central repository for later retrieval and search requests. These archiving products, while providing the ability to manage overall database size, may not, however, lower the actual number of items in folders. It is important to be aware that depending on the mode chosen for archiving, these solutions can leave behind data redirection links (pointers) to the archived content for the purposes of retrieval from the Outlook client. These pointers, sometimes referred to as “stubs,” are generally counted as mail items and therefore will increase the total number of items in a user’s “Inbox” when these users do not proactively manage their critical folders.

Best practice therefore, would be to:

- Educate the end users in the environment in order to assist them in proactively “moving” or archiving mail items away from these critical folders. This means that users can create more subfolders “under” these critical folders and move message to these subfolders, as user-created folders are accessed differently than critical folders. To a large degree, user-created subfolders are only accessed when the user interacts with these folders. As an example, moving all January mail items into a “January” user-created subfolder or perhaps a “To-Do” or a “Save-for-later folder.” This self-induced method of user-based archiving decreases the pressures on Exchange critical folders.

- Deploy an Outlook 20xx Cached Mode Client as cached clients to remove the burdens associated with index creation and index maintenance for high-count critical folders. These cached clients also move user search and find requests to users’ local machines.

Most commonly today, organizations have adopted the Microsoft Outlook 2007 or Outlook 2010 messaging product as their client of choice. Today, it is possible to deploy both versions in either an “Online-mode” or a “Cached-mode” differentiated by the manner in which each client communicates with Exchange Server 2010.
Note: For Microsoft Outlook 2003/2007/2010, cached-mode represents the default mode of operation at installation time.

Microsoft Outlook clients operating in online mode are susceptible to the behaviors outlined in the previous section, primarily due to the fact that the online client relies on the Exchange server itself to perform sorts and views, and create and maintain user-created indexes, while searches and find operations are similarly performed against the Exchange server. In contrast however, the cached-mode client will perform these operations locally on the client workstation copy of cached user Exchange data, that is, once the initial mailbox synchronization with the user client copy on the workstation is completed, which represents a disk intensive operation as well. However, in this mode of operation, as the mailbox size grows, the disk subsystem burden is shifted from the Exchange server to the Outlook client. This means that having a large number of items in a user’s Inbox, or a user searching a mailbox will have little effect on the server.

There are some general considerations for deploying MAPI cached clients and to that end, Microsoft has issued some recommendations and considerations for these client deployments. Because the pressures related to maintaining this Exchange data are shifted to the client workstation hardware for these clients, Microsoft recommends some specific client workstation hardware for very large mailboxes, detailed in Table 23 on page 302.

<table>
<thead>
<tr>
<th>Mailbox Size</th>
<th>Memory Size</th>
<th>Hard disk speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GB</td>
<td>1 GB</td>
<td>5,400 rpm</td>
</tr>
<tr>
<td>2 GB</td>
<td>1 - 2 GB</td>
<td>7,200 rpm</td>
</tr>
<tr>
<td>10 GB</td>
<td>2 - 3 GB</td>
<td>7,200 rpm</td>
</tr>
</tbody>
</table>

When deploying cached-mode clients, client performance can be an issue when this Exchange data becomes very large and the file containing the data (OST) becomes physically scattered over a user’s workstation hard disk. On large cached mailboxes greater than 500 MB, best practice is to keep the OST files (the local data cache) free of physical disk fragments. The maximum supported OST file in Outlook 2007 is 2 GB and for Outlook 2010 is 10 GB.
In Exchange Server 2010, wireless-enabled mailboxes using technologies such as RIM’s BlackBerry devices place additional demands upon the server and ultimately, therefore the Symmetrix storage. Many organizations report a two- to four-fold increase in database disk I/O for these unique mailboxes. This behavior is primarily due to the manner in which these technologies maintain communication with their associated Exchange Server as a persistent network connection. Additionally, many of the users requiring wireless-enabled messaging solutions do so because they generate more mail activity, more often, in general. Finally, these technologies have been further differentiated by the somewhat addictive nature of this technology exhibited with some users that only exaggerates this behavior.

Disk I/O estimates and best practices vary greatly on the incremental disk I/O overhead associated for these mailboxes. Some observations have some wireless-enabled users consuming 2 to 4 times the I/O requirements to those of an average user.

Note: Best practice in this regard, therefore, is to gather as much information from the existing environment in order to better predict any specific incremental overhead associated with wireless-enabled mailboxes. If this data is unavailable or not known, it is prudent to consult http://www.blackberry.com or see the RIM whitepaper for latest recommendations and guidance.

In addition, it is also important to consider all of the other applicable storage components involved in processing Exchange Server I/O, and what proper queue management implies for each of these relative components. Consider that poor I/O response times for Exchange Server are commonly the result of one or any combination of the following issues:

- Queuing at the physical disk level implies an undersized deployment of Exchange Server from a physical disk perspective. Always size the Exchange Server workload for physical spindles.
- Queuing at the Symmetrix Front-end Adapter (FA) level, playing a critical role in providing enough Symmetrix Front-end Adapter bandwidth to sustain the bursts of read and write I/O at the time of Exchange Server database checkpointing operations. Always consider using four EMC Symmetrix FAs for every 8,000 Exchange Server 2010 users profiled at 0.14 IOPS to adequately handle Exchange Server database checkpointing.
Queuing can take place at the HBA level in that a particular HBA's queue depth setting is improperly set. This creates the effect of I/O throttling, sometimes described as I/O starvation.

Configure two fibre HBAs on the server with a queue depth set to 128 per adapter. Such a configuration will also require EMC PowerPath.

Configure four EMC Symmetrix Front-End Adapter slices for every two fibre HBAs and using a “round-robin” approach, load balance across Symmetrix FA processors before wrapping around FA ports on those same processors as depicted in Figure 66 on page 304. This provides the benefit of load balancing the Microsoft Exchange workload during database checkpoints.

At a minimum, configure two Symmetrix RA adapters for SRDF I/O propagation. In this way the replicated write workload may be load balanced across two RDF links while providing high availability.

**Note:** To determine the actual number of RDF adapters required for a given configuration, EMC uses internal tools that analyze link bandwidth, distance (for latency numbers) between disk arrays, user counts an workloads. The results of this analysis will ultimately determine the number and type of RDF adapters required.
In Synchronously replicate environments, it is acceptable to mix Exchange Server Transaction Log I/O with Exchange Server Database I/O on the same physical spindles assuming the environment has been properly sized for IOPS requirements.

Always validate a storage configuration with Jetstress first, then LoadGen for Exchange Server 2010 before deployment.

I/O profiling for disk IOPS requirements

The first step in designing a storage configuration for use with Exchange Server is to calculate the total number of IOPS that the Microsoft Exchange environment will require. The total number of IOPS required is used to determine the number of physical drives necessary to meet the IOPS requirement. Finally, calculate the capacity requirement for the Exchange Server data files and adjust the drive count upward, if necessary.

The best way to provide enough I/O to the Exchange Server, especially in a large Exchange Server environment, is to understand users’ usage profile. Sizing of the storage infrastructure should be based on a careful analysis of the number of current and anticipated users and their messaging habits and patterns as measured I/O in the existing environment provides the best starting point and ultimately leads to a better performing messaging deployment from the beginning.
The fundamental calculation concerns IOPS per user, as shown in Table 24.

### Table 24  Microsoft Exchange I/O profiles

<table>
<thead>
<tr>
<th>Messages sent/received per mailbox per day (~75KB average message size)</th>
<th>Database cache per user (MB) for Single database copy (Stand-alone)</th>
<th>Estimated IOPS per mailbox for Multiple database copies (DAG)</th>
<th>Estimated IOPS per mailbox Stand-alone server</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>3</td>
<td>0.06</td>
<td>0.05</td>
</tr>
<tr>
<td>100</td>
<td>6</td>
<td>0.12</td>
<td>0.10</td>
</tr>
<tr>
<td>150</td>
<td>9</td>
<td>0.18</td>
<td>0.15</td>
</tr>
<tr>
<td>200</td>
<td>12</td>
<td>0.24</td>
<td>0.20</td>
</tr>
<tr>
<td>250</td>
<td>15</td>
<td>0.30</td>
<td>0.25</td>
</tr>
<tr>
<td>300</td>
<td>18</td>
<td>0.36</td>
<td>0.30</td>
</tr>
<tr>
<td>350</td>
<td>21</td>
<td>0.42</td>
<td>0.35</td>
</tr>
<tr>
<td>400</td>
<td>24</td>
<td>0.48</td>
<td>0.40</td>
</tr>
<tr>
<td>450</td>
<td>27</td>
<td>0.54</td>
<td>0.45</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>0.60</td>
<td>0.50</td>
</tr>
</tbody>
</table>

**Note:** These numbers are *estimates* and are based on observed averages. Best practice dictates that it is best to acquire accurate measurements of the actual user profiles for the specific implementation.

