Self Assessment and Strategy Guide for Migrating from Domino Document Manager

Assessing your Domino Document Manager Environment

Understanding your migration options

Best Practice considerations and tools for migrating

Cali Clarke
Femke Goedhart
John Head
Jelan Heidelberg
Wojciech Kowalski
Daniel Lieber
Matthew Mangels
Steve Peters
Cynthia Schiegoleit
Stephan Wissel

ibm.com/redbooks
Note: Before using this information and the product it supports, read the information in “Notices” on page xi.

First Edition (September 2009)

This edition applies to Domino Document Manager, Release 6.5, and Release 7.
Contents

Part 1. Overview ................................................................................................................... 1

Chapter 1. Introduction ......................................................................................................... 3
  1.1 The Transition Plan ...................................................................................................... 4
    1.1.1 Key dates: ............................................................................................................ 4
    1.1.2 Entitlement to replacement product: ................................................................. 4
    1.2 Proposed Timeline ................................................................................................. 4
    1.2.1 Assistance: .......................................................................................................... 5
    1.3 Beginning to think about the transition options ....................................................... 5
    1.4 What is in this IBM Redbooks publication ............................................................... 7

Chapter 2. Lotus Quickr primer ................................................................................................ 9
  2.1 What is IBM Lotus Quickr? ........................................................................................ 10
    2.1.1 Team Places ........................................................................................................ 10
    2.1.2 Standard Team Place ......................................................................................... 12
    2.1.3 Content libraries ............................................................................................... 13
    2.1.4 Blog ................................................................................................................... 16
    2.1.5 Wiki ................................................................................................................... 17
    2.1.6 Rooms ................................................................................................................ 18
    2.2 Connectors ............................................................................................................... 18
      2.2.1 Lotus Notes or Outlook e-mail ........................................................................ 19
      2.2.2 Microsoft Office and Lotus Symphony ............................................................. 20
      2.2.3 Windows Explorer .......................................................................................... 22
      2.2.4 Sametime .......................................................................................................... 23
    2.3 Templates ................................................................................................................. 24
    2.4 Personal file sharing ................................................................................................. 25
    2.5 Lotus Quickr + Enterprise Content Management (ECM). ................................... 28
      2.5.1 Quickr + FileNet ............................................................................................. 29
      2.5.2 Lotus Quickr + Content Manager ................................................................. 30
    2.6 Conclusion ............................................................................................................... 31

Chapter 3. Is Enterprise Content Management in your future? ........................................ 33
  3.1 The growing need for managing content. ................................................................. 34
    3.1.1 Does your organization need ECM? ................................................................. 36
  3.2 The human aspect ....................................................................................................... 37
  3.3 Principles of end-to-end ECM .................................................................................. 38
    3.3.1 Maximize user acceptance ............................................................................... 38
    3.3.2 Minimize the Burden on IT ............................................................................. 38
    3.3.3 Meet diverse organizational needs ................................................................. 39
    3.3.4 Content Federation Across the Enterprise ...................................................... 39
6.3 Are there other big IT projects that might affect the timeline? .......................... 82
6.3.1 Introducing Lotus Connections or Lotus Sametime ................................. 82
6.3.2 Introducing an ECM solution .......................................................... 83
6.3.3 Changing your enterprise e-mail system ............................................. 83
6.3.4 Changing hardware and operating system platforms ............................. 84
6.3.5 Outsourcing your IT function ............................................................ 84
6.4 Given the collaborative capabilities offered through Lotus Quickr, can this be an opportunity to move forward? ................................. 84
6.5 Are there architectures and standards in your organization that are relevant to your DDM migration plan? .................................................. 85
6.5.1 Are you already a Lotus Quickr customer? ........................................... 85
6.5.2 Are you already vested in an ECM solution? ......................................... 85

Part 3. Making the Mental Leap ............................................................... 87

Chapter 7. Hierarchical and structural considerations between Domino Document Manager and Quickr ............................................................... 89
7.1 Hierarchy structure .................................................................................. 90
7.1.1 Basic hierarchy concepts of DDM ....................................................... 90
7.1.2 Basic hierarchy concepts of Lotus Quickr ......................................... 92
7.1.3 Hierarchy comparison ....................................................................... 93
7.1.4 Mapping example .............................................................................. 95
7.2 Security .................................................................................................. 96
7.2.1 DDM security .................................................................................... 96
7.2.2 Quickr Security ................................................................................ 97
7.2.3 Security Comparison ......................................................................... 99
7.2.4 Security mapping ............................................................................. 99
7.3 Database structure .................................................................................. 101
7.3.1 DDM database Structure ................................................................. 101
7.3.2 Lotus Quickr structure ...................................................................... 104
7.3.3 Database structure comparison ........................................................ 106
7.4 Document model ................................................................................... 108
7.4.1 Domino.Doc document model ............................................................ 108
7.4.2 Lotus Quickr document model ........................................................... 109
7.4.3 Data types mapping ......................................................................... 109
7.5 Document management functionality ...................................................... 109
7.5.1 Document management features in DDM ......................................... 109
7.5.2 Document management features in Quickr .................................... 111
7.5.3 DDM features ................................................................................... 114
7.5.4 Lotus Quickr features ...................................................................... 115
7.5.5 Feature comparison between DDM and Lotus Quickr ....................... 116
7.6 User Interfaces ....................................................................................... 117
7.6.1 DDM Interfaces ............................................................................... 117
7.6.2 Quickr Interfaces ............................................................................ 118
7.6.3 Interface Comparison ....................................................................... 118
7.7 Conclusion ............................................................................................. 119

Part 4. Digging into Customized Domino Document Manager Implementations ................................. 121

Chapter 8. Design element review ................................................................. 123
8.1 Reviewing your implementations ............................................................ 124
8.1.1 Site administration database ............................................................. 124
8.1.2 Library .............................................................................................. 124
8.1.3 File cabinets ..................................................................................... 126
12.14.5 Collaborative authoring and versioning ................................. 251
12.14.6 Searching .............................................................................. 251
12.14.7 Performance issues using hook event placebots ....................... 252
12.15 Creating a placetype .................................................................. 252

Appendix A. Domino Document Manager Site Map and Security Analysis Tool ........................................................................ 253
Overview of Domino Document Manager Site Map and Security Analysis Tool ................................................................. 254
Installing the tool .............................................................................. 255
Components of the Domino Document Manager Site Map and Security Analysis Tool ......................................................... 255
How and where to download the tool ................................................... 258

Appendix B. Summary questionnaire for technical assessment .............. 259
Summary: Key questions regarding your DDM implementation. ......... 260
Functional overview ......................................................................... 260
Implementation Details .................................................................... 261
Customizations ................................................................................. 262

Appendix C. Scenario priority matrix ................................................... 263
Determining your Domino Document Manager complexity level ........ 264

Appendix D. Reference hook methods for Lotus Quickr ......................... 265
D.1 Reference: Hook methods for Lotus Quickr .................................... 266
D.1.1 ServicesApproveDraft ............................................................... 266
D.1.2 ServicesApproveOrCheckinDraft ............................................ 266
D.1.3 ServicesCancelDocument ....................................................... 266
D.1.4 ServicesCheckinDocument ...................................................... 266
D.1.5 ServicesCopyDocument .......................................................... 267
D.1.6 ServicesCopyFolder ............................................................... 267
D.1.7 ServicesCreateDraftOrDocument ............................................ 267
D.1.8 ServicesCreateMember .......................................................... 268
D.1.9 ServicesCreateVersion ........................................................... 268
D.1.10 ServicesDeleteDocument ...................................................... 268
D.1.11 ServicesDeleteMember ......................................................... 269
D.1.12 ServicesLockDocument ........................................................ 269
D.1.13 ServicesMoveDocument ....................................................... 269
D.1.14 ServicesMoveFolder ............................................................. 269
D.1.15 ServicesRejectDraft ............................................................. 270
D.1.16 ServicesRenameDocument .................................................... 270
D.1.17 ServicesRestoreVersion ........................................................ 270
D.1.18 ServicesSubmitDraft ........................................................... 270
D.1.19 ServicesUnlockDocument ..................................................... 271
D.1.20 ServicesUpdateDraftContents .............................................. 271
D.1.21 ServicesUpdateDraftOrDocument .......................................... 271
D.1.22 ServicesUpdateFolder .......................................................... 272
D.1.23 h_CreateRoom ....................................................................... 272
D.1.24 h_ChangeACL ....................................................................... 272
D.1.25 h_CreateFolder ..................................................................... 273
D.1.26 h_CreateGroup ...................................................................... 273
D.1.27 h_CreateOffice ...................................................................... 274
D.1.28 h_CreateUsers ....................................................................... 274
D.1.29 h_DeleteGroups ..................................................................... 275
D.1.30 h_DeleteUsers ....................................................................... 275
D.1.31 h_SetAesthetic ...................................................................... 276
D.1.32 h_UpdateGroup ...................................................................... 276
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.1.33 h_UpdateUser</td>
<td>277</td>
</tr>
<tr>
<td>D.1.34 h_DeleteFolder</td>
<td>277</td>
</tr>
<tr>
<td>D.1.35 h_DeleteOffice</td>
<td>277</td>
</tr>
<tr>
<td>D.1.36 h_DeleteRoom</td>
<td>277</td>
</tr>
<tr>
<td>D.1.37 h_DeletePages</td>
<td>278</td>
</tr>
<tr>
<td>D.1.38 h_MoveRoom</td>
<td>278</td>
</tr>
<tr>
<td>D.1.39 h_ReorderRoomArea</td>
<td>278</td>
</tr>
<tr>
<td>D.1.40 h_SendMail</td>
<td>278</td>
</tr>
<tr>
<td>D.1.41 h_SpellCheck</td>
<td>279</td>
</tr>
<tr>
<td>D.1.42 h_CopyMoveFolder</td>
<td>279</td>
</tr>
<tr>
<td>D.1.43 h_ServerBasics</td>
<td>279</td>
</tr>
<tr>
<td>D.1.44 h_GetSkinGroupName</td>
<td>279</td>
</tr>
<tr>
<td>D.1.45 h_DeleteSkinGroup</td>
<td>280</td>
</tr>
<tr>
<td>D.1.46 h_PublishSkinGroup</td>
<td>280</td>
</tr>
<tr>
<td>12.15.1 h_PublishPlaceType</td>
<td>280</td>
</tr>
<tr>
<td>D.1.47 h_RefreshPlaceTypesGallery</td>
<td>280</td>
</tr>
<tr>
<td>D.1.48 h_DeletePlaceType</td>
<td>281</td>
</tr>
<tr>
<td>D.1.49 h_RefreshPlaceFromPlaceType</td>
<td>281</td>
</tr>
<tr>
<td>12.15.2 h_SetNewUsersAllowed</td>
<td>281</td>
</tr>
<tr>
<td>D.1.50 h_RunAgent</td>
<td>281</td>
</tr>
<tr>
<td>D.1.51 h_PreProcessForm</td>
<td>282</td>
</tr>
<tr>
<td>D.1.52 h_DeleteAgent</td>
<td>282</td>
</tr>
<tr>
<td>D.1.53 h_EditAgent</td>
<td>282</td>
</tr>
<tr>
<td>D.1.54 h_CheckMemberName</td>
<td>283</td>
</tr>
<tr>
<td>D.1.55 h_CreateAgent</td>
<td>283</td>
</tr>
<tr>
<td>D.1.56 h_ExternalDirectory</td>
<td>284</td>
</tr>
<tr>
<td>D.1.57 h_CopyAgent</td>
<td>285</td>
</tr>
<tr>
<td>D.1.58 h_MoveCopyPages</td>
<td>285</td>
</tr>
<tr>
<td>D.1.59 h_CopyForm</td>
<td>286</td>
</tr>
<tr>
<td>D.1.60 h_EditIndex</td>
<td>286</td>
</tr>
<tr>
<td>D.1.61 h_EditFolder</td>
<td>286</td>
</tr>
<tr>
<td>D.1.62 h_ConfigureMailSettings</td>
<td>287</td>
</tr>
<tr>
<td>12.15.3 h_LookupMembers</td>
<td>288</td>
</tr>
<tr>
<td>D.1.63 h_ReorderItems</td>
<td>288</td>
</tr>
<tr>
<td>D.1.64 h_DeleteAttachments</td>
<td>288</td>
</tr>
</tbody>
</table>

**Appendix E. Additional material**

- Locating the Web material ........................................ 289
- Using the Web material ........................................... 289

**Related publications** ........................................... 291

- Online resources ................................................. 291
- How to get Redbooks .............................................. 291
- Help from IBM .................................................... 291

**Index** ............................................................. 293
Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

1-2-3®, AIX®, AS/400®, DB2®, Domino®, FileNet®, i5/OS®, IBM®, iSeries®, Lotus Notes®, Lotusphere®, LotusScript®, Notes®, Passport Advantage®, QuickPlace®, Quickr™, Redbooks®, Redpaper™, Redbooks (logo)®, Sametime®, Symphony™, WebSphere®, z/OS®

The following terms are trademarks of other companies:

Adobe, and Portable Document Format (PDF) are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

FileNet, and the FileNet logo are registered trademarks of FileNet Corporation in the United States, other countries or both.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

J2EE, Java, JavaScript, JVM, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

ActiveX, Excel, Internet Explorer, Microsoft, MS, Outlook, PowerPoint, SQL Server, Visual Basic, Visual C++, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

This IBM® Redbooks® publication is intended to help your organization plan and prepare for a migration away from Lotus® Domino® Document Manager (formerly known as Domino.Doc). This IBM Redbooks publication focuses primarily on analyzing, planning, and preparing with a portion devoted to the actual process of migrating to another solution. While there is no exact equivalent to Domino Document Manager (DDM), we discuss IBM Lotus Quickr™ in detail and describe how your organization can achieve many of the same functional goals using Lotus Quickr services for Domino. We introduce the tool for migrating, Lotus Quickr Migrator: Domino Document Manager Edition, and discuss how you can use this for migrating content.

Moving beyond Domino Document Manager requires some new thinking about your organizations’ document management and content management needs. In many cases, Lotus Quickr may offer a collaborative solution that can provide significant benefit to how your organization creates, shares, and stores content. In other cases, this may present an opportunity for you to re-evaluate if you need a larger scale, more robust enterprise content management solution.

This IBM Redbooks Publication is intended to help you in making an informed decision about the best way forward. We help you to better understand and analyze your current Domino Document Manager environment and guide you in determining the best options for moving forward. With proper planning and analysis up front, this book will help you recognize and overcome the potential obstacles and challenges, and help you explore the opportunities opened up by alternate content management solutions.

The team who wrote this book

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

Cali Clarke is a Senior IT Specialist in the UK specializing in all aspects of content from document and team collaboration to Web Content Management. Cali joined Lotus 20 years ago working in Technical Support, as a Notes Trainer and then as a Notes Administrator & Developer before her current role in Technical Sales. She has worked with Domino.Doc since the first beta release in 1997 and prior to that with Lotus Notes® Document Imaging (LN:DI). For many years, she worked with customers and Business Partners as a Document Management specialist. In her current role she works with large and small customers to develop and facilitate their use of Lotus collaborative technologies.
Femke Goedhart is a Senior Consultant working for Silverside (http://www.silverside.nl), situated in Gouda, The Netherlands. Silverside, an IBM award-winning Business Partner, specializes in Document Management, Social Intranet, and Enterprise Search. Femke has worked with IBM Lotus products for 10 years. Her main focus is translating and analyzing complex business requirements into technical Document Management, Collaboration, and Workflow solutions. Apart from her Development and analytical skills, she has been involved in more then 30 projects with IBM products such as Domino Document Manager, Domino Workflow, and Lotus Quickr.

John Head is a Chicago-based consultant. He currently works full-time with PSC Group LLC, where he has served as Director of Enterprise Collaboration since 2000. Throughout his career, he has focused on the development and implementation of enterprise collaboration solutions. For this, and other work he has been recognized with IBM Lotus and IBM Beacon awards and is a regularly-featured speaker at Lotusphere® and other IBM and Microsoft® technology conferences and user groups around the world. In 2008, John was asked to be the technical editor for Lotus Symphony for Dummies. He is currently focused on providing solutions for Notes/Domino & Microsoft Office Integration, Document Generation with ODF and OOXML, and Enterprise Social Media. You can read his blog at http://www.johndavidhead.com or follow him on twitter at http://twitter.com/johnhead.

Jelan Heidelberg is a Senior Offering Manager responsible for document and team collaboration, and social software including Lotus Quickr, Lotus Domino Document Manager, and Lotus Connections. In this role, she works with team members across the organization to drive the overall product strategy, set pricing and packaging, develop sales materials and go-to-market planning, and integrate with the teams for other Lotus products. Before assuming this role, Jelan spent 8 years in the IBM hardware organization in product marketing. She served as the liaison between the hardware organization and Lotus, with a particular focus on building the Domino on AS/400® business and an overall server consolidation product and sales strategy around Lotus products. Jelan spent the middle part of her career working in the AS/400 product development laboratory in Rochester, Minnesota. She was responsible for AS/400 documentation in a variety of areas: security, backup and recovery, CISC-to-RISC upgrades, and Domino. Jelan began her career as an IBM systems engineer. She spent 10+ years working with both large and small organizations, helping them set their technical strategies and plan implementation of application software. She facilitated numerous executive planning sessions with her customers, both to help them set long term strategies and to jointly develop detailed project plans for the implementation of specific applications. Jelan is the author of several books and numerous articles. She is a frequent speaker at customer conferences and sales education events around the world.
Wojciech Kowalski is a Senior IT Specialist with IBM Software Services for Lotus (ISSL). He joined IBM in 2000 and worked for the Lotus division of IBM Software Group, IBM Software Group Marketing and Technical Sales Teams in Poland. Since 2006 he is a member of the ISSL CEEMEA Team, providing consulting services in Europe and the Middle East. His areas of expertise include IBM Lotus Notes & Domino, WebSphere® Portal, and Lotus Quickr on various platforms, and he specializes in customization, interoperability, integration with non-IBM products, and performance tuning. He also serves as a dedicated problem solver, working on complex software-related issues reported by IBM customers.

Daniel Lieber is President of Innovative Ideas Unlimited, Inc. He has over 20 years of experience working in information technology and over 15 years working with Lotus content systems. He has designed numerous solutions for organizations of all sizes around the world and has taught thousands of people how to use Lotus software. He has written and presented extensively on Lotus content systems and compliance. He holds a Master of Business Administration degree with a focus on Management of Technology from Bentley University along with a post-graduate certificate in Business Communication.

Matthew Mangels is Senior Consultant of Innovative Ideas Unlimited, Inc. He has over 12 years of experience working in information technology and with Lotus products. He has extensive experience in analyzing business requirements, developing sophisticated applications, and training technical professionals. He is the principal architect of the IIUI products and has written extensively on Domino and Domino Document Manager. He is a Certified Public Accountant and has a Master of Science in Taxation degree from Bentley University.

Steve Peters is a Principal Consultant for Peters Technology Group, a 14 year IBM business partner headquartered in Minneapolis, MN. For over 20 years, he has been working with companies to help them realize the benefits of document-based collaborative computing. Peters consulting expertise ranges from assisting clients in building business cases to requirements gathering and analysis, architectural design, system integration, and solution implementation. In particular, he has worked extensively with Lotus Notes and Domino as well as the Lotus extended products including: Domino Document Manager, Domino Workflow, Lotus Quickr and Lotus Forms. In 2000, Steve facilitated the IBM acquisition of the “Desktop Enabler Plus” (the Domino Document Manager MS® Windows® interface) a product that he designed. He is the co-author of the IBM Redbooks Publication Wiki: Customizing Lotus Quickr 8.1. Peters holds Bachelors degrees in Computer Science and Mathematics from the University of Minnesota and a Masters of Software Design and Development from the University of St. Thomas in St. Paul, MN.

Cynthia Schiegoleit is a Senior Consultant with IBM Software Services for Lotus and has been with IBM for 9 years. She has extensive experience delivering solutions based upon Lotus' WPLC product portfolio, specializing in Lotus Notes and Domino, Lotus Web Content Management and Lotus Quickr. Cynthia has a passion for working with customers to help them achieve their enterprise collaboration and content management goals. She holds a BS in Computer Science from the University of Massachusetts, Lowell.
Stephan Wissel is the Lotus Technology & Productivity Advisor for IBM AP based in IBM Singapore. As a member of IBM Techworks he helps customers to take full advantage of Lotus’ products. He has worked with Lotus Technology since the days of Lotus 1-2-3® 1.0 and Lotus Notes 2.1. He worked with various IBM Business Partners in Europe and Asia as developer, trainer, consultant, and architect before joining IBM in 2006. On his blog (http://www.wissel.net) he has contributed hundreds of technical articles around Lotus technologies and software development.

John Bergland is a project leader at the ITSO, Cambridge Center. He manages projects that produce Redbooks publications that focus on IBM WebSphere and Lotus software technology. Before joining the ITSO in 2003, John worked as an IT Specialist with IBM Software Services for Lotus, specializing in Lotus Notes and Domino messaging and collaborative solutions. He holds a Master of Business Administration degree and a Master of Science degree in information technology from Boston University.

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

Mustansir Banatwala  
Chief Programmer, Lotus Quickr  
IBM Software Group, WPLC,  
IBM, Westford, MA

Peter van de Graaf  
Program Director, Social Computing Software  
IBM Software Group, WPLC,  
IBM, Westford, MA

Bruce Roberts  
Architect, Lotus Quickr  
IBM Software Group, WPLC,  
IBM, Westford, MA

David E. Kajmo  
Senior Product Manager, Lotus Quickr  
IBM Software Group, WPLC,  
IBM, Westford, MA

Heidi R. Votaw  
Program Director, Social Computing Software  
IBM Software Group, WPLC,  
IBM, Westford, MA

Diane F. Loomis  
Senior Offering Manager, Team Collaboration and Document Services,  
IBM Software Group, WPLC,  
IBM, Westford, MA

Cengiz H. Satir  
Cross Software Group Sr. Product Offering Manager,  
IBM Software Group, WPLC,  
IBM, Westford, MA
Become a published author

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

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

Find out more about the residency program, browse the residency index, and apply online at: ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

▶ Use the online Contact us review Redbooks form found at:
  ibm.com/redbooks

▶ Send your comments in an e-mail to:
  redbooks@us.ibm.com

▶ Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400
Overview
Chapter 1. Introduction

Change is inevitable, especially when it comes to the world of information technology. Certainly, change brings obstacles and challenges to overcome. But it also brings opportunities. This book is about change and the information that you need to cope and to come out on top.

If you are reading this book, you are no doubt aware that IBM has announced the withdrawal of Lotus Domino Document Manager (formerly known as Domino.Doc), with sufficient warning for you to plan, to prepare, and to rise to the challenge of what to do next. Our goal with this book is to get you started, to help you recognize and overcome the potential obstacles and challenges, and to help you explore the opportunities opened up by alternate solutions.

There is no exact equivalent to Domino Document Manager (DDM) that we can offer as your next step. The options available are both less than DDM and more. Perhaps most importantly, they are different. You will have to make some mental leaps to go from what you are used to with DDM to how things might be organized and presented in a replacement product. We will help you make those leaps and to make the adjustments as easy as possible for your users. And along the way, we will also point out the benefits, so that you can say “Yes, I know it doesn't look exactly the same, but look at what else you can do once you learn the new way.”
1.1 The Transition Plan

IBM has announced the end of life for DDM. This section details the basics of the DDM transition plan.

1.1.1 Key dates:

- May 12, 2009: DDM product no longer available for purchase
- September 30, 2012: End of Service. After this date, support is available only by purchasing an extension contract.

1.1.2 Entitlement to replacement product:

For each user license of DDM (on current maintenance), you are entitled to one Lotus Quickr user license. This entitlement should happen automatically in your Passport Advantage® record with no action required on your part. As long as you continue with software maintenance on the Lotus Quickr license, you will also receive support for DDM (through September 2012).

You have the option of exchanging your entitlement to Lotus Quickr for one of the software bundles (listed below) that include both Lotus Quickr and a restricted license to an IBM enterprise content management (ECM) solution. To make this exchange, contact your IBM representative. You have until September 2010 to make the switch, so take some time to evaluate and think about it. Here are the bundles that you can choose:

- IBM Content Manager Collaboration Edition or
- IBM FileNet® Content Manager Collaboration Edition

1.2 Proposed Timeline

The timeline in Figure 1-1 illustrates key events between now and the end of normal support for DDM.

Summary of Time Line

- Withdrawal from marketing (no new sales)
- Entitlement to replacement product(s)
- Publication of Self-Assessment Guide
- Availability of migration tool (through IBMers and BPs)
- Office 2007 fix for Dom Doc 6.5 & Dom Doc 7
- Quickr “next” including document management & API enhancements
- End of normal support
- Service extensions available

Figure 1-1 Timeline of significant events between now and end of life support for DDM
1.2.1 Assistance:

To facilitate the transition from DDM to the appropriate product, we are providing you with the following resources:

- This IBM Redbooks publication
  IBM brought together a group of people who collectively have many years of experience with DDM, working in different parts of the world with many different types of organizations. We spent hours evaluating and debating options to arrive at our best recommendations for the scenarios that we see most frequently in our DDM clients. Our objective is to help you help yourselves, not necessarily through the entire journey, but through much of the preparation phase. Most of us earn our living as consultants, and we believe that you can use your resources and our time most effectively if you spend time analyzing, evaluating, and preparing before you call in the outside expert. Use this book to do your homework before you seek advice.

- The Lotus Quickr Migrator: Domino Document Manager Edition
  This tool will help you move your content from DDM to Lotus Quickr (or part of the way to other repositories if you choose a different path). The tool will be available at no charge from trained IBM employees and IBM Business Partners.

  **Attention:** The Lotus Quickr Migrator: Domino Document Manager Edition is available (at no charge) through trained IBM personnel or IBM Business Partners. Consult the following DDM Web site or your IBM representative for more information about obtaining the tool and assistance with your migration planning:
  

- A wiki
  As the community of people who are knowledgeable about DDM gains more experience during the transition process, we will post advice, tips and techniques in this wadi. You'll find it linked from the DDM home page:
  

1.3 Beginning to think about the transition options

As we discuss in the Redbooks publication (Chapter 7, “Hierarchical and structural considerations between Domino Document Manager and Quickr” on page 89), there is no exact equivalent to DDM that we can offer as your next step. The options available are both less than DDM and more. We will continue to come back to this theme throughout this Redbooks publication, highlighting the benefits and considerations with several of the possible transition options. Figure 1-2 on page 6 illustrates these transition options. This also highlights the focus that for much of this Redbooks publication, we make the assumption that Lotus Quickr (services for Domino) will be the primary migration destination.
When we refer to there being no exact equivalent to Domino Document Manager, Figure 1-3 illustrates how DDM falls in the middle, with functionality ranging between the robust capabilities of a larger scale ECM solution, and the collaborative, content sharing capabilities of Lotus Quickr.

You have likely noticed that your organization’s needs for managing content have most likely become more complex. The quantity of content to manage has grown dramatically, while the types of content span a much broader range than simply documents. Figure 1-4 on page 7 illustrates the concept that as the volume and complexity of content increases, there is also an increasing need for greater collaboration between team members in using this content. This illustrates the range of options from personal file sharing and growing toward a larger scale ECM solution. Figure 1-4 on page 7 illustrates the concept that there is a broad range of solutions, ranging from Personal File Sharing to a larger scale ECM solution. Chapter 3, “Is Enterprise Content Management in your future?” on page 33 goes into depth about the issues from which your organization may benefit in moving to a larger scale ECM solution.
1.4 What is in this IBM Redbooks publication

- Section One: Overview
  - Chapter 2, “Lotus Quickr primer” on page 9. We think that Lotus Quickr will be part of the solution for most customers who have DDM today, and for many of you it will be the whole solution. In this chapter, we introduce you to the document management capabilities of Lotus Quickr and to its collaboration side. In addition, we preview new features that will may make it a better option if you wait until early 2010.
  - Chapter 3, “Is Enterprise Content Management in your future?” on page 33. Some organizations have a fairly large, complex deployment of DDM and should consider stepping up to an ECM solution. In this chapter, we explain what an ECM is and what characteristics might signal an ECM in your future. We also introduce two different ECM options from IBM:
    - IBM Content Manager
    - IBM FileNet Content Manager
  - Chapter 4, “Scenarios: Which might apply to you?” on page 45. A deceptively short title for what is probably the most important chapter in this book. DDM is a flexible product that organizations have used in many interesting and unique ways. This chapter generalizes these varied use cases into a set of scenarios in which we hope that you will find your own use of DDM. These scenarios are central to the guidance and recommendations that we provide throughout the rest of this book.
Section Two: Building your plan for a plan

- Chapter 5, “Building your Plan: Analyze and document your current environment” on page 61. You may have had DDM for a long time. Perhaps the original implementers are long gone, and you are not sure exactly who uses it for what purpose. This chapter will take you through the process of figuring out where you are now, which is essential before you figure out where you want to go next.

- Chapter 6, “Confirming your business goals and key business decisions” on page 79. Moving from DDM to another solution will require some change. Why not take this opportunity to think about how this change relates to some broader business goals. We provide suggestions and food for thought in this chapter. We guide you through some of the key decisions you need to make as you build your DDM transition plan.

Section Three: Making the Mental Leap

- Chapter 7, “Hierarchical and structural considerations between Domino Document Manager and Quickr” on page 89. For better or for worse, the way you think about the structure and security of your documents in Lotus Quickr is different from what you are used to with DDM. We spent many hours in conference rooms with a giant whiteboard, comparing, contrasting, and mapping the two solutions. In this chapter, we lay out the comparisons and help you begin your own mapping of how your DDM hierarchy and security can best be translated to a Lotus Quickr structure.

Section Four: Digging into customized DDM implementations

- Chapter 8, “Design element review” on page 123. This chapter builds upon the fundamental concepts introduced in the previous chapter. It serves as a detailed design element review and discusses many of the customizations developers have made to their DDM environments. It also addresses how similar functionality might be developed in Lotus Quickr Services for Domino (Quickr) where it does not already exist natively. The intended audience includes managers, business analysts, and developers looking to gain a better understanding of the feature differences between the two products from both a user and technical perspective.

- Chapter 9, “Analyzing customizations” on page 137. In this chapter, we help you analyze any customizations you have done to DDM and applications that you have integrated with DDM. We present alternatives and a process for you to evaluate the best course of action.

Section Five: Migration

- Chapter 10, “Overview of the migration tool Lotus Quickr Migrator: Domino Document Manager Edition” on page 177. This chapter provide an overview of the migration tool. We tell you what it can (and cannot) do, and make suggestions for the best ways to use it. We give you an overview of how you can customize the output from the tool to clean up, standardize, or normalize your DDM content before you load it into Lotus Quickr.

- Chapter 11, “Migration Tool: Operational scenarios” on page 187. This chapter introduces the process to relate your DDM paradigm for structuring and securing documents to the Lotus Quickr paradigm. In this chapter, we take you through an actual migration scenario, showing how to translate from your hierarchy and security model in DDM to a model that provides the best match for you in Lotus Quickr.

Section Six: A comprehensive example

Chapter 12, “A Domino Document Manager to Quickr example using a custom placetype” on page 215. This chapter walks you through a real-life DDM to Quickr migration. This chapter is not intended to be a complete step-by-step tutorial, but rather an overview of what is required to complete a migration from DDM to Lotus Quickr (services for Domino). It also illustrates how you could create a custom placetype in Quickr to duplicate some of the existing functionality in DDM.
Lotus Quickr primer

This chapter provides an introduction to Lotus Quickr.

Because Lotus Quickr will be the recommended migration destination for many customers migrating from Domino Document Manager, this chapter provides an overview of the functional capabilities of Quickr and helps build the foundation to understand the functional differences between Domino Document Manager and Lotus Quickr.
2.1 What is IBM Lotus Quickr?

IBM Lotus Quickr is team collaboration software that helps you share content, collaborate, and work more efficiently online with your coworkers. Quickr allows you to store documents in shared content libraries and to manage teams and projects from a central collaborative location.

A Quickr Place is the primary organizing construct for Lotus Quickr. A Quickr Place is a secure, virtual collaborative place that serves as the container for various capabilities. You can easily and instantly create Lotus Quickr Places. A place can be single-purpose, such as a library for a specific grouping of content or a wiki for frequently asked questions. Or a Quick place can be multi-purpose, such as a place to manage a project that has several document libraries, a team calendar, and tasks.

Lotus Quickr offers the following capabilities:

- Team Places
  Virtual project team places for working with colleagues.

- Content Libraries
  Collaborative document management functionality also known as Basic Content Services.

- Collaborative Components
  Capabilities to enhance content sharing and project work, such as a team calendar, task management, team blogs, and wikis.

- Rooms
  A sub area within a place that can contain folders, documents, discussion threads, or nested rooms.

- Connectors
  Plug-ins that allow you to interact with the content libraries from within popular desktop applications, such as Microsoft Office, Windows Explorer, Lotus Symphony™, Notes, and Sametime®.

- Templates
  Easy tools to design your own place types and use them as the basis for creating new Quickr Places.

- Integration with Enterprise Content Management (ECM) solutions
  The ability to use the Quickr user interface to access and contribute content to IBM Content Manager and IBM FileNet Content Manager.

- Personal File Sharing (future preview)
  The ability to contribute and share personal content outside the context of a team place or library.

2.1.1 Team Places

Creating an IBM Lotus Quickr Place is easy for anyone to do and does not require intervention from your busy IT department.

Follow the do-it-yourself guide from the Lotus Quickr home page to help you set up places. Or, you can use the ready-to-use place templates that come with Lotus Quickr to help you get started even faster.
When you set up a Lotus Quickr Place, you can customize it with out-of-the-box components, such as team calendars, discussion forums, blogs, wikis, and other collaboration tools for managing projects.

Three types of teamplaces are provided for you:

- Standard Team Place
- Blog
- Wiki

Quickr Places can be public (anyone can read), or they can be private with an explicit set of members allowed specific levels of authority. You can also specify access control at the room level and at the individual file level.

Figure 2-1   Example of a Quickr Place
2.1.2 Standard Team Place

The Standard Team Place is the most commonly used template for creating Quickr team places. By default, it includes the following capabilities, each implemented in a specific folder type with a user interface tailored to particular user tasks:

- Discussion
- Library
- Calendar
- Tasks
- Index

Figure 2-2 illustrates an example of the welcome page within a Standard Team Place.

![Figure 2-2 Example of a Standard Team Place](image)

The functional components are included within the Standard Team Place:

- Discussion
  
  A simple discussion forum lists the titles of all the items in the folder and allows members to create new content in response to existing content.

- Library
  
  The Library is a repository folder where you can upload a file or a set of files to share with other members in the place.
2.1.3 Content libraries

Content libraries provide basic content services (BCS) within a Quickr Place. Content libraries help you easily organize and share content for your projects, your teams, or yourself.

You can set up libraries for all your documents and rich media files so your teams can share and collaborate with check-in and check-out, version control, and other tools to help manage content.

Content libraries provide the following basic content services:

- A repository for any kind of file type
- Document-level security
- Check-in and Check-out capabilities allowing you to lock a document for editing
- Custom metadata support through document types
- Document versioning allowing you to create a snapshot of an existing document that cannot be edited but can be reverted to.
- Hierarchical structure of folders and subfolders
- Subscription feed (through ATOM) to allow users to be notified of new content in a Library
Figure 2-3 illustrates a basic Library containing various file types.
You can also access content in Lotus Quickr libraries right from your favorite applications using the Lotus Quickr connectors. For example, use the familiar Microsoft Windows desktop to drag files from your C-drive folders to personal or shared places. Or, save documents to Lotus Quickr folders directly from Microsoft Office or from IBM Lotus Symphony. Figure 2-4 illustrates the Quickr Connectors accessing a Quickr Team Place through the familiar Windows Explorer interface.

Figure 2-4  Quickr Connector through Windows
2.1.4 Blog

The Blog template displays information in the style of a journal or diary. Create posts and comments in a blog to share information with your team and to gain quick project solutions. A blog can bring visibility to your ideas and questions.

The Blog template provides for inline commenting, a calendar of blog entries, a blogroll listing other blogs that maybe of interest, blog search, history, and the ability to configure the blog as an RSS feed. You can also customize the blog and manage its members.

Configuring a blog as an RSS feed allows users to follow multiple blogs and be notified of updates to those blogs without having to visit each blog in turn.

![Figure 2-5 Example of a Quickr Blog Place](image-url)
2.1.5 Wiki

The Wiki feature in Quickr enables authors to create new content and solicit feedback, and track revision history. It is usable as is, or you can add wiki features to your existing places (for instance, in a room).

The Wiki template creates a place where content is developed and maintained by members. A wiki enables building a collaborative environment with a mix of formal and informal interactions. You can enter RTF (rich text formatted) content that can include links and images, see a document's revision history and comments on the changes, and configure the Wiki as an RSS feed source. You can also customize the wiki and manage its members.

Configuring a wiki as an RSS feed allows users to easily track multiple wikis and be notified of updates to those wikis without having to visit each wiki in turn.

Figure 2-6 illustrates an example of a wiki contained in a Quickr Place.

![Figure 2-6 Example of a wiki contained within Quickr](image)
2.1.6 Rooms

Rooms are sub areas within a place. You create a room when you want to limit access to certain items and folders, as each room has a members folder that defines who can enter the room and what they can do once inside.

Rooms you create appear as a room within the current place or room. For example, if you are in a room called Employees when you create a room called Biographies, Biographies becomes a room inside Employees. Visitors to Biographies must first enter Employees.

You can also create a room when you want to have a place to put information of a particular type (for example, employee information) where you do not refine the security.

2.2 Connectors

Lotus Quickr Connectors make it easy to use team collaboration along with other applications.

Using Lotus Quickr Connectors for popular business applications, you can access content and collaborate from where you are without switching applications.

With a click, move a Lotus Notes or Microsoft Outlook® attachment to a Lotus Quickr library and send a link instead of large e-mail files. Plus, your recipient is assured of getting the most recent version instead of the stale version attached to an old e-mail.

Lotus Sametime users can exchange links to Lotus Quickr content and collaborate from a chat. Microsoft Office and IBM Lotus Symphony users can check documents in and out of the Lotus Quickr content libraries and more.
2.2.1 Lotus Notes or Outlook e-mail

Using Lotus Notes, you can perform the following actions using the Quickr Connectors:

- Save e-mail attachments in a Lotus Quickr Place and send a link instead.
- Drag entire e-mails with attachments in a Lotus Quickr Place using the Lotus Notes side-shelf. See Figure 2-7.
- If you try to send an attachment, you get a dialog box asking if you want to send a link to your Lotus Quickr content library instead of the attachment.

Figure 2-7 Drag e-mails directly into a Quickr Place
**Outlook**

You can also work with Lotus Quickr documents directly from Microsoft Outlook. Messages and attachments that you receive in Outlook e-mail can be saved to a Lotus Quickr Place as an archive, or to streamline organization of all your work in a central location.

- Save e-mail attachments into a Lotus Quickr Place and send a link instead.
- Drag documents from Outlook into a Quickr Place.

Figure 2-8 illustrates an example of an attachment being saved directly into a Quickr Place.

![Figure 2-8 Using the Quickr Connectors with Microsoft Outlook](image)

### 2.2.2 Microsoft Office and Lotus Symphony

In your Microsoft Office applications, you can perform a number of actions including checking-in your files to a Lotus Quickr content library without leaving the application.

After installing the Lotus Quickr connector, menu options for place tasks are added to Microsoft applications.

Using Microsoft Office, you can:

- Create new documents using Microsoft applications that are stored in the Quickr Place from the Lotus Tools menu option.
- Save existing documents into a Quickr Place.
- Edit existing documents stored in a Quick place in Microsoft applications.
- Check out a document from a Microsoft application to prevent other place users from editing it while you are working.
- Create documents from custom document types in Quickr Places.
Figure 2-9 illustrates an example of a Microsoft Word Document being saved directly into a Quickr Place.

Lotus Symphony

The Lotus Quickr connector for Lotus Symphony will let you access Lotus Quickr content and collaboration services directly from any of the Symphony applications.

Create documents and upload them to Quickr Places for sharing, or use the content management services of Lotus Quickr (check-in and check-out, versioning, security, workflow) for collaborative authoring, without ever leaving Symphony. Collaboration is integrated in a natural way of working.

The following list highlights the integration points between Quickr and Lotus Symphony:

- Lotus Quickr toolbar has been added to Lotus Symphony.
- Create new documents using Symphony applications that are stored in the Quickr Place from the toolbar.
- Save existing documents into a Quickr Place.
- Edit existing documents stored in a Quick place in Symphony applications.
- Check out a document from a Symphony application to prevent other place users from editing it while you are working.
- Set default options for what to do when adding or opening documents from individual places.
- Create documents from custom document types in Quickr Places.
Figure 2-10 illustrates the ability to add the material which is being created in Symphony directly to a Quickr Place.

![Figure 2-10](image.png)

**Figure 2-10  Adding to a Quickr Place directly from within the Symphony application**

### 2.2.3 Windows Explorer

In Microsoft Windows Explorer, you can perform a number of actions, including checking-out a document from a Lotus Quickr content library without leaving Windows Explorer. More specifically, you can perform the following tasks:

- View and navigate Quickr Places using a hierarchical structure.
- Drag documents into a Quickr Place.
- Drag documents out of a Quickr Place onto a personal or shared drive.
- Perform document services such as check out, edit, print and so forth.
- View standard and custom metadata about the item in the Quickr Place.
Figure 2-11 illustrates using Windows Explorer to navigate to a file stored within a Quickr Place, as well as showing the menu of options for what you can do with the file.

2.2.4 Sametime

You can work with documents in places on an IBM Lotus Quickr server right from IBM Lotus Sametime. For example, you can read or edit documents, add documents, and chat about documents.

The Quickr Connector for Sametime will add a new pane to the Sametime window which will allow you to navigate the Quickr hierarchy.