The process of I/O profiling the end user is fundamental in designing the storage solution, yet remains one of the most misunderstood requirements. Studies have shown that if asked, many corporations report a user community that predominantly fits the heavy profile. However, there is empirical evidence that suggests that the average is approximately 0.18 IOPS. Some users use e-mail more than average; some use it less. In a typical environment, 20 percent of the users fit the light profile, 60 percent the moderate profile, while 20 percent fit the heavy profile.

Once the user profiles are defined, it is possible to calculate the total IOPS required by multiplying each user by their predicted use of IOPS.
Storage architecture and I/O principles for EMC Symmetrix

Symmetrix cache

The Symmetrix cache plays a key role in improving I/O performance in the storage subsystem. The cache improves performance by allowing write acknowledgements to be returned to a host when data is received in cache, rather than being fully destaged to the physical disk drives. Additionally, reads benefit from cache when sequential requests from the host allow follow-on reads to be prestaged in cache. The following sections briefly describe how the Symmetrix leverages storage cache for the purposes of processing Exchange Server writes and reads, and additionally discusses performance considerations.

Write operations and the Symmetrix cache

All write operations on a Symmetrix array are serviced by cache. When a write is received by the front-end director, a cache slot must be found to service the write operation. Since cache slots are a representation of the underlying hypervolume, if a prior read or write operation caused the required data to already be loaded into cache, the existing cache slot may be used to store the write I/O. The cache slot is marked write pending if it is not already in this state.

If a cache slot representing the storage area is not currently allocated, a call is made to locate a free cache slot from the global pool, for the write. The write operation is moved to the cache slot and the slot is then marked write pending. At some later point, Enginuity will destage the write to physical disk. The decision of when to destage is based on overall system load, physical disk activity, read operations to the physical disk, and availability of cache.

Cache is used to service the write operation to optimize the performance of the host system. Because write operations to cache often expressed in nanoseconds, are significantly faster than physical writes to disk media often expressed in milliseconds, the write is reported as complete to the host operating system much more efficiently. Battery backup and priority destage functions within the Symmetrix ensure that no data loss occurs in the event of system power failure.

If the write operation to a given disk is delayed because of higher priority operations (read activity is one such operation), the write pending slot remains in cache for longer periods of time. Cache slots are allocated as needed to a volume for this purpose. Enginuity
calculates thresholds for allocations to limit the saturation of cache by a single hypervolume. These limits are referred to as write pending limits.

Cache allocations are based on a per-hypervolume basis. As write pending thresholds are reached, additional allocations may occur, as well as reprioritization of write activity. As a result, write operations to the physical disks may increase in priority to ensure that excessive cache allocations do not occur. This is discussed next in more detail.

In the manner described, the cache enables buffering of write I/Os and allows for a steady stream of write activity to service the destaging of write operations from a host. In a “bursty” write environment, this serves to even out the write activity. Should the write activity constantly exceed the low write priority to the physical disk, Enginuity will raise the priority of write operations to attempt to meet the write demand. Ultimately, if the write I/O load from the host exceeds the physical disk ability to write, the volume maximum write pending limit may be reached. In this condition, new cache slots will only be allocated for writes to a particular volume once a currently allocated slot is freed by destaging it to disk.

Read operations and the Symmetrix cache

As discussed in the previous section, read operations typically have an elevated priority for service from the physical disks. As user processes in general need to wait for an I/O operation to complete before continuing, this is generally a good practice for storage arrays, especially those able to satisfy write operations from cache.

When a read request is received from a host system, Enginuity checks to see if a corresponding cache slot representing the storage area exists. If so, a further check is made to determine if the required data to service the read has been loaded into the cache slot. If so, the read request may be serviced immediately—this is considered a read hit.

If the cache slot has been allocated, but the data is not available, this is considered a short read miss, and a request is made to load the data from disk to the available cache slot. The read request must wait for a transfer from disk.

If a cache slot does not already exist for this storage location, but free slots are available, the read operation must wait for a cache slot to be allocated and for the transfer from disk—this is referred to as a long read miss.
Although cache slots themselves are 32 KB, a cache slot may contain only the requested data. That is, if a read request is made for an 8 KB block, then only that 8 KB block will be transferred into the cache slot, as opposed to reading the entire 32 KB track from disk. The smallest read request unit is 4 KB.

An important performance consideration is to ensure that an appropriate amount of cache is installed in the Symmetrix. All I/O requests from hosts attached to the array are serviced from the Symmetrix cache. Symmetrix cache can be thought of as an extension to other server-side buffering mechanisms. As such, many database application environments can benefit from additional Symmetrix cache. With all Symmetrix arrays, appropriately sizing the cache is performed by the sales team, based upon the number and size of physical spindles, configuration (including number and type of volumes), replication requirements (for example, SRDF), and customer requirements.

Symmetrix cache plays a key role in host I/O read and write performance. Read performance can be improved through prefetching by the Symmetrix if the reads are sequential in nature. Enginuity algorithms detect sequential read activity and prestage reads from disk in cache before the data is requested. Write performance is greatly enhanced because all writes are acknowledged back to the host when they reach Symmetrix cache rather than when they are written to disk. While reads from a specific hypervolume can use as much cache as is required to satisfy host requests assuming free cache slots are available, the limits the number of writes that can be written to a single volume (that is, the write pending limit previously discussed). Understanding the Enginuity write pending limits is important when planning for optimal performance.

As previously discussed, the write pending limit is used to prevent high write rates to a single hypervolume from consuming all of the storage array cache for its use, at the expense of performance for reads or writes to other volumes. The write pending limit for each hypervolume is determined at system startup and depends on the number and type of volumes configured and the amount of cache available. The limit is not dependent on the actual size of each volume. The more cache available, the more write requests that can be serviced in cache by each individual volume. While some sharing of unused cache may take place (although this is not guaranteed), an upper limit of three times the initial write pending limit assigned to a
volume is the maximum amount of cache any hypervolume can acquire for changed tracks. If the maximum write pending limit is reached, destaging to disk must take place before new writes can come in.

In addition to limits imposed at the hypervolume level, there are additional write pending limits imposed at the system level. Two key cache utilization points for the Symmetrix are reached when 40 percent and 80 percent of the cache is used for pending writes. Under normal operating conditions, satisfying read requests from a host has greater priority than satisfying write requests. However, when pending writes consume 40 percent of cache, the Symmetrix then prioritizes reads and writes equally. This reprioritization can have a profound effect on database performance. The degradation is even more pronounced if cache utilization for writes reaches 80 percent. At that point, the Symmetrix begins a forced destage of writes to disk with discernable performance degradation of both writes and reads. If this threshold is reached, it is a clear indicator that both the cache and the total I/O on the array need to be reexamined.

Write pending limits are also established for Symmetrix metavolumes. Metavolumes are created by combining two or more individual hypervolumes into a single logical device that is then presented to a host as a single logical unit (LUN). Metavolumes can be created as concatenated or striped metavolumes. Striped metavolumes use a stripe size of 960K. Concatenated metavolumes write data to the first hyper in the metavolume (metahead) and fill it before beginning to write to the next member of the meta. Write pending limits for a metavolume are calculated on a member-by-member (hypervolume) basis.

---

**Back-end considerations**

Array back-end considerations are typically the most important part of optimizing performance on the Symmetrix. Advances in disk technologies have not kept up with performance increases in other parts of the storage array such as director and bandwidth (that is, Direct Matrix compared to Bus) performance. Disk access speeds have increased by a factor of three to seven in the last decade while other components have easily increased one to three orders of magnitude. As such, most performance bottlenecks in the Symmetrix are attributable to physical spindle limitations.

An important consideration for back-end performance is the number of physical spindles available to handle the anticipated I/O load. Each disk is capable of a limited number of operations. Algorithms in
the Symmetrix Enginuity operating environment optimize I/Os to the disks. Although this helps to reduce the number of reads and writes to disk, access to disk, particularly for random reads, is still a requirement to attempt to optimize I/O operations and counts for individual spindles. If an insufficient number of physical disks are available to handle the anticipated I/O workload, performance will suffer.

In order to reduce or eliminate back-end performance issues on the Symmetrix, take care to spread access to the disks across as many back-end directors and physical spindles as possible. EMC has long recommended for placement of application data to “go wide before going deep.” This means that performance is improved by spreading data across the back-end directors and disks, rather than allocating specific applications to specific physical spindles. Significant attention should be given to balancing the I/O on the physical spindles. Understanding the I/O characteristics of each data file and separating high application I/O volumes on separate physical disks will minimize contention and improve performance. Implementing Symmetrix Optimizer may also help to reduce I/O contention between hypervolumes on a physical spindle. Symmetrix Optimizer identifies I/O contention on individual hypervolumes and non-disruptively moves one of the hypervolumes to a new location on another disk. Symmetrix Optimizer is an invaluable tool in helping to reduce contention on physical spindles, should workload requirements change in an environment.

Placement of data on the disks is another performance consideration. Because of the rotational properties of disk platters, tracks on the outer parts of the disk perform better than inner tracks. While the Symmetrix Enginuity algorithms smooth much of this variation, small performance increases can be achieved by placing high I/O objects on the outer parts of the disk. Of more importance, however, is minimizing the seek times associated with the disk head moving between hypervolumes on a spindle. Physically locating higher I/O devices together on the disks can significantly improve performance. Disk head movement across the platters (seek time) is a large source of latency in I/O performance. By placing higher I/O devices contiguously, disk head movement may be reduced, increasing I/O performance of that physical spindle.
Symmetrix hypervolumes

It is also important to understand that the Symmetrix array is designed to provide shared access to resources. As a result, items such as physical disk spindles are subdivided into hypervolumes, as depicted in Figure 67 on page 312. Multiple hypervolumes exist on any given disk spindle. Each hypervolume may either be presented to a host connected to the Symmetrix, or combined as a part of a metavolume, which is subsequently presented to a host.

![Logical view of mirrored hypervolumes](image)

The key point is that the spindles are not specifically designated to a given host, but rather they can be shared among different hosts. The overall load and performance of any individual spindle will result from the cumulative load represented by all the workloads targeted to all hypervolumes on that spindle. Expectations should be set appropriately for the anticipated performance of these resources. Administrators should understand that their particular system may not be the only system generating workload to the spindles, but they may be seeing the performance impact from the other hosts’ workloads.

Symmetrix metavolumes

Individual Symmetrix hypervolumes of the same RAID type (RAID 1, RAID 5) may be combined together to form a virtualized device called a Symmetrix metavolume. Metavolumes are created for a number of reasons including:

- A desire to create devices that are greater than the largest hypervolume available.
- To reduce the number of volumes presented down a front-end director or to an HBA. A metavolume presented to an HBA only counts as a single LUN even though the device may comprise a large number of individual hypervolumes.

- To increase performance of a LUN by spreading I/O across more physical spindles.

There are two types of metavolumes—concatenated or striped, as shown in Figure 68 on page 313. With concatenated metavolumes, the individual hypervolumes are combined to form a single volume, such that data is written to the first hypervolume sequentially before moving to the next. Writes to the metavolume start with the metahead and proceed on that physical until full, and then move on to the next hypervolume. Striped metavolumes, on the other hand, write data across all members of the device. The stripe size is set at two cylinders or 960 KB.

**Figure 68** Logical view of striped and concatenated metavolumes

**Note:** EMC Symmetrix engineering best practice for Exchange Server 2010 is to deploy striped metavolumes for thick LUNs (not virtually provisioned) and concatenated volumes for thin (virtually provisioned) LUNs.
**Storage provisioning for Exchange Server 2010**

Any sizing exercise of Exchange Server 2010 should take into account and require that the organization understand each of storage components (size, capacity, performance) when sizing the storage devices which will ultimately host Exchange databases. In addition the organization should take the following into consideration:

- A balanced approach to overall workload(s) taking into account all of the EMC Symmetrix storage components
- The request rate (I/O per second) capacity of the physical volumes, ultimately to be presented to the Exchange Server, according to the observed disk read/write ratio.
- The effect of operations when losing a single or multiple volumes, or any other storage component on the Symmetrix involved in the I/O processing
- Time required to reconstruct parity information or mirroring relationship

In general there are two basic approaches to provisioning or allocating EMC Symmetrix storage for use with Exchange Server 2010 each of which are discussed in detail and taken in the following order:

- Traditional Server
- Storage Specific Approach

**EMC sizing approach with Symmetrix**

The first approach may be thought of as the traditional approach to provisioning in that the particular Exchange server(s) transactional throughput is derived by monitoring or trending the existing messaging environment. Subsequently, this workload is decomposed or translated into a cumulative disk workload for transaction logs, databases, item retention policies, growth factors, and free space requirements for occasional maintenance routines. Finally, the EMC Symmetrix volumes may be configured or provisioned with the customer-specific RAID protection schemes, configured for host access, and finally properly aligned and formatted. The information gathered during the organization’s trending and planning phase is crunched through the following basic formula:
Storage provisioning for Exchange Server 2010

Microsoft Exchange Layouts on EMC Symmetrix

Formula: (Base IOPS x Read percentage) + (Base IOPS x Write percentage x Write penalty) = RAID-adjusted required back-end IOPS

Mailbox Size (Mailbox Limit + Whitespace + Dumpster) + Free Space Percentage Requirement = database LUN size.

The fictitious corporation ABC123 has collected all of their relevant messaging trending data, a summary of which is detailed in Example 1 and is undertaking a sizing exercise using an EMC Symmetrix as their storage subsystem in a non-distance-replicated environment.

Example 1  Sample company Exchange profile

ABCD123 Inc. has collected one month of trending data using the Microsoft Windows Server Perfmon for a 16,000 mailbox Exchange Server. (This example is carried forward throughout the remaining storage requirement calculations. The data collected outlined the following requirements:

- Mailbox limit—2 GB.
- Average message frequency per user—150 messages @ 75 KB average message size (0.15 IOPS).
- Outlook mode—100 percent MAPI
- Number of Exchange Server 2010 mailbox servers—8
- Number of active mailboxes per server—4,000 in HA (2,000 active and 2,000 passive)
- Deleted Items retention window—14 days
- Logs protection buffer—3 days
- 24 * 7 database maintenance configuration enabled.
- Database read / write ratio—3:2.

Today, there are three generally accepted RAID protection schemes on the EMC Symmetrix that are applicable for most high-end Exchange deployments. These three RAID-type configurations are: RAID 1/0, RAID 5 and RAID 6. All of these are appropriate for use with Exchange Server data.

The following defines all RAID configurations that are available on the Symmetrix available for deployments with Exchange Server 2010 environments:
RAID 1—These are mirrored devices and are the most common RAID type in a Symmetrix. Mirrored devices require writes to both physical spindles. However, intelligent algorithms in the Enginuity operating environment can use both copies of the data to satisfy read requests that are not already in the cache of the Symmetrix. RAID 1 offers optimal availability and performance but at an increased cost over other RAID protection options.

RAID 5—RAID 5 stripes parity information across all volumes in the RAID group. RAID 5 offers good performance and availability at a decreased cost. Data is striped using a stripe width of four tracks (128 KB). RAID 5 is configured either as RAID 5 (3+1) (75 percent usable) or RAID 5 (7+1) (87.5 percent usable) configurations.

RAID 6—As with RAID 5, RAID 6 has the data and parity tracks for a given Symmetrix logical volume striped horizontally across all disks in the RAID group. The difference in RAID 6 is that in addition to the horizontal parity in RAID 5, a second, independent parity is kept so recovery can be made from dual disk failures in a single RAID group. The first parity, which is similar to RAID 5, is called a “horizontal” parity. The second parity is called a “diagonal” parity. For both RAID 5 and RAID 6, the parities rotate among the RAID members. There are two RAID 6 drive counts implemented, 6+2 (6 data + 2 parity)(75 percent usable) and 14+2 (14 data + 2 parity)(87.5 percent usable) for a total of 8 and 16 drives, respectively.

RAID 1/0—These are striped and mirrored devices. This configuration is only used in mainframe environments. However, RAID 1/0 can also be configured by creating striped metavolumes, as described next.