The Connector will allow you to perform the following tasks:

- View and navigate Quickr Places using a hierarchical structure
- Chat with other users about specific documents directly from Sametime
- Link to documents in a place in a chat invitation
- Link to documents in a place during a chat
- Invite users to download the connectors directly from within Sametime
Figure 2-12 illustrates some of the integrated functionality between Lotus Sametime and Quickr, through the Quickr Connectors for Sametime.

![Image](image_url)

**Figure 2-12** Illustrating integrated functionality between Lotus Sametime and Quickr, via the Quickr Connectors for Sametime

### 2.3 Templates

IBM Lotus Quickr comes with pre-built templates that can provide immediate value and support for common business processes. Once you use a Lotus Quickr Template to create a new Lotus Quickr Place, you can customize the place to fit the needs of its particular team or project.

In addition to the templates that are included with Lotus Quickr, you can also create your own, or turn an existing place into a template to capture best practices and make a project's team place reusable.
IBM Business Partners also provide additional Lotus Quickr Templates, which are available for download from the IBM Solution Catalog.

Figure 2-13 illustrates three of the standard Place Type templates that come out of the box, (Standard Place for Teams, Blog, and Wiki), while the Virtual War Room template represents a custom place type that was added.

![Figure 2-13 Illustration of Place Type Templates to work from](image)

### 2.4 Personal file sharing

Files has been introduced with Lotus Connections 2.5 and provides a straightforward interface for file sharing. This will be available to Quickr customers once Lotus Connections 2.5 ships in Q3 of 2009. The service for both Connections and Quickr is delivered through a WebSphere Application Server based environment.

Files provides a Web-based collaboration tool in which individual users can upload files of any type and let other people see and work with them. Users will also be able to see and work with other users files. It will provide a convenient way to share files, information, communications, and ideas with other users without sending large files through e-mail.

Files is accessed through a Web-based interface. Users are able to upload files using an upload control and manage the sharing of their files. Figure 2-14 on page 26 illustrates the interface and listing of files.
Files can be private, shared, or public. Private means only the owner can see and work with it. Shared means the owner has shared it with specific people. Public means anyone can read the file.

Uploaded files start out private, but users can share a file or make it public any time they want, and other users can share files with them. When a file is shared with specific people the owner can give them Reader or Editor access to the file. Everyone can read a public file. Figure 2-15 on page 27 illustrates the options within a specific file.
Files provides Web 2.0 style capabilities for files. Files can be recommended and commented on and tagged to make them easier to find later on. A tag cloud or list for each user will be shown, as well as a download count for each file so the file owner and other users can see which files are most popular.
As well as being tagged, files can be grouped together in collections. Like files, collections can be private, shared, or public.

Files supports a simple versioning model. When files are edited the system will automatically create a new version of the file. Any user with access to view the file can see the previous versions but only editors can remove versions.

As you can see, Files has been designed to provide a way for users to store and manage their personal and public files, and share those files with other individual users or with everyone using Files.

In comparison with the library and document management functionality provided by Quickr, Files does not provide a structured hierarchy in which to store documents, only simple collections. It does not provide check-in and check-out functionality to support multi-person editing, the ability to apply workflow to documents, or provide access to the file store through the Quickr Connectors.

### 2.5 Lotus Quickr + Enterprise Content Management (ECM)

You can provide a complete, end-to-end solution by integrating Quickr collaboration tools with industry-leading ECM systems to address scenarios across the spectrum of collaboration & content.

**Note:** In addition to the material presented in this section on Lotus Quickr + ECM, refer to Chapter 3, “Is Enterprise Content Management in your future?” on page 33 to gain a more complete understanding of the Enterprise Content Management options available from IBM.

Quickr Connectors can help you unlock your enterprise content so that it is more accessible across your organization from the tools you use every day (such as IBM Lotus Notes, Microsoft Office, Microsoft Outlook, and more).

A combined solution can help make advanced ECM services (such as records management and business process management for example) easier to access and use.

Using Lotus Quickr and the IBM ECM tools lets you combine collaborative authoring and sharing of everyday business content with the structure, business process management rules, and classification and discovery models provided by ECM solutions. Figure 2-16 on page 29 illustrates the ability to publish with a link to the original document stored in an ECM System.
Figure 2-16 Example of publishing with a link to an ECM System

The Lotus Quickr Web interface and the Lotus Quickr Connectors have an easy-to-use, attractive user interface that acts as the front-end to both systems.

**Note:** At the time of writing, Quickr for Portal Services is the supported version of Quickr as a Web UI to ECM. Quickr for Domino Services is expected to provide this support in the first half of 2010.

### 2.5.1 Quickr + FileNet

The integration of IBM Lotus Quickr with IBM FileNet Content Manager provides the most complete end-to-end content management solution in the industry. You have multiple options for integrating FileNet Content Manager and Lotus Quickr.

- FileNet Services for Lotus Quickr is an adaptable Web service component that integrates Lotus Quickr collaboration tools and IBM FileNet Content Manager. For customers who have both Lotus Quickr and FileNet Content Manager licenses, FileNet Services for Lotus Quickr is included at no additional charge.
- FileNet Content Manager Collaboration Edition is an integrated bundle including Lotus Quickr and a restricted version of IBM FileNet Content Manager.
FileNet Services for Lotus Quickr

Using this integration your knowledge workers can use the Web interface and familiar user interfaces for collaboration while taking advantage of the advanced content, compliance, and business process management functions available with the IBM FileNet P8 platform.

The integration between Lotus Quickr and FileNet Content Manager provides the following advantages:

- Enables the discovery of enterprise content and provides access to business documents across the organization. Documents in FileNet can be accessed through the Web interface, e-mail, Microsoft Office, and so forth.
- Manages your organization's content as a secure corporate asset in an enterprise content management (ECM) system, allowing content to participate in Business Process Management, and to be managed in accordance with content discovery, auto classification, and compliance/retention policies.

FileNet Services for Lotus Quickr allows users to perform the following tasks:

- Transfer collaborative content from Lotus Quickr to an ECM repository.
- Incorporate existing ECM content inside a Lotus Quickr team place.
- Search Lotus Quickr and ECM content from a Lotus Quickr user interface.
- Use the Lotus Quickr connectors to interact with content in both Lotus Quickr and IBM FileNet Content Manager.
- Extend advanced ECM services and function to broader content and collaboration deployments.

FileNet Content Manager Collaboration Edition

FileNet Content Manager Collaboration Edition enables users to collaborate on business content with Lotus Quickr while using FileNet Content Manager for its scalability, high-availability, and multi-platform support. It is intended for organizations that want to use the capabilities of ECM through the Lotus Quickr user interfaces.

FileNet Content Manager Collaboration Edition provides users with seamless access to enterprise content across the organization, using their everyday desktop applications such as e-mail, Windows Explorer, desktop productivity tools like Lotus Symphony and Microsoft Office, as well as from the intuitive, easy-to-use browser interface of Lotus Quickr.

2.5.2 Lotus Quickr + Content Manager

New and existing customers have multiple options for integrating IBM Content Manager and Lotus Quickr.

- Content Manager Services for Lotus Quickr is an adaptable Web service component that integrates Lotus Quickr collaboration tools and IBM Content Manager.
- Content Manager Collaboration
Content Manager Services for Lotus Quickr
Content Manager Services for Lotus Quickr helps today's knowledge workers use intuitive Web 2.0 and familiar user interfaces for collaboration while taking advantage of the advanced content, compliance, and business process management functions available with IBM Content Manager. The integration between Lotus Quickr and Content Manager provides the following advantages:

- Enables the discovery of enterprise content and provides seamless access to business documents across the organization. It uses collaboration tools and everyday business applications such as e-mail, Microsoft Office, and so forth.
- Manages your organization's content as a secure corporate asset in an ECM system, allowing content to participate in Business Process Management, and to be managed in accordance with content discovery, auto classification, and compliance and retention policies.

The Services allow users to perform the following tasks:
- Transfer collaborative content from Lotus Quickr to an ECM repository
- Incorporate existing ECM content inside a Lotus Quickr team place
- Search Lotus Quickr and ECM content from a Lotus Quickr user interface
- Use the Lotus Quickr connectors to interact with content in both Lotus Quickr and IBM Content Manager
- Extend advanced ECM services and function to broader content and collaboration deployments

The integration of IBM Lotus Quickr with IBM Content Manager provides the most complete end-to-end content management solution in the industry. For customers who have a license for both Lotus Quickr and IBM Content Manager, IBM Content Manager Services for Lotus Quickr is included at no additional charge.

Content Manager Collaboration Edition
IBM Content Manager Collaboration Edition enables users to collaborate on business content with Lotus Quickr while using IBM Content Manager for its scalability, high-availability, and multi-platform support. It is intended for organizations that want to use the capabilities of ECM through the Lotus Quickr user interfaces.

Content Manager Collaboration Edition provides users with seamless access to enterprise content across the organization, allowing access to their everyday desktop applications such as e-mail, Windows Explorer and desktop productivity tools like Lotus Symphony and Microsoft Office. Content Manager Collaboration Edition also provides access to enterprise content through the easy-to-use browser interface of Lotus Quickr.

2.6 Conclusion

Now that we have provided an introduction to the capabilities in Quickr, review Chapter 7, “Hierarchical and structural considerations between Domino Document Manager and Quickr” on page 89 to understand how Quickr relates to Domino Document Manager, both from a hierarchical and functional perspective. Additionally, review Chapter 8, “Design element review” on page 123 to review specific design element customizations in Domino Document Manager and better understand how you can create some of these customizations in Quickr.
Is Enterprise Content Management in your future?

Because you are considering the proper approach and strategy for migrating your existing Domino Document Management (DomDoc) content repositories, it prompts you reevaluate your content management needs. Fortunately, this IBM Redbooks Publication provides guidance and specific steps on how to best evaluate your existing DomDoc environment, both from a technical perspective (see Chapter 5, “Building your Plan: Analyze and document your current environment” on page 61), and from an underlying business perspective (see Chapter 6, “Confirming your business goals and key business decisions” on page 79).

One aspect is probably clear to you: your organization must manage a rapidly increasing quantity of content. Not only is the sheer quantity of content exploding, but the content types are now much more complex and less structured. You possibly already have a complex structure within your DomDoc implementation and are looking for ways to handle more complex content management needs going forward. While the primary focus of this IBM Redbooks publication is how to migrate your existing DomDoc content repositories over to Quickr, this chapter discusses whether you should consider stepping up to an Enterprise Content Management (ECM) solution. In this chapter, we will explain what ECM is and what characteristics might signal ECM in your future.

IBM can help organizations replace content chaos with an easy-to-use solution that sets the standard for world-class, end-to-end ECM. IBM Lotus Quickr collaboration software includes IBM ECM Services for Lotus Quickr, further helping businesses of all sizes manage, share, and collaborate on important information. IBM Lotus Quickr provides a collaborative environment and intuitive interface, while integrating with documents, spreadsheets, and other files that reside in IBM ECM systems. This new integration between IBM Lotus Quickr and the IBM ECM offerings will allow Lotus Quickr and IBM ECM users (both IBM FileNet P8 and IBM Content Manager 8 users) to collaborate on business content through Quickr while using the IBM comprehensive ECM infrastructure.

This technology advancement will allow employees to access enterprise content and programs so it can be easily accessed across the organization. This content includes everyday business applications like e-mail, presentations, spreadsheets, complicated records management, and business process management programs.
3.1 The growing need for managing content

By now, the notion of ECM is hardly a new concept for most organizations. Many companies have already begun addressing the multitude of formats (such as images, text documents, Web pages, spreadsheets, presentations, graphics, drawings, e-mail, video, and multimedia). The volume of structured and unstructured content can be complex and difficult to manage.

**Note:** The goal of ECM is to bring all of an organization’s unstructured content into a managed environment for sharing, controlled access, findability, and archive. In some organizations, ECM may indeed be a single system capable of dealing appropriately with many different types of content and records requirements. In others, it may be a collection of repositories and applications.

The common goal, however, is to provide users with a single-access capability allowing them to find, retrieve, and process information from wherever it is stored, without needing to log in to multiple applications.a

---

An Accenture study from 2007 found that more content will be created in the next two years than in the entire previous history of mankind. And more than 93 percent of it will be electronic.1 Additionally, the following statistics illustrate the growing quantity of collaborative content:2

> **Internet Usage**
> - 1996: 48 million users
> - 2006: 1.1 billion users
> - 2010: 1.6 billion users (estimated)

> **e-Mail Mailboxes**
> - 1998: 253 million (1 trillion e-mails)
> - 2006: 1.6 billion (22 trillion e-mails)
> - 2010: 2.0 billion (estimated)

> **Corporate Instant Messages**
> - 2002: 41 million users
> - 2010: 250 million users (estimated)

This explosion of content represents your company’s collected intelligence. It represents what you do, and how you do it. In many cases, it also represents what you need to avoid doing. Additional statistics indicate the following facts:

> **E-mail is still out of control, with 55% of organizations having little or no confidence that important e-mails are recorded, complete, and retrievable.**

> **28% of organizations would take more than a month to produce documents for a legal discovery process.**

---

1 IBM Whitepaper on FileNet P8
2 Source, IDC, 2007, IBM
In addition to the growth of content, several unavoidable drivers continue to influence the ECM market.

- Cost Savings and efficiency gains through more efficient document and records management. Since the beginning of 2008, cost savings has taken a position as a lead driver for ECM.
- Legislative requirements such as the Sarbanes-Oxley Act, has moved ECM from being a nice-to-have system to a must-have system. Enforceable document retention policies, systems for managing unstructured information, and consistent records management are all key requirements from a legislative standpoint. Yet while legislative requirements and audit concerns may be key motivators for many companies, several other factors are also influential.
- There is the obvious need to exert control over an abundant volume of records and documents, with the resulting benefit of improving content and document handling.
- There is the need to automate business processes to reduce task duplication and replace existing manual paper processing. Addressing these needs is made even more difficult because most existing content resides in disparate content silos, existing systems, and a widely varying mix of heterogeneous IT platforms and standards. Most large companies have 10 or even hundreds of line-of-business applications that deliver mission-critical content and processes to their business units. While silo applications may have sufficed in the past, today's users need access to necessary information but are unwilling or unable to jump between disconnected systems to find the information.

Figure 3-1 illustrates the concept that as the volume and complexity of content increases, there is also an increasing need for greater collaboration between team members in using and effectively using this content. This illustrates the range of options from Personal File Sharing and growing toward a larger scale ECM solution.
3.1.1 Does your organization need ECM?

If unstructured information, like e-mail, spreadsheets, presentations, forms, and any of the myriad types of content that are produced or received by most enterprises today are critical to your business, ECM might be a worthwhile investment. The following subsections discuss a few reasons.

Security
Content security tightly controlling access to the content, with highly granular rights for different kinds of access granted to users and groups of users. These rights, which determine who can read, edit, annotate, delete, and perform many other actions on content of different classes (at different stages in its life cycle) are more sophisticated than the relatively simple controls that are sufficient for structured data in relational database systems. As with RDBMS security, our many ECM customers with sensitive requirements (like the military) have helped us develop security models that meet the most stringent requirements.

Scalability
The huge amount of unstructured content, both in terms of number of files and the size of the average file, create unique information management challenges for many enterprises. The need to get the right information into the hands of employees, customers and partners quickly (when the right information is one file among many terabytes) requires an infrastructure designed to deal with these challenges. Content repositories can grow more quickly than structured databases. All content-enabled applications, from collaboration suites like Lotus Quickr to case management solutions, can benefit from the support of a highly scalable ECM system.

Storage Cost Reduction
Very large volumes of content entail expensive hardware investments. An ECM infrastructure helps control these costs in several ways. One is deduplication. Recognizing that a PowerPoint® presentation may be copied hundreds or even thousands of times after it is sent as an e-mail attachment, ECM systems can replace the attachment with a pointer to a single copy stored in the repository. An added advantage of this technique is that the pointer can be configured so that changes to the presentation are automatically available to all recipients of the original. An ECM infrastructure thus improves the quality and currency of information within the organization and reduces information management costs at the same time.

Backup and Recovery/High Availability
Structured information management services, such as centralized backup, recovery, failover, and other HA support capabilities are equally important to unstructured information solutions. In fact, there are often special challenges associated with content management because of the larger volumes of bytes involved. ECM systems provide many services that help address these requirements.

Enterprise Search
Finding the right information is always a challenge. ECM systems help with this challenge in a number of ways. First, they provide sophisticated data modeling facilities to capture schemas and taxonomies that index content for convenient discovery. They also provide tools that suggest classifications, based on the content within your organization. Finally, sophisticated search solutions analyze content and help guide the searcher to the relevant documents to assist with business decisions.
Compliant Archival and Discovery
Unstructured information that is deemed as critical business records may need to be archived. This is above and beyond backups made for purposes of assured availability. Archiving is the mechanism that an enterprise uses to comply with various regulatory mandates from governments and internal auditors. In certain industries, these compliance mandates may be quite complex. Archival management also requires implementation of retention policies, to insure that content that may need to be discovered at a later date is available and that content that has expired according to established policy is expunged so that it is no longer discoverable.

Responding to legal requests for records may be a tremendously expensive undertaking as well. Along with the infrastructure for archival and records management, ECM systems provide sophisticated discovery tools that enable internal legal staff to assess exposures and provide for complete response to legal discovery.

Business Process Management and Optimization
The most valuable content to an organization is active content. That is, content that enables business processes. This can be a sales proposal that begins as a template from the central repository, is collaborated on by an account team, and is ultimately approved according to established procedures. Or it can be an electronic form that is begun by a customer and elaborated upon as it passes through various workflow steps.

Some of the largest returns on investment in an ECM system come from the automation of processes that are not as efficient as they can be. Content-enabled workflow and BPM are often the drivers behind an ECM decision.

Once processes are automated, the BPM subsystem of an ECM infrastructure offers tools that allow business analysts to improve efficiency at the process model level, and optimization techniques that assure human and machine systems are optimally resourced to meet the scale needs of the organization.

If your organization has requirements in any of these areas, the odds are that an ECM backend to the collaboration services offered by Lotus Quickr will add tremendous value to your information management solution.

3.2 The human aspect

Many organizations have learned that human factors can be among the greatest challenge in deploying a comprehensive ECM strategy. Even if users are on board with implementing an ECM solution in principle, they will not embrace a system unless it integrates with their current work style. Solutions must be intuitive to use, and cost-of-ownership requirements dictate that users cannot rely heavily on the IT department. The solution must also fit the organizational culture. For instance, an organization that operates with strong individual teams is not likely to embrace a solution that rolls out enterprise-wide at the teams’ expense. The solution must support a variety of user interaction methods, as well as integrating into existing line-of-business applications. Line-of-business units do not have the luxury to start from scratch or disregard the years of existing content and archived information that is critical to their business function. An organization may have one or more content management products already in place to address specific departmental needs. As the organization moves toward having a company-wide ECM strategy, it becomes important for any new solution to interoperate with existing products using industry standards.
An ECM solution that is scalable, easy to use, and integrated with other applications will be successful. Because user adoption is key to the system’s success, ensuring that users recognize ECM’s usefulness in streamlining their own workloads is important.

As we will discuss in upcoming sections, IBM Lotus Quickr provides not only a collaborative environment for sharing of ideas and team-based knowledge, but it provides an easy to use, intuitive interface which can serve as a seamless frontend for a complete end-to-end ECM solution.

3.3 Principles of end-to-end ECM

What is end-to-end ECM? From a user perspective, it is the ability for information workers to conduct their activities faster, with better context and collaboration, without having to think about where the content lives, what process steps are required, or what compliance needs exist. The four major factors to successfully implement end-to-end ECM are examined in greater detail here.

3.3.1 Maximize user acceptance

Perhaps the most critical factor in ECM success is how well a system is embraced by its users. Even if every other ECM goal is met, any system that fails in this regard will have limited success. On the flipside, users will extol a system that they perceive as streamlining their workday, improving their productivity, and making their jobs easier.

Systems that require minimal training and enable workers to proceed on their own will be embraced. Familiar and intuitive interfaces will also help users accept the system as something that is merely an extension of their current workload rather than an entirely new procedure.

Gaining user acceptance, which is inherently necessary to reach the goal of ECM for an enterprise, requires that users feel comfortable with the interface to the system and methods of interaction. Ideally, users should not have to learn new solutions or be asked to jump between environments to complete their tasks.

Solutions that help users find documents more quickly, expedite approval processes and minimize task duplication (regardless of what system is actually managing the work) will be embraced and championed.

3.3.2 Minimize the Burden on IT

Successful ECM implementations minimize IT input and empower users to integrate the solution into their own work schedules.

If the ECM solution requires extensive IT involvement and professional developers or consultants to set up new sites and workflow processes, many business needs will be left unresolved even though the organization has invested in the technology to meet these needs. Providing a solution with a manageable learning curve and minimal IT reliance is therefore key.

Besides empowering users to be self-reliant and minimizing IT support calls, a solution that is well-integrated allows IT to manage a larger system in a centralized manner. Centralized deployment enables the IT department to manage the system consistently across all areas using common tools and deployment models. This is an equally important timesaver for IT.
3.3.3 Meet diverse organizational needs

At a macro level, meeting diverse needs means selecting a solution that is highly configurable and fits an organization’s culture.

If the organization works most effectively in teams, a departmental or team-based deployment may be an effective starting point, with a larger overall deployment to follow. In this type of culture, the teams will be more likely to embrace an organization’s FileNet P8 & Microsoft SharePoint: End-to-End ECM for Improved Control Page wide rollout if they first witness its success within their department. On the other hand, if the organization is centralized and accustomed to systems, an enterprise-wide rollout may be most effective from the start. Choosing a system that is extensible enough to be deployed in either manner will provide the greatest flexibility.

Another reality of an organization’s IT environment is the multitude of applications that are currently in place to support different business processes. As discussed earlier, most enterprises have a broad range of existing content and process solutions that, in many cases, store millions of pages of content and records. An ECM solution must integrate easily with a company’s existing applications using industry standards. This way an organization can continue to reap the benefits of their existing applications that may serve specific functional areas as they deploy an enterprise-wide ECM strategy.

3.3.4 Content Federation Across the Enterprise

By definition, an ECM deployment manages content across an entire enterprise. Often, that content resides in disparate systems, yet it must still interact with an ECM solution to remain vital. Combined with increasing compliance demands to better control corporate content, organizations must find a way to access and manage information stored across the enterprise.

Companies often need to connect multiple point solutions for content management to gain a more comprehensive picture of enterprise content. Delivering end-to-end ECM requires the ability to federate, or unify, content for easier access and management. Therefore, content federation has emerged as a critical requirement for companies wanting to use their entire information assets. Federating content allows for content to be available when and where it is needed, regardless of the underlying source or system serving the content.

Existing silos of content, archiving policies, and mission-critical, line-of-business applications need to be incorporated into enterprise-wide initiatives. End-to-end ECM integration gives users a single point of access to this content, and the attendant business processes, regardless of what desktop environment is used.
3.4 Lotus Quickr and ECM

When your organization uses a combination of the self-service, team collaboration capabilities in IBM Lotus Quickr and the industry leading infrastructure of IBM ECM, you can work smarter, reduce costs, and make better decisions. IBM offers the most complete single-vendor collaboration and content management offering.

- Increase efficiency and productivity for information workers with an intuitive user interface and integration into popular desktop applications
- Use your investment in a content infrastructure by making it more accessible to all your employees
- Ensure your collaborative and ad hoc content is managed appropriately to help meet legal and industry compliance requirements
- Extend business process and records management capabilities to key knowledge workers, wherever they are in your organization
- Make content part of your broader business and compliance processes with easy-to-use collaborative tools
- Combine the flexibility of collaborative content with a highly-scalable and reliable enterprise class repositories
- Strike the ideal balance between centrally-provisioned and managed ECM systems, IT governance, and a self-service and empowered user base.

ECM services for IBM Lotus Quickr software unites collaboration and content management across the enterprise. The solution uses IBM FileNet Content Manager or IBM Content Manager software along with IBM Lotus Quickr software to support intuitive collaboration for users and improved management of content.

3.4.1 Bridging the gap between basic content services and ECM

Lotus Quickr makes it easy for people, both inside and outside the organization, to author ideas and work together on cross-organization projects. Ad hoc content creation fuels innovation, while corporate information proliferates. However, collaboration software is not designed to categorize and repurpose large amounts of information.

To manage this growing wealth of content, many organizations are turning to ECM solutions, like IBM Content Manager and IBM FileNet Content Manager software. These solutions deliver critical capabilities for managing content and support consistency in classifying and categorizing information.

ECM solutions help maintain document revision control and enable users to find relevant information across multiple repositories. And they enforce established compliance and records management rules.

What has been missing is a way to bridge the gap between collaboration and basic content services that meet ad hoc needs and more specialized ECM solutions that adhere to corporate management controls for content. Recognizing the increasing role of collaboration and the proliferation of content creators, IBM developed its business content services (BCS) strategy. BCS is an ECM shared- services approach that helps you take control of your content—from creation and collaboration to reuse and storage. It provides a comprehensive solution to manage content by integrating collaboration and basic content services with industry-leading ECM systems. BCS encompasses a centrally provisioned content infrastructure that addresses the entire spectrum of business content—spanning from

40 Self Assessment Guide for Domino Document Manager Migration
personal content creation and sharing, to team content collaboration and workgroup content management, to content-enabled applications. And it delivers virtually unlimited scalability and multiplatform support because it is based on open standards.

ECM services for Lotus Quickr (either IBM FileNet Services for Lotus Quickr or IBM Content Manager Services for Lotus Quickr) is an adaptable Web service component that provides integration between IBM Lotus Quickr and ECM systems from IBM. Using this integration, today’s knowledge workers use intuitive Web 2.0 technology and familiar desktop user interfaces for collaboration, while taking advantage of advanced content, compliance, and business process management functions available with ECM software from IBM.

3.4.2 Uniting collaboration with content management

ECM services for Lotus Quickr software, together with Lotus Quickr and ECM software from IBM, deliver an integrated, comprehensive ECM and collaboration solution.

Gain seamless access to business documents
Unlock enterprise content and provide users with seamless access to business documents across the enterprise, using Lotus Quickr collaboration tools and everyday business applications. Knowledge workers can now create, share, and collaborate on their business documents using Lotus Quickr connectors embedded into familiar desktop applications (including Microsoft Office, Microsoft Outlook, Microsoft Windows Explorer, IBM Lotus Notes, IBM Lotus Sametime and IBM Lotus Symphony software). Using the Quickr connectors, users perform basic content operations as a natural extension of these applications, including check-in and check out, round-trip editing, versioning, sending and copying links, view-edit properties, approval, publish-draft and create-from-document-type functions.

Manage content to achieve compliance
ECM services for Lotus Quickr software provides a security-rich, scalable, and central content store for compliance, accountability, and reporting purposes.

It effectively manages content to address regulatory requirements, corporate governance mandates, and legal discovery needs, as well as integrates with your business processes.

Organize content across the enterprise
ECM services for Lotus Quickr software provides the content repository and infrastructure necessary for normalizing and dynamically classifying content across the organization. This helps your organization achieve agility and innovation through the use of collaborative tools, without sacrificing the business controls provided by a powerful ECM system.

Extend ECM control over content
ECM services for Lotus Quickr software extends advanced ECM services, including business process management and records management, to broader content and collaboration deployments. This helps organizations connect the right people with the right information at the right time.
3.4.3 Capabilities of ECM services for Lotus Quickr software

ECM services for Lotus Quickr software integrates the robust content management of IBM ECM solutions with the robust collaboration features of Lotus Quickr software and allows users to perform the following tasks:

- Transfer collaborative content from Lotus Quickr to an ECM repository.
- Incorporate existing ECM content inside a Lotus Quickr team place.
- Search Lotus Quickr and ECM content from a Lotus Quickr user interface.
- Use the Lotus Quickr connectors to interact with content in Lotus Quickr, Content Manager and FileNet Content Manager software.
- Extend advanced ECM services and function to broader content and collaboration deployments.

3.4.4 Getting started with ECM services

For clients who have a license Lotus Quickr and ECM software from IBM (either Content Manager or FileNet Content Manager), services for Lotus Quickr are at no additional charge. Clients have a license for IBM ECM software but not for Lotus Quickr can purchase Lotus Quickr to use the integration of collaboration and content management capabilities offered by ECM services for Lotus Quickr.

For customers who are new to ECM, IBM Content Manager Collaboration Edition software is an easy way to get started with the IBM collaboration and ECM solution. Content Manager Collaboration Edition software is an integrated bundle that includes both Lotus Quickr and a restricted version of Content Manager specifically intended for organizations that want to use the capabilities of ECM through the Lotus Quickr user interfaces.

Additional information about ECM services for Lotus Quickr software: can be found at the following Web page:


3.5 ECM Offerings from IBM

In this section, we provide a brief overview of the two key ECM offerings for the audience of this IBM Redbooks publication:

- IBM Content Manager (formerly named IBM DB2® Content Manager)
- IBM FileNet Content Manager P8

IBM FileNet Content Manager Collaboration Edition and IBM Content Manager Collaboration edition both extend ECM services and function to a broader audience. IBM FileNet Content Manager Collaboration Edition and IBM Content Manager Collaboration Edition use new Web service components to provide integration between best-of-breed collaboration tools of IBM Lotus Quickr and IBM ECM repositories.
3.5.1 IBM Content Manager (formerly DB2 Content Manager)

IBM Content Manager (formerly DB2 Content Manager) has been an IBM content management solution for more than a decade. This product and its predecessors have been leaders in the marketplace for many years. You may hear this referred to as CMV8 or CM Version 8. CM customers tend to be large organizations, although the solution can scale down for SMB applications as well. As with all of the ECM offerings, there is a database component and a file storage component. CMV8 includes the licenses needed for a DB2 implementation, and offers an Oracle® option for customers wishing to use their existing Oracle infrastructure.

Content Manager manages all types of digitized content across multiple platforms, databases, and applications. It provides imaging, digital asset management, Web content management, and content integration. Built on a multi-tier distributed architecture, it provides the scalability to grow from a single department to a geographically dispersed enterprise. The products and solutions help large and small organizations meet regulatory compliance requirements, collaborate with greater productivity, and improve customer service.

Solutions built with IBM Content Manager deliver documents and information to users in context, streamline work processing, and improve productivity with automated processes.

Content Manager serves the following functions:

- Acts as the core repository of a portfolio of products that helps manage, share, integrate and deliver critical business information about demand, for multiple platforms, databases and applications.
- Provides a comprehensive, scalable, and secure content management platform for solutions such as production imaging, compliance, records management, document management, Lotus Notes and Microsoft Exchange e-mail management, digital media, Web content management, and case management.
- Manages the life cycle of all content, including images, electronic documents, XML, and streaming audio and video, so that users can work with all types of information anytime, anywhere.
- Offers a powerful and expressive XML-ready data model, Java™, C++ and Web services programming interfaces, and an integrated hierarchical storage manager that supports hundreds of storage devices and media types.
- Scales horizontally and vertically, from Microsoft Windows through mainframes, with enterprise features such as replication to store and manage objects in multiple locations and application-transparent content caching.
- V8.4.1 for Agile ECM provides integration with FileNet P8.
- Operating systems supported: AIX®, HP Unix, Linux®, Sun™ Solaris™, Windows.

3.5.2 IBM FileNet Content Manager P8

In 2007, IBM acquired FileNet, another leading provider of content solutions. In the past several years, FileNet has been extremely successful in its business process management (or BPM) business. This is where content can be used against existing business systems and processes. Like CM, FileNet supports DB2 and Oracle, but also offers a SQL Server® implementation as well.

IBM FileNet Content Manager is the core content management solution for the FileNet P8 platform. It combines powerful document management with ready-to-use workflow and process capabilities to automate and drive your content-related tasks and activities.
FileNet Content Manager offers the following features:

- Provides the content management, security management, and storage management engine for FileNet P8. V4.5 provides new functions for Agile ECM and Business Content Services
- Combines universal content management and advanced document management capabilities with industry-leading Active Content process capabilities to drive unstructured information for better and faster use in your organization
- Delivers active management of enterprise content across the enterprise regardless of what repository it resides in
- Maintains secure control over metadata, process and compliance activities while managing highly customized, transactional content delivered to the right place at the right time
- Streamlines document management tasks by providing extensive versioning and parent-child capabilities, approval workflows, and integrated publishing support
- Provides you with the tools to deploy turnkey solutions that handle complex document types using out-of-the-box workflow and process capabilities
- FileNet Content Manager supports zLinux and z/OS® and DR550
- IBM System Dashboard for Enterprise Content Management (formerly called P8 System Manager) is supported

For further information about FileNet Content Manager, refer to the following Web page:
http://www-01.ibm.com/software/data/content-management/filenet-content-manager/
Scenarios: Which might apply to you?

Before we can determine what the best migration path is for your use of Domino Document Manager (DDM), we need to determine what user scenario best fits your organization’s use of DDM. For the purpose of this IBM Redbooks publication, we have created three typical user scenarios and five typical migration paths. To help you decide which of the user scenarios might apply for your use of DDM, we have created a Priority Matrix. This chapter will discuss how to use the Priority Matrix to measure your DDM implementation and recommendations for each of the migration paths.
4.1 Determining which scenario best fits your organization

To make the most accurate determination of which customer scenario applies to your organization, you need to understand DDM fundamentals and how the DDM hierarchy is structured. You can learn more about these in 7.1.1, “Basic hierarchy concepts of DDM” on page 90. If you are new to DDM, you will want to get a better understanding of the product before you attempt to determine the complexity level of your implementation.

To help you determine what scenario best matches your use of DDM, this IBM Redbooks publication provides you with the following tools:

- Priority Matrix found later in this chapter with a full page version at the end of the IBM Redbooks publication (See Appendix C, “Scenario priority matrix” on page 263.).
- A custom tool called DXL Magic, which will help you compare your DDM implementation to the default installation to determine the level of customization you have in your existing DDM implementation. You can learn more about DXL Magic in Chapter 9, “Analyzing customizations” on page 137.
- Chapter 8, “Design element review” on page 123 will provide some steps on how to review and understand more Binder and Document status, much of which will help you answer questions asked in this chapter.
- Review Appendix A, “Domino Document Manager Site Map and Security Analysis Tool” on page 253 to understand another tool that can help report on your DDM environment site structure and security levels set within the hierarchy.

Remember when making this evaluation of DDM that your implementation is probably not a single company implementation. You may have multiple Libraries for different uses within your company. One used as a basic file store and another for a complex business process. If you have multiple Libraries, you should review each as an individual implementation and go through the scenario process for each one. In many cases, each implementation will have different needs. Different migration strategies and considerations for each are possible.

4.2 Determining your DDM complexity level

To help you determine what scenario you will most likely identify with the most, we have created a priority matrix.

The priority matrix identifies focus areas of DDM on the vertical axis and measures the level of relevance and importance of each focus area on the horizontal. The four items of focus are:

- Customization
- Hierarchy
- Security
- Workflow

The goal in using this priority matrix is to evaluate how the appropriate levels and quadrants apply to your organizations’ DDM implementation. In 4.3, “Customer Scenarios” on page 52, we discuss special considerations based on the combinations of factors.
4.2.1 Customization

Customization is defined as the amount of change you have made to DDM on top of the base product. These customizations include any of the following elements:

- Document types with custom metadata fields
- Binder types with custom metadata fields
- Changes to the default Binder views, including new columns or action buttons
- New views added to the Binder
- Use of the Library or Binder events in the custom Script Libraries
- Changes to the Notes or Web user interface that include adding your own logo or company color scheme
- Use of the DDM APIs outside of the product, such as use within a Visual Basic® or C++ application
- Custom ODMA Dialog User Interface
For purposes of assessing the relevance of your customizations, the type (for example sophistication) of your customizations is more important than how many different customizations you have done. Find your customizations in Table 4-1. Your level of relevance is the highest “X” that you have.

<table>
<thead>
<tr>
<th>Customization</th>
<th>Minimal</th>
<th>Average</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document types with custom meta data</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binder types with custom meta data</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changes to default binder views</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>New views added to binder</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Use of library or binder events in custom script library</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Changes to Notes / Web UI (e.g. for branding)</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Use of the DDM APIs outside the product</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Custom ODMA Dialog User Interface</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

4.2.2 Hierarchy

Hierarchy is defined as the number of levels of categorization that you use to organize the documents in DDM. By default, DDM requires the following levels of hierarchy:

- Library
- File Cabinet
- Binder
- Document

A default File Room of “Default” is used if nothing is selected. A default value of “Not Categorized” is used at the Binder Category level if no Binder Category is selected. If you use every level of hierarchy built into DDM, it may look like the complete hierarchy represented in Figure 4-2 on page 49.
Chapter 4. Scenarios: Which might apply to you?

To determine the level of relevance of your hierarchy in your implementation, look at two factors. First, how much of your existing DDM hierarchy are you using? Second, how important is the ability to set specific security levels within different levels of the hierarchy? To determine the importance of the hierarchy in your implementation, ask yourself the following questions:

- Do you use every level of the hierarchy?
- Do you have anything besides “Default” as your File Room?
- Do you have anything besides “Not Categorized” as your Binder Category?
- If you had the opportunity, would you reorganize your document hierarchy to have less levels?

The images below illustrate a breakdown of the hierarchy within DDM to help you further analyze the structure at each relevant level. As you review these, continue to review the questions above, validating the extent to which you currently use the different levels of the hierarchy.
Figure 4-3 illustrates the highest level of hierarchy at the Library level. Security can be set at the Library level.

![Diagram: Hierarchy at the Library level]

Figure 4-4 illustrates the hierarchy at the File Cabinet Level, with security set at the File Cabinet level.

![Diagram: Hierarchy at the File Cabinet level]

Figure 4-5 illustrates the hierarchy at the Binder level. Security can be set at both the Binder level and the document level.

![Diagram: Hierarchy at the Binder level]

Once you review your existing DDM hierarchy, you need to decide whether you want to maintain the exact levels after you migrate or whether you want to simplify to a less complex hierarchy and use tagging to locate files. If you want to maintain the exact hierarchy you have in DDM, you are in the AVERAGE level of relevance square. If you want to move to a flat hierarchy, like Lotus Quickr supports with just a Place and Library, then you would be in the MINIMAL level of relevance square. If you have customized the hierarchy to increase the number of levels that DDM supports, including adding more than one binder category, you will be in the HIGH level of relevance square.
The hierarchy of Lotus Quickr is more attuned to the modern day, Web 2.0 thought process of “everything is flat and we let the user tag documents” The tagging process is done by the user completely. They generate the tags, enter them, and use them. The user can locate documents through a Tag Cloud or through searching. If you prefer a less complex, flatter hierarchy, consider moving to this methodology of classifying documents.

4.2.3 Security

DDM provides multiple options for managing document security, You can have custom security settings at the following levels:

- Library through Domino.Doc Library Users Groups
- File Cabinet
- Binder
- Document

For many DDM implementations, security is set at the Library level and inherited down to the document without any changes. If your implementation looks like this, then you would be in the MINIMAL level of relevance square.

If you have custom security settings at the File Cabinet or Binder level, but everything else inherits from them, you would be in the AVERAGE level of relevance square. Examples of this are as follows:

- Library with DDM Users groups security settings
- File Cabinets with custom security
- Binders inheriting from the File Cabinet
- Documents inheriting from the Binder
- Library with DDM Users groups security settings
- File Cabinets with security inheriting from the Library
- Binders with custom security
- Documents inheriting from the Binder

If you have security that is custom at every level of the DDM or have modified DDM to have different security settings, you would be in the HIGH level of relevance square.

4.2.4 Workflow

DDM has two workflow processes that are built into the product. There is a review workflow cycle and an approval workflow cycle that puts a document through process. Each document can go through both review and approval processes when a document is in a draft status. What makes DDM different than many of the document management systems in the same class is that the review and approval cycle are done on a working draft while the previous published version is available to anyone with read access to the document.

Workflow for DDM is set at the document type, so various workflow patterns could be intermixed in a single hierarchy. This allows for complex workflow patterns for even the basic of document management implementations.

If your DDM implementation is not using the review or approval workflow process or you are just using the saving of a document as a version as your publishing process, you would be in the MINIMAL level of relevance square.

If you are using review and approval process for documents and have created custom document types to facilitate the workflow process, you would be in the AVERAGE level of relevance square.
If you have done any of the following within DDM, then you would be in the HIGH level of relevance square:

- Using DDM Custom Events to intercept the document type review and approval workflow processes
- Changing the security of a document through customizations during the Workflow process
- Any integration with external workflow systems

Any any of these cases you will need to understand any workflow customizations before any migration or upgrade.

4.2.5 Combinations

If you discover that your DDM implementation has one of the focus areas as HIGH and the others as AVERAGE or MINIMAL, you are not atypical. You might also find that if you have multiple libraries used by different user departments, the priority matrix might be different for each. The priority matrix is a tool to help you get an understanding of your implementation and what effort level will be required to move to another solution.

If you are having difficulty gathering the information to complete the priority matrix or you are unsure of the correct categorization (perhaps because you are not the original implementor or the solution), the best method is to work with an experienced DDM developer to document your environment and help you determine your next steps. Any time and money spent at this stage of your process to move from DDM to any other system will pay major dividends down the road.

4.3 Customer Scenarios

The three scenarios that are typical for DDM customers are as follows:

- Document Store
- Advanced / Team Document Management
- Robust Document Management System

Each one of these scenarios has a different use case, level of complexity, and requirements. This section will detail each of the scenarios and the characteristics that will help you decide which one is right for your implementation.

4.3.1 Document Store scenario

The Document Store scenario is typical of an implementation that was used to replace a team folder or shared network drive. This implementation is used with the minimum amount of customization, hierarchy, and security. In many cases, the complexity of DDM was more of a burden than a feature.

A typical setups for the Document Store scenario has a library for each functional unit in the company and a binder per team, user or client. Examples of how this may be organized are shown in Table 4-2 on page 53.
These implementations were setup by the DDM administrator to mimic an existing shared or network folder layout. In many cases, the users can create their own binders without any interaction with an administrator.

The following characteristics are typical of a DDM implementation that fits the Document Store scenario:

**Minimal customizations**
In the Document Store scenario, implementations usually have minimal customizations. The most common customization in this scenario is custom document types with meta data to provide the extra value over a shared network drive. Implementations that fit this scenario may also have replaced the DDM logo with a company logo and done basic view customizations, such as adding a document type meta data field to a column.