As a matter of Best Practice, EMC generally recommends the RAID protection type most applicable to the system being installed to be the primary choice in RAID configuration. For reasons of reliability and availability and performance for high-end deployments of Exchange Server RAID 1 will always come out on top. However, Exchange Server 2010 databases can be deployed on any of the other RAID configurations mentioned above.

The bottom line in choosing a RAID type is ensuring that the configuration meets the needs of the customer’s environment. Considerations include read and write performance, balancing the I/O across the spindles and the back end of the Symmetrix, tolerance for reduced application performance when a drive fails, and the...
Storage provisioning for Exchange Server 2010

Microsoft Exchange Layouts on EMC Symmetrix

consequences of losing data in the event of multiple disk failures. It should also be noted that this is not an all-or-nothing approach. It is feasible to implement a combination of RAID 5, RAID 6 and RAID 1 devices to support a given database, selecting the optimal configuration for each type of workload is the key consideration. This becomes more apparent with the introduction of FAST, "EMC Symmetrix Fully Automated Storage Tiering (FAST)" on page 124, and FAST (VP), "EMC Symmetrix Fully Automated Storage Tiering for Virtual Pools (FAST VP)" on page 126.

Calculating the EMC Symmetrix RAID-type overhead and the read/write ratio

Regardless of the eventual RAID type chosen, the critical factor in building the solution is to ensure that enough physical drives are configured to accommodate the required I/O demand. Example 2 illustrates the RAID-adjusted required back-end IOPS calculation for the example company.

Example 2 RAID overhead calculation

ABC123 Inc. has chosen to implement EMC Symmetrix for its Microsoft Exchange Server 2010 deployment. Each server will host 4,000 users (2,000 active and 2,000 passive). As a result, carrying forward this example of 600 base IOPS (4,000 users * 0.15 IOPS/user) and 120 additional server IOPS then applying a 3:2 read/write ratio for RAID 1/0, the calculations are as follows:

Formula: (User IOPS x Read percentage) + (User IOPS x Write percentage x Write penalty) = RAID-adjusted required back-end IOPS

The calculations for the different RAID types are listed below:

RAID 1

\[(720 \times 0.6) + (720 \times 0.4 \times 2) = 432 + 576 = 1,008 \text{ IOPS are required}\]

RAID 5

\[(720 \times 0.6) + (720 \times 0.4 \times 4) = 432 + 1,152 = 1,584 \text{ IOPS are required}\]

RAID 6

\[(720 \times 0.6) + (720 \times 0.4 \times 6) = 432 + 1,728 = 2,160 \text{ IOPS are required}\]

The pure I/O requirements may be more than adequate to determine the number of physical drives required. However, there are instances where it may be necessary to account for any specific extra capacity or additional physical space requirements for the databases. For

Additional storage capacity considerations for database volumes
certain environments (for example, Exchange deployments with large mailboxes) the total storage capacity requirements could require more disk space than performance needs actually dictate.

- One example of this is the case where more than one profile, for each category of users is required in the Microsoft Exchange store, multiply the maximum allowed mailbox size by the number of users for each category.

- Allow an additional percentage of the database size based on the organization’s policy for deleted item retention. By default, the deleted item retention period is set to 7 days. For large scale organizations, observations have shown that a 7-day retention period can cause the database to grow 10 to 20 percent, 14-day policy by 30 to 40, while a full 30-day retention policy can mean an 80–90 percent increase. This varies greatly and a specific analysis should be performed relative to organizational specifics.

- Database maintenance and online defragmentation are other considerations. This process is a significant generator of disk I/O. The Exchange database defragmentation will defragment the Exchange database file, but it should be understood that this procedure does not compact the actual physical size of the database file. Compacting or shrinking the physical size of the database is accomplished by performing Exchange offline defragmentation. This process requires that the specific Exchange Server database volume is dismounted while ESEUTIL is used to rebuild a newly compacted copy of the database. To minimize downtime for the Exchange database, this rebuild process may be performed on the same volume as the source database. There must be free space equaling at least 110 percent of the size of the database in order for the rebuild to complete successfully. If multiple databases are stored on the same volume, enough space must be allowed to handle the rebuild of the database with the largest size.

The database size on the physical disk isn’t just the number of users multiplied by the mailbox limit. Space required for the Microsoft Exchange mailbox database would be calculated as follows:

Mailbox Size = Mailbox Limit + Whitespace + Dumpster

Mailbox Limit—This is the storage size limit, known as the mailbox limit, that’s in effect in the organization and is the amount of space available to all users within the database.
Whitespace—The database itself will always have free pages, or white space, spread throughout. This is due to users deleting and receiving mail messages during a normal working cycle. Once an item is deleted by the user it is marked for removal by the background database maintenance process, items marked for removal from the database are removed, which frees these pages. These free pages are known as whitespace. The percentage of white space is constantly changing due to the efforts of the 24x7 online defragmentation process.

Dumpster—Each database has a dumpster that stores soft-deleted items. By default, soft-deleted items are stored for 14 days and calendar items are stored for 120 days in Exchange Server 2010. In addition, Exchange Server 2010 also includes the ability to prevent the purging of data before the deleted item retention window has passed. This functionality is known as single item recovery. When single item recovery is enabled (disabled by default), there is an additional 1.2 percent increase in the size of the mailbox for a 14-day deleted item retention window. For calendar version logging data (enabled by default), there is an additional 5.8 percent increase in the size of the mailbox.

Example 3 shows an example of the calculation for capacity requirements.

Example 3  Calculating capacity for database volumes

Assume ABC123 Inc. has chosen to deploy 2,000 users, evenly distributed across 10 databases. Each database is configured on its own physical volume. Each of these users is assumed to be characterized at the same I/O profile as detailed in Example 1. ABC123 Inc. also maintains a 2 GB quota on all mailboxes.

2,000 users x 2048 MB mailboxes = 4000 GB

Whitespace = (150 messages / day x 75/1024 MB) x 2000 = 14.6 GB

Dumpster = ((150 messages / day x 75/1024 MB * 14 days) + (2048 MB x 0.012) + (2048 MB x 0.058)) x 2,000 = 492 GB

Capacity required 4506.6 GB for active database copies. Capacity required for passive databases copies 4506.6 GB.

Total capacity for all mailbox store copies = 9013.2 GB

Size of each database = 4506.6/10 = 450 GB

Additional 20 percent for OS overhead per LUN.
Total LUN requirement per database: 450 GB x 1.2 = 540 GB
Total server disk capacity: 10,800 GB

To include space for offline defragmentation, add space equal to the size of the largest databases, considering that free space has already been factored into these calculations. The advantage of presenting the storage requirements from a Symmetrix is the ability to present LUNs as and when required. If the database has a requirement to have an offline defragmentation carried out a separate LUN of 540 GB x 110 percent would be used. This LUN would be 594 GB.

The total storage capacity of today’s physical drives continues to dramatically increase over time. However, the total transactional throughput a physical spindle can handle has not grown at a similar rate. That is, the total number of IOPS a physical drive can process has not increased at the same rate as capacities. This is an important deployment concept to consider given a particular Exchange Server workload. From a performance perspective only, I/O throughput is much more of a consideration than the total storage capacities of the drives. As a result, today’s Exchange storage deployments should be driven by their I/O requirements, rather than their pure capacity requirements. Finally, EMC Symmetrix Engineering observations have shown that a properly sized performance configuration (in terms of adequate numbers of physical spindles) will, in most configurations, support the capacity requirements as well.

The EMC Symmetrix Performance group regularly publishes guidelines for the latest disk types by capacity, RAID type, and disk speeds. Additionally, there are published numbers for simulated Exchange Server workloads. Consult your local EMC Performance Guru for the latest information and documentation in regard to this information as it is regularly updated. Table 25 on page 320 presents general guidelines and averages with respect to some common disk types and RAID protections; these numbers represent averages and should be treated accordingly.

<table>
<thead>
<tr>
<th>Disk type and rpm</th>
<th>EMC Symmetrix I/O capacity for use with Exchange Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fibre Channel 15k rpm</td>
<td>Fully stroked 200 IOPS</td>
</tr>
<tr>
<td>Fibre Channel 10k rpm</td>
<td>Fully stroked 150 IOPS</td>
</tr>
<tr>
<td>SATA 7.2K rpm</td>
<td>Fully stroked 55 IOPS</td>
</tr>
</tbody>
</table>
Using the values referenced in Table 25, it is possible to calculate the number of physical drives required for a RAID 1/0 protection scheme supporting the desired number of defined Exchange Server end users. Example calculations are shown in Example 4 which cater for required IOPS.