**Default hierarchy**
Many Document Store implementations used the basic DDM hierarchy because it was a requirement of DDM. The File Room and Binder Category levels were left at the default values. The primary purpose of file cabinets and binders was to segment the number of documents for performance reasons. Often, implementations with the Domino Store scenario feel that the required hierarchy of DDM is limiting.

**Default security**
In the Document Store scenario, the security settings are set at the top levels and everything underneath them inherits that security. For DDM, this means the security is set at the Library or File Cabinet levels. Binders and documents inherit that security.

The Document Store scenario is also typically open to numerous users who have read authority. In some cases, the entire company can read everything in the Document Store. In other cases, the Document Store is specific to a division, location, or functional team.

**No workflow**
Document Store implementations typically have little to no workflow. The majority of workflow in these scenarios is to publish a document by creating a new version. Usually, the Draft functionality is turned off. When a document is edited, a new version is always created. Any removal of documents is done manually.
Result
The majority of implementations that fit into the Document Store scenario are ready to move to IBM Lotus Quickr today. Customers who are using DDM as a Document Store will see the opportunity to move to a flatter file storage model as a benefit. Because these groups do not use the advanced features of DDM, they will not miss them. Many of these customers should also consider using the new Files capability for personal file sharing.

These customers all have one thing in common - they will move to Lotus Quickr with little hesitation. Many will do the migration away from DDM themselves.

4.3.2 Advanced/Team document management

A typical customer with the scenario of Advanced/Team document management uses DDM to manage documents at a complex level.

The document management system is seen as a key business process tool but is self-contained. The entire document life-cycle takes place within DDM. External applications may interact with the repository, but the customizations, security, and workflow for the documents resides completely within DDM.

Basic to moderate customization
For the Advanced/Team document management solution, organizations add customizations to make the repository fit into the business process and company culture. These typically go beyond the standard document types and meta data. Complex view changes with their own action buttons, corporate branding, and even more complex customizations are typical. Often, customers customize both the Notes client and Web browser versions of Domino.Document Manager, adding another level of complexity in the Advanced / Team document management scenario.

Another common customization is to use the Read Only file option on a document to allow attachment of a PDF file. The only way to do this is by using the DDM Custom Events Script Libraries and the Document Object Model. When a document is approved, this customization can automatically create a PDF (on the local workstation or through server software) and attach it. When a user opens the document, instead of having the attachment open in its native editor, the PDF opens. Even with the native application control and read-only status capabilities, having the document open a read only format, one which can even prevent copy and paste of the document text, is a preferred solution. PDFs are also preferred because having PDF viewing software on a machine is likely for most users.

Moderate to complex hierarchy
For customers with the Advanced / Team document management scenario, the hierarchy is the most complex part. A single implementation often has multiple libraries and might use both File Rooms and Binder categories. The File Cabinets and Binders separate documents based on many criteria, such as periods of time (year, quarter, month), customer (by name, by customer ID), regions (country, state, city) or company organization (divisions, location). An implementation might have hundreds, even thousands, of file cabinets and binders. When considering moving off of DDM, flexibility and scaling of the hierarchy is the primary factor for determining the appropriate destination product for the migration.

Moderate security
The Advanced / Team document management solution uses security for controlling who can see and edit documents. In most cases, groups in the Domino Directory (maintained by the Domino administrator or the document management team) provide the mechanism for security. The exact number of groups may vary. The primary value of using groups is to
reduce any security changes required when someone leaves a company or is reassigned. For many implementations, groups also provide role-based assignment of the ability to edit or manage documents (for example, a customer service application that uses DDM to store shipping and billing documents). Any customer service representative can access and edit the documents based on the call queue.

**Basic to moderate workflow**

Customers with the Advanced / Team document management scenario typically use the built-in workflow. Both review and approval processes are used within the default options. Customizations around workflow in this case are limited to using the DDM event script libraries to create custom notifications or a simple publishing of a completed document to an external repository for solutions, such as HR or Websites.

**Result**

There are two possible outcomes when a customer falls into the Advanced / Team Document Management scenario. The first is that the user will migrate to IBM Lotus Quickr and change the way they structure and use a document management system. The other option is that they will decide to leave the DDM system in place as an existing application that does not change.

If the hierarchy structure of your DDM is the area with which you are most flexible, we would suggest that you wait for Lotus Quickr Next (tentatively called Lotus Quickr 8.5), which will be released in the first quarter of 2010. IBM has committed to adding folder-level security to Lotus Quickr. Folder-level security will allow a DDM hierarchy with hundreds or thousands of file cabinets and binders to be replicated within Lotus Quickr. Today, that would need to be done using Rooms. While hundreds of rooms can be created, waiting for folder-level security will be the best option for DDM users who rely on a strict hierarchy.

Each DDM implementation that fits into the Advanced / Team Document Management scenario should be evaluated independently. The implementation can be moved to IBM Lotus Quickr or can stay on DDM.

**Customers willing to move to IBM Lotus Quickr**

Making the decision to move your current DDM implementation to Lotus Quickr for the Advanced / Team document management scenario is one that requires change. You will have to implement your customizations, hierarchy, security, and workflow tailoring in new ways in Lotus Quickr. Besides moving the documents from one to the other, the only piece that you can automate will be the hierarchy. Everything else should be reevaluated in the context of Lotus Quickr. In some areas, there will be better ways to accomplish what you were doing in DDM.

Customizations that provided custom branding or worked with the DDM events will need to be done within the parameters of how Lotus Quickr customization and placebots work.

**Customers keeping DDM implementation**

If you find yourself with a DDM implementation that has moderate customization, complex hierarchy, moderate security, and moderate workflow, you can leave the document management solution alone. Because of how DDM is designed, the Domino server can be configured to allow for DDM to continue on that specific Domino version. You can set up your infrastructure so that the rest of your Domino environment (which may include mail, SMTP, and application Domino servers) can upgrade to the latest version while the DDM server stays at the current level. Changes from the Domino Directory can still replicate to the DDM server even if the Domino Directory is upgraded to the latest version.
4.3.3 Robust Document Management System scenario

During the 10 years that DDM has been available, some customers have used it to implement robust, enterprise-wide document management systems, often for thousands of users. This robust document management scenario typically was used to meet regulatory requirements around documents and how a company authored, approved, and maintained them during their life cycle. These systems were designed with redundancy and high availability in mind.

Advanced customization

For the Robust Document Management System scenario, many implementations have complex customizations. Significant changes to the user interface or a complete custom interface is typical. Advanced use of the DDM Event Script Libraries is found in this scenario. In some cases, the developer bypassed the DDM event model was bypassed and used the Domino API directly for tasks such as creating and editing Binders and Documents directly.

Another version of advanced customization for the Robust Document Manager System scenario is a custom Desktop Enabler user interface. The DDM Desktop Enabler allows the use of a custom interface.

Moderate to complex hierarchy

Hierarchy for the Robust Document Management System scenario can be basic to advanced, depending on specific implementation requirements. If the implementation is based around a regulatory issue, the Hierarchy can be more complex than if it is organized around business unit, location, or customer. The hierarchy is also dependant on the number of documents in the system - which in this scenario can often be millions. In DDM, creative solutions that used every level of the hierarchy had to be used to accommodate this number of documents.

One trend that is common for this scenario is a large amount of meta data based upon business rules. Look-ups to other systems and conditional meta data fields based on user selection are often found.

Moderate to complex security

Any DDM implementation that has regulatory requirements will have high security requirements. Often, the security is the most important part of the implementation. The regulatory rules require that documents can be seen only by specific users. And the quality of the security has legal implications.

One typical security customization is that the manager role for a document is assigned to a new individual after legal approval is complete. An automated process releases the document to a new author. This prevents unauthorized editing and publishing without legal review yet still allows a document to be seen in the system.

Moderate to complex workflow

The Robust Document Management System scenario often uses both the built-in review and approval workflow processes and expanded workflow capabilities (beyond what comes with DDM). The customizations to the workflow can be done within DDM or moved to an external workflow system.

Typically, a document must go through various review and approval steps, but always go to a legal approver as the last step. Once that legal reviewer approves a document, it must not be changed and published externally without the legal reviewer approving the document again. This must be implemented in document management system to prevent new versions from being published without that review process.
**Result**

In the IBM software portfolio, IBM Content Manager and FileNet are the options for enterprise content management. IBM Content Manager is the preferred migration option for DDM when enterprise scalability and regulatory security are fundamental requirements. While this means moving from a Domino solution to a solution that requires an enterprise-ready database and application server, users who prefer the Lotus Quickr user interface have options. Lotus Quickr (J2EE™) today and Lotus Quickr (Domino) in the NEXT release (targeted for the first quarter of 2010), both allow the user to use Lotus Quickr to interact with the enterprise content management solution directly within Quickr. For those implementations that decide to move to IBM Content Manager or FileNet we suggest you look at the Lotus Quickr enterprise content management integration options.

**4.3.4 Other Scenarios**

There are other scenarios that your implementation(s) might fall into. For example, DDM is used as a document repository for another Domino-based application. The only time a user interacts with DDM is when a link to the document is selected by the user within the custom application.

All of these possible scenarios have potential for migration from DDM to Lotus Quickr. Each implementation should be measured using the priority matrix and the DDM Magic tools. These will give you a better understanding of the complexity of the DDM implementation.
Building Your Plan for a Plan
Building your Plan: Analyze and document your current environment

Before you can plan your migration from Domino Document Manager (DDM) to Lotus Quickr services for Lotus Domino, you should have a thorough understanding of your current DDM implementation. This chapter provides an overview of the various aspects of DDM that you should analyze and document. Completing this analysis will provide information necessary to assist you in determining the best migration path for your organization.

Note: Although the analysis discussed in this chapter is focused on migrating from DDM to Lotus Quickr services for Lotus Domino, the same analysis and preparation should be used as part of an assessment for migrating to other content management systems such as IBM Content Manager Collaboration or IBM FileNet Content Manager.

This chapter provides an overview of various methods for gathering the information about your DDM implementation. However, other chapters in the book provide information about tools that you can use to obtain the details of your DDM implementation.

Each section contains a list of key questions to help guide you through the analysis. Appendix B, “Summary questionnaire for technical assessment” on page 259 contains a complete questionnaire.
5.1 Functional Overview

Before planning your DDM migration, you should have a good understanding of the existing DDM implementation, the business processes it supports, and any customizations made to the application. To achieve this understanding, you need to perform a systematic analysis of your DDM implementation. Your analysis should capture and document the aspects of your DDM implementation that are relevant to a migration.

Before you start the analysis of your DDM implementation, determine who the DDM experts are in your organization. These experts may include the individuals involved in the initial planning and deployment of DDM, the IT administrator, developers responsible for maintaining DDM, and the current business users of DDM. Involve these individuals and use their combined knowledge as much as possible throughout the assessment and migration process. If outside experts were also involved, such as an IBM services team or an IBM business partner, consider engaging them to assist in the assessment process as well.

5.1.1 Formal Documentation

The first activity in your discovery process is to gather and review existing formal documentation relating to your DDM implementation, including the following items:

- Business process flow diagrams
- Functional and non-functional requirements specifications
- Use cases
- Design documentation
- Security requirements
- Server architecture documents/diagrams
- Replication topology diagram
- Enterprise architecture document that describes DDM integration with other systems

For many DDM implementations, some documentation may be out of date, or worse, simply not in existence. Regardless, in addition to reviewing any documents relating to the DDM implementation, a more hands-on approach to gathering information is also necessary. The remainder of this chapter reviews the aspects of your DDM implementation to focus on as part of this hands-on approach.

5.1.2 The Role of DDM in supporting the business process

The first step in your analysis is to research and document the business processes in which DDM is involved. A business process is a set of coordinated activities, conducted by both people and systems, with the purpose of accomplishing a specific organizational goal. Some examples of business processes are obtaining government approval for a drug or medical device, managing legal documents, or publishing technical drawings. Often, document-related business processes include complex review and approval workflows, publishing documents to external applications and deriving information from other business applications.

Note: Keep in mind that your organization may have multiple implementations of DDM supporting different business processes or a single DDM implementation that supports different business processes. Each business process should be identified and analyzed as it relates to DDM.
To obtain information about a business process, you must first identify the subject matter experts (the business stakeholders and process participants). The stakeholders are those individuals that define the business process and are responsible for making sure the business goals of a given business process are met. The participants are the business user that execute one or more of the core functions of the process. They typically have unique insight into the actual workings of the business process and associated applications. For the purposes of this analysis, these business users are the DDM users. Understanding how DDM fits into the business process facilitates communications with the business stakeholders and provide a better understanding of the business areas that will be affected by the migration.

Engage the business stakeholders in the initial stages of your assessment process. The details they contribute may uncover specific areas in which you need to focus your analysis such as DDM customizations or integration with other applications.

One approach for obtaining the business process details is to host information gathering sessions in which you ask a series questions to elicit the details of the business process. You may want to start by capturing the inputs and outputs of the process and then describe the user/system actions that occur within the process. The output of this would be a document that provides an overview of the entire business process, with special attention to the parts of the process that incorporate DDM.

**Note:** Chapter 6, “Confirming your business goals and key business decisions” on page 79, discusses the examination of your organization’s current business goals and the key business decisions that drive the requirements of your content and document management solution going forward.

### 5.1.3 DDM functionality

Once you understand and document the business process, you need to identify and document the corresponding DDM functionality as it relates to the business process. You need to document the DDM functional requirements to identify how DDM fulfills its role in the business process. This should include a description of the DDM features being used as well as how they are used. Documenting the DDM functionality can be accomplished by employing informal approaches such as interviews and user stories (natural language documentation) or more formal techniques such use cases and UML diagrams.

#### Document life cycle

You should have an in-depth understanding of the extent to which your organization is using the document life cycle capabilities of DDM. In your analysis, describe the following document life cycle functionality:

- **Document authoring**
  
  How are documents created? Do most document have a single author or do multiple users collaborate on the authoring of a document? Are there naming conventions and categorization guidelines for documents? Are documents frequently updated or are most documents fairly static?

- **Workflow**
  
  Is a review and approval process being used for documents? If so, what document types require review and approval? Is your organization using the review/approval capabilities provided out-of-the box or are custom workflows being used? Describe the review/approval processes being used for document authoring.
Major or minor versioning of documents
Is the ability to create working drafts (also known as minor version) being used for certain document types? What is the versioning strategy used for the various document types?

Document publish and release
What actions are triggered when a new document version is released? Once released, are documents distributed to other systems or made available to a wider audience?

Document archiving
Are documents being archived? If so, what is the archive repository? What is the criteria for archiving? Can documents be retrieved from the archive repository (that is, un-archived)

User experience
You should capture the details about the various methods by which users interact with the DDM libraries, specifically:

Access to DDM libraries
How do users access the DDM libraries?
- Using the Desktop Enabler from within desktop applications (ODMA-enabled applications such as Microsoft Office applications)
- From Windows Explorer
- From the Notes client
- From a Web browser
- From a custom user interface (for example, within another application)

Mail integration
Do DDM users store Lotus Notes e-mails in DDM? Do users replace e-mail body text and file attachments with links to DDM documents. Is DDM integrated with Microsoft Outlook?

Sametime integration
Is DDM integrated with Sametime? Is the Who Is Online feature being used? Do users save chat transcripts to DDM?

Discussion Forums
Are document-specific discussion threads being used within DDM libraries?

Offline access
Do users make use of the Attache Case to work with documents while offline.

Bookmarks
Do user create links to documents in other binders using bookmarks?

Favorites
Do users create “favorite” documents and binders to access from the library?

Search
How do users typically search for documents? Is Lotus Domino Domain Search used?

Note: During your research of how DDM fits into the business process, keep in mind that this is an excellent opportunity to identify areas of improvement and ultimately drive the requirements for the new solution. In the next chapter, we focus specifically on analysis of the business requirements and explain how the migration presents a strategic opportunity for an organization to re-visit its business processes and how a Quickr Solution or Quickr and ECM solution might support a more efficient business process.
5.1.4 Nonfunctional requirements

In addition to the functional requirements, the nonfunctional requirements of DDM should be documented. Nonfunctional requirements dictate the qualities and constraints of the application and include the criteria for things such as usability, availability, performance, scalability, and so forth.

5.1.5 Summary: Key Questions

The key questions you should be able to answer relating to the functionality of your DDM implementation are as follows:

- Who are the DDM experts inside and outside your organization (IT Analysts involved in the initial planning and deployment of DDM, IT administrator and developers responsible for maintaining DDM, current DDM users, business partners)?
- What are the functional requirements of the existing DDM implementation? What out-of-the box functionality is being used and how is it being used? Is there existing documentation describing your DDM implementation?
- What are the business process(es) that DDM is involved in. What part of the business process does DDM support? How does DDM fulfills its role in the business process? Is there documentation describing the business processes DDM supports?
- Who are the subject matter experts (business stakeholders and business users) regarding the business processes that DDM supports?

5.2 Implementation details

The following section discusses how to review specific implementation details of your DDM hierarchy, including how to review the structure currently in place, a high level security setting review, and a review of document types being used.

5.2.1 The DDM hierarchy

Some DDM implementations have a fairly simple hierarchy that incorporates just the required levels corresponding to the library, file room (default), file cabinets, binders and documents. Other implementations have a more complex hierarchy that include numerous levels of categorization for both file cabinets (file rooms) and binders (binder categories). Having a solid understanding of your DDM hierarchy will be critical when determining the mapping of the DDM paradigm to the Quickr paradigm. See Chapter 7, “Hierarchical and structural considerations between Domino Document Manager and Quickr” on page 89 for more details on understanding your DDM hierarchy and how it maps to Quickr.

Document your DDM hierarchy. A good way to capture the hierarchy is to use Windows Explorer or the DDM site map. Your document should include an explanation of how the DDM content is organized (by geography, business unit, project, customer, and so forth) and provide a short description of each level of the hierarchy.
Figure 5-1  DDM hierarchy
Note: You should make certain that you are using a DDM site administrator id to ensure that you are able to see the entire DDM hierarchy.

For a more detailed view of the DDM libraries hierarchy, you can also use the Security Site Map tool discussed in Appendix A, “Domino Document Manager Site Map and Security Analysis Tool” on page 253.

5.2.2 DDM security model

Document the DDM security model, including a description of security at the library, file cabinet, binder, and document levels.

The DDM Site Map tool provides the ability to generate reports that capture the DDM library structure and access control lists on file cabinet and binders (see Appendix A, “Domino Document Manager Site Map and Security Analysis Tool” on page 253 for a complete description of the DDM Site Map tool). Figure 5-2 contains an example report.

<table>
<thead>
<tr>
<th>File Cabinet</th>
<th>Security</th>
<th>Binder Category</th>
<th>Binder Name</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domino Doc 001</td>
<td>Managers</td>
<td>Not Categorized</td>
<td>Default Binder</td>
<td>Managers</td>
</tr>
<tr>
<td></td>
<td>Lotus Administrator</td>
<td></td>
<td></td>
<td>Editors -</td>
</tr>
<tr>
<td></td>
<td>Editors</td>
<td></td>
<td></td>
<td>Readers -</td>
</tr>
<tr>
<td></td>
<td>Domino.Doc Users</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Readers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not Categorized</td>
<td>2nd Binder</td>
<td>Managers</td>
<td>Lotus Administrator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Editors</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Readers</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not Categorized</td>
<td>3rd Binder</td>
<td>Managers</td>
<td>Wojciech Kowalski</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lotus Administrator</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Editors</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Readers</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 5-2  Sample report on security set for each of the hierarchical objects inside of Domino Document Manager

See Chapter 7, “Hierarchical and structural considerations between Domino Document Manager and Quickr” on page 89 for more details on understanding your DDM security model and how it maps to Quickr.

While it is not feasible to capture security settings for all of your DDM documents, you can get an idea as to whether document level security is being implemented by checking the security settings of randomly selected documents. To check the security on a document, open the document profile and select the Security tab, as in Figure 5-3 on page 68.
5.2.3 Document types

Document types apply identifying attributes to documents. For example, a contract document might have information like contract number, company name and start date. These attributes, referred to as metadata, are used to search, retrieve, and manage documents and binders. You should capture all of the document types being used within your DDM implementation.

To determine which document types are included in your DDM implementation, perform the following steps for each library:

1. Open the Library database.
2. Select Library Administration in the left navigator.
3. Select document types in the left navigator.

This view displays a list of all of the document types available in the library. However, not all of the document types listed are necessarily being used. To determine which document types are actually used (and are the default within each file cabinet), perform the following steps for each file cabinet:

1. Open the Library database.
2. Select Library Administration in the left navigator.
3. Open the File Cabinet profile document.
4. Review the following values in the Security section:
   - Allowable binder types
   - Allowable document types
   - Default document type

Figure 5-4 on page 69 illustrates this approach to determine the allowable binder types and document types within the file cabinet.
Additionally, a Custom Subform can be applied to all documents within a library or file cabinet. To determine this information, for each library perform the following steps:

1. Open the library database.
2. Select library administration in the left navigator.
3. Select System Profile in the left navigator.
4. Review the following values in the Advanced File Cabinet Setup section:
   - Name of custom subform to be included on all documents
   - List of required fields on custom subform for documents

Figure 5-5 illustrates this approach.

5.2.4 Site profile

For each DDM implementation, you should capture the site profile. The site profile provides summary information about your DDM installation. To access the site profile, perform the following steps:

1. Using the Notes client, open the Domino.Doc Site Admin database
2. Open the hidden view (Domino.Doc Site Profile) (To display hidden views, press the Ctrl and Shift keys and select View/Go To)
3. Open the document displayed in the view.
5.2.5 System profile (library configuration)

For each library, you should capture the system profile. The system profile provides configuration details for a given library. To open the system profile document, perform the following steps:

1. Using the Notes client, open the library database.
2. In the left navigation panel, select **Library Administration**.
3. In the left navigation panel, select **System Profile**.

5.2.6 Volumetrics

For each DDM implementation, you should capture the following volumetrics:

- Number of users
- Number of libraries
- Number of file cabinets
- Number of binders per file cabinet (average/max)
- Total number of binder (document) databases
- Binder database sizes (average/max)
- Number of documents per binder (average/max)
- Total number of documents
- Total size of DDM managed content

See Chapter 7 for details about the DDM on server structure.

5.2.7 Systems architecture

You should document the overall systems architecture as it relates to the DDM implementation. Typically this information is in the form of diagrams and tables. The systems architecture details include the following information:

- **Software inventory**
  DDM, Desktop Enabler, Notes clients, Domino servers, browsers, desktop applications
- **Operating Systems**
- **Server architecture**
  The hardware, platforms and servers and how they are distributed (include CPUs, RAM, Disk Storage, network types and bandwidth, cluster/fail-over, master replicas)
- **Replication topology**
- **Interfaces between DDM and other applications (events and data flows)**
- **Links between DDM and other entities within and outside of the organization - intranet, extranet, Internet, eCommerce**
- **Local and wide area networks**
- **Geography**
  The physical locations of users that access DDM directly
- **Programming Languages and APIs used for interactions between DDM and other subsystems (that is, OLE Automation API, Visual Basic, Visual C++®, and so forth)**
- **System Security architecture**
  LDAP, Domino Directory, Single Sign-on, SSL, and so forth.
5.2.8 Summary: Key questions

The key questions you should be able to answer relating to the details of your DDM implementation are:

- How are file cabinets, binder and documents organized in DDM (what is the DDM hierarchy)?
- How are file cabinets, binder and documents secured in DDM (what is the DDM security model)?
- Are there corporate standards or guidelines that authors follow when working on DDM documents?
- What document types are being used and what metadata is associated with each document type?
- What are the volumetrics associated with your DDM implementation?
- What are the details of the overall system architecture as they relate to DDM?

5.3 Customizations

Often, organizations customize DDM to meet both business requirements and user needs. Some modifications are merely cosmetic, but most are made to modify or extend the out-of-the-box DDM functionality. For the purposes of the DDM migration assessment, you need to know both what customizations have been made and why they were made. Specifically, you need to understand and describe the relationship between the customizations and the business process being supported. Having this understanding is critical to determining the most appropriate migration path for your organization.

To help you gain some insight into DDM customizations, this section provides an overview of the common types of DDM customizations and some methods for gaining a high-level perspective of these customizations. Both Chapter 8, “Design element review” on page 123 and Chapter 9, “Analyzing customizations” on page 137 provide more information about methods for obtaining the design level details of the DDM customizations. Individuals performing this part of the analysis should have a good understanding of Notes and Domino development concepts.

**Note:** Typically, a DDM implementation will have undergone numerous design changes over time, making it difficult to determine all of the changes that have been made. The DXL Magic tool discussed in Chapter 9, “Analyzing customizations” on page 137 provides detailed reports on the differences between your deployed DDM templates and the original out-of-the-box templates. These reports can be used to assist you in determining the customizations made to the various DDM design elements.

Some of the most common reasons for customizing DDM are as follows:

- Customizing the look and feel
  Adding a company logo, adding or changing graphic elements, button, menu and action bar changes, modifying terminology (static text), and enhancing the user interface.

- Custom processing of documents
  Changing or extending the way a document is processed, including both back-end processing and front-end actions taken by the user.
Integration with desktop applications
Providing seamless integration with ODMA-enabled desktop applications such as Microsoft Word, or providing a custom client for interacting with DDM.

External application integration
Integrating DDM with custom Domino applications and other Domino product such as Lotus Workflow, or with third party records managements systems or applications such as archiving, scanning, and publishing systems.

5.3.1 Common areas of customizations

Document types and binder types
In addition to using the document and binder types that come with DDM, many organizations create their own custom document and binder types that are specific to their business processes. Document types and binder types are implemented with subform design elements in the file cabinet template (filecab.ntf). The fields (and formulas) on these subforms correspond to the fields that are in the document or binder profiles.

Make a list of all custom document types and binder types created by your organizations. See Chapter 9, “Analyzing customizations” on page 137 for details on how to gather that information. For each custom document type and binder, you should capture the details associated with it. This information can be obtained by performing the following steps:

1. Open the Library database.
2. Select Library Administration in the left navigation panel.
3. Select Document Types in the left navigation panel.
4. Open the custom document type displayed in the view.

Perform the same for the binder types, selecting Binder Types in the navigator for step three.

You should make a list of all custom document types and binder types actively being used by your organizations See Chapter 5.2.3 for details on how to gather this information.

Design element additions and modifications
One of the most basic methods for customizing DDM is to add or modify existing design elements such as forms, subforms, views, and navigators. These types of customizations are made using Domino Designer to change the behavior or look and feel of DDM when accessed from the Notes client and a Web browser.

Note: Customizations that are purely aesthetic in nature are typically not pertinent to a migration but are included here to provide a complete representation of common customizations made to DDM.
Forms and subforms
There are countless ways of customizing DDM forms and subforms. Some are significant to change the behavior of DDM, others modify the look and feel. The following list provides a sampling of some common customizations made to DDM forms and subforms.

- Modifying document-related forms or subforms
  - This type of customizations is done to add or make a change to a feature that is common to all DDM documents. Examples include:
    - The addition of a button used for integration with an external application
    - Addition of a new field (for example, a Computed for Display field to display corporate compliance information to the user).
    - The addition of subform to capture metadata dependent on other metadata.
    - Addition of a company logo or other graphics
    - Reorganization of the form layout

- Modifying search-related forms or subforms
  - This type of customizations is used to make a change to the behavior or layout of the forms used to perform DDM searches through the Notes client and a Web browser. Examples include:
    - Adding code to extend the search capabilities
    - Adding a button to integrate with an external search engine
    - Building customized and parameterized search forms

- Other miscellaneous types of forms/subforms modifications
  - Customizing the behavior of review and approve actions
  - Customizing the e-mail notification relating to document life cycle events
  - Customizing the dialog boxes and messages presented to the user
  - Customizing the discussion forum functionality

Views and Folders
View and folder modifications are probably the simplest and most common type of DDM customizations. Organizations make changes to existing views and folders as well as add new ones. The following list details some typical types of changes:

- Addition of new columns, or change the existing columns
- Changes to the view categorization
- Changes to the view selection formula
- Rearranging of the action buttons, change their wording or icons
- Changes to the look and feel - colors, fonts, and so forth.

Navigators
Modifications to navigators are done to manage the way by which users access the features of DDM from a Notes client. The following list details some examples of Navigator customizations:

- Hide/show navigator actions based upon the type of user
- Add button/action to navigator to provide a link to an external application
- Addition of the company logo or other graphics
- Creation of image map
Custom agents
Custom agents are typically used to extend the functionality of DDM to meet business requirements. Some examples of the uses of custom agents are as follows:

- Batch updates to documents
  A custom agent could be developed to offload certain types of document updates to occur during off-peak hours.

- Document distribution
  Agents are sometime developed to move documents to other document management systems.

- Report Generation
  Agents can be run to gather information about documents for the creation of reports.

- Archiving
  An agent can be used to run at certain intervals and move documents into an archive repository based up certain criteria.

- Offline storage
  An agent can be used to export\move data to DVD or other long term storage device.

Custom event handling
DDM has specialized events that are used to affect the way DDM reacts to certain actions. For each of these events, there is a corresponding event handler in which code can be added to provide custom handling of the event. Event handlers are basically the actions that are triggered before or after an event occurs (known as QueryEvent and PostEvent).

It is important to have a comprehensive understanding of any custom event handling in your DDM application, because this is where much of the business logic is implemented. Having this information will help you determine the best approach for implementing the business processes in your Quickr environment.

Document events
DDM provides document specific events. The document event handlers provide the ability to trigger a custom action to run before or after a document event occurs. Specifically, before or after these document-related events, these custom actions include the following items:

- Creation
- Check-in
- Review and approval (setup, submittal, and completion)
- Deletion
- Retrieve from archive (This is a special type of event that is used to customize the actions relating to a document retrieval requested by the user.)

Administration and configuration events
DDM also provides events that correspond to administration and configuration activities related to file cabinets, binder types and document types. These event handlers provide the ability to trigger a custom action to run before or after one of the following events occur:

- File cabinet creation, modification, and deletion
- File cabinet replication
- File cabinet contents retrieval
- File cabinet connecting
- File cabinet cloning
- Binder and document type creation, modification, and deletion
Search events

DDM has events that correspond to the display of search results in a Web browser. The event handlers for these events can be used to customize the actions that run before or after one following events:

- Results for a search within a single file cabinet have been returned
- Results for a search across multiple file cabinets have been returned
- Search results for a specific Notes database within a file cabinet have been returned

In the library template the event handlers are defined in the LibEvents script library, and in the file cabinet template the event handlers are located in the DocEvents script library. Figure 5-6 illustrates a listing of the events within each script library.

![Figure 5-6 Events in the LibEvents and DocEvents script libraries](image)

**Note:** If the green scroll image next to a LibEvent is filled in, that is an indication that the event contains custom code. However, the DocEvents include some stub code out-of-the-box, thus the indication that they contain code.

Some common implementations of event handing are as follows:

- After a document is checked in and versioned, publishing it to an external system (that is, a corporate intranet).
- Notifying a specific group of people when a document is updated (that is, notifying a marketing team of updates to product literature).
- Before a new file cabinet profile document is saved, validating the information supplied by the user and updating it based upon specific configuration settings.
Integration with Other Applications

Often, a business process spans multiple systems. As such, it is a common requirement to integrate DDM with other applications to implement business processes. As part of the analysis of DDM implementation, you should provide a thorough description of how DDM is integrated with other applications and the methods used to provide the integration. DDM provides the following mechanisms for integrating with other applications.

**Domino.Doc API programming**

DDM comes with an extensive API that provides the ability to access DDM libraries from external application, extend the DDM user interface or create a completely custom interface, develop custom applications that interact with DDM, as well as to perform background processing of DDM documents.

Some common uses of the DDM API are as follows:

- Integrating DDM with an external workflow application
- Developing a document import/export tool
- Archiving documents to an external storage system.
- Generating custom reports
- Implementing scanning software to directly scan documents into the DDM environment

**Open Document Management API (ODMA)**

The Open Document Management API (ODMA) is a vendor-independent standard that enables desktop applications to interface with document management systems. DDM comes with an ODMA-enabled interface called the Desktop Enabler. From within Windows Explorer or an ODMA-enabled desktop application such as Microsoft Word, users are able to interact directly with DDM, performing all of the document-related activities such as creating, saving, and checking in a new document and opening, checking out, editing, and checking in an existing document. Using ODMA, some organizations have created a custom user interface that bypasses the Desktop Enabler. If your organization uses a custom ODMA interface, you should understand and document how the custom interface is used, the functionality it provides and the business process it supports.

### 5.3.2 Summary: Key questions

The key questions you should be able to answer relating to DDM customizations are as follows:

1. Does your DDM implementation have custom document types and binder types? If so, what metadata is associated with each?
2. Does your DDM implementation have customized design elements (forms, subforms, views, navigators, agents)?
3. Do your libraries and cabinets inherit their design from more than one file cabinet template?
4. Does your DDM implementation have custom event handling?
5. Is your DDM implementation integrated with other applications? If so, how is this integration implemented?


7. Does your DDM implementation have a custom ODMA interface? If so, how is this interface implemented?

For each customizations, provide details of the customization and a description of the business process(es) it supports.

**Note:** For a complete summary of key questions to assess your technical implementation, refer to Appendix B, “Summary questionnaire for technical assessment” on page 259.
Confirming your business goals and key business decisions

This chapter is about looking forward. Moving from Domino Document Manager (DDM) to another solution requires some change. So why not take this opportunity to step back and think about how this change might tie into some broader business goals. We give you suggestions and food for thought in this chapter. And we guide you through some of the key decisions you need to make as you build your DDM transition plan.

It is likely that the software and hardware layers of your DDM environment are only a small portion of your overall solution. Much of your implementation is sure to revolve around business goals and processes. Here are some of the key factors to think about as you examine your organization’s underlying business goals and how these needs align with a document and enterprise content management (ECM) system.

» What are the key business processes that are currently supported by DDM?
» Are there any major changes planned for your organization as a whole or for the groups that are your key DDM users?
» Are there other big IT projects that might affect the timeline? (For example is it better to make plans now, but do the work next year?)
» Given the collaborative capabilities offered through Lotus Quickr, can this be an opportunity for this project to be a positive thing instead of just more work?
» Are there architectures and business standards in your organization that are relevant to your DDM plan?
6.1 Existing business processes being supported by DDM

Whether you are a small, midsize, or large business, chances are that you have used the DDM to support a variety of business processes. If you are a 25-person law firm, for example, you might have a single server where you use DDM for its basic library services (that is, check-in, version control, document sharing, review and approval, and so forth) to house all of your firm and client documents.

If you are a Fortune 500 business, (a 15,000 employee pharmaceutical company, for example), you might have many servers, each supporting their own unique business processes. One DDM server might be using the DDM/Domino Workflow integration to package and route groups of documents needed for FDA drug approval. Another DDM server might be used by the HR department for managing all personnel documents through an employee portal. A third DDM server might drive a customized Web application used by the employees on the manufacturing floor for managing standard operating procedures.

Regardless of how your organization uses DDM, your migration to Quickr will bring about some interesting challenges. Think of this as an opportunity to review these business processes and reengineer them for the Web 2.0 world.

6.1.1 Where does your organization fit into the matrix in terms of hierarchy, security, and customization?

Chapter 4, “Scenarios: Which might apply to you?” on page 45 includes a matrix to classify the complexity of your current DDM implementations. After completing the matrix for a given library, you will be able to map that library to one of the following scenarios:

- Document Store
- Team Document Management
- Robust Document Management

Use Table 6-1 to help you create a high level view of the complexity for the DDM implementations in your organization. This will bring some perspective to the complexity of your DDM to Quickr Migration effort.

<table>
<thead>
<tr>
<th>Department</th>
<th>Library Name</th>
<th>Matrix Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Resources</td>
<td>Personnel</td>
<td>Document Store</td>
</tr>
<tr>
<td>Operations</td>
<td>FDA Approval</td>
<td>Team Document Management</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>Standard Operation Procedures</td>
<td>Robust Document Management</td>
</tr>
</tbody>
</table>

Appendix C, “Scenario priority matrix” on page 263 provides you with a blank table that you can use for your planning purposes.
6.2 Major changes in the works for your organization?

Worldwide economic, global, and technical changes are just some of the things that force companies to refocus their business plans. Refocusing plans almost always means change. As you plan your migration strategy, take a step back to align your strategy with any major changes that are planned for your organization. The following sections detail a few areas to consider.

6.2.1 Technological changes

- Are there any changes planned to your IT infrastructure?
  - Are you planning any desktop refreshes?
  - Is your network/server infrastructure being upgraded/consolidated?
- Are you planning to upgrade or implement any new software platforms?
- Are you considering using more “open standard” software? (that is, Java, Open Office, Eclipse, Web 2.0, and so forth)
- Does your organization already use an ECM system or are you considering one?
- Are there new requirements for content repository types/vendors?
- Are you moving to a centralized LDAP directory?

6.2.2 Organizational changes

- Have you recently downsized the personnel within your organization so you have to do more with less people?
- Are you becoming more of a virtual organization? (that is, work from home)
- Are there new regulations that require you to keep audit trails about your documents?
- Do you need to implement compliance policies for archiving, document classification, records management, record retention or eDiscovery?
- Have you merged with or acquired another company? Does migrating your DDM environment to Quickr allow you to better share documents and collaborate with your new colleagues?

6.2.3 Financial and budget changes

- Have your travel budgets been cut, forcing you to consider alternatives to travel?
- Do you need to get products out the door faster? With less resources?

6.2.4 Changes in organizational philosophy

- Are you encouraging your employees to do a better job of sharing their documents and work product?
- Is there a new focus on collaboration or social networking?
- Is your organization stressing the importance of doing a better job of capturing intellectual property rendered in electronic documents?
6.3 Are there other big IT projects that might affect the timeline?

Do not plan your DDM migration in a vacuum. If there are other large organizational or departmental IT projects in the works, determine if they will impact your DDM migration plans. The following sections list some examples which may or may not apply to your organization. If they do, they should give you a good starting point for consideration. If not, they will get you thinking so you can more easily identify projects within your company.

6.3.1 Introducing Lotus Connections or Lotus Sametime

Lotus Connections is social software for business that empowers you to be more innovative and that helps you execute more quickly by using dynamic networks of coworkers, partners, and customers. Sametime is the IBM on ramp to unified communication and collaboration. Sametime technologies include integrated presence awareness, instant messaging, e-mail, telephony, Web conferences and audio/video.

Table 6-2 is a matrix that shows how Lotus Quickr can be integrated with Lotus Connections and Lotus Sametime

<table>
<thead>
<tr>
<th></th>
<th>Lotus Sametime</th>
<th>Lotus Quickr</th>
<th>Lotus Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lotus Sametime</td>
<td>na</td>
<td>Online awareness Instant chat Schedule online meetings Access content from Lotus Sametime Connect client</td>
<td>Online awareness Instant chat Community synchronization Access content from Lotus Sametime Connect client View business card</td>
</tr>
<tr>
<td>Lotus Quickr</td>
<td>Online awareness Instant chat Schedule online meetings Access content from Lotus Sametime Connect client</td>
<td>na</td>
<td>Consume RSS feeds from Lotus Quickr Consume RSS feeds from Lotus Connections</td>
</tr>
<tr>
<td>Lotus Connections</td>
<td>Online awareness Instant chat Community synchronization Access content from Lotus Sametime Connect client View business card</td>
<td>Consume RSS feeds from Lotus Quickr Consume RSS feeds from Lotus Connections</td>
<td>na</td>
</tr>
</tbody>
</table>

If you are planning to implement Lotus Connections or Lotus Sametime, be sure to consider how the integration of these technologies with Quickr can benefit your users.
6.3.2 Introducing an ECM solution

You can use Lotus Quickr and the ECM tools to combine collaborative authoring and sharing of everyday business content with the structure, business process management rules, and classification and discovery models provided by ECM products. There are many ECM solutions available in the market today. In particular, IBM offers two ECM solutions:

- IBM Content Manager
- IBM Content Manager FileNet P8

Both solutions are tightly integrated with Lotus Quickr and provide an easy-to-use, attractive user interface that acts as a front-end to both the CM and FileNet ECM system.

See 2.5, “Lotus Quickr + Enterprise Content Management (ECM)” on page 28 for more information about how Quickr integrates with these two ECM systems.

More specifically, you can use these ECM services to perform the following tasks:

- Publish Lotus Quickr documents to the two ECM systems.
- Create and manage links to documents stored in these ECM systems from Lotus Quickr libraries and rich text areas of blogs, wikis, lists, forums, and comments.
- Store and retrieve documents of the ECM systems directly from within desktop applications, such as IBM Lotus Notes, IBM Lotus Sametime, Microsoft Windows Explorer, Microsoft Outlook, Microsoft Office applications, and IBM Lotus Symphony by using Lotus Quickr connectors.
- Use the Enterprise Library Viewer portlet to view and download the documents stored in the ECM systems.
- Search and retrieve documents available in the ECM systems from Lotus Quickr Search Center.

If any of the items listed above would benefit your organization, you should incorporate these items into your DDM migration planning process. Be sure to identify your business requirements and plan accordingly.

6.3.3 Changing your enterprise e-mail system

Lotus Quickr provides integration points with both Lotus Notes and Microsoft Outlook. But, because they are different products, they have their own unique features and functions.

Regardless of whether your organization is moving from Outlook to Lotus Notes or Lotus Notes to Outlook, it is a good idea to take a step back and see how this change will affect your DDM migration plans. You may find it best to combine the rollout of both systems in hopes of utilizing your resources more efficiently. This gives you the advantage of training users on both systems at the same time, rather than disrupting their work patterns with two separate training sessions.

On the other hand, consolidating the rollout may only complicate the introduction of both systems, only to end up with delays, higher expense, and frustrated users.