**Example 4  Sample drive count calculation based in IOPS requirements**

ABC123 Inc. has chosen to implement EMC Symmetrix RAID 1/0 metavolumes for its Exchange Server deployment. Carrying this example forward and applying a 3:2 read/write ratio for all of the available RAID types:

**RAID 1**

\[(720 \times 0.6) + (720 \times 0.4 \times 2) = 432 + 576 = 1,008 \text{ IOPS are required}\]

**RAID 5**

\[(720 \times 0.6) + (720 \times 0.4 \times 4) = 432 + 1,152 = 1,584 \text{ IOPS are required}\]

**RAID 6**

\[(720 \times 0.6) + (720 \times 0.4 \times 6) = 432 + 1,728 = 2,160 \text{ IOPS are required}\]

**Formula:** Total RAID-adjusted IOPS / IOPS for each disk-type = Drive count for Exchange database volumes.

In this example ABC123 Inc will evenly distribute the 2,000 users across 10 mailbox stores, 2,000 users will have 10 active mailbox stores deployed with the other 2,000 users will have 10 passive copies deployed on this server. **Table 26 on page 321** gives the number of drives required to support this Exchange deployment per RAID type and drive speed and have been rounded up to meet the disk numbers for the specific RAID type.

**Table 26  Sample drive count calculation based on drive speed and RAID type**

<table>
<thead>
<tr>
<th>RAID Type</th>
<th>Fibre Channel 15K RPM</th>
<th>Fibre Channel 10K RPM</th>
<th>SATA 7.2K RPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID 1 1008 IOPS</td>
<td>6</td>
<td>8</td>
<td>18</td>
</tr>
<tr>
<td>RAID 5 3+1 1584 IOPS</td>
<td>8</td>
<td>12</td>
<td>32</td>
</tr>
<tr>
<td>RAID 5 7+1 1584 IOPS</td>
<td>8</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>RAID 6 6+2 2160 IOPS</td>
<td>16</td>
<td>16</td>
<td>40</td>
</tr>
<tr>
<td>RAID 6 14+2 2160 IOPS</td>
<td>16</td>
<td>16</td>
<td>48</td>
</tr>
</tbody>
</table>
As well as calculating the number of drives to meet the performance criteria the capacity requirements also have to be taken into account. Using the capacity calculation detailed in Example 3 the following table details the number of disks required to support this 9013.2 GB deployment.

<table>
<thead>
<tr>
<th>RAID Type</th>
<th>Fibre Channel 600 GB (536 GB usable)</th>
<th>SATA 1 TB (917 GB usable)</th>
<th>SATA 2 TB (1823 GB usable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID 1</td>
<td>42</td>
<td>24</td>
<td>12</td>
</tr>
<tr>
<td>RAID 5 (3+1)</td>
<td>28</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>RAID 5 (7+1)</td>
<td>24</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>RAID 6 (6+2)</td>
<td>32</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>RAID 6 (14+2)</td>
<td>32</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

Comparing both Table 26 on page 321 and Table 27 on page 322 the optimum drive count that meets both the database IOPS and the required capacity. The drive and RAID options are:

- 32 x 1 TB SATA drives in RAID 5 (3+1) to meet the performance and will exceed the capacity.
- 32 x 1 TB SATA drives in RAID 5 (7+1) to meet the performance and will exceed the capacity.
- 24 x 1 TB SATA drives in RAID 1 would meet the capacity and exceed the performance.

The I/O performance to the Exchange Server transaction log volume is critical to Exchange Server database performance. That is, transactional throughput to the Exchange transaction log volume generally and fundamentally translates to Exchange Database performance. For this reason, EMC best practice dictates deploying Exchange Server transaction log volumes on RAID 1/0, as it provides the best response times and requires the least amount of rebuild time in component failure scenarios, thereby minimizing data loss, however with Exchange Server 2010 it is acceptable to locate the log files on the same RAID type that is deployed for the database volumes due to the reduction in I/O that the log files generate.
The transaction log files are a record of every transaction performed by the database engine. All transactions are written to the log first, and then lazily written to the database. Transaction log files in Exchange Server 2010 are 1 MB in size.

Transaction log IOPS calculation

Log volume I/O is I/O associated with database logging read/write activity. Log volume I/O is sequential in nature and not a significant factor for Exchange storage sizing. In Exchange Server 2010, a transaction log for a database requires approximately 10 percent as many I/Os as the databases in the database LUN. For example, if the database LUN is using 1,000 I/Os, the log LUN would use approximately 100 I/Os. With the reduction in database reads in Exchange Server 2010, combined with the smaller log file size and the ability to have more databases, the log-to-database write is 40 percent for stand-alone databases and 50 percent for databases participating in mailbox resiliency. For example, if the database that’s participating in mailbox resiliency is consuming 20 write I/Os, the log LUN will consume approximately 10 write I/Os. On Mailbox servers that are hosting databases that are participating in mailbox resiliency, there is overhead associated with using continuous replication. Closed transaction logs must be read and sent to the target database copies. This overhead is an additional 10 percent in log reads for each active database copy that’s hosted on the Mailbox server. For example, if the Mailbox server is hosting 10 active database copies, and each transaction log stream is generating 10 write I/Os, you can expect an additional 1 read I/Os for each of those 10 active database copies (or a total of 10 read I/Os). After measuring or predicting the transactional log I/O, apply a 20 percent I/O overhead factor to ensure adequate room for busier than normal periods.

Transaction Log capacity calculation

The value for the number of transaction logs generated per day is based on the message profile (number of messages sent / received) and the average message size. It indicates how many transaction logs will be generated per mailbox per day. Table 28 on page 324 can be
used to estimate the number of transaction logs that are generated on an Exchange Server 2010 Mailbox server per mailbox user, when the average message size is 75 KB.

<table>
<thead>
<tr>
<th>Message profile (75 KB size)</th>
<th>Number of Logs generated per user per day</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>150</td>
<td>30</td>
</tr>
<tr>
<td>200</td>
<td>40</td>
</tr>
<tr>
<td>250</td>
<td>50</td>
</tr>
<tr>
<td>300</td>
<td>60</td>
</tr>
<tr>
<td>350</td>
<td>70</td>
</tr>
<tr>
<td>400</td>
<td>80</td>
</tr>
<tr>
<td>450</td>
<td>90</td>
</tr>
<tr>
<td>500</td>
<td>100</td>
</tr>
</tbody>
</table>

If the average message size doubles to 150 KB, the logs generated per mailbox increases by a factor of 1.9. This number represents the percentage of the database that contains the attachments and message tables (message bodies and attachments).

Thereafter, as message size doubles beyond 150 KB, the log generation rate per mailbox also doubles, increasing from 1.9 to 3.8.

Other considerations in calculating the capacity of the logs generated are:

- Move mailbox operations—If a user is moved from one mailbox store to another one within the Exchange Organization the operation of moving the mailbox will create logs on the target mailbox server. Typically 1 percent of users per week.

- Log growth overhead factor—For most deployments, it is recommended that additional overhead factor of 20 percent to the log size (after all other factors have been considered) when creating the log LUN to ensure necessary capacity exists in moments of unexpected log generation.
Database copy replay lag—High availability in Exchange Server 2010 provides the option to lag log replay on passive database copies (configured on a per copy basis). This feature is used to provide a delay for when logs get played in to lagged database copies. This delay can be useful to protect against events which would cause undesirable content to be replicated to all database copies. The content can be stopped from being played in to the lagged database copy by suspending replay before the logs with the undesired content are played in to the database. When replay lag is enabled for a database copy, the log capacity requirements change accordingly. If a 14-day lag is configured, the need to provision for 17 days worth of logs is required. The additional log capacity is only required for the database copy that has the lag configured, other copies of that database, which don’t have a lag, will have normal (non-lagged) log capacity requirements.

The calculation of Exchange transaction log IOPS is shown in Example 5. The calculation of Exchange transaction log capacity is shown in Example 6.

**Example 5 Exchange transaction log IOPS calculation**

ABC123 Inc. has determined that the following number of backend IOPS and subsequently the number of drives required dependent on the RAID type. Table 29 on page 325 details the amount of backend Log IOPS that are required. It must be noted that Logs for Exchange Server 2010 no longer have to reside on a RAID 1 volume and locating them on the same volumes as the databases is an acceptable solution. If, however a hardware VSS backup is deployed the logs and database files must be located on separate volumes.