There is no right or wrong answer. The decision depends on your organization’s unique circumstances. Take the time to access the impact of migrating your DDM environment while at the same time rolling out a new e-mail system.
6.3.4 Changing hardware and operating system platforms

Because of the IBM open standards approach, you can typically lift software systems off of one platform and place them on another without too much trouble. Keep in mind that there is some complexity involved in changing your hardware or operating systems platform.

A change like this is an excellent time to migrate your DDM environments. Quickr runs on a number of hardware and operating systems. Take this opportunity to set up your new Quickr environment with the newly selected hardware platforms and operating systems before migrating your DDM documents.

6.3.5 Outsourcing your IT function

In general, outsourcing your IT functions has little impact on your DDM migration. If this is in the plans, be sure that your new IT support organization is prepared to handle the DDM migration should it happen after the outsourcing transition is complete.

You may want to consider completing your migration before the transition. This would make the outsourcing transition simpler, but would require extra work prior to the switch.

Again, each organization would have its own unique set of circumstances which would govern the decision. The important thing is that you give it sufficient thought while planning your DDM migration strategy.

6.4 Given the collaborative capabilities offered through Lotus Quickr, can this be an opportunity to move forward?

DDM is strictly a collaborative document management platform. It provides you with basic library features, like the following items:

- Shared document repository
- Check-in
- Version control
- Search
- Basic review and approval

Lotus Quickr is a team collaboration and content sharing system that provides the following features:

- Content libraries for sharing documents
- Virtual team places
- Personal file sharing
- Integration with ECM (FileNet P8 and CM8)

Assuming that a Quickr or a Quickr + ECM solution offers you all of the features available from Quickr plus the basic library features of the DDM, is this an opportunity for you to make a change?
In reality, you will not be making your decision to proceed with the DDM migration base solely on the benefits of Quickr’s collaborative capabilities. There will be many items to consider:

- Cost and budget
- Effort
- Needed resources
- Impact on users
- Impact on system admin resources
- Hardware and network
- Product support
- Expected gains in productivity and efficiency
- Migration complexity
- Strategic direction with respect to collaboration, document management and ECM

You might find it useful to create a matrix that outlines the pros and cons of each of these items. Add any other items to the list as you see fit. You may even want to weigh the importance to each item. In the end, this will help you develop your business case for making the change.

6.5 Are there architectures and standards in your organization that are relevant to your DDM migration plan?

Most organizations have defined a set of architectures and standards to ensure consistency in their system infrastructure. Not only does this help you minimize the number of disparate skillsets needed by your organization, but it also provides your users with consistency in how they interact with systems.

As you plan your DDM migration, you should consider how the new Quickr environment it will work within your current architectures and standards. Here’s a couple of questions to ask yourself:

6.5.1 Are you already a Lotus Quickr customer?

If you are already a Lotus Quickr customer, it is likely already included in your IT architecture. If this is the case, the discussion around “architecture and standards” as it relates to Quickr is likely to be short. However, if you are not an existing Quickr customer, you will need to access the impact of adding Quickr to your IT infrastructure. You will need to determine how it affects your existing architecture and whether or not it alters your current set of IT and business standards.

Even if your organization is not currently using Lotus Quickr services for Domino, chances are that it will fit within the constraints of your existing architecture because DDM was housed on a Domino server.

6.5.2 Are you already vested in an ECM solution?

This question is more likely to affect your architectures and standards regardless of whether or not you are already using Lotus Quickr. If you intend to migrate your DDM environment to a Quickr-only environment, you do not need to give consideration to the ECM question. However, if you intend to migrate to a Quickr + ECM environment, you need to address the issue of whether or not your organization is already using an ECM Solution and if so, how easily Quickr can integrate with the ECM environment.
If the answer is yes, what ECM solution are you using?
Quickr is designed to integrate seamlessly with both the IBM CM8 and FileNet's P8 systems. So, if you are using either one of these systems, again, the Quickr + ECM solution should not adversely affect your architecture and standards. If you are not using either one of these systems you will have to devise a way to integrate Quickr with the system you are using or you will need to add either CM8 or FileNet to your infrastructure.

If the answer is no, what ECM solution are you considering?
Quickr is designed to integrate seamlessly with both the IBM CM8 and FileNet's P8 systems. We recommend using either one of these system to avoid the cost and complexity of integrating a non IBM ECM solution.

Regardless of whether you pick CM8 or FileNet P8, you will need to incorporate the technology into your organization's infrastructure and standards. The following list details some issues to consider:

- Personnel and skillsets
- User and system administrator training
- Software and hardware costs
- Outside assistance with installation and configuration
Part 3

Making the Mental Leap
Hierarchical and structural considerations between Domino Document Manager and Quickr

In this chapter we analyze the architecture of IBM Lotus Domino Document Manager (DDM) and IBM Lotus Quickr Services for Domino and describe logical and functional differences between the two products. We also help to map elements of both products with each other, as part of the migration planning.

This chapter contains several sections that address a specific part of the functionality that DDM and Quickr offer. We also make a comparison and offer some suggestions as to how to translate a DDM environment into Quickr.

Quickr and DDM are not systems that can be easily compared. DDM is a document management system, whereas Quickr is a collaboration tool with document management capabilities. This means they differ on multiple levels and have different strengths and weaknesses. The suggestions in the comparison sections can help you bridge the gaps and adjust your thinking to find the similarities that might not be obvious at first glance. In addition, Lotus Quickr might offer many benefits beyond document management for your particular situation, so it is up to you to determine the potential additional value to your company.
7.1 Hierarchy structure

In this section, we address the hierarchy for both DDM and Quickr. We provide a description of the elements in each product's hierarchical structure, then compare them.

As you will see, you have multiple ways of addressing the hierarchy, which gives you an opportunity for re-evaluating your current hierarchy when considering migrating into Quickr.

In Chapter 12, “A Domino Document Manager to Quickr example using a custom placetype” on page 215, we provide a specific example of how to create an actual mapping example. Such an example is also introduced in 7.1.4, “Mapping example” on page 95.

**Note:** What is different between the discussion in 7.1.4, “Mapping example” on page 95 versus Chapter 12, “A Domino Document Manager to Quickr example using a custom placetype” on page 215?

Section 7.1.4, “Mapping example” on page 95 focuses on mapping between DDM and Quickr elements, while Chapter 12, “A Domino Document Manager to Quickr example using a custom placetype” on page 215 provides a larger, more involved example, discussing the following points:

- A scenario of an existing DDM Library and mapping strategy for moving the library to Quickr.
- An example of how to create a custom Quickr document type form
- An example of how to create a Quickr check-in event.
- A discussion on how to add tagging functionality to your migrated Quickr Library.
- A list of caveats that you need to consider during your migration.

7.1.1 Basic hierarchy concepts of DDM

DDM is a document management solution, which means it is a document-central system. It revolves around the files (called documents in DDM) and where they are placed or located. With DDM, it is important to understand that the actual documents are placed in folders called binders, and that each binder has several hierarchy levels above it. This means that you always have several hierarchy levels within a DDM environment.

Understanding the structure of the hierarchy, and the different elements that might exist within the hierarchy is therefore a critical step before you try to make a comparison to other alternatives like Lotus Quickr.

Figure 7-1 illustrates the hierarchical structure of the elements contained in DDM.

![Figure 7-1  DDM hierarchy example](image-url)
Library
A library is the top-level hierarchy element for a DDM environment. It contains all the other elements and forms a basic searchable document storage facility. It allows access rights to be set, which determine who can access the lower hierarchy levels. If you do not have authority to the library, you cannot access anything below it in the hierarchy.

File room
A file room is a virtual layer in the taxonomy. As opposed to being represented by a profile (like binders or file cabinets), file rooms are implemented through a simple text field on the file cabinet profile that is then used in a categorized view. The same concept applies to binder categories. The label entered in a field on the file cabinet profile is used to categorize file cabinets.

File cabinet
A file cabinet is a physical location for a specific set of binders. Security can be set at this level. A file cabinet also allows an administrator to set the master server. So within a single library one file cabinet can have server A as its master server while another file cabinet could have server B as its master server. Both file cabinets will reside on both servers but for check out procedures on documents they will look at their respective master server.

Binder category
By default each binder is in at least one binder category. There can however be up to three (supported) binder category levels. A binder category is not a physical location. It simply allows for categorization of the binders themselves. Any binder can be in multiple categories.

Binder
A binder is the lowest hierarchy level in the DDM hierarchy scheme apart from the document itself. A binder allows for customized access rights and can hold meta data information of its own.

Categorization concept
Categorization is a special concept within DDM. This functionality, which is based on the Lotus Notes categorization, allows the same content to be categorized in more than one way. Within the DDM hierarchy there are two categorization levels: The file room and the binder category.

In Figure 7-2 you see an example of how categorization can work.

![Figure 7-2  Categorization example](image-url)
Within this company example, there are two departments that deal with customer contracts:

- Sales, who communicates with the customers and makes sure the contracts are signed.
- Finance, who checks on customer payments.

Therefore, the company has chosen to insert all contracts for a given year into a single binder called Contracts.

By using categorization, this one binder called Contracts can be depicted both under the Finance Department as well as the Sales Department categories. It might look as though there are two binders called Contracts, but there is only one, referenced by both categories. Figure 7-3 shows how the categorization is listed in the binder profile.

![Figure 7-3  Showing how the categorization is listed in the profile](image)

### 7.1.2 Basic hierarchy concepts of Lotus Quickr

DDM is a document-centric tool where the hierarchy is a critical and fundamental capability that many customers rely on heavily. Lotus Quickr is a collaboration tool with document management capabilities. It can support a hierarchy but the hierarchy and documents in the hierarchy are not the central concept of Lotus Quickr. In Quickr, the organization is not based on the contents but on the people who will be working in the place, and the topic they will be working on. Finding documents and information is based on searching as well as on following a hierarchy structure. This does not mean that there is no hierarchy, but the importance of the hierarchy is less significant than in DDM. Another important difference is that in DDM, files can only be placed in a binder (essentially at the bottom of the hierarchy). In Quickr, files do not need to be added to a specific part of the hierarchy, but can be placed at any level of the hierarchy. Quickr even supports a flat hierarchy (basically, no hierarchy) with all documents at the top level.

![Figure 7-4  Hierarchy available in Quickr](image)

### Place

Within Quickr, a place is the top-level hierarchy element. A Quickr Place is where a group of people can collaborate and share files, thoughts, and schedules. It is not just a place to store documents (like a DDM library). Apart from documents (referred to as files or pages in Quickr), a place can contain much more (discussions, pages, links to other Web sites, custom forms, and calendar and task information).
Room
A room is more or less like a sub place. It allows for specialized access levels within a place so that only specified users get access to the room and its contents. A room also has its own navigation in the Web user interface that allows for more specialized tasks. Rooms can be nested within a place or other rooms. They cannot, however, be nested within a folder. A nested room within another room can have its own security or inherit it from its parent. Content added to a room does not show up in the index of the parent, but will show up in the searches if security permits access.

Folder
A folder allows for grouping of information or documents. Currently, you cannot set specific access at the folder level, so the access to the folder is determined by either the room or place in which the folder resides. Folders can be nested within places, rooms, and other folders. When you create a folder, you have options for how it will look and feel (for example, simple list or nested topics) and any form (meta tags) that are associated with new content.

Content of folders will show up in the index of the parent room or place (depending on where it was created).

Note: The ability to specify folder-level security is a function that is commonly requested by Lotus Quickr customers. IBM intends to provide this capability in the next version of Lotus Quickr, currently planned for the first half of 2010. Refer to Chapter 1, “Introduction” on page 3 for more information (and cautions) about IBM plans for future releases of Lotus Quickr.

7.1.3 Hierarchy comparison

The biggest difference between how the DDM and Quickr hierarchies work is in the placement of files. With Quickr, you can place a file at any level in the hierarchy. With DDM, you can only place a file in a binder (lowest hierarchy level). In other words, having an hierarchy is mandatory in DDM, but it is optional in Quickr. This opens up a lot of new opportunities when determining how to migrate your DDM data into a Quickr environment, but it also means you need to spend time analyzing and evaluating.

The hierarchy in Quickr is much more flexible than it is in DDM. DDM requires a library to have file rooms, cabinets, categories, and binders, Quickr gives you the flexibility to have anything from a single level (the Place) to dozens of nested levels of rooms and folders.

This means that you have choices. If you are using the full DDM hierarchy only because it was required, you might choose to flatten your hierarchy. Or, you might expand the number of levels if your organization requires even more levels than DDM originally offered.

The following sections discuss some considerations that you might take into account to determine what would apply best to your organization when you design a hierarchy in Lotus Quickr for your DDM migration.

Hierarchy dependency
Today, a lot of information is stored in non-structured environments. Finding the right information is based on performing search queries, not on navigating through extensive hierarchy structures. In fact, storing information in a hierarchy structure might even be counter-productive because it limits data to a specific location even though it might be applicable to more than one location in the hierarchy.
A good example of hierarchy independency is the Internet. The information about the Internet is not structured in a consistent way overall but is fragmented over millions of sites and locations. To access the information, you use a search engine like Google or Yahoo to find the specific information you are after. If you do not find what you are seeking right away, you refine or redefine your search query until you find what you are seeking.

Removing your dependency on a hierarchy for locating information might sound revolutionary if you are accustomed to the DDM way of thinking (which mandates that everything must be in a hierarchy of up to seven levels deep), but it actually opens up a lot of new possibilities. For instance, you now have the possibility of consolidating the contents of multiple binders, file cabinets, and libraries into one place. Alternatively, you could also chose to split them up into multiple places.

Another big advantage of Quickr is that information can now be stored at any level within the hierarchy. This means that you are no longer required to store all data at the lowest level (binders in DDM) but can distribute data throughout the whole hierarchy tree.

**Static content versus non-static content**

Static content is content that is not changed often (for example of regulations, archives, manuals, and so forth). This content is often accessed through the Web browser interface by using search functionality. You know what you are seeking and you find it by entering the search criteria. A hierarchy for this type of information is usually of lesser importance.

Users are more likely to access content that is frequently changed (non-static content), like project documentation through a connector directly from applications like MS Word or Symphony. For this kind of content, a hierarchy is more important and advisable, because it will help your users more easily locate the content through the connectors.

Most likely, you will find that both types of content (static and non-static) within your organization. If so, you could consider splitting the content into multiple places or rooms. Use a flatter hierarchy model for the static content and a more extensive hierarchy model for the non-static content.

**Security dependencies**

Another consideration as you think about Lotus Quickr compared to DDM is that in Quickr security is not restricted to specific parts of the hierarchy like it was in DDM (binders and cabinets). Instead, in Quickr, you can set security at any level of a room (or folders in the next release of Lotus Quickr). This gives you much more flexibility.

Whether you require security on multiple levels of content is an important factor in determining how you will organize your information in Quickr. If all your content is readily available to everyone then this is easy. You could create a single place and deposit all files into the index of the place.

If information is sensitive, consider using rooms and secured folders to restrict access to the content. Review 7.3, “Database structure” on page 101, to get a good idea of the effect of using folders and rooms to the server.
Data relations

Each place in Quickr has its own data storage. By default, searching happens within the place. Methods are available, of course, to search across places, but, in general, users will search within a specific place.

It is important to determine how you organize your content into places because the place layout will most likely determine where your users will start their search.

A Quickr Place is much more than just a document storage facility. It also forms a team’s working area. While creating places, it is important to take into account how your organization’s teams are organized. Are they function-driven (as in departments) or goal-driven (as in project teams), or both?

Example: Take for instance a manufacturer. They service multiple clients. To do so, they have multiple departments and cross-department project teams. Their information could be organized in several ways. They could simply create a place per department and each keeps their own data. Or they could chose to create a place per customer, customer-group, or project in which people of several departments can collaborate on that project or customer. They could chose both and create a department place in which each department can store information that is not project-related and separate places per customer or project for the project-related information.

As you can see the possibilities are numerous. The choices really depend on your organization’s current situation, needs, and goals.

7.1.4 Mapping example

So how do the hierarchy elements compare? That depends on what you want to do. If you want to recreate your current DDM environment into Quickr, then a migration could look like Table 7-1.

Table 7-1  Mapping example between DDM and Quickr

<table>
<thead>
<tr>
<th>DDM</th>
<th>Quickr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library</td>
<td>Place</td>
</tr>
<tr>
<td>File Room</td>
<td>Room</td>
</tr>
<tr>
<td>File Cabinet</td>
<td>Room</td>
</tr>
<tr>
<td>Binder Category</td>
<td>Folder</td>
</tr>
<tr>
<td>Binder</td>
<td>Folder</td>
</tr>
</tbody>
</table>

However, you are not required to keep the same hierarchy. Keep in mind that certain things (like categorization under multiple categories) are not available in Quickr.

A library could become a place but could also become a room or even a folder within another place that contains content from multiple libraries. The same goes for every other level. You are therefore not restricted to the above example. This is the time to re-evaluate your current hierarchy structure and make choices as to how you want to proceed into the future.

If you get frustrated that you cannot duplicate your DDM hierarchy exactly, do not forget that you are also getting a lot more functionality, particularly when it comes to collaboration. If you use the additional features of Quickr, like having the possibility of adding information into all levels of the hierarchy instead of just one, this could benefit your organization and users.
Figure 7-5 shows the various containers available to both DDM and Quickr along with the mapping possibilities.

Notice that Quickr has fewer containers than DDM. In reality, that is not true, because a fileroom is simply a category used to further classify a file cabinet and a binder category is a category used to further classify a binder. Quickr supports the concept of subcontainers (subrooms and subfolders), allowing you to mimic your DDM hierarchy in Quickr more closely.

The decisions on how to map containers from DDM to Quickr should not be taken lightly. In determining the most appropriate mapping strategy, consider both your security and your navigational requirements.

7.2 Security

As an application built on IBM Lotus Domino (well known of its unique, robust security infrastructure), DDM provides excellent security features. Lotus Quickr services for Domino uses the same infrastructure, and the security features provided by this product are similar. In this paragraph we will explain how they compare and how we can map security levels of both products to each other.

7.2.1 DDM security

In this section, we begin discussing the key aspects which define DDM security.

**User directory**

DDM supports two types of repositories for user identify information:

- Domino Directory
- LDAP Directory

Like most of the applications using external user directories, DDM does not provide any of its own user management functionality and does not modify directory content, except for creating and maintaining a set of user access groups when adding new libraries.
DDM security levels and roles

DDM gives administrators and users the ability to set access on the following levels, for the predefined roles listed in Table 7-2.

Table 7-2  Levels at which you can set security in DDM

<table>
<thead>
<tr>
<th>Level</th>
<th>Reader</th>
<th>Editor</th>
<th>Manager</th>
<th>Binder Creator</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Cabinet</td>
<td>View binders and read documents</td>
<td>View and potentially create binders if binder creator level is not used. Also allowed to view, create, and edit documents</td>
<td>View, create, edit, and delete binders as well as create, edit and read documents. Can also set security on the file cabinet</td>
<td>If used, has the same abilities as editor and is allowed to create binders</td>
</tr>
<tr>
<td>Binder</td>
<td>View binders and read documents.</td>
<td>View, edit, and optionally create binders. Also allowed to view, create, and edit documents</td>
<td>Create, edit, and delete binders as well as documents. Also allowed to change security settings for the binder</td>
<td>If used, has the same abilities as editor and is allowed to create binders</td>
</tr>
<tr>
<td>Document</td>
<td>Read documents</td>
<td>Read and edit documents</td>
<td>Read, edit, delete, and set security on documents.</td>
<td></td>
</tr>
</tbody>
</table>

Permissions can be inherited from the file cabinet and binder levels, or set explicitly on file cabinet, binder, and document levels.

DDM typically generates access control list groups in the user directory (Domino Directory or LDAP), for newly created libraries. Group membership is typically automatically maintained.

DDM also uses a Domino.Doc Site Administrators group in the Domino Directory. This group is used to store all global Domino.Doc administrators.

7.2.2 Quickr Security

Now that we have defined the key aspects of DDM security in the previous section, we discuss the model for Quickr for the sake of comparison.

User Directory

Lotus Quickr supports two different types of user repositories:

- Domino Directory
- LDAP Directory

Additional users can be added to a local directory within a place. LDAP can be connected directly and used as an LDAP, or used through Domino Directory Assistance. The first option is recommended if you want to benefit from using the real LDAP user repository.

Although Lotus Quickr allows administrators to create new users from its administration interface, this functionality is usually disabled, and user management is done on the Directory level (Lotus Domino or LDAP server).
Quickr Security Levels and Roles
Lotus Quickr allows administrators and users to set an access at the following levels, using the following predefined roles:

- System (global settings)
  - Quickr Administrator: Administers Quickr, allowed to create places
  - Place Creator: Allowed to create places

- Roles available at hierarchy levels

Table 7-3 illustrates a detailed view of the levels of security which can be set within Lotus Quickr.

<table>
<thead>
<tr>
<th>Level</th>
<th>Reader</th>
<th>Author</th>
<th>Editor</th>
<th>Manager</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place</td>
<td>Allowed to access place and read pages</td>
<td>Allowed to access place, read pages, create new pages, and edit their own pages</td>
<td>Allowed to access place, read, create, and modify pages.</td>
<td>Allowed to read, create, modify, and delete pages, create and delete rooms and sub-rooms, manage membership, set ACL on all levels and delete a place</td>
<td>Allowed to read, create, modify and delete pages, create and delete rooms and sub-rooms, add members, set ACL on all levels, delete a place</td>
</tr>
<tr>
<td>Room and sub-room</td>
<td>Allowed to access room or sub-room, and read all pages</td>
<td>Allowed to access room or sub-room, read pages, create new pages, and edit their own pages</td>
<td>Allowed to access room or sub-room, read pages, create and modify pages</td>
<td>Allowed to read, create, modify and delete pages, create sub-rooms, add members, set ACL on all levels and delete a room</td>
<td>Allowed to read, create, modify and delete pages, create and delete rooms and sub-rooms, add members, set ACL on all levels, delete a room</td>
</tr>
<tr>
<td>Folder and sub-folder (security functionality planned for next version of Lotus Quickr)</td>
<td>Allowed to access folder or sub-folder, read all pages</td>
<td>Allowed to access folder or sub-folder, read pages, create new pages, and edit their own pages</td>
<td>Allowed to access folder or sub-folder, read pages, create and modify pages and files</td>
<td>Allowed to read, create, modify and delete pages, create sub-folders, add members, set ACL on all levels and delete a Folder</td>
<td>Allowed to read, create, modify and delete pages, create and delete folders and sub-folders, add members, set ACL on all levels, delete a Folder</td>
</tr>
<tr>
<td>Document</td>
<td>Allowed to read pages</td>
<td>Allowed to read, modify, and delete page, restrict access to page.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.2.3 Security Comparison

The following sections address how to think about security level settings between DDM and Lotus Quickr.

**User directories**

When migrating from DDM to Lotus Quickr, or any other product, you will be moving secured data. Some of the elements that you migrate will have access control lists (ACLs) containing names of users and groups that are stored in the user directory set on DDM. Therefore, if you want to keep permissions, you have the following options:

- Set the same user directory on the target product (for example, Lotus Quickr).
- Re-create the user directory on a new user repository.
- Manipulate the DDM data during migration to match the target user directory content.

The Migration tool (described in Chapter 10, “Overview of the migration tool Lotus Quickr Migrator: Domino Document Manager Edition” on page 177) allows you to manipulate security entries during migration. For example, you write code to change the base distinguished name (that is, o=ITSO,c=US to o=IBM,c=US), or you can completely replace selected user or group distinguished names with the new ones.

Lotus Quickr supports the Domino Directory user registry that DDM uses. However, if you have never used LDAP in your organization, your migration to Lotus Quickr might provide a good opportunity to switch from the Domino Directory to LDAP. LDAP has benefits, such as the ability to integrate Lotus Quickr with other IBM collaboration and social software, as well as with third party applications that use LDAP as their user registry.

**Security levels and roles**

As already mentioned, DDM and Lotus Quickr both use the security infrastructure of the underlying Lotus Domino platform. Therefore, both products provide a similar set of security levels and roles. However, they cannot be mapped precisely one to one. In the next section, we provide an example of mapping roles from DDM to Quickr.

7.2.4 Security mapping

Due to differences between DDM and Lotus Quickr, we cannot map all security elements of one product to the other. You need to make decisions in cases where exactly corresponding elements aren’t available. Table 7-4 on page 100 shows a mapping example that you can use for your migration planning. Mapping is based on one possible migration scenario. More scenarios are described in Chapter 11, “Migration Tool: Operational scenarios” on page 187.
As for the ACL groups used by DDM, Lotus Quickr provides one group by default, QuickPlaceAdministratorsSUGroup, used for storing the names of global Lotus Quickr super user administrators.

When integrated with LDAP, Lotus Quickr allows you to set a global super user administrator, that has full access and management authority for all places hosted by the Quickr server. For more on how to configure Lotus Quickr security, refer to the Quickr product documentation, see the following Web page:

http://publib.boulder.ibm.com/infocenter/lqkrhelp/v8r0/index.jsp
7.3 Database structure

When migrating from DDM to Quickr, you need to understand how the on-server database structure works for both products. If you do not plan and manage your imported DDM environment correctly, you could have scalability or performance issues in your Quickr environment.

This IBM Redbooks publication does not describe the best practices for installing and setting up the basic Quickr servers and DAOS support. Refer to the Quickr installation guide and online Wikis for information. This chapter focuses on the specific issues that accompany a migration from DDM to Quickr from the point of view of database planning.

7.3.1 DDM database Structure

A DDM implementation changes and adds certain information, settings, and databases to a Domino server during installation and use of the application. The design elements and all the content of DDM are spread across a number of databases.

A typical DDM on-server database structure contains the following databases, as shown in Figure 7-6.

![Figure 7-6 Illustration of a typical DDM on-server database structure](image-url)
- Site Administration database
  The Site Administration database (ddadmin.nsf) is used to create or rename libraries on the server.

- Transaction database
  The transaction database (ddmtrans.nsf) is used in multi-server environments to ensure actions such as check out take place on a master copy of a given document and do not cause replication or save conflicts.

- Configuration database
  A separate configuration database (librarynameCfg.nsf) will be available for each library. This database holds document type and binder data definitions for the library and is meant for programmatic reference. Users do not interact with this database directly.

- Log database
  A log database (librarynameLog.nsf) will be available for each library. It holds logging information for that library.

- Library database
  Each library will have its own library database (librarynameLib.nsf) that holds configuration data, security data, and file cabinet definitions for the library.

- Binder database
  Each file cabinet within a library has a separate binder database that holds all binder definitions for that file cabinet (.NSF filename is based on a unique ID. The database name is the same as the file cabinet name).

- Document databases
  Each file cabinet can have one or more document databases. The number of databases is determined by the threshold settings specified at the file cabinet level. In general, most of these databases are located in the DDM directory (.NSF filename is based on unique ID. The database name is the same as the file cabinet name with a follow up number).

From an administration point of view the most important databases to look at when migrating are the databases that influence your security, database distribution and database sizes. Figure 7-7 on page 103 shows this in a graphic.
Figure 7-7  Databases which influence security settings

All data within DDM is stored in the lowest database level, the document databases. So totalling the size of these will give you an idea of how large your document storage is and how it is distributed over the different file cabinets.

**Spawning**

One of the stronger points of DDM is its ability to spawn the document databases to reduce individual database size. This ability was important when DDM was originally developed because individual database size was restricted to a maximum of four gigabytes.

Spawning was done automatically after it was set up by the file cabinet administrator. The settings for this can be found in the file cabinet administration document.

**Note:** To open this document go to the library database, select **Library Administration** and select the file cabinet that you are interested in (use the Lotus Notes client for this). In the file cabinet administrations settings, find the section called **Document Storage Options** that contains the thresholds on document databases.

To view how many document databases are in use, open the library database in a Notes client and double-click a file cabinet to open that cabinet's binder database. In the navigation ribbon on the left, click **File Cabinet Administration**. On the next window, click the option for **Binder Administration**. The view presented is categorized by replica ID of the document databases.
7.3.2 Lotus Quickr structure

A typical Lotus Quickr installation uses much more than databases. It consists of database files, templates, static html files, servlets, and other elements. For a full and extensive description of all design elements that are used in the Quickr setup, consult the Lotus wiki documentation at the following Web page:


Here we focus on the databases used within the Quickr design structure because they have the most affect on your server capacity planning when migrating from DDM to Quickr. Figure 7-9 on page 105 provides a conceptual illustration of databases which make up the Quickr structure.
Chapter 7. Hierarchical and structural considerations between Domino Document Manager and Quickr

A typical Place consists of the following databases:
- main.nsf: The main Library database holding the place characteristics
- contacts1.nsf: User or group authentication information
- search.nsf: Used for searches within the place
- pagelibrary<UNIQUEID>.nsf: contains the room characteristic. A separate database is created for each room.

Unlike DDM, where the library database acts as a portal and does not store any of the managed content, Quickr stores content in the Place database by default (main.nsf) and optionally in a series of room databases (pagelibrary<UNIQUEID>.nsf) depending on where in the Quickr hierarchy you store your data.

**DAOS**

Quickr does not allow for spawning like DDM. However, beginning with Quickr 8.2 (which uses Domino 8.5), Quickr provides an option of using DAOS to manage database size. The IBM Lotus Domino Attachment and Object Service (DAOS) reduces the total cost of ownership and database scalability by storing all file attachments in a separate repository on the server and retrieving them by reference. This means that the files are no longer within the database but stored separately. This limits the size of the database files considerably.
Another advantage of DAOS is that it will check if attachments are already stored, and if so, it will reuse the existing file instead of storing the same file twice.

Refer to the following Web page for more information about DAOS:

7.3.3 Database structure comparison

When migrating from DDM to Lotus Quickr, several things are important for the server administrator to keep in mind:

- Backend-database sizes
- Document spread per backend-database (number of documents per backend database and the total number of databases per server)
- Document spread per folder (number of documents in each folder)
- Server distribution

The choices that you make during migration might determine the flexibility and scalability of your server as well as the overall performance of your new Quickr environment.

Database size

In DDM, spawning is used to distribute large amounts of data over several databases. This is done transparently, meaning the user is not aware of it. Because of the limitations on database sizes in older version of Notes (for example, four GB per database), spawning was an absolute necessity. However, even with support for larger database sizes (up to 64 GB as of Lotus Domino 6.5.1), it is still advisable to limit the size of the DDM database to manage the response time when opening a database.

For Quickr, the recommended solution for storing attached files that represent large amounts of data is to use DAOS. DAOS reduces the database sizes and allows you to put more data into one database. We suggest you use DAOS for your Quickr environment when importing DDM data into it.

To get an idea of the amount of data in your current file cabinets, you can total the sizes of all the document databases per DDM file cabinet. This is not an exact or definite number because the Notes documents themselves are also included (as well as any design elements), but this gives you a good indication of the size. This number includes all stored versions and drafts of each document. If you are planning to migrate only the latest version of each document, the total amount of imported data could be significantly smaller.
By document spread in databases, we mean the number of documents located in any single backend database. This is separate from the size of the documents.

Limiting the number of documents per database helps to improve the overall performance of your Quickr environment. Putting too many documents into one single backend database can make opening that particular place or room slow. However, do not go overboard in creating multiple places to avoid too many documents in any one place. Having thousands of databases on a single server can also have a negative impact on performance.

Within Lotus Quickr, the following elements each represent a single backend databases:

- Place (including the place library and any folders created in the place directly)
- (Sub) Room (and any folders created in the room directly)

If you have a DDM library with hundreds of thousands or even millions of documents, it might be a good idea to split that up into several rooms or even several places. For instance, consider creating a place per file cabinet or per binder category depending on the document spread over the individual binders. On the other hand, if you have a library with only a few hundred of documents or a couple of thousand, then putting the whole library into a single place would most certainly be an option as well.

This does not mean that you cannot put larger numbers of documents into one place. It only requires that you use rooms and sub-rooms to divide the total number of documents over multiple databases if you exceed this. The advantage of doing this is that you can also include security at this level.

### Document spread in folders

A folder is not a separate database but included in the place or room (or sub-room) database it was created in. The number of documents within a folder counts directly towards the total number of documents for the database.

The main consideration when determining how many documents you should put in a single folder is based on considering the user experience. You can put hundreds of documents in a single folder. However, having large lists of documents is not something users find easy to work with. By default, a typical Quickr view will list from 10 to 100 documents on a single page (based on user preference). Having many documents in a single folder will require the user to scroll to navigate and will increase the page load time for loading the lists.

---

**Note:** Determine the file cabinet data size by getting the database size of each of the document databases. To locate the document databases for the file cabinet, go to the DDM directory on your DDM server and locate the databases that have the name of a file cabinet plus a number (the database with just a file cabinet name are the binder database and shouldn’t be included).

For each of the document databases, check the database size through the database properties.

To view how many document databases are in use, open the library database in a Notes client and double-click a file cabinet to open that cabinet's binder database. In the navigation ribbon on the left, click **File Cabinet Administration**. On the next window, click the option for **Binder Administration**. The view presented is categorized by replica ID of the document databases.
You can take a lead from Google here, too. When searching with Google, you can get millions of hits for your search query, but if you cannot find what you are looking for within the first two pages, you are most likely to search again or lose interest. Nobody goes through all thousands or millions of entry pages that the search produced.

A Quickr folder is, of course, not a search result, but the same principle still applies. If a user is looking for a particular document, he or she is not likely to scroll through pages and pages of listed documents to find the document they are looking for. The whole purpose of having a hierarchy structure (in addition to using it for security) is to lead your user through different levels of categorization to the specific document they are looking for without having to use the search functionality. If they reach the end of the hierarchy and still find hundreds of documents, they could lose interest or the need to use the search functionality anyway.

Use the folders to distribute the documents in such a way that the hierarchy supports a logical path to the user, as well as any security measurements that you require.

Do not nest too many folders. It is considered an annoyance for users. Windows OS (for example, usage of Quickr through the connector) limits the total file path to 254 characters, after which names are truncated and some programs might have problems accessing the files.

### 7.4 Document model

In terms of document management, both DDM and Lotus Quickr for Domino store files and associated data on the servers. To do so, both products use the Lotus Domino server and use NSF databases as a main storage. But, because DDM is a pure document management solution and Lotus Quickr is a collaboration platform, they store files and associated data in different ways.

#### 7.4.1 Domino.Doc document model

DDM treats an uploaded file as a managed document. It treats additional information entered in the document profile (properties) as meta-data associated with the document. Profiles are based on the document type subforms, containing specific fields appropriate to the document type. For example, the field Contract_Name is a field on the document type Contracts subform and contains a piece of information describing the actual managed document associated with that profile. DDM can also store OLE objects and Notes documents.

A DDM document profile supports the following field data types:

- Text
- Rich text
- Date & time
- Numeric
- Selection Lists (text)

A DDM field value can be computed, as well as generated programmatically. Profiles and fields are configurable. With DDM, you can create custom document types (for more on customizing DDM, see Chapter 12, “A Domino Document Manager to Quickr example using a custom placetype” on page 215.)
7.4.2 Lotus Quickr document model

Lotus Quickr is a Web collaboration platform, designed to be used as a team workplace. You can host team Web sites, build rooms (sub-sites), folders, and pages that contain rich text content and attachments. When you use Quickr as a document management system, each uploaded file is attached to a new page. The page is based on the simple Quickr upload form and contains the document title and attachment itself. You can enhance this default behavior by creating custom Quickr Forms, which can be used as an equivalent of the DDM document types. For more on how to create dedicated DDM custom forms on Quickr, refer to Chapter 12, “A Domino Document Manager to Quickr example using a custom placetype” on page 215.

Quickr Forms can contain a variety of data types including the following types:

- Text
- Rich text (including text formatting and images)
- Pick lists (single- and multi-valued)
- Attachment

You can enter values in Quickr fields manually, or you can generate or modify fields programmatically using placebots, written in LotusScript® or Java languages. When using HTML-based forms, Quickr document fields can be also computed by the client-side JavaScript™ code provided in the form. It is possible to implement a client-side database or dictionary lookups, to enhance the functionality even more.

7.4.3 Data types mapping

During a migration, the Migration Tool will create a custom Quickr Form for each DDM document type, then create a single page for each DDM managed document (a file with associated meta-data). Once migrated, each page will have a file attached and all meta-data placed in the custom fields, available in the custom Quickr Form. By default, custom forms created by the Migration Tool will provide simple text fields to store meta-data and an attachment field to store files.

It is possible to create a custom Quickr Forms, simple, or HTML-based, and use them for viewing and manipulating migrated DDM data. Simple forms equal simple DDM profiles, containing only few simple text fields. HTML-based forms should be considered when migrating DDM content that contains complex meta data fields where values or selection list components are calculated.

7.5 Document management functionality

As you plan your migration from DDM to Quickr, you should become familiar with the specific functionality surrounding document management and how they differ between the two products. You will need to understand and determine whether the capabilities in Quickr cover the requirements of your organization.

7.5.1 Document management features in DDM

In this next section, we discuss the document management features specific to DDM, to then use as a basis for comparison with the document management features provided by Quickr.
Versioning

In DDM, the versioning capability provides both versions and drafts. This is also referred to as major or minor versioning. A major version is typically identified by its version number being a whole number (for example: 1.0, 2.0, 3.0, 4.0) while a draft (minor) version would be an intermediate number (0.1, 0.2, 1.1, 1.2, 2.1). The check in and check out function may be used on both drafts and versions. You can also specify individual security settings for both.

By default, a version will get the full document security (usually based on the binder it is in or on the document security that was set on the previous full version). A draft will have limited access security allowing access only to users with editor, draft editor, or manager access to the document.

Check in and check out

Check in and check out is a mechanism that prevents different users from editing the same document at the same time. When opening a document, it opens in read mode by default. Saving an edited document back into the system as a new draft or version is allowed only if it was first checked out.

DDM allows for a distributed system with replica databases spread out over multiple servers. To ensure two people would not check out the same document at the same time, check out is allowed only on the master server that was specified for the file cabinet. This approach ensures the integrity of the check-in process. However, it can sometimes cause delays. Sometimes, users do not have direct access to the master server so the check-out request has to go through the transaction database on the master server before it is completed.

Life cycle management

DDM provides the capability for fairly complex approval and review options. On a document basis, users can specify the following elements:

- Who should review the document
- In which order the document should be reviewed
- If routing should happen parallel or serial
- After how many days a reviewer or approver will be reminded (by mail) if he or she does not respond.

With review cycle, a user can also set whether the reviewers can edit the original document or only add comments.

With approval cycles, the user has the option to specify what should happen with the document after it is approved by all approvers:

- Checked in as version.
- Remain a draft.

This gives the user optimal control over the process.

Administrators can enforce certain settings by specifying whether users can change approval and review settings per document type and whether they can dynamically determine who should do the review or approval.

Document naming

In DDM, a document is always referenced by the document title that is specified in the document profile. The filenames of any files are included in the document profile, but are not referenced in any of the views or enablers. Version numbers are also visible in all of the views, so that users can visually ensure they are working with the intended version.
Metadata
Within DDM, you can create document types that determine what metadata users enter when they are creating documents. Users are prompted for required and optional fields, depending on the document type that they choose. The creation of metadata fields is done through the use of subforms and requires basic Domino Designer skills.

Out-of-the-box DDM comes with a set of document types that can be used or customized to the needs of an organization.

7.5.2 Document management features in Quickr

In the previous section we described the document management capabilities with DDM. Now we discuss the document management capabilities provided by Lotus Quickr to compare the functionality and highlight key considerations between the two products.

Versioning
Versioning in Lotus Quickr works differently from DDM. Quickr allows for both versions and drafts, but a draft is automatically removed after it has been upgraded to a version. It is not possible to store multiple intermediate draft versions before going to the next version.

When a user saves a document as a draft in Quickr, the document is only accessible and visible to that particular user. It is not possible to authorize other users to view or edit a draft.

An administrator can determine per document type how check-in influences the document versioning. For instance a Quickr Form can be set as follows:

- Never allow versioning
- Allow versioning (user has the option of creating new versions or performing in place edits)
- Enforce versioning (a new version is automatically created each time the file is checked in).

Check in and check out
Check-in and check-out in Quickr works similar to DDM. A document is locked for editing to all users except the user that checked out the document.

Quickr provides the ability to create replicated environments over multiple clustered servers. No mechanism exists to ensure that two users on two different servers do not check out the same document. This could potentially result in save or replication conflicts or data being overwritten. If the DDM environment used the multi-server checkout functions, you might consider server consolidation or slightly modifying business rules as appropriate to accommodate the rules and functionality of Quickr (for example, placing reference or read-only copies of documents in one folder and editable copies in another).

Life cycle management
Like with DDM, you configure the built-in Quickr workflow and approval capabilities on a form-by-form basis. However, with Quickr, users do not have the ability to change workflow settings on a per document base.

The approval workflow allows you to specify up to four names of approvers. It is not dynamic, which means that you need to change the Quickr Form to change the workflow. An advantage, on the other hand, is that you do not need developer knowledge to change these settings. It can easily be done by a place manager or owner.

Quickr supports the types of workflows described in Table 7-5 on page 112.
Table 7-5  Workflows supported in Quickr

<table>
<thead>
<tr>
<th>Workflow</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple submit workflow</td>
<td>When you use the Simple submit option, the author has only one publication option: he or she can publish the content (by clicking a button with a name you specify) or cancel its publication (by clicking Cancel). The content appears in a folder you specify. Use this option when you do not want content created with the form to undergo review. When you do not want members who use the form to have the option of saving content in draft mode; and when you want the ability to rename the button members use to publish.</td>
</tr>
<tr>
<td>Editor-in-chief workflow</td>
<td>Use this workflow if you want a particular member to review each item created with the form.</td>
</tr>
<tr>
<td>Approval workflow</td>
<td>Use this workflow if you want more than one member to review each item created with the form. You set up the order in which the reviewers review the content.</td>
</tr>
<tr>
<td>Multiple editors workflow</td>
<td>Use this workflow to grant all members who have author privileges in the current place/room the ability to edit each item created with the form.</td>
</tr>
<tr>
<td>No special workflow</td>
<td>Choose this workflow if you want to allow members who create content with the form to publish immediately (in either final or draft form) without approval from any other member in your organization.</td>
</tr>
</tbody>
</table>

**Document naming**

In Lotus Quickr, you have the ability to view a list of your content either from the Web user interface or from the Quickr connectors (for example, Windows Explorer or a sidebar in your Notes client). Remember that in Quickr, every file is attached to a page. A page has a name, independent of the name of the attached file. And a page can have more than one file attached to it.

In the Quickr Web user interface, you see page names. You have to navigate to a page to see the actual files names of any attached files.

The Quickr connectors are file-oriented, rather than page-oriented. When you look at a Quickr Place through the connectors, you see the folder hierarchy with actual file names beneath the folder names. In the case where a page has more than one attached file, the page appears as a folder with the attached files beneath it.

**Metadata**

You can use a Quickr Form to specify custom data fields that become associated with the page and the attached files. Three types of forms are available:

- **Simple Form**
  - Create a form using standard Quickr fields such as plain or rich text, attachments, date pop-up, task information, and so forth.

- **Microsoft Office Form**
  - Use a Microsoft Word, Excel®, or PowerPoint document as a template for new content.

- **Imported HTML Form**
  - Create a form using your own HTML.
Simple Form is the easiest but also gives the least functionality. Place managers or owners can easily create simple forms themselves. Capabilities that you might have in DDM that are not possible with simple forms include dynamic choice lists or name selection lists.

Microsoft Office Forms can be handy if you already have forms in an external format (for example, Word, Excel, and so forth). Take, for instance, an organization that has some standard forms based on Word documents that each new employee needs to fill out. You can create a named Microsoft Office Form in Quickr for each of your existing standard Word forms. Each time a user creates a document in Quickr based on one of these forms, the user is presented with a blank version of the Word form to fill in. This option is handy if your organization already has a lot of stand-alone forms and does not have developer capacity.

Imported HTML forms are the most complex to build because you would need to know HTML, JavaScript, and CSS to benefit from all the options available. With HTML forms, you have significant flexibility in what you can create, including lookups from external data sources and name list selections (both of which are often used in DDM applications). When adding documents through the connectors, you are not able to choose an imported HTML form. Comparison of document management features can be reviewed in Table 7-6.

<table>
<thead>
<tr>
<th></th>
<th>DDM</th>
<th>Quickr</th>
</tr>
</thead>
</table>
| **Versioning**| Supports major and minor versions (including multiple shared drafts). As you plan your migration, make sure you give careful consideration to your business requirements around major and minor versions. | Supports major versions and a single active draft accessible by a single author.  
Note: Migration Tool will import all DDM drafts as Quickr versions. |
| Check-in / check-out | Functionally similar in both products. Centralized management in distributed environments to avoid conflicts. | Functionally similar in both products. No centralized management in replicated environments so conflicts are possible and need to be managed. |
| **Review / Approval** | - Supports both serial and parallel workflow.  
- No limit on maximum # of approvers.  
- Approver list can be programmatically generated.  
- Approval settings can be specified at the document level.  
- Can specify time limits and reminders.  
- Requires expertise to build workflow. | - Supports only serial workflow.  
- Maximum of 4 approvers.  
- Approver list must be specified ("hard coded") in the form associated with the document.  
- Approval settings can be specified only in the form.  
- No time limits and reminder support.  
- Workflow forms can by created by users / place managers without programming skills. |
| Document naming | - Always uses the document name (not the file name).  
- A document with more than one file attachment does not appear in the desktop enablers, only in Web user interface and the Notes client. | Uses the page name in the Web UI and the name of the attached file in the connectors.  
- If more than one file attached to a page, connectors show the page as a folder and the file names beneath it. |
| Metadata | - Comparable function available.  
- Domino Developers can create forms with complex metadata logic. | Comparable function available  
- users can create forms that provide simple metadata prompting.  
- Skills in HTML, CSS, and Javascript to develop forms with complex metadata logic. |
Other features
Aside from the standard document management functionality and advanced security, DDM provides a set of useful features that make it best-of-breed among document management solutions. Lotus Quickr is also equipped with several advanced features, making this collaboration product user-friendly and easy to customize.

Both products give users ability to search for resources, work off-line and archive data that is no longer used. Both can be integrated with other systems and both can be accessed through several interfaces, including native Windows applications, Web user interfaces, and APIs. In this section we describe key features of both products and explain which Quickr features match DDM features and which features will be missing after switching to Quickr.

7.5.3 DDM features

Searching
You can search for documents across multiple file cabinets within the library. Search returns results in a single view. Searching respects all access permissions on the DDM-managed content. Search is available from all DDM user interfaces, Desktop Enabler, Web and Notes Client, as well as from the applications integrated with DDM through DDM API.

Archiving
You can archive managed files to an external storage facility where they can be reached. The original document is replaced with a proxy document that contains the profile and security information. This allows retrieval of the document retrieved from the archive anytime.

Bookmarks
It is possible to make a virtual copy of a document placed in one binder, in another binder, by creating a document bookmark. Bookmarks can be created for any document in any binders.

Favorites
With Domino Document Manage, you can create favorite documents and binders for quick access from the library. The favorites list can be accessed from any DDM interface.

Collaboration tools
The DDM discussion forum feature gives users the ability to initiate a threaded discussion around a managed document. Visibility of topics and responses is limited to the user's access to the related document.

Integration
DDM and its desktop interface application, DDM Desktop Enabler, integrate with Microsoft Windows Explorer and the Office suite (using ODMA). The DDM server can also be integrated with IBM Lotus Sametime. This provides presence awareness (the ability to see the real-time online status of users who are associated with a document) and the ability to save Sametime chat transcripts directly to DDM.

You can also integrate DDM with other applications through DDM API. For more on this, see 7.6, “User Interfaces” on page 117.

Offline
The attache case allows you to work with documents offline. It is possible to pull documents from the library and work with them while disconnected from the network. Any document placed in the attache case is copied to the user workstation, so the user can work on it offline.
Multiple user interfaces
DDM allows users to access its resources in several ways, as follows:

- Notes Client
- Web browser
- ODMA-enabled applications (like Microsoft Office)
- Through custom applications and interfaces that use the DDM API.

For more information about interfaces, see 7.6, “User Interfaces” on page 117.

Agents
DDM is shipped with a set of Domino agents, responsible for basic document management tasks. You can also create and run custom agents, such as the following items:

- Archiving documents
- Notifying users of document events
- Generating statistics
- Transferring documents within a DDM library or outside DDM.

Events
DDM contains script libraries in the library and file cabinet database templates that allowed developers to insert custom code that executes just before or after most DDM events (for example, PostCheckIn, QueryDeleteDocument, QueryAddToBinder, and so forth)

Customizing
Customize DDM for integration with desktop applications, custom look and feel, external application integration, custom processing, and integration with third-party technology.

7.5.4 Lotus Quickr features

Searching
You can search across a single place or multiple places within the Quickr server, using a Web interface. Search returns results in a single view and respects all access permissions on the Quickr content. Cross-place (cross-library) searching requires setting up a Domino Domain Search and Domain Catalog domain indexing. It also possible to search for Quickr content from the Windows desktop. This is limited to the places added to locally-installed Quickr Connectors client, and uses standard Windows search functionality.

Archiving
You can archive places which are no longer used and consume disk space.

Collaboration tools
As a collaboration platform, Lotus Quickr provides number of collaboration tools, like group calendar, wikis, blogs and discussion forums.

Integration
Lotus Quickr and its desktop interface application (Quickr connectors) integrates with Lotus Symphony suite, Microsoft Office, Lotus Notes, Microsoft Outlook, Lotus Sametime, and Windows Explorer. You can integrated the Lotus Quickr server with Lotus Sametime to provide presence awareness anywhere in a Quickr Place where you see user names.

Lotus Quickr can be also integrated with other applications, using REST- or SOAP-based protocols.
Offline
You can use Domino off-line services (DOLS) to allow users to take whole places offline, work with the content through a Web browser, and synchronize the changes with the online version on the Quickr server.

Multiple user interfaces
Lotus Quickr allows users to access its resources in several ways:

- Web browser (Internet Explorer®, Firefox, and Safari)
- Desktop connectors (Lotus Symphony, Microsoft Office, Lotus Notes Client, Windows Explorer, Microsoft Outlook, and Lotus Sametime)
- Custom applications integrated with Quickr through REST- and SOAP-based protocols.

For more about interfaces, see 7.6, “User Interfaces” on page 117.

Agents
Lotus Quickr can run placebots (agents written in Java or LotusScript) to perform tasks used to access, process, and manage the data stored in Quickr Place (for example, to create a placebot that automatically exports some data from a Quickr document to external application whenever a new document is uploaded).

Events
Event handlers allow customizations that permit expanded logic for workflow automation, external system integration and advanced programmatic control. Information about Quickr event handlers and other customization interfaces can be found on the Lotus Quickr wiki.

Customizing
You can create custom Quickr PlaceTypes (a set of Notes databases which can be used as place templates). By creating a custom PlaceType, you can change the look and feel of the places and add custom functionality, based on placebots agents and custom client-side JavaScript. It is also possible to change a limited number of global theme elements of the Quickr.

7.5.5 Feature comparison between DDM and Lotus Quickr

Lotus Quickr is not a one-to-one replacement for DDM. You will not find an exact equivalent in Lotus Quickr for every DDM-specific feature. For example, one value feature in DDM is the ability to archive documents on the file system (which is not available in Quickr). However, the main reason for using this feature is to keep the size of Notes libraries and total number of databases as small as possible. Using Domino Attachment Object Store (DAOS) with Lotus Quickr meets a similar goal and more: It stores all files uploaded to the Quickr, on the file system, which keeps the size of the Quickr databases small.

Another feature that will be missing after migrating to Lotus Quickr will be the ability to create bookmarks (virtual document links) that can be placed in any binder. Lotus Quickr has the ability to insert links into pages in folders, but it will not feel like creating a bookmark to a document to a user accustomed to the other process.

Because the migration tool will not migrate DDM agents or any custom DLL libraries, all custom functionality provided by them needs to be re-created on the Quickr by developing placebots. At the time this book went to press, additional customization options that compare to DDM functionality were under consideration for the next release of Lotus Quickr in 2010.
Cross-place search functionality in Lotus Quickr is not available out of the box. It needs to be manually configured after deploying the product. Also, although it is possible to search for the Quickr documents from the user's Windows desktop, it is limited to the content available in the places added to user's local Quickr Connectors client, and it is available from the Windows search feature, not directly in the desktop client.

One of many Quickr advantages over DDM is its ability to integrate this collaboration platform with other applications, like WebSphere Portal, Lotus Connections, Lotus Sametime, and many non-IBM products.

DAOS support allows Quickr to handle hundred thousands of documents but keep the sizes of the Notes databases small, comparing to large DDM libraries, spanned across multiple Notes databases.

Lotus Quickr is a relatively new product that continues to be enhanced. As of the printing of this book, a new release of Lotus Quickr is being planned for the first half of 2010. It will likely include several capabilities that are particularly important to customers planning migration from DDM:

- Folder-level security
- User-friendly prompting of metatag data from the Quickr connectors
- Enhanced options for customizing Quickr Places

User interfaces are compared in 7.6, “User Interfaces” on page 117.

7.6 User Interfaces

DDM and Lotus Quickr can be accessed with a range of clients. Both provide Web access, for regular and mobile browsers, like Firefox, Opera, or Internet Explorer. Both are shipped with native Windows32 desktop clients, able to be integrated with Windows Explorer and selected Windows applications, and both provide access through APIs. In addition, DDM can be also accessed by users through the Notes client, just like any other Domino application.

7.6.1 DDM Interfaces

**Web**
Web access is the primary DDM user interface. It allows users to access DDM libraries through HTTP and HTTPS protocols, using popular browsers.

**Desktop**
DDM Desktop Enabler is an optional user interface. It integrates with Microsoft Windows Explorer, Microsoft Office, and other ODMA-compliant applications. It gives users fast access to DDM libraries from the applications where they're most likely working with content.

**Notes clients**
DDM is a pure Domino application, so it can be fully accessed by the users through the Notes client. The Notes client also serves as the main DDM administration interface. Because DDM runs on the Domino Server, Lotus Domino Administrator and Domino Server Console are used for the Domino administration.
API
DDM supports integration with other applications and systems through OLE Automation API. It allows full programmatic access to the DDM storage and functionality.

7.6.2 Quickr Interfaces

Web
Web access is the main Quickr user interface. Modern design and Web 2.0 technologies make this product friendly and intuitive. It gives users full access to the Quickr content and functionality. Web interface is also used for basic administration tasks.

Desktop
Lotus Quickr Connectors is a universal Quickr user access application. It integrates with Microsoft Windows Explorer, Windows Context Menu, Microsoft Office suite, Microsoft Outlook, Lotus Symphony, Lotus Sametime, and Lotus Notes clients. It allows full access to the Quickr-managed documents and limited functionality, like check in and versioning. Lotus Notes integration lets you store messages and attachments in a Quickr Place.

Notes clients
Lotus Domino Administrator and Domino server console are used for advanced Quickr administration tasks. Both are also used for administration of the underlying Domino Server.

API
Lotus Quickr can integrate with enterprise content management systems, such as IBM Content Manager and IBM FileNet, as well as other applications, through REST and SOAP technologies.

7.6.3 Interface Comparison

Both products reply on a Web interface for the primary user experience with the most robust set of features and a Windows add-on component to facilitate quick access to limited elements of each system. DDM also had a Notes client interface, but given the relatively small percentage of users that relied on it, and the additional abilities of the Quickr Web interface to work with multiple attachments and limited rich text, this is not thought to be a material issue.

The interfaces have an obvious difference in their approach to accessing the content they manage. DDM focuses on the user experience around documents (for example, a home page that allows users to create links to favorite documents and link to recently edited documents; flexible workflow options with process initiator retaining complete control; and robust event model for programmatic enhancements).

Quickr Places primary focus on collaboration and secondary focus on document management functions. Instead of maintaining an internal list of user favorites, users can use the regular bookmarking features within their browsers. Folders allow users to store content at any level (not just the lowest level) and thus create taxonomies that can sometimes better align with business processes. Like in DDM, there are multiple sort options that can assist users when working with large collections of documents in a single container.

One of the significant functional differences between the interfaces is the differing uses of titles and filenames. In DDM, binder tables of contents always displayed the value in the title field stored on the document profiles. Documents containing rich text or multiple file attachments were not visible in the Desktop Enabler. In Quickr, the title item is displayed when users view folder content through the Web browser, but when looking at folder contents using
the Quickr Connectors, filenames are displayed. This is an important difference to understand and an issue whose impact should be carefully analyzed before performing a migration. As part of the migration, the provided utility can assist with updating either filenames or document titles as appropriate to better ensure a consistent user experience. Making this change post-migration is substantially more difficult.

In Windows integrated clients, both the DDM Enabler and Quickr Connectors provide a subset of the features available in their Web counterparts. Most notably, the Quickr Connectors allow for fast creation of new folders and the drag functionality for moving documents and manipulating the taxonomy that DDM has been missing. DDM users should be aware of some significant functional differences, such as the inability to select different forms (or in DDM terms, document types) when uploading a file. By default, Quickr uses a standard upload form. Meta data is not easily associated with documents nor can it be displayed to users as of the time of this writing. This is another aspect of the environment that deserves careful consideration and understanding prior to attempting a migration.

There is no search available from the Quickr Connectors. Using Windows search functionality, users can only search places added to the locally installed Connectors application. Also, cross-place searching on the Web interface is not available from the main banner search field by default. This can be changed only by customizing Quickr (development).

Compared to the old look and feel of all DDM user interfaces, Lotus Quickr services for Domino provides modern, fast, and ergonomic clients, blending with other applications with which Quickr can integrate. Because this product has been designed mainly as a collaboration platform, it can be used not only to store documents and files, but also to host wikis, blogs and discussion forums.

Quickr APIs also allow the building of custom Quickr Connector clients, similar to custom DDM GUI clients.

### 7.7 Conclusion

This chapter has described many of the fundamental and architectural level differences between DDM and Lotus Quickr and hopefully helped you to understand better how to think about the two in terms of features and benefits.

Chapter 8, “Design element review” on page 123 looks at specific design elements in DDM, and discusses how to think about these within the context of Lotus Quickr.
Part 4

Digging into Customized Domino Document Manager Implementations
Design element review

This chapter serves as a design element review and discusses many of the customizations developers have made to their Domino Document Manager (DDM) environments. It also addresses how similar functionality might be developed in Lotus Quickr Services for Domino (Quickr) where it does not already exist natively. The intended audience includes managers, business analysts, and developers looking to gain a better understanding of the feature differences between the two products from both a user and technical perspective. As such, the chapter is divided into multiple sections. The first portion of each section is meant to give a general overview of the feature differences and how to begin thinking about them. The latter half of each section provides more technical information about where in the DDM templates developers might look for customizations and the skills and effort that may be required to create similar functionality in Quickr.
8.1 Reviewing your implementations

This section is an element review. It discusses how to locate customizations for reuse and documentation.

8.1.1 Site administration database

To create a library, the individual that set up your DDM system would have used the site administration database (typically domdoc\ddadmin.nsf). Users with a high level of access to the server click Create Library, complete a short profile, and click Create, which creates a library, log, and configuration database for the new library. An example of this is shown in Figure 8-1. This database is not likely to be customized in small to moderately complex DDM environments that only have one or even a few libraries. There is no meaningful data here that is unique or important to the migration process.

This database should be examined to develop a list of libraries that need to be reviewed for potential migration to Lotus Quickr. However, this list should not be considered comprehensive, as this database may have been replaced over time due to reinstallations, corruption, upgrade problems, and so forth. In addition, look at the DDM server for any additional databases with the last seven characters lib.nsf to develop a comprehensive list of libraries on the system.

![Figure 8-1 Site administration database](image)

Customizations to this database are considered rare and most likely to be discovered during the business process review than as part of a technical review (for example, the automation of new library creation).

8.1.2 Library

The library database acts as a portal to access various elements of the DDM system. Each DDM library will have a single database, typically ending in lib.nsf (for example, ITSo1ib.nsf). The library database does not contain any of the content data managed by DDM. Rather, it holds a series of pointers to file cabinets containing data as well as user experience information such as bookmarks, favorites, recently checked out items, and so forth. See Figure 8-2 on page 125.
Typical customizations are changes to the user interface (logos, colors, labels), adding links to other applications, and search.

Business analysis will reveal the most important changes to this database, what functionality users have come to rely upon, and what customizations may need to be made to the new Lotus Quickr environment to create feature parity for critical items. One of the core differences between DDM and Quickr involves how users relate to the managed data. The core DDM product focused on DDM being a key reference point for user access to a vast array of data across the enterprise. It included a portal interface that allowed users to create their own list of favorite documents and binders they frequently reference. Additionally, it included a list of items currently checked out to them and documents in a review and approval workflow awaiting their action.

Quickr provides more focus on allowing teams to collaborate around smaller collections of content and less on helping users aggregate data from the larger collection. As such, the lists of DDM favorites, recently edited document history, checked out items, and life cycle status are not migrated. In most cases, this is not expected to cause significant hardship, but dependencies on these features should be examined prior to the migration.
For technical customization, it is hoped that your developers and consultants have left a well-organized list of elements they have customized over time for your organization. However, if you have no such list and need to make your comparison manually, consider changes that may have been made to the following elements:

- **LibEvents Script library**
  Look through the objects in this library for custom calls that might reference additional script libraries or outside resources.

- **System Profile form**
  A central form used to create the system profile document. This was a common place to store switches or other bits of information that might control feature customization. Look for any additional fields that may have been added.

- **Agents**
  Custom agents were often added to facilitate any number of functions from automatically checking documents back in after a period of time to review cycle reminders and countless others.

When looking at the design, it would be helpful to have a copy of an unmodified version of the DDM templates against which to compare. Reference copies of those databases are available for download with the rest of the content in this IBM Redbooks publication. See Appendix E, “Additional material” on page 289 for instructions on how to download this template.

When looking at a list of design elements, sorting the list by last modified date may be helpful in identifying some elements that have been changed. See Figure 8-3. This is far from a fool-proof method, as actions like signing a database or recompiling all Lotus script will reset all dates to the same value. But if you have information about what changes you are looking for, it can be helpful.

![Figure 8-3 Sorting the list by last modified date](image)

### 8.1.3 File cabinets

File cabinets contain the data being managed by DDM. A file cabinet is composed of at least two databases (a binder database and at least one document database) both created from the same template. By default, file cabinet databases inherit their design from the DominoDocFileCabinet template with the filename `filecab.ntf`. However, unlike a library
which typically inherits design from a single template (domdoc.ntf by default), there can be many file cabinet templates used in a single library. Each template would likely have its own set of customizations.

To determine what templates are in use, look at the entry in the "Inherit design from master template" field in the “Database” dialog box for each file cabinet. See Figure 8-4.

![Figure 8-4  Design template name](image)

If you have a large DDM system and the server's catalog database has been kept current, you may find it faster to check the template inheritance on the catalog document entries. See Figure 8-5.

![Figure 8-5  Database catalog document entries](image)
Figure 8-6 illustrates the inheritance from the template name.

 DATABASE CATALOG - Database Entry

<table>
<thead>
<tr>
<th>General Database Information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Database title:</td>
<td>Philippines</td>
</tr>
<tr>
<td>Database server:</td>
<td>DDQUICKR03Lotus</td>
</tr>
<tr>
<td>Database filename:</td>
<td>domdocLADR-7QAP6X.nsf</td>
</tr>
<tr>
<td>Database type:</td>
<td>0</td>
</tr>
<tr>
<td>Reply ID:</td>
<td>8825757E006327E9</td>
</tr>
<tr>
<td>Database Admin Server:</td>
<td></td>
</tr>
<tr>
<td>Database size:</td>
<td>19,136,512</td>
</tr>
<tr>
<td>Database percent used:</td>
<td>84</td>
</tr>
<tr>
<td>Database creation date:</td>
<td>03/19/2009 02:03 PM</td>
</tr>
<tr>
<td>Database last modified date:</td>
<td>09/01/2009 07:08 AM</td>
</tr>
<tr>
<td>Link to database Policy document:</td>
<td></td>
</tr>
<tr>
<td>Number of documents:</td>
<td>18</td>
</tr>
<tr>
<td>Number of design documents</td>
<td>573</td>
</tr>
<tr>
<td>Database Design Template Information</td>
<td></td>
</tr>
<tr>
<td>Template to inherit design:</td>
<td>DominoDocFileCabinet</td>
</tr>
<tr>
<td>from:</td>
<td></td>
</tr>
<tr>
<td>Design last modified date:</td>
<td>09/10/2009 04:01 AM</td>
</tr>
</tbody>
</table>

Figure 8-6  Inheritance from the template name

8.1.4 Binders

Binders are containers created within a file cabinets. Assuming a default implementation where a file cabinet is represented by a binder database (for example, Philippines in Figure 8-4 on page 127) and a single document database (for example, Philippines 1 in Figure 8-4 on page 127), a binder will be represented by a profile document in the first database and a collection of documents in the second. If Notes Folders is selected on the file cabinet profile in the library database, a folder design element will also exist in the document database (for example, Philippines 1) and contain the collection of documents in the binder. If ActiveX® is selected, notes folders are not created.

In Quickr, a folder is represented by a profile document which can be seen in the System\Folders view, as well as a corresponding Notes folder design element. If attempting customization in this area, it will be helpful to know that these two elements are linked together through the h_FolderDocument item which is added to the folder design note and contains the universal id of the corresponding folder profile.

There were a number of customization opportunities for binders within DDM. Custom metadata could be added to the binder profile documents:

- The BinderCategory field typically held text strings that controlled the order in which binders were displayed. This field was natively available and often programmatically manipulated to determine where the binder appeared in the table of contents.

Folder levels in Quickr are controlled using the h_Ancestry item, which instead of containing a list of names delimited with back slashes (as in BinderCategory), contains a tilde-delimited (~) list of UNIDS that represent folders higher up in the structure.
 Binders in DDM had the ability to appear under multiple categories within the same cabinet. By default, the BinderCategory field was able to hold multiple values and display the same container under multiple levels within the table of contents.

This feature is not currently available in Quickr. Because folders contain an explicit list of their ancestry levels, a given folder may only appear once in the hierarchy. For it to appear multiple times, the folder and its contents would have to be duplicated within the Quickr system.

Custom binder types were subforms that could be created in the file cabinet template, enabled in the library database, and applied to binder profiles as a way to collect additional metadata. (for example, a binder might have a title "Expense Reports" and the binder type might contain metadata fields to collect which department and period these reports were for.

Quickr does not allow for the collection and maintenance of additional data on folder profiles. However the most uses of such fields involved attempts to work around customer-perceived limitations of how DDM presented binders in the table of contents. Because Quickr is more flexible in terms of creating taxonomy layers, this is not expected to be a material issue for most customers.

The presentation of the list of binders as well as the way individual binders presented their lists of documents were also able to be customized. See 8.1.5, “View customizations” on page 129 for more information.

### 8.1.5 View customizations

Views were commonly customized in DDM to better facilitate workflow needs, provide additional context, add additional functionality and more. Quickr offers the ability to customize the look and feel of the environment through place types. While working with place types requires more technical knowledge than editing the formula in a DDM view column, they also offer substantially more options. There is a substantial amount of information available regarding how to use place types with custom html, css, Javascript, and other Web-oriented code elements to create user interfaces that are better suited to allow for advanced customization than was available in DDM.

In addition, many of the common customizations DDM developers made to views are natively available within the standard Quickr Place type. See Figure 8-7.

**Figure 8-7  Customized table of contents in Quickr**

- Quickr provides a link to download attachments from the list of documents contained in a folder. DDM was often customized to have this feature so users did not have to open the document profile and select View from the Document menu.
- Custom sorts based on created, last updated by, author, and type.
- Functions to take action on the document from the view such as checkout, create response, notify, and more.
- Documents can be printed, copied, moved by using check boxes, and natively available actions without opening the individual profiles.
In determining what modifications may have been made to your DDM system, consider checking the following features:

- **Binder type definition documents**
  Within the library database, access the view of binder types and look at the individual documents to see if they are set to inherit their design from a specific view. If so, look for that view in the file cabinet template as it is likely to have been customized from the default. Compare it to the BinderTOC view in an unmodified file cabinet template to find changes.

- **BinderTOCTemplate**
  This is the view that is used as the template for creating Notes folders and was often customized in cabinets using the Notes folders table of contents presentation style to provide global customizations. Look for additional columns added to the left of the “Title” column.

Other customizations may have been made to the Binder Task Bar menu that allow actions to be taken on the entire set of documents within a binder, or perhaps on selected documents if a check box selection column was added in a view customization. See Figure 8-8.

---

**Figure 8-8  Binder Task Bar menu customization**

Code for these customizations is typically referenced on the subform WebTaskBarBinder. Typically, menu items were added as fields toward the bottom of the form. Code within the fields would define labels within the menu and the action to be performed, often calling an agent or custom Javascript. The graphic below represents the area where such fields would be located on the WebTaskBarBinder subform. The field “iiuiCustomAction” would represent a custom feature added to the bar. See Figure 8-9.

---

**Figure 8-9  BinderTask Bar subform**
### 8.1.6 Document types

Document types are one of the core elements that was meant to be customized with the DDM environment and have been customized by almost every DDM customer. The list of active document types is visible in the library database by accessing library administration document types from the blue navigation column. See Figure 8-10.

![Figure 8-10 Document types view](image)

<table>
<thead>
<tr>
<th>Document Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>- No Type</td>
</tr>
<tr>
<td>- Application</td>
</tr>
<tr>
<td>- Archive Sample</td>
</tr>
<tr>
<td>- Asset Valuation</td>
</tr>
<tr>
<td>- Client Correspondence</td>
</tr>
<tr>
<td>- Contract</td>
</tr>
<tr>
<td>- Correspondence</td>
</tr>
<tr>
<td>- Deposition</td>
</tr>
<tr>
<td>- Design Document</td>
</tr>
<tr>
<td>- DWF Archive Cover</td>
</tr>
<tr>
<td>- DWF Archive Document</td>
</tr>
<tr>
<td>- DWF Client Correspondence</td>
</tr>
<tr>
<td>- DWF Expense Report</td>
</tr>
<tr>
<td>- Employment Verification</td>
</tr>
<tr>
<td>- Generic Document</td>
</tr>
<tr>
<td>- Letter of Intent</td>
</tr>
<tr>
<td>- Loan Document</td>
</tr>
<tr>
<td>- Market Research</td>
</tr>
<tr>
<td>- Marketing Plan</td>
</tr>
<tr>
<td>- Product Information</td>
</tr>
<tr>
<td>- Proposal</td>
</tr>
<tr>
<td>- Request for Proposal</td>
</tr>
<tr>
<td>- Review and Approval Sample</td>
</tr>
<tr>
<td>- Standard Document - Dublin Core</td>
</tr>
</tbody>
</table>

Figure 8-10  Document types view
Each entry in this list represents a DDM document type and should correspond to a subform in a file cabinet template. Simpler systems likely use a single file cabinet template while systems of moderate size, where different cabinets were likely to have different workflow functionality, user interface elements, or appearance are more likely to use multiple templates. Besides the techniques already mentioned, another way to determine this would be to open each of the documents in this list and look at the value in the File Cabinet Template field. See Figure 8-11.

![Figure 8-11](image)

This is not a fool-proof test as some document types may have been deleted from this view when they were no longer appropriate to use for new content being created. However, this technique should be helpful in determining what templates might be in use.

Each document type will correspond to a single subform in the listed template. Typically, these subforms are quite simple with a few text, data, number, and selection list fields. However, these subforms can also be quite complex with default values being looked up from a Notes database, or external sources. Adding to the complexity, these subforms can also contain computed list values screened according to where in the taxonomy a document is being created, and so forth. Business analysis should be performed to determine how each of these document types are being used prior to the migration.

Quickr uses the term “form” in a similar fashion to how DDM refers to document types. There are some significant differences that should be noted:

- Document types were created as subforms using the Domino Designer client. Quickr Forms are created through the Quickr Web user interface. As such, there are more limitations on the functionality that can be implemented through simple forms. If more complex functionality is required, custom place types should be considered. Refer to Quickr product documentation and the Quickr support Wiki for more information about creating place types.

- The document type subform was created once in a template that could then be used to create as many cabinets as needed. In Quickr, forms are unique to a given place database (main.nsf) or room database. This means that a form added in a place is not available to rooms created within that place. Rather, the form element must be recreated in each room where you would like it to appear.

- Document type subforms can be modified as needed over time. In Quickr, once a form is created, it is not easily modified and changes passed to documents using that form. In addition, because forms are independent in each main.nsf and room database, if a form was created in five rooms, the same change would need to be made in all five rooms. (This paragraph is limiting commentary to the native Quickr features of Simple Forms.)
Quickr offers the ability to create HTML forms. These allow for improved design options over their DDM counterparts and allow for the creation of an environment that better fits a corporate or project theme.

Like DDM, Quickr Forms control basic revision and workflow options, configured when the form is created within the place or room.

In terms of skills, the majority of DDM customizations were achieved with LotusScript, Formula language, and a bit of HTML. Comparable customizations within the Quickr environment are likely to require knowledge of HTML, and Javascript for front end work and LotusScript or Java for the back end routines.

### 8.1.7 Documents

Documents are typically modified through custom agents, additions to the DocEvents script library, or additions to the document form. This section will discuss some of the fundamental changes DDM customers have been known to make to the document form and the DocEvents script library. Modifications such as document type subforms, security, and custom agents are covered in separate sections.

Two forms were typically used to manipulate DDM documents:

- **DocumentNotes**, for interacting through the Notes client
- **DocumentWeb**, for interactions through Web browsers

If a custom field were to be added, it would typically be added to both forms to provide continuity between the two interfaces (though this choice was left up to the individual developer performing the work).

Common customizations include the following items:

- **Adding additional metadata fields to appear on the upper portion of the form**

  Similar in effect to global subforms, adding fields directly to the form placed them near the top of the page next to fields such as “Title,” “Filename,” and “Description.” These fields would appear on every document created in a file cabinet associated with the custom template. Quickr does not have a similar global element that can be changed and inherited to a select portion of a place. However, similar functionality can be achieved using customized place types.

- **Inclusion of additional subforms**

  Much like the custom fields above, when it was desired to apply a grouping of fields to all documents in cabinets associated with a given template, an additional subform could be added to the DocumentNotes/Web form. This was a key point of integration for some external systems, such as records management that might require additional input for each document submitted.

- **Better defined breadcrumb trail**

  The DDM breadcrumb trail was only able to depict the physical layers containing a given document (library, cabinet, and binder). However, material context was often provided by the layers of binder category information that contained a binder. Quickr does not use the same style of virtual layers DDM uses for rooms and categories. All layers are physical. As such, a full folder path is available to users.

- **Bookmarks**

  DDM allowed for the same document to appear to exist in multiple binders within the same file cabinet and some customers customized their system to allow bookmarking across file cabinets, thus allowing multiple paths of access to access a single data element. Quickr
does not have parallel functionality. To provide multiple access paths for a given
document, the document must be stored multiple times in the system or a workaround will
need to be implemented.

8.1.8 Script Libraries

DDM allowed the creation of custom script libraries. This was one of the standards that
helped centralized code and made it easier to port customizations to additional file cabinet
templates or new templates during system upgrades. In addition, DDM provided a script
library, called DocEvents, which served as a launch point for custom code based on a series
of user triggered events. See Figure 8-12.

When looking for customizations to a DDM system, click through each of the events listed in
this library. Some calls are included by default—especially in the 6.5.1 and 7.0 releases of
DDM, however most custom calls should be easily identifiable as in the example above. Also,
be sure to look in the Options event for "Use" statements that might use custom script
libraries.

DocEvents was a key customization point for DDM developers. With intercepts for events
such as QueryCheckOut, and PostCheckIn, it was an easy place to call code that provided
enhanced functionality (for example, writing names of users accessing documents to a
special log, notifying a select group of users that a document has been changed/added, and
governing the deletion of documents based on business rules rather than simple user access
to the data). Quickr does not yet have parallel functionality. The current release supports
some similar events using the QuickPlace® C++ API and additional functionality through
REST services, but the skills required and ease of implementation will make duplicating these
types of customizations more difficult.

Additional programmatic intercepts for LotusScript and JavaScript are expected in the
upcoming releases of Quickr.
8.1.9 Custom agents

Custom agents were another common place developers added code to enhance DDM functionality. Scheduled agents may look for new or changed documents and send notifications or advance a document to the next stage in a workflow. Other scheduled agents might look for documents that have not been accessed in a few years and e-mail a request that this content be deleted. Agents may be invoked when a user clicks a button in the Web user interface and launch code to download automatically the source file from a document or upload a view file. In short, agents could be written to accomplish just about any data verification or manipulation organizations needed to perform on their DDM data.

Agents were also one of the known performance killers of DDM. Because each file cabinet has its own set of scheduled agents, a large system with hundreds of cabinets typically required multiple instances of the Agent Manager running on the server to keep up with the day-to-day burden. Custom agents added to this backlog and further degraded system performance, sometimes noticeably.

DDM developers are likely to be surprised the first time they open the Quickr templates and discover there are no agents in Quickr databases by default. In contrast to DDM, where a set of file cabinet databases contain the complete set of design elements needed to present and manipulate content as well as the data being managed, Quickr databases contain little design overhead. Most design information is centralized elsewhere on the server, leaving Quickr databases to contain almost all user created data. Compared with the release 7.0 DDM library template that had grown to a massive 65 MB, a new Quickr Place is less than two MB.

Custom agents can be written in Quickr and are still useful in updating document status flags, normalizing meta data, moving documents between places, and more. Agents can be written in either LotusScript or Java, and are often referred to as placebots. See the Quickr documentation for information about how to manipulate individual elements of the Quickr system.

As noted in 8.1.6, “Document types” on page 131, Quickr does not have the same type of perpetual template relationship that DDM maintains between a template and containers created from that template. Once a place or room is created, changes to templates are not automatically passed to that place or room. If attempting a moderate-sized migration with code elements (such as agents which are likely to be updated over time), it is suggested that you take the time to understand the relationship between templates and containers.
Chapter 9. Analyzing customizations

This chapter describes possible ways discover and analyze the customizations made to your Domino Document Manager (DDM) environment. It also provides step-by-step instructions on how to use DXLMagic to discover and document changes to your databases compared to a baseline database you choose.

We will discuss the following topics:

- Tools to compare code differences
- Which code changes are significant?
- Step-by-step instructions how to setup and use DXLMagic
- How to document and create reports of the changes
9.1 Overview

We discussed the different types of customizations that might be made to DDM in Chapter 8, “Design element review” on page 123. Often these customizations are not fully documented. Comparing your database to a baseline (the original template shipped with the product) will allow you to get a complete view of the changes that have been made.

Domino stores design information in the binary Note format, so you either need a specialized tool that can compare the Note format or you need to export the Note into something that can be understood by a general purpose compare tool. Domino allows you export the design as XML documents for inspection.

Script code comparison can be done with a regular comparison tool. You might choose to document script code as flow charts or using JavaDoc style libraries.

9.2 Visual comparison tools

Initially, a fully visual comparison tool might yield the best results when looking for the difference between the original IBM templates and the templates or databases that are in use in your environment.

However, a full analysis might provide more data than is needed to identify and list the design elements that require action and attention. Nevertheless, once these elements are identified, a visual tool becomes invaluable.

9.2.1 Binary comparison tools

The first class of comparison tools are those that read the note format directly. They can provide every detail of change on a C-Record level. This generates a large amount of information that may not be needed.

Teamstudio Delta

Currently the only tool allowing a full comparison of the binary design elements is Teamstudio Delta. It displays differences nicely as a tree structure clearly indicating what has changed, has been removed and what was added. For more information about Teamstudio Delta, visit the following Web page:

http://www.teamstudio.com/

For finding final details it is an invaluable tool. However, it provides more details than needed. For example, we do not need a changed data display when only signatures have changed, because it is common practice in Notes installations to re-sign all templates.

Figure 9-1 on page 139 illustrates a typical report from Teamstudio Delta, illustrating what has been changed in the file cabinet at the design level.
Figure 9-1  Sample report of changes from Teamstudio Delta (Courtesy of Teamstudio, Inc.)
Figure 9-2 illustrates how using Teamstudio Delta, you can navigate directly to the modified design elements.

Notes Peek

Notes Peek was available for R4-R6 but has never been released for R7.x or R8.x. While it does not allow automated comparison, it does allow you to run two instances of the program next to each other on the window and compare the results manually.

Notes Peek is available from the following Web page:

DXL Peek/DXL Explorer

Because Notes Peek has not been released for R7 or R8, Keith Smilli created DXLPeek as plug-in for the Notes R8 client. To avoid a name conflict with another OpenNTF project the tool was named DXL Explorer. The tool is unique because it runs as a plug-in the Notes client, Domino Designer is not required.

DXL Explorer is limited by the current DXL fidelity in what it can show.

DXL Explorer is available for download at the following Web page:
9.2.2 XML comparison tools

Tools more likely to be of use to compare two versions of a Notes database design are XML-based. You can export the relevant design elements (for example, all of them) and use a XML comparison tool to see the differences. The export can happen using a small script, the Tools menu in Domino Designer or a tool like DXL Magic.

**Alphaworks XMLDiff**

One of the oldest available tools (published in 1999) is the IBM Alphaworks XMLDiff and merge tool. It is available on the IBM Alphawork Web site:

http://www.alphaworks.ibm.com/tech/xmldiffmerge

It works best when comparing distinct design elements rather than an entire design. You would typically use it after you have identified the changed design elements for a closer inspection.

**Diff Tools in XML Editors**

Most XML Editors have XML Compare Functionality. This includes the following tools:

- XML Spy provides DiffDog
  
  http://www.xmlspy.com/

- oXygen XML () has a compare module
  
  http://www.oxygenxml.com/

- Liquid XML Studio ()
  
  http://www.liquid-technologies.com/

- Editix ()
  
  http://www.editix.com/

- Stylus Studio ()
  
  http://www.stylusstudio.com/

For the sole purpose of comparing Domino Designs, the investment into an XML editor might not be justified. However, XML technologies are a convenient tool for data manipulation, so you might need one anyhow.

What these tools have in common is that they all need DXL files to work with. These files are most easily created with DXLMagic.
9.3 Changes to look for

Looking at the schema for Domino reveals a huge amount of information. For example, in Figure 9-3, we see an example of a Domino Schema. Not all of it is relevant. One must look for the essential changes.

![Diagram of Domino Schema](image)

*Figure 9-3 The 14 first entries of more than 200 elements in the Domino Schema Definition*

We are interested in the following elements:

- Additional subforms
- Additional views
- Removed, altered, or added view columns
- Added fields to existing (sub)forms
- Events code
- Agents
- Hide when formula

Less relevant are the following elements:

- Fonts and colors
- Margins
- Design element editors, signers
- Static elements like pages and framesets

These items do not have an equivalent in other systems or cannot be automatically converted, so you will need to recreate them on your target platform anyhow.
9.4 Introducing DXL Magic as a tool for comparison

When trying to understand what changes and customizations have been made in your Domino Document Manager Environment, it should be expected that many of the people who implemented these changes (and the knowledge about how or why they implemented a change), will no longer necessarily be with the organization. Unfortunately, much of this customization knowledge will be lost. Fortunately, DXL Magic can help you analyze the extent of changes within your DDM environment.

DXLMagic is a tool containing sample code to export, import, document, and manipulate DXL. It consists of the DXLStudio database and a set of Java command-line tools.

As an example of the what the tool does, refer to the following three figures.

As shown in Figure 9-4, DXL Magic provides a summary showing all of the design elements which have been changed or modified from the original DDM Template.
Figure 9-5 illustrates one of the design element changes in detail. In this case, it uses the color purple to indicate new code inserted or modified within the design element.

Figure 9-6 illustrates another code change. In this case, it is illustrating a specific element of code that was removed from the design element.

### 9.5 Setting up DXLMagic

DXLMagic requires a few steps to setup. It comes with a graphical installer for Windows and with full source code for customization to your needs.