<table>
<thead>
<tr>
<th>RAID Type</th>
<th>Database IOPS</th>
<th>Log IOPS single server</th>
<th>Log IOPS DAG member</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID 1</td>
<td>1008</td>
<td>108</td>
<td>118</td>
</tr>
<tr>
<td>RAID 5</td>
<td>1584</td>
<td>158</td>
<td>173</td>
</tr>
<tr>
<td>RAID 6</td>
<td>2160</td>
<td>216</td>
<td>237</td>
</tr>
</tbody>
</table>

Whilst the IOPS required to support the Exchange deployment have greatly reduced from previous versions of Exchange it can be seen that they still need to be taken into account.
Example 6  Exchange transaction log capacity calculation

ABC123 Inc. has an average sent / received profile of 150 messages per day. Based on the figures in Example 1 and Table 28 on page 324 we can deduce that each user creates 30 MB of logs per day. The formula used to calculate the capacity required per user for logs is:

Number of users x number of logs created x 1 MB = capacity of logs generated.

In our example of 4,000 users per server will create (4000 x 30/1024 user log generation) + (4000 x 1/100 for move mailboxes) + (20 percent Log growth overhead factor) = Log capacity.

((120) + (40)) x 1.2 = 192 GB

In the performance and capacity solutions detailed in “Optimum number of drives required for databases” specify the number of drives required to meet both the database performance IOPS and the capacity. For the options derived for RAID 5 there is enough spare capacity and performance available so that no additional drives would be required to host the logs, however and additional 2 drives would be required for the RAID 1 option due to capacity.

Total number of drives required for databases and logs, for this example of 2,00 active and 2,000 passive users on a single Exchange Server 2010 server, are:

32 x 1 TB SATA drives in RAID 5 (3+1) or (7+1).
26 x 1 TB SATA drives in RAID 1.
It can be seen that any of the 3 main RAID types (RAID 1, RAID 5 and RAID 6) could host an Exchange Server 2010 environment and the costs involved for the differing RAID types had to be weighed up to determine the best overall cost effective solution. The method detailed in this section has been proven to produce consistent results when the information gathered and used as inputs for the storage design were properly collected. There are a number of challenges presented to both the Exchange and storage administrator when using this approach. First, this approach implies that this process could be required to be completed for every Exchange server within the enterprise. Building a “balanced” configuration on Symmetrix where Exchange Server A requires 17 x RAID 1/0 physical spindles, Server B requires 41 spindles, and Server C requires 23, and so on, can be a complicated, tedious, and time-consuming process that is further complicated when taking into account other disk-intensive operations such as Exchange VSS backup checksum workloads as well as other applications that may be configured to run on the Symmetrix as is not uncommon given its architecture.

Even when the calculations have been made on the data available at the time, any Microsoft Exchange Server 2010 mailbox database environment will often exhibit a perceived skewed workload across any given Exchange server due to the nature of the users’ e-mail workloads, I/O profiles, and concurrency. The storage design for any given Exchange Server 2010 Mailbox Server Role server is not always an exact science and while the number of physical disk drives can be derived from various sources, the exact nature of the workload cannot be fully analyzed until the server has been commissioned into production and detailed monitoring carried out. There can be one of three outcomes to this analysis:

- The I/O performance and latency statistics for the given number of physical drives and the layout are accurate and are performing to expectations and the storage within the array is utilized to the maximum.
- The I/O performance is low and the drive latency is too high, resulting in users’ e-mail performance experience being compromised.
The I/O performance and latency statistics for the given number of physical drives and the layout are accurate and are performing to expectations; however, after detailed analysis after commissioning into production it is found that the workload could have been achieved by using fewer physical drives.

For the two scenarios where the storage subsystem is either performing below par or has been over configured, the use of FAST, FAST VP, and Symmetrix VMAX can enable the system to be better balanced to meet overall storage I/O requirements. FAST and FAST VP will also enable Exchange and storage administrators to use storage types within a storage array to allow the use of the correct class of storage for the correct class of e-mail users.

For the current (Exchange Server 2010) and future versions of Exchange Server the amount of storage capacity needed to fulfill users’ mailbox requirements of greater than 10 GB lends itself to the use of Virtual Provisioning for the initial deployment and to grow, in staged increments, when the need arises. Whilst the figure of a “10 GB” user mailbox is currently the “standard” there will be many users within organizations that will never reach that limit. Thus having to provision the total capacity upon initial deployment is less cost-effective and Symmetrix VMAX Virtual Provisioning in conjunction with FAST VP is the ideal fit.
This appendix shows how Symmetrix device groups and composite groups are created for the TimeFinder family of products including TimeFinder/Mirror, TimeFinder/Clone, and TimeFinder/Snap. Topics include:

- TimeFinder/Mirror example.......................................................... 330
- TimeFinder/Clone example ........................................................... 332
- TimeFinder/SNAP example........................................................... 334
TimeFinder/Mirror example

This example shows you how to build and populate a device group and a composite group for TimeFinder/Mirror usage.

Device group

1. To create the device group, execute the command:
   ```
   symdg create dbgroup -type regular
   ```
2. The standard devices need to be added to the group. The database containers reside on five Symmetrix devices. The device numbers for these are 0CF, 0F9, 0FA, 0FB, and 101:
   ```
   symld -g dbgroup add dev 0CF
   symld -g dbgroup add dev 0F9
   symld -g dbgroup add dev 0FA
   symld -g dbgroup add dev 0FB
   symld -g dbgroup add dev 101
   ```
3. Associate the BCV devices to the group. The number of BCV devices should be the same as the number of standard devices. They should also be the same size. The device serial numbers of the BCVs used in the example are 00C, 00D, 063, 064, and 065:
   ```
   symbcv -g dbgroup associate dev 00C
   symbcv -g dbgroup associate dev 00D
   symbcv -g dbgroup associate dev 063
   symbcv -g dbgroup associate dev 064
   symbcv -g dbgroup associate dev 065
   ```

Composite group

1. To create the composite group, execute the command:
   ```
   symcg create dbgroup -type regular
   ```
2. The standard devices need to be added to the composite group. The database containers reside on five Symmetrix devices on two different Symmetrix arrays. The device numbers for these are
0CF, 0F9 on Symmetrix with the last three digits of 123, and device numbers 0FA, 0FB, and 101 on the Symmetrix with the last three digits of 456.

3. Associate the BCV devices to the composite group. The number of BCV devices should be the same as the number of standard devices. They should also be the same size. The device serial numbers of the BCVs used in the example are 00C, 00D, 063, 064, and 065:

```bash
symbcv -cg dbgroup associate dev 00C -sid 123
symbcv -cg dbgroup associate dev 00D -sid 123
symbcv -cg dbgroup associate dev 063 -sid 456
symbcv -cg dbgroup associate dev 064 -sid 456
symbcv -cg dbgroup associate dev 065 -sid 456
```
TimeFinder/Clone example

This example shows how to build and populate a device group and a composite group for TimeFinder/Clone usage.

Device group

1. To create the device group dbgroup, execute the command:
   symdg create dbgroup -type regular

2. The standard devices need to be added to the group. The database containers reside on five Symmetrix devices. The device numbers for these are 0CF, 0F9, 0FA, 0FB, and 101:
   symld -g dbgroup add dev 0CF
   symld -g dbgroup add dev 0F9
   symld -g dbgroup add dev 0FA
   symld -g dbgroup add dev 0FB
   symld -g dbgroup add dev 101

3. The target clone devices need to be added to the group. The targets for the clones can be standard devices or BCV devices. In this example, BCV devices are being used. The number of BCV devices should be the same as the number of standard devices. They should also be the same size or larger than the paired standard device. The device serial numbers of the BCVs used in the example are 00C, 00D, 063, 064, and 065:
   symbcv -g dbgroup associate dev 00C
   symbcv -g dbgroup associate dev 00D
   symbcv -g dbgroup associate dev 063
   symbcv -g dbgroup associate dev 064
   symbcv -g dbgroup associate dev 065

Composite group

1. To create the composite group dbgroup, execute the command:
   symcg create dbgroup -type regular
2. The standard devices need to be added to the group. The database containers reside on five Symmetrix devices on two different Symmetrix arrays. The device numbers for these are 0CF, 0F9 on Symmetrix with the last three digits of 123, and device numbers 0FA, 0FB, and 101 on the Symmetrix with the last 3 digits of 456:

   symcg -g dbgroup add dev 0CF -sid 123
   symcg -g dbgroup add dev 0F9 -sid 123
   symcg -g dbgroup add dev 0FA -sid 456
   symcg -g dbgroup add dev 0FB -sid 456
   symcg -g dbgroup add dev 101 -sid 456

3. Add the target for the clones to the device group. In this example, BCV devices are added to the composite group to simplify the later symclone commands. The number of BCV devices should be the same as the number of standard devices. They should also be the same size. The device serial numbers of the BCVs used in the example are 00C, 00D, 063, 064, and 065:

   symbcv -cg dbgroup associate dev 00C -sid 123
   symbcv -cg dbgroup associate dev 00D -sid 123
   symbcv -cg dbgroup associate dev 063 -sid 456
   symbcv -cg dbgroup associate dev 064 -sid 456
   symbcv -cg dbgroup associate dev 065 -sid 456
**TimeFinder/SNAP example**

This example shows how to build and populate a device group and a composite group for TimeFinder/SNAP usage.

---

**Device group**

1. To create the device group `dbgroup` execute the command:
   ```bash
   symdg create dbgroup -type regular
   ```

2. Add the standard devices to the group. The database containers reside on five Symmetrix devices. The device numbers for these are 0CF, 0F9, 0FA, 0FB, and 101:
   ```bash
   symld -g dbgroup add dev 0CF
   symld -g dbgroup add dev 0F9
   symld -g dbgroup add dev 0FA
   symld -g dbgroup add dev 0FB
   symld -g dbgroup add dev 101
   ```

3. Then the virtual devices or VDEVs need to be added to the group. The number of VDEVs should be the same as the number of standard devices. They should also be the same size. The device serial numbers of the VDEVs used in the example are 291, 292, 394, 395, and 396:
   ```bash
   symld -g dbgroup add dev 291 -vdev
   symld -g dbgroup add dev 292 -vdev
   symld -g dbgroup add dev 394 -vdev
   symld -g dbgroup add dev 395 -vdev
   symld -g dbgroup add dev 396 -vdev
   ```

---

**Composite group**

1. To create the composite group `dbgroup` execute the command:
   ```bash
   symcg create dbgroup -type regular
   ```
2. The standard devices need to be added to the composite group. The database containers reside on five Symmetrix devices on two different Symmetrix arrays. The device numbers for these are 0CF, 0F9 on Symmetrix with the last three digits of 123, and device numbers 0FA, 0FB, and 101 on the Symmetrix with the last three digits of 456:

```bash
symcg -g dbgroup add dev 0CF -sid 123
symcg -g dbgroup add dev 0F9 -sid 123
symcg -g dbgroup add dev 0FA -sid 456
symcg -g dbgroup add dev 0FB -sid 456
symcg -g dbgroup add dev 101 -sid 456
```

3. Then the virtual devices or VDEVs need to be added to the composite group. The number of VDEVs should be the same as the number of standard devices. They should also be the same size. The device serial numbers of the VDEVs used in the example are 291, 292, 394, 395, and 396:

```bash
symld -cg dbgroup add dev 291 -sid 123 -vdev
symld -cg dbgroup add dev 292 -sid 123 -vdev
symld -cg dbgroup add dev 394 -sid 456 -vdev
symld -cg dbgroup add dev 395 -sid 456 -vdev
symld -cg dbgroup add dev 396 -sid 456 -vdev
```
This appendix includes the following topic:
- Related documents .......................................................... 338
Related documents

The following is a list of related documents that may assist readers with more detailed information on topics described in this TechBook. All EMC documentation can be found at http://powerlink.emc.com.

**SYMCLI**

- Solutions Enabler Release Notes (by release)
- Solutions Enabler Support Matrix (by release)
- Solutions Enabler Symmetrix Device Masking CLI Product Guide (by release)
- Solutions Enabler Symmetrix Base Management CLI Product Guide (by release)
- Solutions Enabler Symmetrix CLI Command Reference (by release)
- Solutions Enabler Symmetrix Configuration Change CLI Product Guide (by release)
- Solutions Enabler Symmetrix SRM CLI Product Guide (by release)
- Solutions Enabler Symmetrix Double Checksum CLI Product Guide (by release)
- Solutions Enabler Installation Guide (by release)
- Solutions Enabler Symmetrix CLI Quick Reference (by release)

**TimeFinder**

- Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide (by release)

**SRDF**

- Solutions Enabler Symmetrix SRDF Family CLI Product Guide (by release)
- Symmetrix Remote Data Facility (SRDF) Product Guide
Replication Manager

- Replication Manager Product Guide
- Replication Manager Support Matrix

Microsoft

- http://www.microsoft.com/exchange
This glossary contains terms related to disk storage subsystems. Many of these terms are used in this manual.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>actuator</strong></td>
<td>A set of access arms and their attached read/write heads, which move as an independent component within a head and disk assembly (HDA).</td>
</tr>
<tr>
<td><strong>adapter</strong></td>
<td>Card that provides the physical interface between the director and disk devices (SCSI adapter), director and parallel channels (Bus &amp; Tag adapter), director and serial channels (Serial adapter).</td>
</tr>
<tr>
<td><strong>alternate track</strong></td>
<td>A track designated to contain data in place of a defective primary track. See also &quot;primary track.&quot;</td>
</tr>
<tr>
<td><strong>cache</strong></td>
<td>Random access electronic storage used to retain frequently used data for faster access by the channel.</td>
</tr>
<tr>
<td><strong>cache slot</strong></td>
<td>Unit of cache equivalent to one track.</td>
</tr>
<tr>
<td><strong>channel director</strong></td>
<td>The component in the Symmetrix subsystem that interfaces between the host channels and data storage. It transfers data between the channel and cache.</td>
</tr>
</tbody>
</table>
Glossary

**CKD**  Count Key Data, a data recording format employing self-defining record formats in which each record is represented by a count area that identifies the record and specifies its format, an optional key area that may be used to identify the data area contents, and a data area that contains the user data for the record. CKD can also refer to a set of channel commands that are accepted by a device that employs the CKD recording format.

**controller ID**  Controller identification number of the director the disks are channeled to for EREP usage. There is only one controller ID for Symmetrix.

**D**

**DASD**  Direct access storage device, a device that provides nonvolatile storage of computer data and random access to that data.

**data availability**  Access to any and all user data by the application.

**delayed fast write**  There is no room in cache for the data presented by the write operation.

**destage**  The asynchronous write of new or updated data from cache to disk device.

**device**  A uniquely addressable part of the Symmetrix subsystem that consists of a set of access arms, the associated disk surfaces, and the electronic circuitry required to locate, read, and write data. See also "volume."

**device address**  The hexadecimal value that uniquely defines a physical I/O device on a channel path. See also "unit address."

**device number**  The value that logically identifies a disk device in a string.

**diagnostics**  System level tests or firmware designed to inspect, detect, and correct failing components. These tests are comprehensive and self-invoking.

**director**  The component in the Symmetrix subsystem that allows Symmetrix to transfer data between the host channels and disk devices. See also "channel director."

**disk director**  The component in the Symmetrix subsystem that interfaces between cache and the disk devices.
dual-initiator  A Symmetrix feature that automatically creates a backup data path to the disk devices serviced directly by a disk director, if that disk director or the disk management hardware for those devices fails.

dynamic sparing  A Symmetrix feature that automatically transfers data from a failing disk device to an available spare disk device without affecting data availability. This feature supports all non-mirrored devices in the Symmetrix subsystem.

E

ESCON  Enterprise Systems Connection, a set of IBM and vendor products that connect mainframe computers with each other and with attached storage, locally attached workstations, and other devices using optical fiber technology and dynamically modifiable switches called ESCON Directors. See also “ESCON director.”

ESCON director  Device that provides a dynamic switching function and extended link path lengths (with XDF capability) when attaching an ESCON channel to a Symmetrix serial channel interface.

F

fast write  In Symmetrix, a write operation at cache speed that does not require immediate transfer of data to disk. The data is written directly to cache and is available for later destaging.

FBA  Fixed Block Architecture, disk device data storage format using fixed-size data blocks.

frame  Data packet format in an ESCON environment. See also “ESCON.”

FRU  Field Replaceable Unit, a component that is replaced or added by service personnel as a single entity.