DXLMagic is available on OpenNTF from the following Web page:

http://www.openntf.org/Projects/pmt.nsf/ProjectLookup/DXLMagic

An introduction to DXL is available on SlideShare from the following Web page:

http://www.slideshare.net/NotesSensei/ad215-practical-magic-with-dxl

The tools is licensed under a Apache 2.0 licence and its source code is available for re-use and customization to specific needs.
9.5.1 Downloading DXLMagic

Open the following Web site and select the Releases tab:

http://www.openntf.org/Projects/pmt.nsf/ProjectLookup/DXLMagic

![Download of DXLMagic](image)

Make sure you download the latest version. You either download the DXLMagicWindowsInstaller.exe (for automatic installation on Windows) or DXLMagic.jar for manual installation on any operating system. You will need a free registration on OpenNTF to download. OpenNTF is an IBM-sponsored foundation that makes OpenSource tools around Lotus products available.

DXLMagic is written in Java and will run on Windows, Linux and Apple Macintosh.

For the internal working of DXLMagic we make use of XMLUnit. The Windows installer includes XMLUnit. For installation on Linux or Mac you will need to download it separately from its project location:

http://xmlunit.sourceforge.net/

Make sure you download the latest stable version. We have used XMLUnit 1.2 (June 2008), so the file is xmlunit-1.2.jar.

9.5.2 Configuring DXLMagic

For Windows we provide an installer file DXLMagicSetup.exe. Before you run it, make sure that you have the following prerequisites:

- Java 6.0 installed (you can install it from http://www.java.com/)
- Notes 8.5x client installed (you can use older version of the Notes client, but DXL will not be fully accurate, so items might be missing in your comparison).
When you run the installer, it will copy all the necessary jar files into the Notes client program directory and create a cmd file in your Windows Start menu. See Figure 9-8.

![Figure 9-8 The DXL Magic Installer Welcome window](image)

After you begin the installation, you will be prompted to confirm the location for the installation files, as shown in Figure 9-9.

![Figure 9-9 Select the installation directory](image)
We recommend that you follow the default recommendation of the installer which points to your Lotus Notes Client Java Client library directory. After the installation is complete, you will also find a menu entry under the Programs menu, as shown in Figure 9-10.

![Figure 9-10](image)

**Figure 9-10** The completed installation creates a menu entry

### 9.5.3 Manually configuring DXLMagic

If you would like full control over what the installer is doing or you are running Mac OS/X or Linux, you will need to manually install DXLMagic.

Perform the following steps to manually configure DXLMagic:

1. Replace `[AppDir]` with the path to your Notes application (typically something like `C:\Notes`, `C:\Program Files\IBM\Lotus Notes`, `/applications/lotus/notes` or `/opt/ibm/lotus/notes`, depending on your platform.

2. Set `[JarDir]` where you copied the `DXLMagic.jar` and the `[JarDir]/xmlunit-1.2.jar` file.

Once you have completed these steps, perform these steps for configuring it:

1. Download Java 6.

   **Note:** It should work with Java 5 too, but this has not been specifically tested.

2. Install and configure Java 6.

   **Note:** It should work with Java 5 too, but this has not been specifically tested.

3. Copy the `DXLMagic.jar` and `xmlunit-1.2.jar` file into the `[JarDir]`
4. Edit your environment to include Notes on the path and the jar files on the classpath and add an environment parameter LD_LIBRARY:

   - Windows: notepad c:\autoexec.bat and add these lines shown in Example 9-1.

   **Example 9-1  Editing your environment**
   
   ```
   SET PATH=%PATH%;[AppDir]
   SET CLASSPATH=.;[AppDir]\jvm\lib\ext\*;[JarDir]\DXLMagic.jar;[JarDir]\xmlunit-1.2.jar
   SET LD_LIBRARY=[AppDir] (Reminder: if you have spaces in your path you need to put the path in quotes)
   ```

   - Linux: sudo nano /etc/environment and add or edit these lines, as shown in Example 9-2.

   **Example 9-2  Editing the PATH and Classpath in a Linux environment**
   
   ```
   PATH="[whatever was there]:[AppDir]"
   CLASSPATH=".:([AppDir]/jvm/lib/ext/*;[JarDir]/DXLMagic.jar;[JarDir]/xmlunit-1.2.jar"
   LD_LIBRARY="[AppDir]"
   DYLD_LIBRARY="[AppDir]"
   ```

   - Mac: vi ~/.profile and add or edit these lines, as shown in Example 9-3

   **Example 9-3  Editing the PATH and Classpath in a Mac environment**
   
   ```
   export DYLD_LIBRARY_PATH=[AppDir]/Contents/MacOS (where AppDir is going to look something like: /Applications/Notes.app)
   export CLASSPATH=$CLASSPATH$:.:[AppDir]/Contents/MacOS/jvm/lib/ext/*;[JarDir]/DXLMagic.jar;[JarDir]/xmlunit-1.2.jar"
   ```

   **Tip:** When you copy the DXLMagic.jar and xmlunit-1.2.jar into the [AppDir]/jbm/lib/ext you do not need to add these two explicitly to the CLASSPATH, because they are then covered by [AppDir]/jvm/lib/ext/*

Now we can proceed. Try the following command in a command window:

```
java com.ibm.sg.dxlmagic.DesignExporter
```

You should get a reply stating the required parameters for this command. If you get an error message either your path is not correct or you might need to reboot your machine.

**Note:** Depending on the size of your templates and your Java configuration, the DXLMagic tools might run out of heap space. To avoid this, you can add a startup parameter to the Java command line:

```
java -Xmx1024m com.ibm.sg.dxlmagic.[NameOfTheTool]
```

This will allow Java to consume more memory. The parameter is case sensitive.
9.6 Using DXLMagic

In Windows, select **DXL Magic for Windows** from the Start Menu. This entry calls the DXLMagic.cmd file with the content shown in Example 9-4.

**Example 9-4  The DXLMagic.cmd file**

```bash
@echo off
REM DXLMagic Utility functions for Lotus Notes
ECHO Starting DXLMagic
SET DYLD_LIBRARY_PATH=C:\Notes\n
SET LD_LIBRARY_PATH=C:\Notes\n
SET PATH=C:\Notes\;%PATH%
SET CLASSPATH=.;C:\Notes\jvm\lib\ext\*;CLASSPATH
java -Xmx1024m com.ibm.sg.dxlmagic.UI compare %2 %3 %4 %5 %6 %7 %8 %9
exit
```

**Tip:** The installer adjusts the file path entries in the command files to match the installation directory of your Notes client. If you have multiple different Notes clients installed, you might need to adjust them manually.

DXLMagic is a set of Java classes delivered in a jar file. You can run most commands from the command line or integrate them into your own applications. To get an overview of available commands the command shown in Example 9-5.

**Example 9-5  List out available DXLMagic commands**

```bash
java com.ibm.sg.dxlmagic.Help
```

DXLMagic comes with an user interface for comparison. To start that from the command line use the UI or CompareUI class. See Example 9-6.

**Example 9-6  Graphical UI for file comparison**

```bash
java -Xmx1024m com.ibm.sg.dxlmagic.CompareUI
```

9.6.1 The documentation workflow

Documenting changes using DXLMagic is performed in three distinct steps. Each step has file inputs and file outputs that can be performed independent from each other. Only the first step needs to be performed in the Domino environment. Steps 2 and 3 are pure XML operations can happen in their own environment.

Figure 9-11 on page 150 illustrates the flow of documentation.
9.6.2 Using the comparison user interface

The comparison user interface automates the three steps and generates two reports for you. The most time-intensive task is the export of the database design into DXL. The user interface only supports the exporting of local databases. If you want to export from a server, use the command line version.

All files are stored in the directory structure below your working directory. You can use both the DXL files as well as the comparisonresult.xml file to run additional reports. From our consulting practice we know that the field and hide-when reports are popular.

Figure 9-12 illustrates the comparison user interface.

The buttons behind the entry fields allow you to open a selection dialog box to pick the file or directory for processing. If you pick a file of a unsupported type, the comparison will fail. The file selection dialog box is the standard cross-platform Java file dialog box. Figure 9-13 on page 151 illustrates the file selection dialog box.
You can select a summary and a detail report. The stylesheets that generate these reports are inside the jar file. You can alter them by opening the jar file as ZIP (rename it) or you can copy them and create your own versions that you use with the command line tools.
Once you start the comparison process, the user interface switches to the messages tab to keep you updated on the progress of the tool. Figure 9-14 illustrates the export progress display.

![Figure 9-14: The export progress display](image)

At the bottom of the window there is a small progress bar that indicates the completeness of the action. If your Notes client is not running when you start the process you will get prompted for your password, as shown in Figure 9-15.

![Figure 9-15: Password prompt if the Notes client isn't running](image)

**Note:** Because the process has four or five distinct steps: 2x export 1x compare, 1-2 reports the bar will complete four to five times. Typically the longest part is the comparison process. The report generation on the other hand completes in a few seconds.

After the whole process has completed, the Message tab indicates in a color line that the reports are ready. You might need to scroll down to see that line.
Chapter 9. Analyzing customizations

9.6.3 Exporting databases to DXL from the command line

You can export DXL from any Notes database or template. You can export from your local Notes data directory or from a server. We use the regular Notes client functionality, so you want to have your Notes client running before you export databases.

**Tip:** Exporting DXL is short but I/O intensive. When exporting many files you want to use local replicas or copies to avoid high network use.
Databases are exported one by one. However you can use a command file to do a batch export. The syntax for the exporter is shown in Example 9-7.

Example 9-7  Syntax for a Batch Export

```
java com.ibm.sg.dxlmagic.DesignExporter [OutputPath] [PathToDatabase] optional
[Server]
```

- OutputPath is the directory where the DXL file will be created. You should avoid having spaces in the file name. Put a directory name into quotes if it contains spaces.
- PathToDatabase is the database filename relative to the Notes/Domino data directory. Include the extension (.nsf, .ntf)
- Server. The name of the server your databases or templates reside. You can omit the server when you export DXL from your local Notes client (recommended). The Notes client must be able to reach this server. You can use Ctrl+O in your Notes client and type in that name to see if the connection is valid.

Example 9-8 exports the design of your log.nsf from your Notes client into the c:\temp directory.

Example 9-8  Command for exporting the design of the log.nsf from the Notes client

```
java com.ibm.sg.dxlmagic.DesignExporter C:\temp\ log.nsf
```

The directory C:\temp must exist before you run the tool. If a database is located in a subdirectory of the Notes data directory, it also will be placed into a sub directory of the output directory. Missing subdirectories are created automatically.

The export process will write its progress onto the console using dots. Every dot represents a successfully exported top-level element. You might get series of warnings (for example, “failed assert”) that represent the current shortcomings of the DXL process. These warnings are not relevant for our purpose. The only errors you need to act on are genuine crashes if Java runs out of space or your database has a problem. You can use the discussion section on OpenNTF’s project site to obtain help.

A window could look like Figure 9-17 on page 155.
When the export starts it will state the number of found design elements as Collection members. For each member, there will be one dot for the extraction and one dot for the assembly. You can redirect that output to a file or the nul device.

If you have multiple databases to export, you can use a OS level batch command to export them in one run. A good idea is to have all nsf/ntf in a subdirectory of your Notes data directory and run the command from there, as shown in Example 9-9.

Example 9-9 Exporting all local databases from a folder

```
for $x in (*.*) do java -Xmx512 com.ibm.sg dxlmagic.DesignExporter c:\dxl\ $x
```

The export process will create a file with the same name as your database with the extension dxl. The extension is appended to the full file name, so your database test.nsf will create a dxl file test.nsf.dxl

Once you have exported the base line templates along with your changes, you will be able to run comparisons. You need to export every file only once because the result is not changed in the following steps.
You can compare any two DXL files for their differences. The best results, however, will be achieved when comparing files that have the same ancestor or are related. For example, comparing names.nsf.dxl to pernames.ntf.dxl will highlight changes made to a personal address book after it has been created. See Example 9-10.

Example 9-10   Command syntax compare 2 DXL files

```java
java com.ibm.sg.dxlmagic.CompareXML [BaseLine] [SecondFile] [OutputFile]
```

- BaseLine is the path to the dxl file representing the original template
- SecondFile is the path to the dxl file representing the altered template/database
- OutputFile is the path to the result file. An existing file will be overwritten

Example 9-11 shows the command comparing the Address Book to the original template.

Example 9-11   Comparing Addressbook to template

```java
java com.ibm.sg.dxlmagic.CompareXML c:\dxl\pernames.ntf.dxl c:\dxl\names.nsf.dxl c:\dxl\namecomp.xml
```

The comparison process uses a two step approach. In step one, both DXL files are subjected to an internal XSLT transformation that removes XML elements we do not need or want to compare. Typically we do not need to look at the signer of an element or the modification date or font-sizes. We have selected a default set of elements to exclude. If you want to define your own criteria, you will find instructions in 9.7, “Customize DXLMagic” on page 165. Step two involves running the comparison. Figure 9-18 on page 157 illustrates the comparison running inside of the command wind.

---

**Note:** As part of the available downloads for this IBM Redbooks publication, we also provide a baseline, out of the box version of the DDM 7 template. Refer to Appendix E, “Additional material” on page 289 for information about how to download these templates.
Figure 9-18  Sample comparison run

While running the comparison, the command window shows the design elements found, the matching condition, and the differences inside the design elements. You do not need to pay attention to these values, because they are written into the result file.

If you also like to capture the results of the console output you can use a standard redirector to write it into a file. An example of this is shown in Example 9-12.

Example 9-12  Saving the console output into a file

```
java com.ibm.sg.dxlmagic.CompareXML c:\dxl\pernames.ntf.dxl c:\dxl\names.nsf.dxl c:\dxl\namecomp.xml > namecomp.console.txt
```

The resulting XML file (namecomp.xml in our example) can be used for various reports to document your changes.

### 9.6.5 Generating additional reports from DXL or the comparison results

You might want to run additional reports either against the exported DXL or the comparison results. DXLMagic uses XSLT stylesheets to transform these XML files into different formats. In the DXLMagic user interface, select Run Reports from the DxLMagic Menu. Select the Settings tab to select your parameters.
The report panel has a number of built-in reports that are all catered to run against a DXL file. These are shown in Figure 9-19. You can select multiple reports by typing Ctrl and the report name. If you run your own report, use the file picker dialog panel to navigate to the XSLT file of your choice.

![Figure 9-19 The report panel selection](image.png)

The build in reports are as follows (as of the time this book is published):

- **HideWhen**
  This report lists all the HideWhen formulas used in a form or subform in tabular form. Great resource to find hidden business logic.

- **ActionButtonList**
  This report is a list of all actions and buttons in forms and views to provide an overview what interactions your applications provide.

- **FieldReport**
  This report lists all the fields and columns used in forms and views to provide an overview of the datamodel used in your applications.

- **CodeReport**
  This report lists LotusScript code used in your database by event.

- **Synopsis**
  This report lists the complete and comprehensive documentation of a database. Stylesheet created by Cesar Mugnatto in an OpenNTF project:
  
  ![http://www.openntf.org/Projects/pmt.nsf/ProjectLookup/Comprehensive%20XSLT%20for%20database%20design](http://www.openntf.org/Projects/pmt.nsf/ProjectLookup/Comprehensive%20XSLT%20for%20database%20design)
9.6.6 Generating reports from the comparison results from the command line

The comparison result file contains all comparison results in XML format. Additions, changes, and deletions are documented including the XPath expression to find these changes in the original XML file. XML files are not so easily readable, so, as a final step, you would convert this file into one or more XML reports. We use an XSLT transformation for this process. Because XSLT allows you freedom in creating output, you can render the result into other formats (for example, for upload to a bug tracker) as well.

We use the TransformXML component of DXLMagic to execute the transformation, as shown in Example 9-13.

Example 9-13 Generating a report

```
java com.ibm.sg.dxlmagic.TransformXML [Source] [Style] [Target] optional [parameter=value] [parameter=value]
```

- **Source**
  
  This is the path to the XML file generated in the previous step. Typically it is an XML file. But you also could use a DXL file as source to generate usage reports.

- **Style**
  
  This is the XSLT Stylesheet containing the report definition. You can specify one report defining stylesheet at a time. A stylesheet could generate more than one report in one or more files (see the XSLT tutorials for that). When you need multiple reports you run the transformation multiple times with different stylesheet names.

- **Target**
  
  This is the path to the resulting file. For our reports this would be a html file, but you can pick whatever file name you wish. The transformation step doesn't alter the file extension.

- **parameter**
  
  These are name value pairs that can be handed over to the XSLT stylesheet. Depending on your stylesheet you might point to external resources like a css file or to a query parameter. In our stylesheets we do not use parameters for now.

Example 9-14 illustrates how to generate the report using the command shown in Example 9-13, but this time using actual parameter values.

Example 9-14 Calling the report with a short overview

```
java com.ibm.sg.dxlmagic.TransformXML c:\dxl\namecomp.xml c:\dxl\ComparisonReportShort.xslt c:\dxl\namecompshort.html
```

**Note:** The list of reports will change over time because DXLMagic is used beyond this IBM Redbooks publication. Refer to OpenNTF to access new report types.
Figure 9-20 illustrates the report being generated within the command console.

![Executing the report generation](image)

The report generation is comparably fast with little feedback on the console. The result can be used in a browser instantly.

There are two built in stylesheets inside the jar file. You can modify them to suit your needs or add completely new reports. Additional stylesheets will be available from OpenNTF in the DXLMagic project over time.

**ComparisonReportShort.xslt**

This file provides you with a list of added, altered, and removed design elements. For altered design elements the number of changes compared to the baseline is listed. The nature of the changes is not outlined. An example of the Comparison Report is shown in Figure 9-21 on page 161.
The report has three sections with gray headings:

- **Modified**
  These elements are in both databases but they are different in key areas like text, code, and number of fields.

- **Added**
  These elements do not exist in the base line. Typically, you can see your custom document types (= subforms) and your additional script libraries here.

- **Missing**
  Elements that exist in the base line but not in your template. Typically you can expect the subforms of the default document types here that would have been removed when you defined your own elements.

The report has three columns:

- **Element**
  Type of design element (form, subform, view, agent and so forth)

- **Name**
  Name (or if missing: alias) of the design element

- **Changes**
  Number of identified changes and Document unique ID of the design element in brackets. You can use that ID to identify a design element in the property box in Domino Designer.

**ComparisonReportDetail.xslt**
This file lists the changes in greater detail. Specifically, the source code of the base template and your database are displayed next to each other so you can visually compare the changes. Figure 9-22 on page 162 illustrates a comparison report.
Depending on the number of DDM templates you need to analyze, you might want to automate the process. You have 2 possibilities here: use a batch file or use DXLMagic command file.

Using a batch file has the advantage, that for each step the jvm is loaded separately and you can include checks to skip existing files. Using DXLMagic has the advantage, that the jvm is loaded only once, so execution speed is faster. In Example 9-15 we assume that your standard template names are DomDoc70.ntf and FileCab70.ntf.

**Example 9-15  DOS/Windows command file (ddmcompare.cmd)**

```bash
@echo off
REM the standard templates are domdoc70.ntf and filecab70.ntf
REM the two modified databases are %1 and %2
REM the resultpath comes from DXLPATH
if x%DXLPATH%x==xx SET DXLPATH=C:\dxl
cd %DXLPATH%
if not exist domdoc70.ntf.dxl java com.ibm.sg.dxlmagic.DesignExporter %dxlpath% domdoc70.ntf
if not exist filecab70.ntf.dxl java com.ibm.sg.dxlmagic.DesignExporter %dxlpath% filecab.ntf
if not exist %1.dxl java com.ibm.sg.dxlmagic.DesignExporter %dxlpath% %1
if not exist %2.dxl java com.ibm.sg.dxlmagic.DesignExporter %dxlpath% %2
if not exist %1.compare java com.ibm.sg.dxlmagic.CompareDXL %dxlpath%\domdoc70.ntf.dxl %dxlpath%\%1.dxl %dxlpath%\%1.compare
if not exist %2.compare java com.ibm.sg.dxlmagic.CompareDXL %dxlpath%\filecab70.ntf.dxl %dxlpath%\%2.dxl %dxlpath%\%2.compare
REM We generate the reports unconditionally
java com.ibm.sg.dxlmagic.TransformXML %dxlpath%\%1.compare %dxlpath%\ComparisonReportDetail.xslt %dxlpath%\%1.detail.html
REM More reports here....
java com.ibm.sg.dxlmagic.TransformXML %dxlpath%\%2.compare %dxlpath%\ComparisonReportDetail.xslt %dxlpath%\%2.detail.html
REM More reports here....
echo Done!
```

**9.6.7 Putting it all together**

Figure 9-22  Sample report provided by DXL Magic
You would call the file like this:

```
ddmcompare mylibrary.nsf mycabinet.nsf
```

The command file will then compare your library and cabinet file to the base line library and cabinet. The base line gets exported only once. Example 9-16 illustrates the command file to run the compare on either a Macintosh or Linux environment.

**Example 9-16  Macintosh/Linux command file ddmcompare.sh**

```bash
#!/bin/sh
#
# Exports File Cabinets and Libraries from a local Notes client and compares them to a base line
#
# the standard templates are domdoc70.ntf and filecab70.ntf
# the two modified databases are $1 and $2
# the result path comes from $DXLPATH
if [ -z "$dxlPath" ]; then
    export dxlPath=$HOME/dxl
fi
echo Operating from basePath $dxlPath
cd $dxlPath
if [ -z "$2" ]; then
    echo echo echo
echo Usage: $0 [nameOfLibraryFile] [nameOfCabinetFile]
echo fileName needs to be relative to the Notes data directory and include the extension
echo exit
fi
if [ ! -e domdoc70.ntf.dxl ]; then
    echo Retrieving base line library domdoc70.ntf
    java com.ibm.sg.dxlmagic.DesignExporter $dxlPath domdoc70.ntf
fi
if [ ! -e filecab70.ntf.dxl ]; then
    echo Retrieving base line cabinet filecab70.ntf
    java com.ibm.sg.dxlmagic.DesignExporter $dxlPath filecab70.ntf
fi
if [ ! -e $1.dxl ]; then
    echo Retrieving library to compare $1
    java com.ibm.sg.dxlmagic.DesignExporter $dxlPath $1
fi
if [ ! -e $2.dxl ]; then
    echo Retrieving cabinet to compare $2
    java com.ibm.sg.dxlmagic.DesignExporter $dxlPath $2
fi
if [ ! -e $1.compare ]; then
    echo Generating library comparison baseline to $1
    java com.ibm.sg.dxlmagic.CompareDXL $dxlPath/domdoc70.ntf.dxl $dxlPath/$1.dxl
    $dxlPath/$1.compare
fi
if [ ! -e $2.compare ]; then
    echo Generating cabinet comparison baseline to $2
```
Do not forget to make the script executable using `chmod +x ddmcompare.sh`. You then can use that script like this: `./ddmcompare.sh mylibrary.nsf mycabinet.nsf`

You can modify the command files to accommodate your specific needs.

The other possibility to automate the process is to use DXLMagic's command files. You can call them like this: `java -Xmx1024 com.ibm.sg.dxlmagic.DXLMagic [PathToCommandFileName]`

Using DXLMagic with a command file name loads the JVM™ only once and executes all commands unconditionally. The command file usually gets generated by the DXLMagic Notes front-end (see the OpenNTF download for that), but you can create that file also with a text editor. Example 9-17 shows an example of the command file.

**Example 9-17  DXLMagic command file (for example, things2do.txt)**

```
# Command File for dxlMagic
workingdirectory=/home/stw/dxl/
projectdirectory=/home/stw/dxl/DDM/

# Action: EXPORT
export,domdoc70.ntf
export,filecab70.ntf
export,myLib.nsf
export,myCabinet.nsf

# Action: COMPARE
compare,domdoc70.ntf.dxl,myLib.nsf.dxl,myLib.nsf.compare
compare,filecab70.ntf.dxl,myCabinet.nsf.dxl,myCabinet.nsf.compare

# Action: TRANSFORM / CREATE REPORT
transform,myLib.nsf.compare,ComparisonReportShort.xslt,myLib.nsf.short.html
transform,myLib.nsf.compare,ComparisonReportDetail.xslt,myLib.nsf.detail.html
transform,myCabinet.nsf.compare,ComparisonReportShort.xslt,myCabinet.nsf.short.html
transform,myCabinet.nsf.compare,ComparisonReportDetail.xslt,myCabinet.nsf.detail.html
```
One difference in the command files is a distinction between the DXL directory containing stylesheets and static files versus the project directory, which is the default destination for all generated output and source for the variable input files.

9.7 Customize DXLMagic

The database export exports the entire NSF, so you will not want to change the export process. However, you are free to look at the source code and adjust it to your needs.

**Note:** DXLMagic can be used to compare any Notes database. You are not limited to DDM templates and databases.

More likely you might want to change the comparison process and most likely you want to change the reports.

To do that you need to understand the workings of DXLMagic to customize it to your needs. The skill you need for doing that is a understanding of XML and XSLT. Good learning sources for this universal skills are as follows:

- Tutorials for XML and XSLT on the following Web page:

9.7.1 Customizing reports

All reports provided by DXLMagic are basically XSLT stylesheets that transform XML files. The format of the comparison result consists of five elements inside the `<CompareResult>` root element:

- **unmodified**
  - All Design Elements that have been found to be equal
- **missing**
  - Elements that are in your base line but are absent
- **modified**
  - Elements that have differences, differences are listed
- **added**
  - Elements that are not in the base line but added in the custom template
The XML Schema for the report looks like Figure 9-23.

![Figure 9-23 XML Schema of comparison results]

Using an XML editor you can easily modify one of the existing reports to suit your needs. The basic mechanism for XSLT transformation apply. Example 9-18 illustrates the XSLT of a report.

**Example 9-18 The XSLT of the compact report**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
  <xsl:template match="/">
    <html>
      <head>
        <title>Comparison Result</title>
        <style type="text/css">
          html, body {
            font-family : Verdana, Arial, Helvetica, sans-serif;
            font-size : xx-small
          }
          table.modtable {
            width : 100%;
            border : 1px solid gray
          }
          table.modtable th {
            border-bottom : 2px solid red;
            text-align : left;
            font-weight : normal;
            font-size : x-small
          }
        </style>
      </head>
      <!-- Body of the report goes here -->
    </html>
  </xsl:template>
</xsl:stylesheet>
```
Chapter 9. Analyzing customizations

```html
<table class="modtable">
  <tr>
    <th>Element</th>
    <th>Name</th>
    <th>Changes</th>
  </tr>
  <tr>
    <th colspan="3" class="changetype">Modified</th>
  </tr>
  <xsl:apply-templates select="modified" mode="summary"/>
  <tr>
    <th>Element</th>
    <th>Name</th>
    <th>Unid</th>
  </tr>
  <th colspan="3" class="changetype">Removed</th>
  <xsl:apply-templates select="missing" mode="summary"/>
  <tr>
    <th>Element</th>
    <th>Name</th>
    <th>Unid</th>
  </tr>
</table>
```
Looking at the stylesheet in detail you can see that it consists an HTML layout extended by XSLT commands.

9.7.2 Customizing the Compare process

The compare process internally has three steps:
1. Find two elements to compare
2. Perform the comparison
3. Process the result and write it out
CompareDXL, the routine used in DXLMagic to compare elements, first tries to find matching elements using the UniversalID of the note. This works well if the custom template was derived from a copy of the original template. If a match is not found it looks for a element with the same alias if the current element has an alias and then with the same name.

Figure 9-24 illustrates the flow used for the comparison routine.

The easiest way to customize the compare process is altering the “Transform to ‘leaner’ XML” step in the flowchart. It uses an XSLT transformation to achieve this. The XSLT file is part of the DXLMagic.jar file.

You can open any JAR file with an extraction utility. Windows Explorer can do this for you after you renamed DXLMagic.jar to DXLMagiX.zip.
Navigate to com/ibm/sg/dxlMagic/resources. There you find CompareCleanup.xslt, which is used to simplify the DXL into something more manageable. Example 9-19 shows the code for the CompareCleanup.xslt for your review.

**Example 9-19 The simplification stylesheet**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0"
  <xsl:output method="xml" indent="yes" xml:space="preserve" />
  <xsl:template match="/">
    <database>
      <xsl:apply-templates />
      <xsl:sort select="name()"/>
      <xsl:sort select="@name"/>
    </database>
  </xsl:template>
</xsl:stylesheet>
```

<!-- Elements we want to suppress but get the decendants, we have them one by one for easier maintenance -->

```xml
<xsl:template match="d:run"><xsl:apply-templates /></xsl:template>
<xsl:template match="d:table"><xsl:apply-templates /></xsl:template>
<xsl:template match="d:tablecell"><xsl:apply-templates /></xsl:template>
```

<!-- Stuff we want to suppress including any content -->

```xml
<xsl:template match="d:acl" />
<xsl:template match="d:databaseinfo" />
<xsl:template match="d:launchsettings" />
<xsl:template match="d:helpaboutdocument" />
<xsl:template match="d:helpusingdocument" />
<xsl:template match="d:noteinfo" />
<xsl:template match="d:imageresource" />
<xsl:template match="d:wassignedby" />
<xsl:template match="d:updatedby" />
<xsl:template match="d:revisions" />
<xsl:template match="d:note" />
<xsl:template match="d:tablecolumn" />
<xsl:template match="d:font" />
<xsl:template match="d:border" />
<xsl:template match="d:actionbarstyle" />
<xsl:template match="d:actionbuttonstyle" />
<xsl:template match="d:tablerowstyle" />
<xsl:template match="d:agentdata" />
<xsl:template match="d:rundata" />
```

<!-- We don't compare RAWITEMS and item not inside the body tag. Duplicate rules to avoid ambiguous matches -->

```xml
<xsl:template match="d:item[d:rawitemdata]" />
<xsl:template match="d:form/d:item" />
<xsl:template match="d:subform/d:item" />
<xsl:template match="d:form/d:item[d:rawitemdata]" />
<xsl:template match="d:subform/d:item[d:rawitemdata]" />
```
Inspecting the simplification stylesheet closer, you can distinguish four sections:

- The starting section with basic definitions and the root and database template starting the processing. The root template has the selection formula match="/". The database template specifies two sort keys for all top level elements: name() and @name. This will sort the resulting XML by element types (<agent ...><form ...><view ...>) and then by the name of the element. The @ sign stands for "Attribute".

Example 9-20 shows the code for the start section of the XSLT.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0"
    xmlns:d="http://www.lotus.com/dxl"
    xmlns="http://www.lotus.com/dxl">

    <xsl:output method="xml" indent="yes" xml:space="preserve" /
    <xsl:template match="/">
        <xsl:apply-templates />
    </xsl:template>

    <xsl:template match="d:database">
        <database>
            <xsl:apply-templates>
                <xsl:sort select="name()" />
                <xsl:sort select="@name" />
            </xsl:apply-templates>
        </database>
    </xsl:template>

    <xsl:template match="d:attribute">
        <attribute name="{$curAttName}">
            <xsl:apply-templates />
        </attribute>
    </xsl:template>

    <xsl:template match="d:element">
        <element name="{$curTagName}">
            <xsl:apply-templates />
        </element>
    </xsl:template>

</xsl:stylesheet>
```

- One section with elements we are not interested in, but need to analyze the content. The template tells the XML processor to look into the child elements (xsl:apply-templates) but doesn't define any output for the element itself.
Example 9-21 illustrates the section of code for Elements we want to suppress but get the descendants,

Example 9-21  Elements we only check the content

```xml
<xsl:template match="d:run"><xsl:apply-templates /></xsl:template>
<xsl:template match="d:table"><xsl:apply-templates /></xsl:template>
<xsl:template match="d:tablecell"><xsl:apply-templates /></xsl:template>
```

- One section with elements or attributes we do not intend to compare and are not interested in the inner workings. For a customized template updated by and was signed by are definitely different, so stating the obvious does not help us, so we remove these elements (and all the others). As a general rule in XSLT, the more precise the match is, the higher the priority for a template, with `match="*"` being the lowest priority. A template that directly ends without a body leads to no output in the result file.

Example 9-22 illustrates how this suppresses the elements.

Example 9-22  Suppressing design elements

```xml
<xsl:template match="d:wassignedby" />
<xsl:template match="d:updatedby" />
```

- The final section with the weak matches `match="*"` (any element) and `match="@*"` (any attribute) copies the elements 1:1 from the source including all attributes. The attributes get sorted by the attribute name in that process. This eliminates “element has changed” hit for merely changed sequence of attributes which by definition of XML must not be relevant for a comparison result. One specialty that eases the retrieval of matching elements is the treatment of the noteinfo element. When we find an element containing a noteinfo element (which are all top level elements) we pull the @unid attribute element out and store it with the noteid's parent element. The noteid element itself gets suppressed.

Example 9-23 illustrates the details of this code and how the matching works as described in the paragraph above.

Example 9-23  XSLT 1:1 copy with sorted attributes

```xml
<!-- Copy everything 1:1 to the result if not specified otherwise -->
<xsl:template match="*">
  <xsl:variable name="curTagName" select="name()"/>
  <xsl:element name="{$curTagName}">
    <!-- Walk through the attributes -->
    <xsl:if test="d:noteinfo">
      <xsl:attribute name="unid"><xsl:value-of select="d:noteinfo/@unid" /></xsl:attribute>
    </xsl:if>
    <xsl:apply-templates select="@*">
      <xsl:sort select="name()" /></xsl:apply-templates>
  </xsl:element>
</xsl:template>
```

```xml
<xsl:template match="@*">
  <xsl:variable name="curAttName" select="name()"/>
  <xsl:attribute name="{$curAttName}"><xsl:value-of select="." /></xsl:attribute>
</xsl:template>
```

```xml
</xsl:stylesheet>
```
You could add or remove template statements from this stylesheet to tweak your comparison operation. You would need a good understanding of both XSLT and DXL to do that successfully.

9.8 Integrating DXLMagic in your own code

So far we have shown how to use DXLMagic from the command line. This is not the only way to invoke DXLMagic, however. DXLMagic are Java classes under an Apache 2.0 licence that allows you to reuse this in your own applications. Check the licence information inside the jar file as well as the licence conditions for XMLDiff.

The complete JavaDoc documentation can be downloaded from OpenNTF.org to gain a better understanding. The following section will show you how you can integrate DXLMagic into your own Java programs. This integration capability allows you, for example, to make DXLMagic part of an automated build system or a project management solution.

The code snippets will omit import definitions and sometimes variable declarations. The full source code is contained in the DXLMagic.jar file where you can take a look.

9.8.1 Exporting DXL

```java
Example 9-24  Exporting DXL

DesignExporter ex = new DesignExporter();
// Variables you want to adjust
String basePath = "/home/user/dxl/";
String nsfToExport = "mydatabase";
String server = ""

ex.setBasePath(basePath);
ex.setServer(server);
ex.setNsfToExport(nsfToExport);

// What wait delay is needed to prevent crashes
// 88 is ok for local unless you have a really fast box
// Eventually this needs to be much longer
// for server based databases
ex.setCrashPreventSleepTime(88);
// Don't write out
ex.setWriteToFile(false)
// Start processing
ex.start();
ex.join(); // Wait until finished
// The ready exported DXL as a String
String result = getDxlResult();
```
9.8.2 Comparing two DXL files

**Example 9-25  Comparing two DXL files**

```java
CompareDXL tx = new CompareDXL();
// Parameters to set
String baseName = "/home/user/dxl/file1.dxl";
String compareName = "/home/user/dxl/file1.dxl";
String targetName = null; // we don't want to write

// You could fill properties directly if you want an in-memory operation
tx.setBaseName(baseName);
tx.setCompareName(compareName);
tx.setTargetName(targetName);

// We use a cleanup stylesheet to narrow down
tx.setUseCleanupStyleSheet(true);
tx.setCleanupStyleSheetResourceName(CompareDXL.DEFAULT_CLEANUP_STYLESHEET_NAME);

// Execute
tx.process();
System.out.println(tx.getTargetXML()); // a DOM with the result
```

9.8.3  Running transformations and reports

**Example 9-26  Running transformations and reports**

```java
TransformXML tx = new TransformXML();
String sourceName = "somestuff.compare";
String styleName = "somereport.xslt";
String targetName = "someresults.html";

// Add additional parameters
for (String parName : allParameters) {
    String parValue = allParameters.get(parName);
    tx.addTransformationParameter(parName, parValue);
}

// Execute the transformation
tx.process(sourceName, styleName, targetName);
```

DXLMagic can do a few things more, like splitting design into individual file components or updating design back into a template. Refer to the DXLMagic source, available on OpenNTF, for updated versions and releases:

http://www.openntf.org/Projects/pmt.nsf/ProjectLookup/DXLMagic
Migration
Overview of the migration tool
Lotus Quickr Migrator: Domino Document Manager Edition

IBM has partnered with Innovative Ideas Unlimited, Inc. to deliver the Lotus Quickr Migrator: Domino Document Manager Edition. This tool provides a means to export an entire or partial Domino Document Manager (DDM) library structure (including taxonomy, data, and security model) into Quickr.

This chapter provides an overview of how the tool works, to help you better understand how this can help in your migration.

**Important:** This chapter is intended to make you aware of the tool and to help you understand how it works. This chapter is not intended to provide low level, step-by-step instruction on how to use the tool. These steps are discussed in more detail with the documentation provided directly with the Lotus Quickr Migrator.

**Attention:** The Lotus Quickr Migrator: Domino Document Manager Edition is available at no charge through trained IBM personnel or IBM Business Partners. Speak with your IBM representative for more information about obtaining the tool and assistance with your migration planning, or visit the following Web page:

http://ibm.com/software/lotus/products/dominodocumentmgr/
10.1 Overview of the Lotus Quickr Migrator

The Quickr Migrator Tool provides a means to export an entire or partial DDM Library structure (including taxonomy, data, and security model) into Quickr. Figure 10-1 illustrates conceptually how the Quickr Migrator Tool allows you to extract and view your DDM taxonomy and review and make adjustments to the DDM model in preparation for importation into Quickr Domino.

10.1.1 Components of the Quickr Migrator Tool:

The Quickr Migrator Tool is comprised of two main databases:

- DDM Migrator Main
- DDM Migrator Working NTF

The Quickr Migrator Tool is comprised of two main databases, as shown in Figure 10-2.

It also includes a number of working databases. The working databases provide users with an intermediary step to preview how their DDM taxonomy would look if migrated directly into Quickr.

The following list describes the high level functions of the databases and where and how profiles and export information is stored.

- Main database or “Main”

The main database contains all of the information the DDM Migration tool needs to migrate the DDM libraries to Quickr. All migration actions are launched from this database.
Export Profiles

Export Profiles is used to specify the DDM library (or partial library) to be exported and the target Quickr Place. Export profiles are created and maintained in the main database.

Working database

The working database is a Domino database file (NSF) that is associated with an Export Profile in the main database. One working database is created for each export profile in the main database. Through the course of the multi-stage migration process the Migrator tool populates the working database with documents that represent the DDM taxonomy as well as the DDM documents to be migrated.

Working database template

The working database template is the Domino template file (NTF) that is used to create the new working database. This template can be installed (copied) anywhere on the Domino server. Its path is specified on the Export Profiles database.

Working documents

Working documents can represent either the DDM taxonomy or DDM document profiles. The working documents provide a means to adjust the taxonomy or normalize the DDM data before it is imported into Quickr.

Figure 10-3 is intended to illustrate the relationship between the components within the tool.

---

**Figure 10-3**  Relationship of key components within the Quickr Migrator tool

---

### 10.2 Overview of the migration process

Figure 10-4 on page 180 illustrates the steps the user and the migration tool performs during the migration process. These steps are detailed in 10.2.1, “Steps involved in a typical migration” on page 181.
Figure 10-4  Overview of the Migrator process

Customer Installs database and template (A) & (B) on Quickr server.

Complete Export Profile and click “Process Current Step” in main database (A). One profile will need to be completed for each export.

User clicks “Process Current Step” on profile in main database (A). Stub documents are created in (C) representing DDM content to be migrated. File attachments are exported to file path defined on Export Profile.

(Optional) If XML option is used, XML files are created from source DDM data and can be manipulated to reflect desired changes.

Views in (C) show how content will look in new Quickr taxonomy. Supplied Actions and optionally custom-developed agents can be used to manipulate proposed Quickr taxonomy layers.

Working database (C) is created from template (B) if it doesn’t exist. Documents representing each DDM taxonomy level (Library, File Room, File Cabinet, Binder Category, Binder) are created in working database (C).

(Optional) User can write custom agents to manipulate data in (C) prior to import. (Examples: document type consolidation, name changes, data normalization)

User clicks “Process Current Step” on profile in main database (A). Quickr containers (rooms and folders) are created.

User clicks “Process Current Step” on profile in main database (A). Security is applied to Quickr containers.

Verify information is correctly migrated. Optionally clean-up files on operating system, delete working database (C), and DDM system when no longer necessary. Congratulations!

User clicks “Process Current Step” on profile in main database (A). Documents are imported into Quickr.
10.2.1 Steps involved in a typical migration

The DDM Migration Tool breaks down the DDM to Quickr conversion into a multi-stage process. This allows administrators to review the results of each process and make modifications as necessary. Throughout the migration process, the administrator can monitor the steps. Figure 10-5 illustrates an overview of the steps, as well as the status for the current processing step.

![Figure 10-5 Status display](image)

The typical migration involves the following steps:

1. Create an export profile in the main database identifying the DDM content to be migrated and the target Quickr Place (Quickr Places must already be created.)

2. From the export profile, launch the action to create working documents that represent the current DDM taxonomy. Additional views show how that same taxonomy would look if migrated into Quickr. Some actions are provided to make basic adjustments. Many programmatic hooks are provided to allow agents to be written for more advanced customization.

3. Extract data from the DDM system. Documents are created in the working database containing all metadata to be migrated in to Quickr. Files are placed in a directory on the operating system. Additional code can be run to normalize or manipulate the data before bringing it into Quickr. See Appendixes A and B in the Quickr Migrator Tool documentation for the list of available items.

4. Rooms and folders are created in the Quickr system according to the documents in the working database created in step 2.

5. Documents are created in Quickr and files are copied from the file system into Quickr.

6. Security is updated on documents, rooms, and the Quickr Place.

7. Clean up. Once satisfied with the migrated content, the working database and the directory containing exported files on the file system can be deleted.

Note: Some actions are provided directly within the tool to make basic adjustments to your basic DDM hierarchy and data prior to the migration into Quickr. Many programmatic hooks are provided to allow agents to be written for more advanced customization. Details of some of these examples are provided in Chapter 11, “Migration Tool: Operational scenarios” on page 187, while specific details on the hooks and fields that can be modified are contained in the documentation provided with the Quickr Migrator Tool.
10.3 Key components

The following section discusses the purpose and functionality of the primary components and configuration parameters within the tool. This is intended to give a preview of how DDM data and the corresponding DDM hierarchy can be extracted and manipulated prior to importing into Quickr. For specific details on each parameter within export profiles and the details of configuring the tool, refer to the Quickr Migrator Tool documentation.

10.3.1 Export profiles

The export profile provides the Quickr Migrator with the location of the DDM and Quickr systems, the file path to use for exported files, and various options unique to this export. Figure 10-6 illustrates an example of an export profile and some of the information it contains.

![Sample export profile](image)

Figure 10-6 Sample export profile
From within the “Export Options” tab, the administrator can specify the scope of the export if they wish to export an entire library, a room, a file cabinet, or only a binder. Details of the Export Options tab are shown in Figure 10-7.

**Figure 10-7  Sample export options**

### 10.3.2 Working databases and views into the DDM taxonomy

The Migrator tool populates a working database during the “Export DDM Taxonomy” step with working documents. The working database contains a number of views for review and adjustment of the DDM model in preparation for import into Quickr Domino.

### 10.3.3 The DDM export View

The user is presented with the DDM Export View. This view, shown in Figure 10-8 on page 184, displays the taxonomy in the current DDM library installation. Column headers and icons distinguish the different DDM components.

*Note:* As the documents in the working view are modified, the contents and display of this view may change.
10.3.4 The Conversion and working view

This view, shown in Figure 10-9 on page 185, contains all of the tools needed to adjust the taxonomy for import into Quickr Domino. By default, all DDM components (file rooms, file cabinets, binder categories, and binders) are imported into Quickr Domino as folders. The **Convert to Room** action is used to change a folder to a room, thereby allowing DDM lower-level security in Quickr. The working view contains a number of actions and sub-actions that can be used to adjust the DDM taxonomy to fit client needs in Quickr. If additional custom code is needed to complete other tasks, refer to the documentation provided with the Quickr Migrator Tool to understand specific field values, parameters and how these can be changed using custom Domino agents and Lotus Script.
10.3.5 Quickr preview view

This view, shown in Figure 10-10, displays a preview of the Quickr Place with the modifications made in the working view. Plus sign (+) icons indicate that the section is expandable.

Important: Refer to the Quickr Migrator Tool documentation for recommendations on how to work with taxonomy documents and manipulate fields within the working taxonomy document.
Migration Tool: Operational scenarios

In this chapter, we provide a working example of a scenario migration using the Lotus Quickr Migrator: Domino Document Manager Edition (hereafter referred to as the Migration Tool). For the sake of covering a broad set of migration criteria, we will use two different scenarios for moving our ITSO example library into Lotus Quickr.

- The first scenario is a straightforward move from Domino Document Manager (DDM) to Quickr. We will change little in security and taxonomy and simply recreate a similar structure as in DDM.
- In the second scenario, we move the same ITSO library, but we show some more complex options and highlight some special circumstances and situations you might encounter.

11.1 General migration considerations

There are several things to keep in mind while migrating from DDM to Quickr. Some of these already have been named in other chapters. Keep in mind the following issues during the migration:

▶ Time aspect

Key in all migrations is to remember that the time you need to do the migration can vary with the amount of data you are migrating. Even a small migration can take several hours to complete. Therefore, make sure you have enough time to do the migration.

▶ Data consistency and data security

Make sure you take adequate measurements to protect your data during the migration. Do not allow users to add or change anything during the migration as that can cause data inconsistencies. Another reason for not allowing your users access during the migration is that security is not fully implemented on the documents migrated into Quickr until the final step. It is recommended that during the migration, you lock users out of the target Quickr library to avoid accidental data exposure.

Keep in mind that DDM and Quickr work differently. Things like the filenames of attachments, or version numbers can change during the migration. Make sure you are aware of the consequences of this for your environment, especially if other applications or procedures are in place that use the documents.

▶ Communication and training

Make sure your users are aware of what is going to happen and how to work with their new Quickr environment before you migrate. Going back is almost never an option, so make sure you have discussed everything with the users and make sure the users are aware of changes, new functionality, and how to access their data after the migration.

▶ Additional information and help

During each of the steps in the export profile, help text explaining what will happen in that step is displayed below the Process button on the export profile. Read these messages carefully. Also, check the product site for additional information and documentation:

http://ibm.com/software/lotus/products/dominodocumentmgr/

**Attention:** The Lotus Quickr Migrator: Domino Document Manager Edition is available (at no charge) through trained IBM personnel or IBM Business Partners. Consult your IBM representative for more information about obtaining the tool and assistance with your migration planning, or visit the following Web page:

http://ibm.com/software/lotus/products/dominodocumentmgr/

11.2 Scenario I: Straight forward migration

This scenario describes a situation in which you migrate the complete ITSO DDM library into a single Quickr Place without making any major changes to either the taxonomy or the security structure. The one change that will be made is that security will be set on what was previously the file room level to ensure that each global region has access to its own section.

Figure 11-1 on page 189 illustrates the sitemap in DDM of the ITSO library that will be migrated to Quickr.
11.2.1 Preparation

The first step of any migration should be to determine what you want to migrate and clean up your current DDM environment.

1. Remove anything you do not want to migrate, make sure all documents are checked in, and make sure you have a plan as to how you want to create the new situation. Chapter 7, “Hierarchical and structural considerations between Domino Document Manager and Quickr” on page 89 and Chapter 8, “Design element review” on page 123, both focused on comparing the DDM and Quickr functionality, can help you in doing this.

2. Limit access to the DDM environment so no one can work in it while you are migrating.

3. Keep in mind the time required for the migration. Migrating from DDM to Quickr requires many actions by the system and depending on the size of your DDM environment, take quite a while. To get an idea on how much time the migration requires and how it all works, first test with a limited portion of your environment. For instance, by migrating a single binder or a smaller file cabinet first would give you an idea of what to expect in terms of time and actions required.

4. Keep in mind that during migration you will have to take security measurements to prevent unauthorized access to the data. During the migration process documents in Quickr do not have all of the security settings they had in DDM until the process has completed. Make sure that overall access to the Quickr servers and databases has been restricted to the servers and the administrative ID performing the migration.
11.2.2 Installing the migration tools

The next step in any migration is to set up the tools required for the migration. We use the DDM Migration Tool described in 10.1, “Overview of the Lotus Quickr Migrator” on page 178.

The Migration Tool requires two databases, as shown in Figure 11-2.

- DDM Migrator Working NTF database
- DDM Migrator Main database

![Figure 11-2 Databases](image)

Note: Refer to 10.1, “Overview of the Lotus Quickr Migrator” on page 178 for more information about how the tool works. Use the instructions supplied with the Migration Tool to install these.

During the migration, data is moved from the DDM environment to the migrator database and local hard drive before being transferred to the Quickr Place. This means that data is temporarily stored at the location where you installed the Migration Tool. You need to make sure that there is enough disk space for this. Make sure you have checked this in advance.

The migration is recommended to be performed from a client machine that is located as physically close to both servers as possible. A substantial amount of data is transferred over the network and working remotely can cause significant latency and potential data integrity issues. If a mobile computer cannot be plugged into the server room, a virtual machine running the 8.5 client can also be used. As a last resort, the migration code can also be run from the client on the Quickr server itself.

11.2.3 Setting up Quickr

The Migration Tool migrates the data into Quickr, but it does not create the actual Quickr Place. For a successful migration it is important to have the Quickr Place ready. This can either be an existing Quickr Place or a new one. For our example, we create a new place called “ITSO” that will be used to import the data into.

1. Log in to Quickr with sufficient access to create new places and create a new place by selecting New Place.

2. Select the place type you want to use. You can chose to use a template your company has created, but for this scenario we use the standard place template (Figure 11-3 on page 191) supplied with Quickr.
Chapter 11. Migration Tool: Operational scenarios

3. Enter the name and a description for your place and select Create to have the place created, as shown in Figure 11-4.

4. Log in to the new place and make sure it works. You do not have to set security manually. This will be all done by the Migration Tool.
11.2.4 Configuration

The next step is to create a migration profile.

1. Go into the Migrator main database and click Create Export Profile. This will open a profile document in which you can specify the settings for the export. The tool requires a new profile for each export, so give it a descriptive name.

The first tab of the export profile has three sections:

- **Library Information**
  Here you enter the server name and location of the DDM Library from which you want to export data.

- **Working Database**
  This is the database that will be created for this particular migration and holds all the information regarding hierarchy, security and document profiles until they are imported.

- **Quickr Upload**
  Select the name of the server and the name of the place where the data will be migrated into, as shown in Figure 11-5.

![Figure 11-5  Entering the configuration information](image)

Note: If you enter the Quickr server name with "Http://" or "WWW" in front of it, the tool might have problems locating it. Make sure you list the DNS name only.
The second tab, “Export Options” allows you to specify which data you would like to export. In our example we will be exporting the whole library into one Quickr Place, but you could also choose to export only a section of the data (for instance a cabinet, file room, or even a single binder).

- In the “Directory & Path” field, enter the location were the file attachments will be stored during the migration.

**Note:** Make sure this location has enough disk space to hold the data that you are exporting.

Figure 11-6 illustrates the export options.

**Versioning and file naming settings**

Another important setting on this tab is the **File naming** setting. In DDM, documents were always portrayed with the document title. The actual file name was not important as users never saw these. In Quickr, users will see the filenames (attachment names) when using the Connector instead of the profile title, so it is important to have filenames that are recognizable to the users. For now we use the option to use document titles, which means all attachments will be renamed to the title that was set on the document profile.

We are exporting all versions of documents. This means that every version will become a version in Quickr. This also means that when a document had draft versions, these will become full versions in Quickr, because Quickr does not have stored minor versions.

**Note:** Versioning and file naming are important factors to keep in mind while migrating. If you have software or procedures that are dependent on version numbers or specific file naming then make sure you look into the consequences of the migration before doing it.
When you have finished all the fields, the system is ready. Click **Process Current Step** to check your data. See Figure 11-7.

![Figure 11-7 Process the current step](image)

If everything is acceptable, the **Complete Export Profile** action will indicate completion by having a check in front of it, as in Figure 11-8.

![Figure 11-8 Exporting DDM Taxonomy](image)

If an error occurs, check whether all the entered information is correct. Errors usually occur when either the DDM location or Quickr location cannot be found. Correct these path values if necessary and try again.

### 11.2.5 Manipulating the hierarchy and security

You are now ready to proceed to the next step of the migration.

1. To proceed, click **Process Current Step** to activate the “Export DDM Taxonomy” migration step. Depending on how large your library is, this can take a couple of minutes. It will create the migrator working database and documents representing the layers of the DDM taxonomy.

   When finished, you will see that on the first tab of the profile, a link is displayed to the newly created working database. See Figure 11-9 on page 195.
2. Click the link to open the working database.

3. Once in the Working database, go to the “Conversion Working View”. This view allows you to interact with the taxonomy by using the Add, Remove, Rename Level and Convert to Room options, as shown in Figure 11-10.
Each level of the hierarchy is depicted by a single document.

Binders are depicted with a brownish folder icon or a blue icon if they have custom security.

All other layers are portrayed with yellow folder icons.

Creating rooms

If we import the DDM environment directly into Quickr now, all levels would become folders. In this example, we want to set security on the geographic region level.

1. Select the “Americas” folder level and click **Convert to Room**. The folder icon will change to a doorway icon indicating it has become a room within Quickr. See Figure 11-11.

![Figure 11-11 Convert folder to room option](image)

Now that a room has been created, we can illustrate the benefits of setting room level security.

Setting room security

Within the fictitious scenario (ITSO company), let us assume that the departments are organized by global region instead of by city or country because the employees travel a lot and use each others’ resources. Users are therefore allowed access to all binders within their global region. This means that users located in Rio de Janeiro would also be allowed access data in New York. Within DDM however, it was not possible to set security on the global Region level (that is “Americas”) as that was a file room which did not allow for setting security. This meant security had to be set on the file cabinet level (for example, Brazil, USA, and Canada) instead. Each of these file cabinets had the same security groups (Americas Editors, Readers, and Managers) added to it. It would be more logical to set the security just once on the higher level, but as mentioned above, DDM did not allow setting security on the file room level.

In Quickr, however, we can set security on different levels, so we will change the security so it is located on the global regions level instead. This limits the chance of administration errors and is more in line with the actual setup of the fictitious ITSO company’s security model.
1. To set room security in Quickr, open the “Americas” room document. The security settings are located on the second tab. Specify which security you would like to be set on the room.

2. Add the groups “Americas (Editors)”, “Americas Readers (readers)”, and “Americas Managers (managers)”. These rights are set on the room in the new Quickr Place. See Figure 11-12.

3. Repeat these steps for the other global regions (namely APAC and EMEA).

The big advantage of these additional security levels is that where previously the fictitious company ITSO had to maintain security on the country level (even though they were given by Global Region), we can set security directly on the level where it was meant to be.

Remove unwanted levels within the hierarchy

The other step we illustrate here is how to remove the unwanted uncategorized levels. For the sake of example, we are assuming that these are no longer necessary and can be removed by using Remove. The Selected level and all lower levels option will remove the “not categorized” level and default binder at the same time in this example. Do not use this option if there are lower levels (beneath the level you will be removing) that you want to keep. Use Remove - Selected Level only (adjust lower levels) in those cases. See Figure 11-13 for reference.
11.2.6 Exporting the documents from DDM

When you are ready to begin changing and setting your taxonomy, start importing the document profiles.

Go to the DDM Migrator main database and go to your Export profile (it should still be open), and click **Process current Step** to import the document profiles. This may take a long time, so make sure you are confident that everything is ready. Depending on the size of the library you are exporting, this can take several hours to complete. Make sure that no other processes, such as scheduled backups or reboots are planned as these could cause the process to be interrupted half way through. See Figure 11-14.

*Figure 11-14   Creating the document profiles*

Document profiles will be created in your working database and simultaneously all the files will be copied from the DDM environment to the location you specified in the “Path & directory” field on the second tab of your export profile.

Each document is stored under the name that will be used in the new situation. In our example we have chosen to use the profile titles as names so the attachments are already renamed to this value. If there are multiple versions, each version gets a follow up number indicating its rank in the version ranking. See Figure 11-15.

*Figure 11-15   Illustrating where profiles are stored*
The profile documents can be found in the Working database under the “Documents” (all versions) or “Current Documents” (only last versions) views. We will not change anything about the documents in this scenario and let the system import them as they are.

### 11.2.7 Creating the folders and rooms in Quickr

To review where we are: Until now we have performed the following steps:

- Prepared a Quickr Place
- Created an Export Profile
- Imported the taxonomy
- Changed the taxonomy and security
- Imported the document profiles

The next logical step is to actually create the Quickr Containers, in this case, the rooms and folders in which the documents will be placed.

1. To create the Quickr Containers, return to the export profile and click **Process current step**. See Figure 11-16.

![Figure 11-16  Create the Quickr Containers](image)

2. Make sure everything is done correctly by logging into the Quickr Place and checking the folders and rooms. See Figure 11-17.

![Figure 11-17  Verifying that rooms and folders have been created](image)

As you will see, the rooms and folders have been created, but security has not yet been set yet. Setting security will be done as a last step. See 11.2.9, “Set security” on page 200.
11.2.8 Import documents

Now that the folders and rooms are created, the next step is to import the documents. This step can take a considerable amount of time to process—typically similar to the time taken in the Create Document Profiles step.

Go to your export profile and click **Process current step** to activate the step. See Figure 11-18.

![Figure 11-18 Process the current step](image)

When everything is imported, we suggest you review the contents in the Quickr Place, to validate that everything went ok during the import process.

11.2.9 Set security

The last step to perform is to set the security. This will update the security on the Place, documents as well as rooms, if necessary. See Figure 11-19.

![Figure 11-19 Setting security](image)

Ensure the security was set correctly by going into the rooms and checking whether the right groups were added to the correct access rights. For instance, in our example the group "Americas" should be editor in the Americas room, as in Figure 11-20 on page 201.
Once security has been set, this completes our scenario of a straightforward document migration. After verifying the migration, the working database can be deleted and the files stored on the local file system should be destroyed.

11.3 Scenario 2: Exploring more complex migration options

In this next scenario, we will go into some of the more complex possibilities that the Migration Tool offers you while migrating to Quickr. Specifically, we will discuss how to use the Migration Tool and custom code samples to perform customizations in the following areas:

- Manipulating the hierarchy and security
- Using programmatic options to influence migration taxonomy
- Influencing document profiles

As a starting point, we base this on the same fictitious company and library ITSO. In this scenario, however, the company has chosen to redefine their document management hierarchy. They want to create three different places for each of the departments:

- Finance
- Sales
- Legal

Within each Quickr Place, a differentiation will be made between the different branches (cities), so each branch has their own room. Additionally, security will be tightened to prevent users from other branches accessing the data of a different branch.
11.3.1 Preparation

The first step is the same as in scenario 1 (See 11.2.1, “Preparation” on page 189). Check your current DDM environment, clean up, make sure the documents are checked in and verify that no one else but you can work in the library while you are migrating.

11.3.2 Setting up the Quickr Places

The next step is to prepare the three different places we will be using to import the documents. This is done the same way as described in scenario 1. (See 11.2.3, “Setting up Quickr” on page 190 for reference.) Figure 11-21 shows the resulting creation of the Quickr Places.

![Figure 11-21 Preparing the Quickr Places](image)

11.3.3 Configuration within the Migration Tool

To get the data from the one ITSO Library (stored in DDM) into the three different Quickr Places we will need to split the migration. We do this by using three different export profiles instead of simply one. This is necessary as each of the places requires its own export profile.

1. For each of the three departments, create an export profile with the name of the department (ITSO Finance, ITSO Sales, and ITSO Legal). We then adjust each profile for each of the departments.

2. On the first tab we refer to the respective working database and Quickr Place. See Figure 11-22 on page 203. Specify a different working database for each of the export profiles to ensure data is not mixed up.
Chapter 11. Migration Tool: Operational scenarios

3. On the second tab, specify a specific location for the directory and path where the attachments will be stored. See Figure 11-23.

![Figure 11-23](image)

**Figure 11-23  Specific location for the Directory and Path where the attachments will be stored**

*Note:* We recommend not to use the same attachment storage locations or working database for the three different departments, as it could result in unwanted results when migrating.

You can specify different settings for each department but in our situation we will keep all other settings the same.

4. After finishing the profiles, we run the first step on each of the profiles to validate the export profiles.
Through the remainder of this chapter, we concentrate on the ITSO Sales export for the sake of example, but the same actions would be taken for the two other departments as well.

### 11.3.4 Manipulating the hierarchy and security

Within this next section, we will illustrate how to modify the original hierarchy being imported from DDM, and how to modify the security settings for the different levels. The steps we will discuss are as follows:

- Removing levels we do not want to include in this migration
- Removing intermediate levels
- Renaming levels
- Inserting Levels
- Changing the branch levels into rooms and setting security

#### Removing levels we do not want to include in this migration

The first step is to remove all the levels you do not want to import. Because we are working in the ITSO Sales working database, we remove all the Finance and Legal levels and everything below them. For the import of these documents, we will use the other profiles in a later step.

**Note:** For this migration example, we are using this as an opportunity to now split the information between the different departments, and ultimately have this end up in three different Quickr Places. Refer to the background in 11.3, “Scenario 2: Exploring more complex migration options” on page 201. It is for this reason that we are removing all levels not pertaining to the Sales Department.

1. To remove the levels, select one of the Finance or Legal levels and use the **Remove - Selected level and all levels below** option. (See Figure 11-24) Selecting this option is quicker than removing all levels individually.

![Figure 11-24](image-url)  
**Figure 11-24** Removing unwanted levels
Removing intermediate levels

After removing all the Finance, and Legal levels, remove all the intermediate levels we no longer want. The destination Quickr Place will be called "ITSO Sales" and all the Finance and Legal documents will be placed in different Quickr Places. Therefore, the Sales level really does not add any value and can be removed. Additionally, for the sake of our example, the company has decided to remove the Country and the Global Region levels. For the sake of the scenario, we assume that the company has only nine branches worldwide and there are no plans for expansion, so naming the branches by their city name should be fine.

Figure 11-25 highlights the levels that will be removed from the Rio de Janeiro taxonomy.

To remove the levels, perform the following steps:

1. Select the levels one-by-one and click **Remove-Selected level only (adjust other levels)**. This ensures that lower levels that need to stay in place remain.
2. Repeat step 1 for all the global region, country and sales levels.

The resulting taxonomy looks like Figure 11-26.
Renaming levels
Within the fictitious scenario, we will assume that the Cambridge office recently moved to Boston. We will use the Migration Tool to rename its level.

1. Select Cambridge, and select Rename.
2. Enter the new name and click OK. The change will be immediately visible. See Figure 11-27.

![Figure 11-27 Renaming the level](image)

Inserting Levels
In this section, we illustrate how to insert a new level.

In this scenario, we assume that when the DDM environment was set up, the administrator created separate folders per year (2008, 2009, 2010, and so forth) for each of the branches, except for some of the smaller branches. The Cape Town level therefore does not have any subsidiary year folders. All the Cape Town documents have been placed into the Cape Town binder (indicated by the fact that the Cape Town folder consists of a brown folder instead of a yellow folder). If we migrate this to Quickr, all the documents for Cape Town would be put directly into the index of the room instead of into a folder. To maintain a similar hierarchy and structure with the other branches, we will correct this and create a new intermediate level.

To insert the level, perform the following steps:

1. Select Cape Town level and click Insert - Level above. Enter the name as Cape Town.

![Figure 11-28 Inserting level above](image)

This will create a level above the current Cape Town level with the same name.
2. Select the lower Cape Town level (the one with the brown folder icon) and click **Rename Level** to rename it to 2008-2009. See Figure 11-29. By inserting a new level above and renaming existing level, this provides you with a more uniform environment to work with.

![Figure 11-29 Renaming the lower level](image)

### Changing the branch levels into rooms and setting security

In this section, we change the branch level (cities) into rooms to allow specialized security settings in them.

1. Click **Convert to Room** in the view, as described in “Creating rooms” on page 196.
2. Open each of the room documents to set the security on the second tab. We designate the "Americas Managers" group as the managers for the Calgary room, and the "Sales Calgary" group as editors.

![Figure 11-30 Setting room security](image)

### 11.3.5 Using programmatic options to influence migration taxonomy

Until now we have only used the default options to influence settings. You can also create your own programmatic tools to set parameters and determine the settings. In doing so, keep in mind that changing fields can have unwanted results if done incorrectly, so check the Migration Tool documentation for field references and tips before doing anything.
Scenario example

In our example, we want to change all the level 2 names to uppercase (for example, "NEW YORK" instead of "New York"). We could do this by using the Rename option on each of the level 2 documents, but this can get tedious if you need to do it for many documents, because you can only do one document at a time. Therefore, we are going to use an agent to do it all at once.

We could try to determine which fields need to be changed but that is unnecessary, as there is already code for renaming under the Rename button, so we simply took that code from the ddmMigrationWorkingToolsLib script library and modified it.

Example 11-1  Code for renaming taken from the ddmMigrationWorkingToolsLib script library

`'Agent Change Level to Uppercase:

Option Public
Use "ddmMigrationWorkingToolsLib"
Sub Initialize

'*********************************************************************************
'   This agent helps change the name of the taxonomy levels to uppercase
'*********************************************************************************

Dim session As New NotesSession
Dim db As NotesDatabase
Dim view As NotesView
Dim colUD As NotesDocumentCollection
Dim vecol As NotesViewEntryCollection
Dim ve As NotesViewEntry
Dim doc As NotesDocument
Dim docSelected As notesdocument

Dim strNewName As String
Dim intLevel As Integer

Set db=session.CurrentDatabase
Set view=db.GetView("QTaxSettings")
Set colUD=db.UnprocessedDocuments
Set docSelected=colUD.GetFirstDocument

Do While Not docSelected Is Nothing
' determine level, get value and change it to the uppercase variant
intLevel=docSelected.intQuickrLevels(0)
strNewName = Ucase(docSelected.GetFirstItem("textQuickr_" & intLevel).values(0)) 'get the actual level name and change it to uppercase

' update the dispColumn and textQuickr.
Call docSelected.ReplaceItemValue("dispColumn_" & intLevel,strNewName)
Call docSelected.ReplaceItemValue("textQuickr_" & intLevel,strNewName)
docSelected.textQuickrPath=CalcTextQuickrPath(docSelected)

' update ComponentName for this document (needed for rooms to be created with proper name)
docSelected.ComponentName=strNewName

docSelected.Save True,True`
'process other documents that have docSelected in their ancestry tree
Set vecol=view.GetAllEntriesByKey(docSelected.UniversalID)
Set ve=vecol.GetFirstEntry
Do Until ve Is Nothing
Set doc=ve.Document
Call doc.ReplaceItemValue("textQuickr_" & intLevel,strNewname)
doc.textQuickrPath=CalcTextQuickrPath(doc)
doc.Save True,True
Set ve=vecol.GetNextEntry(ve)
Set doc=Nothing
Loop

Run this agent against all the taxonomy documents and the level names will all be set to uppercase text. This is just one example of what you can do using script capabilities. For more possibilities check the Migration Tool documentation.

11.3.6 Influencing document profiles

After finishing the taxonomy changes, go back to the export profile in the Migration Tool database and run the next process step to import all the documents. Make sure you do this after you are satisfied that the taxonomy is structured how you want it to be.

In general, the Migration Tool does not allow you to do much manipulation with individual documents and it does not display any of the field values. Still this does not mean you cannot use other means to achieve your goal. You can develop your own code and script to influence the documents and their profiles. For instance, you could choose to add the indication DDM-Draft to each document that was checked in as a minor version (draft version such as 1.1, 1.2, 1.3, ...) in DDM. In Quickr, all checked in versions will have major version numbers (1, 2, 3, 4, ...) so doing this could help your users identifying which of the old documents were actual major versions and which were minor versions previously in DDM. The Migration Tool documentation includes an appendix in which the different fields and their purposes are mentioned. Refer to this and review it carefully before writing your own code.

Scenario example

As an example, using the context of our ITSO scenario, we will create an agent that will add the year of document creation to the actual filename of each of the selected documents. Users can identify the documents that are in the combined “2008-2009” folders that were included for the cities where there was not a distinction by year previously.

This is complex, as it entails changing both the document profiles in the working database, as well as the actual files that are stored on the hard disk. The following discussion presents an example script (Example 11-2 on page 210) that performs this action.

Upon reviewing the script example, update two separate fields that hold filename information.

- The “DDM_Profile_FileName” field that holds the original filename and,
- The “DDM_FInfo_FilePath0” field that holds the name the associated file has on the file system during the migration.

Initially, you would expect these name values to be the same, but that is not the case. For the export purposes, the system adds the version number of each version of the file to the actual filename (for example, version 1 of the Eula.pdf doc becomes Eula_1.pdf in the migration).
This is necessary to prevent duplicate files when there are multiple versions. It is the reason we have to update both fields separately. Keep in mind that the code shown in Example 11-2 is just a sample. If you are trying this in your own environment, we strongly advise including some error trapping and testing to ensure everything works well prior to running code like this on your production environment.

**Example 11-2  Example of code to rename documents and their profiles**

```vba
Option Public
Option Explicit

Dim session As NotesSession
Dim db As NotesDatabase
Dim col As NotesDocumentCollection
Dim doc As NotesDocument
Dim dateCreated As NotesDateTime

Dim strYear As String
Dim strOldFilename As String
Dim strNewFilename As String
Dim strExtension As String
Dim strNewFilePath As String
Dim strOldFilePath As String

Sub Initialize
    'Process each selected document to include the year the document was created to the Filename
    Set Session=New NotesSession
    Set db=session.CurrentDatabase
    Set col=db.UnprocessedDocuments
    Set doc=col.GetFirstDocument
    Do Until doc Is Nothing
        'test to be sure this document had a file attachment associated with it
        If Not doc.DDM_FInfo_FilePath0(0)="" Or doc.DDM_Profile_FileName(0)="" Then
            Set dateCreated=New NotesDateTime(doc.DDM_ProfileCreateDate(0))
            strYear=Cstr(Year(dateCreated.DateOnly))
            'udpate DDM_Profile_Filename item
            strOldFilename=doc.DDM_Profile_FileName(0)
            strNewFilename=Strleftback(strOldFilename,".")
            strExtension=Strrightback(strOldFilename,".")
            strNewFilename=strNewFilename & "_" & strYear & "." & strExtension
            doc.DDM_Profile_FileName=strNewFileName

            'update DDM_FInfo_FilePath0 Item
            strOldFilePath=doc.DDM_FInfo_FilePath0(0)
            strNewFilePath=Strleftback(strOldFilePath,".")
            strNewFilePath=strNewFilePath & "_" & strYear & "." & strExtension
            doc.DDM_FInfo_FilePath0=strNewFilepath
        End If
    End Do

    'Copy the source file to create a new file with the new name
```
Filecopy strOldFilePath,strNewFilePath

'delete the source file
Kill strOldFilePath

doc.save True,True
End If
Set doc=col.GetNextDocument(doc)
Loop
End Sub

11.3.7 Importing the documents into Quickr

After making all the changes discussed in the previous sections, we perform the last steps of the Export Profile document to create the Quickr containers, import the documents into Quickr and set security. These steps are similar to those used in the scenario 1 example (see 11.2.7, “Creating the folders and rooms in Quickr” on page 199, 11.2.8, “Import documents” on page 200 and 11.2.9, “Set security” on page 200), so we will not go into detail here, but after performing these final steps, the ITSO Sales Quickr environment is created and ready for use. Repeat these steps for the Legal and Finance departments.

An example of how the end result looks is included in Figure 11-31. It shows the Tokyo room with the “2008-2009” folder containing the documents.

![Figure 11-31 A Quickr room after importing documents with the Migration Tool](Image)

Notice the years behind the file names that we added with the agent code (Example 11-2 on page 210) to change the document profiles.

**Document types and simple forms**

In the DDM environment two document types were used to enter the documents into the system:

- Contract
- Generic Document

During the migration, the Migration Tool created simple forms in the Quickr Place and rooms to ensure that you can still view the data that was entered using these forms.
Forms

You can easily create and edit custom forms for authors to fill out. Each form you create will be listed as an option when an author clicks Create...

> Create a new form by clicking the New Form tab on this page.
> Edit an existing form by clicking its name below.

- Contract
- Generic Document

You can change and modify these forms, but you will have to change these for each of the places and rooms where they are used. Simply changing the form used at the place level will not be enough as simple forms are not synchronized between rooms and places like document type subforms were in DDM.

11.4 Migration clean up

When everything is imported and you have checked that everything is set correctly, remove certain remnants from your system.

1. Navigate to the file location where all the attachments are stored and remove those. The file location for the attachments is specified on the second tab of the Export Profile. Leaving the attachments on the system would be a severe security risk, as the information is not secured other than by the access rights of who has access to that specific location. Depending on the security needs of your organization, consider using an advanced file shredding utility to be sure files stored here during the migration are not recoverable.

2. Remove the working database. After the migration is complete, this has served its purpose and is no longer required if the import was done correctly.

   If you want to keep it for a while for reference purposes, make sure you set appropriate ACL access to it. The filename and location of the Working database were specified on the first tab of your Export Profile.

3. The Export Profile document in the Migrator Main database does not hold any sensitive information and can be left there for reference but is no longer necessary.

4. The old DDM environment (in our case the ITSO Library) is also no longer necessary. Make sure you remove access to it to prevent users from working with the old environment instead of the new Quickr environment. Remove the environment completely when you are sure it is no longer required and it has been backed up.
A comprehensive example
A Domino Document Manager to Quickr example using a custom placetype

Sometimes its easiest to learn by example. This chapter walks you through a real-life Domino Document Manager (DDM) to Quickr migration which uses a Quickr custom placetype to mimic the structure and functionality of DDM. This chapter is not intended to be a complete step-by-step tutorial, but rather a detailed overview of what is required to complete a migration from DDM to Lotus Quickr (services for Domino). In some situations, we will provide code snippets, but not necessarily complete code samples.

In this chapter, the following topics are discussed:

- A scenario of an existing DDM library and mapping strategy for moving the library to Quickr.
- An example of how to create a custom Quickr document type form
- An example of how to create a Quickr check-in event.
- A discussion on how to add tagging functionality to your migrated Quickr Library.
- A list of caveats that you need to consider during your migration.
12.1 Overview

You have all of your documents in a DDM library and you want to migrate them to Lotus Quickr. Where do you start?

This chapter covers the steps for converting a simple DDM library to a Quickr Library. As you have learned from other chapters in this book, one set of instructions will not cover all scenarios. Even the simplest DDM libraries have their own set of unique issues. Use this scenario as an example to help you frame the conversion for your DDM to Quickr migrations.

In this chapter, you will see a number of code snippets that you can use to build your own Quickr template (placetype). If you would like the complete ITSO solution, you can download the IBM Global Solutions Directory at the following Web page:

http://www.ibm.com/software/brandcatalog/portal/lotus

12.2 Scenario

Our ITSO DDM sample library has little customization and falls into “Column 1” of the matrix. The library as viewed from the DDM Desktop Enabler is shown in Figure 12-1.
The ITSO DDM Library consists of three file rooms and six file cabinets. Each file cabinet contains yearly binders that are categorized by city and department.

A custom file cabinet template was created that uses all of the document types from the out-of-the-box file cabinet template (filecab.ntf) but with some customization to the contract document type and the DocEvents script library. All of the document types have been enabled for review and approval. One event has been added that renames documents upon checkout to indicate that they are a work in progress.

The conversion details will be covered in subsequent sections. Here’s a summary of what we will be doing:
- Map the library structure from a DDM library to a Quickr Library
- Create a Quickr contract form that mimics the DDM contract document type for capturing metadata
- Convert the DDM checkout event to a Quickr checkout event
- Add tagging functionality to the Quickr Library

### 12.3 Mapping considerations

The first step in any DDM to Quickr Library migration is determining a container mapping strategy. In other words, how do you map from the library/fileroom/file cabinet/binder category/binder structure that is used in the DDM to the teamplace/fileroom/folder structure of Quickr?

Figure 12-2 shows the various containers available to both DDM and Quickr along with the mapping possibilities.

![Figure 12-2 Containers available to both DDM and Quickr](image)

At first glance, you will notice that Quickr has fewer containers than what you find in DDM. In reality, that is not true because a fileroom is simply a category used to further classify a file cabinet and a binder category is a category used to further classify a binder. Quickr supports the concept of subcontainers (subrooms and subfolders), allowing you to mimic your DDM hierarchy in Quickr more closely.
The decisions on how to map containers from DDM to Quickr should not be taken lightly. In determining the most appropriate mapping strategy, you must consider your security and navigational requirements. A detailed discussion of these topics can be found in Chapter 7, “Hierarchical and structural considerations between Domino Document Manager and Quickr” on page 89.

12.4 Outlining the mapping strategy

In our ITSO example, security beyond the file cabinet level is not significant (inheritance is used), so mapping will be relatively straightforward. Here’s our strategy:

1. Map the ITSO DDM library to a Quickr teamplace named ITSO.
2. Map each ITSO DDM fileroom to a Quickr room with the same name as its corresponding fileroom name.
3. Map each ITSO DDM file cabinet to a Quickr folder with the same name as its corresponding file cabinet name.
4. Map each ITSO DDM binder category to Quickr folder with the same name as its corresponding binder category name.
5. Map each ITSO DDM binder to a Quickr folder with the same name as its corresponding binder name.

Once you have mapped out your strategy, you can use the Quickr Migration Tool to perform the actual mapping. See Chapter 11, “Migration Tool: Operational scenarios” on page 187, for details on using the DDM to Quickr Migration Tool.

The picture below shows the ITSO Library mapped to a Quickr Teamplace for part of the DDM library.

![Figure 12-3  ITSO Library mapped to a Quickr Teamplace for part of the DDM library](image)
12.5 DDM mapped to Quickr

You have defined your strategy, set up the Quickr Migration Tool based on your strategy, clicked Go. So what does it look like? Let us take a look at it using the Lotus Quickr Connector Windows Explorer interface.

From a hierarchical perspective, things look similar. You can navigate through your Quickr Library, just as you did in DDM. You recognize the names of your file rooms, file cabinets, binder categories, and binders. The navigational experience may feel a bit clunky at first because visually, you cannot distinguish between folders and subfolders in the way you could between binder categories and binders. Remember that the navigation is virtually identical to navigating through the folder hierarchy on your local hard drive or a network drive using Windows Explorer.

Figure 12-4 shows the ITSO library after it has been mapped to a Quickr teamplace. Compare this snapshot to the DDM snapshot in 12.2, “Scenario” on page 216.

![Figure 12-4 ITSO library after it has been mapped to a Quickr teamplace](image-url)
12.6 DDM customizations

Any DDM implementation could be easily customized to meet a customer’s specific business requirements by making alterations to the base library or file cabinet templates (domdoc.ntf or filecab.ntf). Usually, a designer would make a copy of the base templates and name them accordingly. For the ITSO DDM library, the templates were named domdocitso.ntf and filecabitso.ntf. The ITSO library was created using these two templates.

Ordinarily, a designer would create custom document types or custom binder types to support the implementation. Other customizations would include programatic additions that would execute based on events (like document check in, document check out or the initiation of an approval cycle). Some of these customizations are simple, while others can be complex.

This section outlines the customizations that were made to our ITSO DDM library.

12.7 Contract document type subform

It is common practice to define custom document type subforms for any DDM implementation. These subforms are defined inside the file cabinet template.

The ITSO DDM library uses a custom file cabinet template that contains a custom subform used for gathering all necessary meta data about a contract. The subform holds four metadata fields:

- Manager
- Date
- Priority
- Contract Type

Each field has its own unique characteristics. We will dissect each field, and in a later section, demonstrate how you would replicate the functionality in Quickr.

Figure 12-5 illustrates the subform as viewed from the Domino Designer.
12.7.1 Manager

The manager field is a picklist used to capture the person responsible for the document. This person may or may not be the person inserting the document into the library. The picklist is populated from the Domino Directory and the field value defaults to the name of the logged in user who is creating the document.

12.7.2 Date

The date field simply captures the document date. This may or may not be the date the document is inserted into the library. The value of the date field defaults to today's date.
12.7.3 Priority

The priority field is used to assign some level of priority to a contract. The picklist values (Low, Medium, and High) are hard coded into the picklist design, as shown in Figure 12-7.
12.7.4 Contract type

The contract type picklist allows the user to further classify a contract. The picklist is populated from a separate Domino lookup database by using a `@DBColumn` formula.

In Figure 12-8 you will see the design of the lookup database.