G

gatekeeper  A small logical volume on a Symmetrix storage subsystem used to pass commands from a host to the Symmetrix storage subsystem. Gatekeeper devices are configured on standard Symmetrix disks.

GB  Gigabyte, $10^9$ bytes.
H

head and disk assembly  A field replaceable unit in the Symmetrix subsystem containing the disk and actuator.

home address  The first field on a CKD track that identifies the track and defines its operational status. The home address is written after the index point on each track. See also “CKD.”

hyper-volume extension  The ability to define more than one logical volume on a single physical disk device making use of its full formatted capacity. These logical volumes are user-selectable in size. The minimum volume size is one cylinder and the maximum size depends on the disk device capacity and the emulation mode selected.

I

ID  Identifier, a sequence of bits or characters that identifies a program, device, controller, or system.

IML  Initial microcode program loading.

index marker  Indicates the physical beginning and end of a track.

index point  The reference point on a disk surface that determines the start of a track.

INLINES  An EMC-provided host-based Cache Reporter utility for viewing short and long term cache statistics at the system console.

I/O device  An addressable input/output unit, such as a disk device.

K

Kilobyte, 1024 bytes.

L

least recently used algorithm (LRU)  The algorithm used to identify and make available the cache space by removing the least recently used data.

logical volume  A user-defined storage device. In the Model 5200, the user can define a physical disk device as one or two logical volumes.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>long miss</strong></td>
<td>Requested data is not in cache and is not in the process of being fetched.</td>
</tr>
<tr>
<td><strong>longitude redundancy code (LRC)</strong></td>
<td>Exclusive OR (XOR) of the accumulated bytes in the data record.</td>
</tr>
<tr>
<td><strong>M</strong></td>
<td>Megabyte, $10^6$ bytes.</td>
</tr>
<tr>
<td><strong>mirroring</strong></td>
<td>The Symmetrix maintains two identical copies of a designated volume on separate disks. Each volume automatically updates during a write operation. If one disk device fails, Symmetrix automatically uses the other disk device.</td>
</tr>
<tr>
<td><strong>mirrored pair</strong></td>
<td>A logical volume with all data recorded twice, once on each of two different physical devices.</td>
</tr>
<tr>
<td><strong>P</strong></td>
<td>Physical identification number of the Symmetrix director for EREP usage. This value automatically increments by one for each director installed in Symmetrix. This number must be unique in the mainframe system. It should be an even number. This number is referred to as the SCU_ID.</td>
</tr>
<tr>
<td><strong>primary track</strong></td>
<td>The original track on which data is stored. See also “alternate track.”</td>
</tr>
<tr>
<td><strong>promotion</strong></td>
<td>The process of moving data from a track on the disk device to cache slot.</td>
</tr>
<tr>
<td><strong>R</strong></td>
<td></td>
</tr>
<tr>
<td><strong>read hit</strong></td>
<td>Data requested by the read operation is in cache.</td>
</tr>
<tr>
<td><strong>read miss</strong></td>
<td>Data requested by the read operation is not in cache.</td>
</tr>
<tr>
<td><strong>record zero</strong></td>
<td>The first record after the home address.</td>
</tr>
<tr>
<td><strong>S</strong></td>
<td>The process of reading, checking the error correction bits, and writing corrected data back to the source.</td>
</tr>
<tr>
<td><strong>Glossary</strong></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td><strong>SCSI adapter</strong></td>
<td>Card in the Symmetrix subsystem that provides the physical interface between the disk director and the disk devices.</td>
</tr>
<tr>
<td><strong>short miss</strong></td>
<td>Requested data is not in cache, but is in the process of being fetched.</td>
</tr>
<tr>
<td><strong>SSID</strong></td>
<td>For 3990 storage control emulations, this value identifies the physical components of a logical DASD subsystem. The SSID must be a unique number in the host system. It should be an even number and start on a zero boundary.</td>
</tr>
<tr>
<td><strong>stage</strong></td>
<td>The process of writing data from a disk device to cache.</td>
</tr>
<tr>
<td><strong>storage control unit</strong></td>
<td>The component in the Symmetrix subsystem that connects Symmetrix to the host channels. It performs channel commands and communicates with the disk directors and cache. See also &quot;channel director.”</td>
</tr>
<tr>
<td><strong>string</strong></td>
<td>A series of connected disk devices sharing the same disk director.</td>
</tr>
<tr>
<td><strong>U</strong></td>
<td></td>
</tr>
<tr>
<td><strong>unit address</strong></td>
<td>The hexadecimal value that uniquely defines a physical I/O device on a channel path. See also &quot;device address.”</td>
</tr>
<tr>
<td><strong>V</strong></td>
<td></td>
</tr>
<tr>
<td><strong>volume</strong></td>
<td>A general term referring to a storage device. In the Symmetrix subsystem, a volume corresponds to single disk device.</td>
</tr>
<tr>
<td><strong>W</strong></td>
<td></td>
</tr>
<tr>
<td><strong>write hit</strong></td>
<td>There is room in cache for the data presented by the write operation.</td>
</tr>
<tr>
<td><strong>write miss</strong></td>
<td>There is no room in cache for the data presented by the write operation.</td>
</tr>
</tbody>
</table>
Microsoft Exchange Server on EMC Symmetrix Storage Systems

Index

A
Adaptive copy 59
Asynchronous SRDF 59
AUTOGROW 158
Autopropatinoning Groups 116

B
BCV 72, 73, 86
bin file 112, 113, 151

C
Cache 49, 58
Change Tracker 47, 56
CKD 61
Composite groups 59
Con group trip 60, 62
Concurrent SRDF 64
Consistency group 62
Consistency groups 46, 59, 60, 61, 68, 86
Crash recovery 78

D
Data devices 147, 149
Dependent-write consistency 62, 67
Device group 59
device move 126
device swap 127
DR 65

E
Enginuity 46, 49, 51, 151
Enginuity Consistency Assist 60, 61, 62, 76, 77
ESCON 49, 57

F
FAST 123
FAST Policies 124
FAST Policy 126
FBA 61
Fibre Channel 49, 57, 112, 113
FICON 49

G
Gigabit Ethernet 49, 57

H
HBA 112, 113, 115, 118, 148

I
Instant File Initialization 153
iSCSI 49, 112

L
LUN masking 112, 115
LUN Offset 115

M
Metavolumes 151
Mirror positions 73
<table>
<thead>
<tr>
<th>O</th>
<th>Open Replicator 159</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Path failover 93 Path load balancing 92, 93 Path management 92 PowerPath 48, 60, 92, 121</td>
</tr>
<tr>
<td>R</td>
<td>RA group 57, 60, 68 RAID 1 49 RAID 5 49 RAID 6 49 Remote adapter 57 Restartable databases 68 Rolling disaster 61, 62</td>
</tr>
<tr>
<td>S</td>
<td>SAN 112, 113, 114 SNMP 160 Solutions Enabler 46, 47, 53 SRDF 46, 56, 57, 58, 59, 62, 63, 65, 67, 72, 159 SRDF adaptive copy 59 SRDF Data Mobility 65 SRDF establish 66, 67 SRDF failback 69, 70 SRDF failover 69 SRDF restore 68 SRDF Split 67 SRDF/A 59 SRDF/AR 59 SRDF/CE 71 SRM 47 Storage Class 123 Storage Group 123, 125 Storage Type 124, 126 SYMAPI 46, 53, 77, 160 SYMCLI 53, 56, 61, 72, 85, 88, 89, 115 symclone 72, 79, 80 symmir 72, 73, 75, 76 symsnap 72, 81, 83, 84 Synchronous SRDF 58</td>
</tr>
<tr>
<td>T</td>
<td>Temporary table spaces 158 Thin device 147, 148, 149, 154, 159 Thin pool 147, 149 TimeFinder 47, 56, 65, 72, 159 TimeFinder/CG 72 TimeFinder/Clone 47, 72, 79 TimeFinder/Mirror 47, 72, 73 TimeFinder/Mirror establish 73 TimeFinder/Mirror restore 75 TimeFinder/Mirror split 73 TimeFinder/Snap 47, 72, 81, 83</td>
</tr>
<tr>
<td>V</td>
<td>VDEV 81, 84 Virtual Provisioning 104, 146</td>
</tr>
<tr>
<td>W</td>
<td>WWN 113, 114, 115, 118, 148</td>
</tr>
<tr>
<td>Z</td>
<td>Zoning 112</td>
</tr>
</tbody>
</table>