```
<table>
<thead>
<tr>
<th>Document Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidentiality</td>
</tr>
<tr>
<td>Consultant</td>
</tr>
<tr>
<td>Escrow</td>
</tr>
<tr>
<td>Evaluation</td>
</tr>
<tr>
<td>Indemnification</td>
</tr>
<tr>
<td>Lease</td>
</tr>
<tr>
<td>Letter of Intent</td>
</tr>
<tr>
<td>License</td>
</tr>
<tr>
<td>Maintenance</td>
</tr>
<tr>
<td>Master</td>
</tr>
<tr>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>Services</td>
</tr>
<tr>
<td>Statement of Work (SDW)</td>
</tr>
</tbody>
</table>
```

*Figure 12-8  Design of the lookup database*
Within the contract type field, you will see a formula that performs the actual lookup used to populate the picklist as shown in Figure 12-9.

![Formula that performs the actual lookup used to populate the picklist](image)

**Figure 12-9** Formula that performs the actual lookup used to populate the picklist

### 12.8 Library and file cabinet events

DDM provides an extensive event model that enables you to customize processing that occurs during various DDM events. Each event has a corresponding event handler: a blank subroutine that you can populate to provide custom processing.

DDM provided events that occurred at both the library and file cabinet levels. The library database template (`domdoc.ntf`) comes with a script library named LibEvents and the file cabinet template (`filecab.ntf`) comes with a script library named DocEvents.

The event handlers Table 12-1 are found in the LibEvents script library located in the library template; they are also known as LibEvents.

<table>
<thead>
<tr>
<th>LibEvents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SearchStartDisplay</td>
</tr>
<tr>
<td>SearchProcessResults</td>
</tr>
<tr>
<td>SearchEndDisplayFileCabResults</td>
</tr>
<tr>
<td>SearchEndDisplay</td>
</tr>
<tr>
<td>SearchStartDisplayFileCabResults</td>
</tr>
<tr>
<td>QuerySaveDocumentType</td>
</tr>
<tr>
<td>PostSaveDocumentType</td>
</tr>
<tr>
<td>QueryDeleteDocumentType</td>
</tr>
</tbody>
</table>
The event handlers in Table 12-2 are found in the DocEvents script library located in the file cabinet template; they are also known as DocEvents.

<table>
<thead>
<tr>
<th>LibEvents</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuerySaveBinderType</td>
</tr>
<tr>
<td>PostSaveBinderType</td>
</tr>
<tr>
<td>QueryDeleteBinderType</td>
</tr>
<tr>
<td>PostDeleteBinderType</td>
</tr>
<tr>
<td>QuerySaveFileCabinet</td>
</tr>
<tr>
<td>PostSaveFileCabinet</td>
</tr>
<tr>
<td>QueryDeleteFileCabinet</td>
</tr>
<tr>
<td>PostDeleteFileCabinet</td>
</tr>
<tr>
<td>PostDeleteDocumentType</td>
</tr>
<tr>
<td>QuerySaveReplica</td>
</tr>
<tr>
<td>PostSaveReplica</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DocEvents</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryCheckOut</td>
</tr>
<tr>
<td>PostCheckOut</td>
</tr>
<tr>
<td>QueryCheckIn</td>
</tr>
<tr>
<td>PostCheckIn</td>
</tr>
<tr>
<td>QueryAddToBinder</td>
</tr>
<tr>
<td>PostAddToBinder</td>
</tr>
<tr>
<td>QueryDeleteDocument</td>
</tr>
<tr>
<td>PostDeleteDocument</td>
</tr>
<tr>
<td>QuerySubmitReview</td>
</tr>
<tr>
<td>QuerySubmitApproval</td>
</tr>
<tr>
<td>GetEventErr</td>
</tr>
<tr>
<td>PostReviewComplete</td>
</tr>
<tr>
<td>PostApprovalComplete</td>
</tr>
<tr>
<td>QuerySetupReview</td>
</tr>
<tr>
<td>QuerySetupApproval</td>
</tr>
</tbody>
</table>

**Why use DDM events?**

There are many reasons why you would use LibEvents and DocEvents. For example, you may use the PostApprovalComplete DocEvent to automatically generate a PDF rendition of an approved document and publish it to your corporate intranet. You might integrate a third party workflow system into your application using the QuerySubmitReview DocEvent. Or, you
could use the SearchProcessResults LibEvent to format the search results list in a more meaningful way.

Using the PostCheckOut DocEvent in the ITSO library
The ITSO library uses the PostCheckOut DocEvent to append the words ((Work In Progress)) to the title of a document immediately after it is checked out. The event code is shown in Example 12-1.

Example 12-1  PostCheckOut

```vbnet
Sub PostCheckOut(ddoc As DocDocument)
    If Not EELogger Is Nothing And loggingEnabled Then
        Call EELogger.logEntryExit("Sub PostCheckOut" & delimiter & DEScriptLib &
        delimiter & ScriptLibNameDocEvents ,1)
    End If

    If Not EELogger Is Nothing And loggingEnabled Then
        Call EELogger.logEntryExit("Sub PostCheckOut" & delimiter & DEScriptLib &
        delimiter & ScriptLibNameDocEvents ,0)
    End If

    ddoc.Title = ddoc.Title + " ((Work in Progress))"
End Sub
```

Figure 12-10 shows the title of a checked out document.
12.9 Lotus Quickr customizations

Much like the DDM, you can customize Quickr to meet the specific needs of your organization. In this section, we cover how to customize the ITSO Quickr ITSO library to provide functionality similar to that of the DDM ITSO library.

We will be adding functionality to support the contract document type, including lookups to external databases. We add functionality triggered by the check-in event. And finally, we add a tagging capability, a custom view, and a custom theme for our ITSO Quickr Library. The ability to tag documents and create custom themes is not readily available in DDM. It is possible to create custom views in DDM, but the ITSO library did not contain any custom views.

12.10 Creating a custom Quickr theme

A theme controls the look and the layout of a place. That is, where the navigational controls appear, how an element looks when it is selected, background colors, fonts, and so on. Quickr delivers several out-of-the-box themes you can use. When you create a place, you can accept the default theme or change the theme by choosing from a gallery of predefined themes.

Each theme is composed of a group of layouts that define the appearance of specific Lotus Quickr Place components. For example, the layout for an item differs from the layout of a folder, but they will probably share some style elements as part of a common theme. Using a custom theme, you can give your place a strong identity, design it to look like other corporate sites, or supply additional functionality as well as a unique look.

We will be using a custom theme to provide a navigational experience similar to the DDM Web interface. In particular, we create an expandable table of contents that mimics the site map of DDM. Additionally, we will be adding a tagging capability to our migrated place that will give us the ability to create collapsible views.

It is not our intent to cover the entire process for creating a custom Quickr theme. Several articles already exist for doing this. Instead, we will be pointing out the areas of theme customization that are likely to be most important for users migrating their DDM libraries to Quickr.

12.10.1 Adding the site map functionality

DDM Web users have become accustomed to a tree style navigational experience. In DDM, this tree style navigator was known as the site map. This tree style control allows users to quickly navigate through the library to locate documents. A sample Site Map* is shown in Figure 12-11 on page 228.
The default Quickr theme does not offer this type of navigational experience. Fortunately, you can add a similar type of experience by creating a custom theme that uses Quickr’s expandable table of contents component.

Figure 12-12 on page 229 shows the user experience using a custom Quickr theme that uses the expandable table of contents QuickPlaceSkin component.
Example 12-2 shows the code that you would insert into your custom theme files to provide the expandable table of contents functionality.

**Example 12-2  ExpandableTOC QuickPlaceSkin component**

```html
<script type="text/javascript" src="/qphtml/html/common/dynamic_toc.js"></script>

<QuickPlaceSkinComponent
    name=ExpandableTOC
    format={<Item class="roundedBox">}
    prefixHTML={<div id="toc">}
    postfixHTML={</div>}>
12.11 Creating custom Quickr Forms

Lotus Quickr allows you to create custom forms to capture metadata about documents that you are placing into a Quickr Library. Quickr Forms are analogous to DDM document types.

There are several ways to create custom Quickr Forms, but for our purposes, we will focus on two ways:

- Simple forms
- HTML Imported forms

Simple forms require no programming and can be easily created by a common user. But as you might expect, Simple forms contain limited functionality. HTML imported forms, on the other hand, offer endless possibilities for customization, but must be developed using HTML, CSS, and Javascript.

12.11.1 Creating the correspondence form

The base file cabinet template (filecab.ntf) came equipped with a document type called “Correspondence”. This document type contained four fields:

- To
- From
- Subject
- Date

The To and From fields were defined as basic text fields. The Subject field was defined as a picklist with several hard-coded values. The Date field was defined as a date type field.

In Quickr, a place manager or owner can create a custom correspondence form that mimics the functionality of the correspondence document type. A sample custom correspondence form is shown in Figure 12-13 on page 231.
12.11.2 Creating the contract form

The contract form is significantly more complex than the correspondence form and its required functionality cannot be obtained when built as a simple form. Therefore, we will need to construct it as an HTML imported form.

Recall that the DDM contract subform had four metadata fields:

- Manager
- Date
- Priority
- Contract Type

We will need to create each of these fields using HTML. We will use Javascript to deliver the dynamic lookup capability to the DDM lookup database.
The basic contract.html form
If we just wanted to create the basic fields for the contract form, we would create a text file named contract.html with the code shown in Example 12-3.

Example 12-3 contract.html

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<!-- /*
 'Contract Form
 */ -->
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Contract</title>
</head>
<body>
<input type="hidden" name="c_Form" id="c_Form" value="Contract" />
</div>

<form>
  <strong>Title</strong><br />
  <input type="text" id=h_Name name=h_Name style="width:300px;" />
</div>

  <strong>Date</strong><br />
  <input type="text" id="c_Date" name="c_Date" style="background-color:#FFFFE0" />

  <strong>Priority</strong><br />
  <select name="c_Priority" id="c_Priority">
    <option value="">-Select Priority-</option>
  </select>

  <strong>Manager</strong><br />
  <select name="c_Manager" id="c_Manager">
    <option value="">-Select a Manager-</option>
  </select>

  <strong>Contract Type</strong><br />
  <select name="c_ContractType" id="c_ContractType">
    <option value="">-Select Contract Type-</option>
  </select>

  <QUICKPLACECONTROL type="attachment" name="attachment">
</form>
</body>
</html>
```
Once the file has been created, you can import the form into your ITSO Quickr Teamplace by clicking **Customize** and selecting the `contract.html` file, as shown in Figure 12-14.

**Figure 12-14** Clicking **Customize** and selecting the `contract.html` file

**Populating the contract type picklist**

Now that you have created the basic fields for our contract form, populate the contract type picklist with the values from the same lookup database used by the contract subform in our ITSO DDM library.

Domino has direct support for JSON in views with the `&outputformat=JSON` parameter. Using JSON, we can obtain the contents of a view column from an external Domino database.

Example 12-4 on page 234 shows the Javascript that you would add to the `contract.html` file to populate the contract type field.
Example 12-4  Populating the contract type picklist from an external lookup database

```javascript
<script type="text/javascript" src="/qphtml/html/templatecommon/qp_ajax.js"></script>

<dojo.require("dojo.json");

function getContractTypes(){
    var sURL = 'http://QuickrDomDocLookup.nsf/h_ContractType?readviewentries&Count=1000&OutputFormat=JSON';
    dojoIO(sURL, "populateContractTypeField", null, "POST", "text/json", false);
}

function populateContractTypeField(oJSON){
    var sFieldName = "c_ContractType";
    var oField = dojo.byId(sFieldName);
    if(oJSON[@toplevelentries] && oJSON[@toplevelentries] > 0){
        var viewentries = oJSON.viewentry;
        for(var i = 0; i < viewentries.length; i++){
            var entrydata = viewentries[i].entrydata;
            var sValue = returnJSONValue(entrydata[0]).items[0];
            iIndex = oField.options.length;
            oField.options[iIndex] = new Option(sValue, sValue);
            if(eval(sFieldName) == sValue){
                oField.options[iIndex].selected = true;
            }
        }
    }
}

function initCustomRead(){
}

function initCustomEdit(){
    getContractTypes();
}

dojo.addOnLoad(function(){
    if(h_isBeingEdited){
        initCustomEdit();
    }else{
        initCustomRead();
    }
});
</script>

This code can be added to the contract.html file anywhere between the <body></body> tags.

The dojo.addOnLoad function is automatically called when a new, blank contract form is loaded. Because new documents are always placed in edit mode, the initCustomEdit function is called, which in turn calls the getContractTypes function.
The getContractTypes function performs the JSON lookup to the h_ContractType view of the external database and returns the list of items from the first column of the view. Finally, the populateContractTypeField function is called to populate the options for the c_ContractType HTML field.

**Populating the priority picklist**

In the ITSO DDM contract subform, we hard-coded the picklist values directly into the priority field. In our ITSO Quickr teamplace, we populate the picklist from a configuration document that is stored in the home folder.

![Figure 12-15 Populating the picklist from a configuration document that is stored in the home folder](image)

**Note:** The priority configuration document was created using a custom HTML imported form. We will not discuss the specifics of how that form was created.

We use JSON to locate the priority values from a view. This view was placed directly in the ITSO Main.nsf database and the design is shown in Figure 12-16.

![Figure 12-16 Getting at the priority values from a view](image)
The code for the Priority lookup is shown in Example 12-5. Notice that the code is slightly different because we are not reading the picklist values from the view column. Instead, we are pulling the configuration document from the view and then splitting the values stored in the priorities field into separate values which in turn, are used to populate the picklist.

**Example 12-5  Populating the priority picklist from an internal configuration database**

```javascript
<script type="text/javascript" src="/qphtml/html/templatecommon/qp_ajax.js"></script>
<script type="text/javascript"/>

dojo.require("dojo.json");

function getPriorities(){
  var sURL = '
    ../../../Main.nsf/s_Priorities?ReadViewEntries&count=1&OutputFormat=JSON';
  dojoIO(sURL, 'populatePriorityFields', null, "POST", "text/json", false);
}

function populatePriorityFields(oJSON){
  if(oJSON["@toplevelentries"] && oJSON["@toplevelentries"] > 0){
    var viewentries = oJSON.viewentry;
    var entrydata = viewentries[0].entrydata;

    aPriorities = returnJSONValue(entrydata[0]).items;
    populateSelectOptions("c_Priority", aPriorities);
  }
}

function populateSelectOptions(sFieldName, aValues){
  var iIndex, oField;
  if(typeof(aValues) == "string"){
    aValues = aValues.split();
  }

  for(var i = 0; i < aValues.length; i++){
    oField = dojo.byId(sFieldName);
    if(oField){
      iIndex = oField.options.length;
      oField.options[iIndex] = new Option(aValues[i], aValues[i]);
      if(eval(sFieldName) == aValues[i]){
        oField.options[iIndex].selected = true;
      }
    }
  }
}

function initCustomRead(){
}

function initCustomEdit(){
  getPriorities();
}

dojo.addOnLoad(function(){

if(h_isBeingEdited){
    initCustomEdit();
} else {
    initCustomRead();
}
}

</script>

Similar to what you saw with the contract type picklist, the dojo.addOnLoad function is automatically called when a new, blank contract form is loaded. Because new documents are always placed in edit mode, the initCustomEdit function is called, which in turn, calls the getPriorities function.

The getPriorities function performs the JSON lookup to the s_Priorities" view of main.nsf and returns the priority configuration document. Finally, the populatePriorityFields function is called to populate the options for the c_Priority HTML field.

**Populating the manager picklist**

Earlier, we discussed the procedure for populating the contract type picklist. Populating the Manager picklist works similar to what we did for the contract type picklist except that we will be pulling the Manager names from the contacts1.nsf database for the Quickr Place. Additionally, we will be defaulting the field value to person currently logged in.

The code for the manager lookup is shown in Example 12-6. You will notice that we are looking into the contacts1.nsf database for the list of possible managers. As we populate the picklist, we check to see if the user is the current logged in user (haiku.userName). If it is, we set that person to the selected picklist value.

**Example 12-6  Populating the manager picklist from contacts1.nsf**

```javascript
<script type="text/javascript"
src="/qphtml/html/templatecommon/qp_ajax.js"></script>
<script type="text/javascript">
  dojo.require("dojo.json");

  function getManagerNames(){
    var sURL = '/..../Contacts1.nsf/h_MembersList?readviewentries&Count=1000&OutputFormat=JSON';
    dojoIO(sURL, "populateManagerNameField", null, "POST", "text/json", false);
  }

  function populateManagerNameField(oJSON){
    var sFieldName = "c_Manager";
    var oField = dojo.byId(sFieldName);
    var tmpIndex = 0;
    if(oJSON["@toplevelentries"] && oJSON["@toplevelentries"] > 0){
      var viewentries = oJSON.viewentry;
      for(var i = 0; i < viewentries.length; i++){
        var entrydata = viewentries[i].entrydata;
        var sValue = returnJSONValue(entrydata[0]).items[0];
        iIndex = oField.options.length;
```
oField.options[iIndex] = new Option(sValue, sValue);
if(eval(sFieldName) == sValue){
    oField.options[iIndex].selected = true;
}
if (sValue == haiku.userName){
    tmpIndex = iIndex;
}
}
oField.options[tmpIndex].selected = true;

function initCustomRead(){
}

function initCustomEdit(){
    getManagerNames();
}
dojo.addOnLoad(function(){
    if(h_isBeingEdited){
        initCustomEdit();
    }else{
        initCustomRead();
    }
});

</script>

12.12 Adding tagging functionality

A tag is a non-hierarchical keyword or term assigned to a piece of information. This kind of metadata helps describe an item and allows it to be found again by browsing or searching. Tagging was popularized by Web sites associated with Web 2.0 and is an important feature of many Web 2.0 services.

DDM users are accustomed to using binder categories to organize binders and to simplify library navigation. Quickr uses folders and subfolders to provide similar functionality.

By adding tagging functionality to our migrated DDM to a Quickr migrated place, we can give users another method of document organization. The example below shows how you could tag documents by company name to provide an additional method of library navigation.
Figure 12-17   Example of adding tagging functionality

To add tagging functionality to our DDM-migrated Quickr Place, perform the following steps:

1. Add a tag field to our custom Quickr Forms
2. Add the expandable tag view form to the Quickr Place
3. Add a tag view to the Quickr Place
4. Update the theme files to show the tag cloud
12.12.1 Adding a tag field to a Quickr custom form

Documents are tagged by setting a value in the tag field when the document is created or edited. Figure 12-18 shows how you would tag a document associated with the contract form.

To tag a document, a tag field must be present on your custom Quickr Form. In our case, we will add the field to the contract form.

The code in Example 12-7 shows the tag field added to the HTML along with the code that is executed during the dojoAddOnLoad function.

Example 12-7 Contract form with tag field

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<!-- /*
"http://www.w3.org/1999/xhtml"
</head>
<]]> /*/
```
function populateTags(xmlDoc){
    var children = 0;
    var entrydata, sTag;
    var viewentries = xmlDoc.getElementsByTagName("viewentry");
    var n_viewentries = viewentries.length;
    var sHTML = '';
    for (var i = 0; i < n_viewentries; i++) {
        children = parseInt(viewentries[i].getAttribute("children"));
        entrydata = viewentries[i].getElementsByTagName("entrydata");
        sTag = returnEntryValue(entrydata[0])[0]; //Tag
        if (sTag != "") {
            if(sHTML != "){
                sHTML += ', ';
            }
            sHTML += '<a href="javascript:addTag(\'' + sTag + '\');" ';
            sHTML += 'title="' + sTag + ' (' + children + ')">' + sTag + '</a>';
        }
    }
    dojo.byId("tagsLinkDiv").innerHTML = sHTML;
}

function addTag(sTag){
    var oTagField = dojo.byId("c_Tags");
    if(sTag != "){
        oTagField.value += (oTagField.value!=''?,:'') + sTag;
    }
}

dojo.addOnLoad(function(){
    if(h_isBeingEdited){
        initCustomEdit();
    }else{
        initCustomRead();
    }
});

function initCustomRead(){
}

function initCustomEdit(){
    dojo.byId("tagsLinkDiv").style.display = "";
}
12.12.2 Creating the tagging view

In the example above, we used a view named s_Tags and a dojoGetXML call to populate a new contract form with the existing tags. This view must be placed in all places and rooms where the contract form is used.

As you can see in Figure 12-19 on page 243, the view is quite simple. The view selects all documents of type contract where the c_Tag field is not empty. The Javascript in the form parses through the results and displays them as options on a contract form that is in edit mode.
The expandable tag view is created using a custom form (called the enhanced view form) that contains all of the logic to create an expandable view document for displaying tagged documents. The enhanced view form was developed by SNAPPS and can be downloaded from the following Web page:

http://QuickrTemplates.com

Once you have added the enhanced view form to your place, you can create a document based off the form and set the column widths to meet your needs. Figure 12-20 on page 244 shows how you would create an enhanced view form to display your documents in a categorized view by tags.
12.12.4 Adding the tag cloud to the custom theme

Your theme files (listfolder.htm, page.htm and headlines.htm) must be updated to display the tag cloud. Typically, the tag cloud is displayed in the left column of the page, but you have the flexibility to display it anywhere within your custom theme files.

The code shown in Example 12-8 will give you an idea of what you would add to your theme files to display tag clouds. The first part shows the Javascript code that you would use to populate the tag cloud. The second part shows a style and associated HTML that you can use for displaying the tag cloud.

Example 12-8  Adding a tag cloud to a theme file

```javascript
<script type="text/javascript" src="/qphtml/html/templatecommon/qp_ajax.js"></script>
<script type="text/javascript">
    var g_TagArray = [];
```
function addTagTheme(sTag){
    var oTagField = dojo.byId("c_Tags");
    if(sTag != ""){
        oTagField.value += (oTagField.value!='?','') + sTag;
    }
}

function displayTags(xmlDoc){
    var viewentries = xmlDoc.getElementsByTagName("viewentry");
    var n_viewentries = viewentries.length;
    var children = 0;
    var entrydata, sTag;
    var sHTML = '';
    var oObject;
    var gFound = false;
    var aTags = [];
    if(typeof(fieldNames.c_Tags) != 'undefined'){
        aTags = fieldNames.c_Tags.split(',');
    }
    for (var i = 0; i < n_viewentries; i++) {
        children = parseInt(viewentries[i].getAttribute("children"));
        entrydata = viewentries[i].getElementsByTagName("entrydata");
        sTag = returnEntryValue(entrydata[0])[0]; //Tag
        if (sTag != "") {
            oObject = {
                "value": sTag,
                "children": children
            };
            g_TagArray.push(oObject); //Add to global array
            gFound = false;
            for (var a=0; a < aTags.length; a++){
                if(sTag == aTags[a]){
                    gFound = true;
                    break;
                }
            }
            if(sHTML != ""){
                sHTML += ', ';
            }
            sHTML += (gFound?'<strong>'+sTag+'</strong>':sTag) + ' (' + children + ');
        }
    }
    dojo.byId("theme_tags").innerHTML = sHTML;
}

function initThemeTags(){
    if(typeof(fieldNames.h_SetReadScene) != 'undefined' &&
    fieldNames.h_SetReadScene == 'h_StdPageRead'){
        dojo.byId("theme_tags").style.display = ''; //Get Tags from s_Tags View
        var sURL = '..','/s_Tags?ReadViewEntries&CollapseView&Count=1000';
        dojoGetXML(sURL, 'displayTags');
    }
12.13 Capturing a Quickr event

With Lotus Quickr V8.2 or later, you are able to intercept and modify certain events. This allows you to log or modify certain operations. For example, you can create a hook that captures the document checkout and document checkin events. For each event, you can create a hook that executes both before and after the event occurs. A complete list of events can be found in Appendix D, “Reference hook methods for Lotus Quickr” on page 265.

In this section, we will discuss how you would replicate the DDMs PostCheckOut event in Quickr.

12.13.1 Two ways of creating hooks for Quickr events

You can use either C/C++ or Domino agents to intercept various events. Before writing hook interfaces, you need to decide which method is most appropriate for your environment. You also can combine C/C++ code and Agent code on one server.

Using C/C++ for hooking into the events is the fastest method and reduces the impact on the server to a minimum. When writing C/C++ hooks, make sure the C/C++ code is clean and error codes are checked properly, because a crash in your C/C++ hook will crash the Quickr Domino server as well.

Using placebots (Domino Agents) to hook into the events is easy to use. You will also be able to modify existing hooks without restarting the Lotus Quickr server to take effect. You can choose LotusScript, Java, or @Functions for your hooks. Placebots or Domino agents do not perform as fast as C/C++ code does, however the risk of bringing down the server due to incorrect code is minimized to a minimum.

12.13.2 Creating a placebot hook for Document check out

To create a placebot hook, add a placebot to a place from the WebUI or create a Domino agent in a Domino database using Domino Designer. Keep in mind that all users must have Read access to this place or database. The placebot (or agent) has to be named according to the hook method you want to intercept if you want it being called before and after the operation is being performed. To execute the placebot only BEFORE the operation is performed, append _Before to the agent name. To execute the placebot only after the operation is performed, append _After to the agent name.

In our case, we want to capture the document check out event and append the words ((Work in Progress)) to the end of the document title. To do this, create a Lotusscript placebot hook named ServicesLockDocument_After. The code for this placebot is shown in Example 12-9.

Example 12-9 ServicesLockDocument_After placebot

```
Sub Initialize
'--------------- get document context ------------------------------------
Dim session As NotesSession
Set session = New NotesSession
Dim doc As NotesDocument
Set doc = session.DocumentContext

doc.h_Name = doc.h_Name(0) + " ((Work in Progress))"
Call doc.Save( False, False )

End Sub
```
12.13.3 Registering the placebot hook

To register the placebot hook add this place to the notes.ini of the Quickr server by using the LotusQuickrPlacebotDb environment variable. For example: LotusQuickrPlacebotDb1=LotusQuickr\itso\Main.nsf. Multiple databases can be specified by incrementing the number after LotusQuickrPlacebotDb. The numbers start with 1 and has to be incremented for each module by one. Do not skip numbers, because this will not activate the ones after the skipped number.

After installing the placebot into your place and registering the place in the notes.ini file, restart your Domino server.

Figure 12-21 shows the title of a checked out document after the Quickr ServicesLockDocument event is triggered.

Figure 12-21   Title of a checked out document
12.14 Caveats and functionality gaps

This section is included to give you information regarding the gaps in functionality between DDM and Quickr. There are several areas of functionality that are commonly used in DDM implementations that are not readily addressed in Quickr 8.2. This section identifies those gaps so you can plan accordingly.

12.14.1 Custom document types

The foundation of any DDM implementation is the set of custom document types that are used to classify each and every document that is placed in the library. In a DDM library, users had a tremendous amount of flexibility in how they designed their custom document types. Using LotusScript, designers could populate picklists from keyword databases, define fields as text, numeric, date, and so forth, and validate data at entry time.

Much of the same functionality can be replicated in Quickr by creating custom forms. However, the form designer must be skilled in HTML, CSS, and JavaScript. It will take your LotusScript Developers some time to master these development skills if they do not already have them.

12.14.2 Workflow (Review and approval)

**DDM**

With DDM, you can set up a document for review or approval, manage the review cycle, and perform the review. When the review cycle is complete, the initiator can delete the review copies and all edits and comments associated with them.

Any draft document can be submitted to the review or the approval process. Reviewers and approvers are notified by e-mail.

Review and approval cycles can be setup to follow a serial or a parallel review process. Under a serial review, all reviewers can edit and comment on the document in sequence in the order established by the initiator. In a parallel review, each reviewer edits and comments on the document at the same time.

**Quickr**

Like DDM, the built-in Quickr workflow capabilities are configured on a form by form basis. Quickr supports three types of workflows, described in Table 12-3 on page 250.
Table 12-3  Workflows supported

<table>
<thead>
<tr>
<th>Workflow</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple SubmitWorkflow</td>
<td>When you use the Simple Submit option, the author has only one publication option: he or she can publish the content (by clicking a button with a name you specify) or cancel its publication (by clicking Cancel). The content appears in a folder you specify. Use this option when you do not want content created with the form to undergo review; when you do not want members who use the form to have the option of saving content in draft mode; and when you want the ability to rename the button members use to publish.</td>
</tr>
<tr>
<td>Editor-In-Chief Workflow</td>
<td>Use this workflow type if you want a particular member to review each item created with the form.</td>
</tr>
<tr>
<td>Approval Workflow</td>
<td>Use this workflow type if you want more than one member to review each item created with the form. You set up the order in which the reviewers review the content.</td>
</tr>
<tr>
<td>Multiple Editors Workflow</td>
<td>Use this workflow type to grant all members who have author privileges in the current place/room the ability to edit each item created with the form.</td>
</tr>
<tr>
<td>No Special Workflow</td>
<td>Choose this if you want to allow members who create content with the form to publish immediately -- in either final or draft form -- without approval from any other member in your organization.</td>
</tr>
</tbody>
</table>

The differences between the DDM and the Quickr workflow functionality are as follows:

- Quickr workflow is always serial, meaning that a document cannot be reviewed or approved by two people simultaneously. In other words, Quickr does not support parallel workflow.
- You cannot place time limits and reminders on documents placed in a review or approval cycle.
- Drafts are private and cannot be edited by multiple people.
- Approvers are defined on the form and not individually setup for each document that is created based on the form.

12.14.3 Quickr connectors

The DDM Desktop Enabler is powerful and virtually every DDM implementation uses the Desktop Enabler in some fashion. The Enabler provides the following areas of functionality:

- A Windows Explorer user interface to the DDM Library
- A method for directly interacting with a DDM library using the File menu of virtually any Windows application that creates and displays documents (that is, ODMA or Office Dialogs)
- The ability to interact with Lotus Notes or Outlook e-mail.

In general, the functionality offered by the Desktop Enabler is equivalent to the functionality offered in the DDM WebUI and the DDM Notes user interface.
Quickr connectors provide the same two areas of functionality, but fall short of what is available in the Quickr WebUI. In particular the Quickr Connectors offer the following features:

- Do not allow the user to set or modify document or folder security.
- Have no provisions for searching a document library.
- Do not support document-centric applications outside of the MS Office and Symphony product suites. In other words, you cannot interact with a Quickr Place from the file menu of applications like Adobe®, WordPad or Notepad.
- Only allow you access to Quickr from a custom menu bar, not from the File menu.
- Will not surface metadata and field logic for custom HTML imported forms.
- Lack the ability to check out documents and take them offline.
- Fields setup as required are not respected.

Although additional functionality is expected in future Quickr releases, this is the current state as of Quickr 8.2. Therefore, you will need to consider the impact these gaps in functionality as you plan your DDM to Quickr migration.

12.14.4 Security model

DDM security was built upon an inheritance model. In other words, a document generally inherited its security parameters from its parent binder, while a binder generally inherited its security parameters from its parent cabinet. If a user did not want a document to inherit its security, he or she could impose custom security upon the document under the parameters allowed by the binder in which it was placed.

At Quickr 8.2, folders are unsecured meaning that users must explicitly impose security at the document level. Short of document level security, you must rely on place or room level security to govern document access.

Security is a major concern in any implementation, so it must be well understood during your migration planning. See Appendix A, “Domino Document Manager Site Map and Security Analysis Tool” on page 253 to help you analyze your security needs.

12.14.5 Collaborative authoring and versioning

A major feature of the DDM is the ability for several authors to work collaboratively on a draft document before publishing it as a final version. This functionality is commonly referred to as maintaining major and minor versions.

Quickr does allow for versioning, but does not allow users to manage document drafts between versions.

As you plan your migration, make sure you give careful consideration to your business requirements around major and minor versions.

12.14.6 Searching

DDM allows library designers to create custom search forms that can be used to conduct parameterized searches. With Quickr, a skilled HTML, CSS, and Java designer would be required to create similar functionality.
12.14.7 Performance issues using hook event placebots

DDM events are implemented in Domino script libraries. Using LotusScript, you can easily write custom code to perform actions when an event is triggered. Because they are implemented as inline server-based scripts, they run efficiently on the DDM server.

Quickr allow you to implement hooks using placebots (Domino agents) that are triggered based on an event within Quickr. Because they are triggered by user behavior, it is difficult to anticipate the load they will put on the server. Additionally, Domino agents consume more processing power than inline code. So, you will need to carefully analyze your implementation needs to ensure that the Quickr event model scales to your environment.

If, for example, you are a 500-person pharmaceutical company using DDM to track FDA documentation, and you are logging every check in and check out and review and approval event to a separate audit database, you might see as many as 500 DomDoc events called per hour. Using Quickr events to accomplish the same functionality could mean triggering 500 agent request per hour. Obviously, this would likely be a serious drain on your server resources.

On the other hand, if you are a 500-person law firm using DDM to manage typical office and client documentation where the only events you use are for submitting occasional documents to a third party review and approval application (say 25/day), the Quickr Placebot event model may work perfectly fine.

The bottom line is that when developing your migration strategy for porting your application from DDM to Quickr, you must pay special attention to how your application uses events. You will need to determine what events the application uses, how often the events are triggered and the amount of processing done by the events.

12.15 Creating a placetype

You were able to customize the DDM by customizing the library and file cabinet templates. With Quickr, you first customize a place and then create a placetype from the place to make it reusable.

To create a reusable place, you must allow a placetype to be created from the place. A placetype is a template, or blueprint, for a particular place. A placetype contains all the elements and customizations of that place. When users create a new place, they are given a list of placetypes to base their new place on.

You must have manager access to a place to allow it to be a placetype. With manager access, you can allow a placetype to be created from a place, and specify some aspects of the placetype, but only an administrator can create a placetype from the place.

If you created a place from a placetype, your place can be regularly updated with any new or changed elements and customizations from that placetype.

For details on how to create a placetype, refer to the Lotus Quickr services for Lotus Domino Administrator's Guide.
Domino Document Manager Site Map and Security Analysis Tool

This appendix discusses the Domino Document Manager Site Map Security Tool, which can help you review and determine both the Domino Document Manager (DDM) hierarchy and access rights of a particular document manager library.
Overview of Domino Document Manager Site Map and Security Analysis Tool

The Domino Document Manager Site Map was originally by Dirk Bomans, ISSL, created for Lotus Domino.Doc 3. It returns DDM library structure and access control lists on file cabinet and binder levels. A sample report is shown in Figure A-1.

![Sample report](image)

It consists of a form and agent, which runs through the lib structure, generates a text report and opens it once created. Reports need to be generated for each DDM library separately. Once generated, a report is placed in the library and it is available in the SysAdmin view.

**Note:** The Security Site Map tool is based on domdoc.ntf v7 template, and has been tested on IBM Lotus Domino Document Manager 7.0.
Installing the tool

Domino Document Manager Site Map Security Tool functionality can be added either to a particular library or to all libraries managed by DDM. The following instructions will help you to make the tool available to all libraries, by modifying the domdoc.ntf template.

1. Download domdoc-mod.ntf (contained within DDM-security_site_map.zip) from the FTP site associated with this IBM Redbooks publication. (See Appendix E, “Additional material” on page 289 for specific instructions on how to download this from the site.)

2. Open the Notes Client and login as a Domino administrator,

3. Shut down the Domino server

4. Create a backup of your current domdoc.ntf template and store it outside the Domino data directory,

5. Open a domdoc.ntf and domdoc-mod.ntf in Domino Designer,

6. Copy the BuildSecuritySitemap agent from domdoc-mod.ntf agents view to the domdoc.ntf agents view,

7. Copy the SecuritySitemap form from domdoc-mod.ntf forms view to the domdoc.ntf forms view,

8. Replace the administration navigator in domdoc.ntf navigators view, with the administration navigator from domdoc-mod.ntf,

9. Sign the modified domdoc.ntf using current admin ID (select database in IBM Domino Administrator, right-click, choose Sign)

10. Start a Domino server and run a design task (Domino admin console → load design).

Components of the Domino Document Manager Site Map and Security Analysis Tool

To generate a report, open Domino Document Manager library, go to Library administration and click Generate report. A report is then placed in the Library SysAdmin view and automatically opened. In case it is not opened automatically, go to the SysAdmin view (click either Ctrl+Shift+View or, Go to from the Notes Client menu), select SysAdmin, then Site Map Security.

Note: Due to the nature of Domino Document Manager Library design, reports cannot be removed. If you click Generate report again, another report will be created.

The following screens illustrate the components which make up the Security Site Map tool.

Figure A-2 on page 256 illustrates the Security Site Map Form.
Figure A-2  Security Site Map Form

Figure A-3 on page 257 illustrates the Security Site Map LotusScript Agent.
Figure A-3  Security Site Map LotusScript Agent

Figure A-4 displays the Generate Report button, available in the Library Administration Navigator.

Figure A-4  Generate report button
Figure A-5 illustrates the Security Site Map reports and how these are placed in the library once generated.

![Figure A-5](image)

**How and where to download the tool**

We have included a copy of this tool as an available download from the FTP site associated with this IBM Redbooks publication. See Appendix E, “Additional material” on page 289 for specific instructions on how to download this from the site.
Summary questionnaire for technical assessment

This appendix provides a summary questionnaire from the questions raised in Chapter 5, “Building your Plan: Analyze and document your current environment” on page 61.
Summary: Key questions regarding your DDM implementation

Functional overview

► Who are the DDM experts inside and outside your organization (IT Analysts involved in the initial planning and deployment of DDM, IT administrator and developers responsible for maintaining DDM, current DDM users, business partners)?

► Is there existing documentation describing your DDM implementation?
  – Business process flow diagrams
  – Functional and non-functional requirements specifications
  – Use cases
  – Design documentation
  – Security requirements
  – Server architecture documents and diagrams
  – Replication topology diagram
  – Enterprise architecture document that describes DDM integration with other systems

► Who are the subject matter experts regarding the business processes that DDM supports?
  – Business stakeholders
  – Business users

► Is there documentation describing the business process(es) DDM supports?

► What DDM out-of-the-box functionality is being used?

► Document authoring: How are documents created? Do most document have a single author or do multiple users collaborate on the authoring of a document? Are there naming conventions and categorization guidelines for documents? Are documents frequently updated or are most documents fairly static?

► Workflow: Is a review and approval process being used for documents? If so, what document types require review and approval? Is your organization using the review/approval capabilities provided out-of-the-box or are custom workflows being used? Describe the review/approval processes being used for document authoring.

► Major/minor versioning of documents: Is the ability to create working drafts (also known as minor version) being used for certain document types? What is the versioning strategy being used for the various document types?

► Document Publish/Release: What actions, if any, are triggered when a new document version is released? Once released, are documents distributed to other systems or made available to a wider audience?

► Document Archiving: Are documents being archived? If so, what is the archive repository? What is the criteria for archiving? Can documents be retrieved from the archive repository (that is, un-archived)?

► How do users access the DDM libraries?
  – Using the Desktop Enabler from within desktop applications (ODMA-enabled applications such as Microsoft Office applications)
  – From Windows Explorer
  – From the Notes client
  – From a Web browser
  – From a custom client
Have any additional desktop applications been added to Doc Manager Neighborhood (that is, modifications to ddmallowedapps.reg file)?

Mail integration: Do DDM users store their Lotus Notes e-mails in DDM? Do users replace e-mail body text and file attachments with links to DDM documents. Is DDM integrated with Microsoft Outlook?

Sametime integration: Is DDM integrated with Sametime? Is the Who Is Online feature being used? Do users save chat transcripts to DDM?

Discussion Forums: Are document specific discussion threads being used within DDM libraries?

Offline access: Do users make use of the Attache Case to work with documents while offline.

Bookmark: Do user create links to documents in other binders using bookmarks?

Favorites: Do users create “favorite” documents and binders to access from the library?

Search: How do users typically search for documents? Is Lotus Domino Domain Search being used?

What are the DDM nonfunctional requirements?

Implementation Details

How are File Cabinets, Binder and Documents organized in DDM (what is the DDM hierarchy)?

How are File Cabinets, Binder and Documents secured in DDM (what is the DDM security model)?

Are there corporate standards or guidelines that authors follow when working on DDM documents?

What document types are being used and what metadata is associated with each document type?

What are the volumetrics of your DDM implementation?

- Number of users
- Number of libraries
- Number of file cabinets
- Number of binders per file cabinet (average/max)
- Total number of binder (document) databases
- Binder database sizes (average/max)
- Number of documents per binder (average/max)
- Total number of documents
- Total size of DDM managed content

What are the details of the overall system architecture as they relate to DDM

- Software inventory: DDM, Desktop Enabler, Notes clients, Domino servers, browsers, desktop applications
- Are Mac clients being used?
- What platform(s) is DDM deployed on (Windows, i5/OS®, AIX or Solaris)
- If the platform iSeries®, is HSM (Hierarchical Storage Management) being used ("BRMS" agents for archiving/retrieving documents)?
- Server architecture: the hardware, platforms and servers and how they are distributed (include CPUs, RAM, Disk Storage, network types and bandwidth, cluster/fail-over, master replicas)
- Replication topology
- Interfaces between DDM and other applications (events and data flows)
- Links between DDM and other entities within and outside of the organization: intranet, extranet, Internet, eCommerce
- Local and wide area networks
- Geography: the physical locations of users that access DDM directly
- Programming Languages and APIs used for intercalation between DDM and other subsystems (that is, OLE Automation API, Visual Basic, Visual C++, and so forth)

- What external systems provide input to DDM databases and how?
- What external systems consume information from DDM and how?

**Customizations**

- Have any customizations been made to the out-of-the-box DDM templates? Are the customizations documented? Do your libraries or cabinets inherit their design from more than one file cabinet template?
- Does your DDM implementation have custom document types and binder types? If so, what metadata is associated with each?
- Does your DDM implementation have customized design elements (forms, subforms, views, navigators, agents)?
- Does your DDM implementation have custom event handling?
- Is your DDM implementation integrated with other applications? If so, how is this integration implemented?
- Is Lotus Workflow being used?
- Is there any 3rd party (BP) application integration? If so, document the product names and contact names of these integrated applications.
- Are there links to DDM document from other applications?
- Does your DDM implementation have a custom ODMA interface? If so, how is this interface implemented?
- For each customizations, provide details of the customization and a description of the business process(es) it supports.
Scenario priority matrix

This appendix provides a blank priority matrix to be used for your planning and determining your level of complexity in terms of your Domino Document Manager (DDM) implementation. This refers to the concepts discussed in Chapter 4, “Scenarios: Which might apply to you?” on page 45.
Determining your Domino Document Manager complexity level

To help you determine what scenario you will most likely identify with the most, we have created a priority matrix.

The priority matrix (Figure C-1) identifies focus areas of Domino Document Manager on the vertical axis and measures the level of relevance/importance of each focus area on the horizontal. The four items of focus are:

- Customization
- Hierarchy
- Security
- Workflow

The goal in using this priority matrix is to evaluate the appropriate levels and quadrants apply to your organizations’ DDM implementation. Refer to 4.3, “Customer Scenarios” on page 52, where we discuss special considerations based on the combinations of factors.

![Priority matrix](image)

*Figure C-1 Priority matrix*
Reference hook methods for Lotus Quickr

With Lotus Quickr Services for Domino version 8.2 or later, you are able to intercept and modify certain events. This allows you to log or modify certain operations.

This appendix documents the events available in Lotus Quickr.
D.1 Reference: Hook methods for Lotus Quickr

D.1.1 ServicesApproveDraft

This method is called when a workflow document is approved.

Document context
The document being approved.

Parameters
None

D.1.2 ServicesApproveOrCheckinDraft

This method is called when a draft document is either approved or checked in.

Document context
Draft document being approved or checked in.

Parameters
None

D.1.3 ServicesCancelDocument

This method is called when a locked (checked out) document is canceled.

Document context
The draft document being canceled.

Parameters
None

D.1.4 ServicesCheckinDocument

This method is called when a locked (checked out) document is checked in.

Document context
The draft document being checked in.

Parameters
None
D.1.5 ServicesCopyDocument

This method is called when a document is copied to a different location.

**Document context**
The document being copied.

**Parameters**
h_HookParam1: the destination path where document should be copied.

D.1.6 ServicesCopyFolder

This method is called when a folder is copied to a different location.

**Document context**
The folder being copied.

**Parameters**
h_HookParam1: the destination path where document should be copied.

D.1.7 ServicesCreateDraftOrDocument

This method is called when either a draft or published document is being created.

**Document context**
The draft or published document being created.

**Parameters**
- h_HookParam1: XML formatted creation date
- h_HookParam2: XML formatted last modified date
- h_HookParam3: language (that is, "en")
- h_HookParam4: temporary file name of file being uploaded. Replace this file with another one if so desired. If the value is empty, h_HookParam9 and h_HookParam10 will contain the in-memory copy of the file being uploaded.
- h_HookParam5: HuPath formatted location of the Form (document type) used to create this document
- h_HookParam6: "1" if a draft document is created, "0" if document is published directly.
- h_HookParam7: Title
- h_HookParam8: Summary
- h_HookParam9: if value is other than "0", it contains the address pointer (in hex) where the "in-memory-copy" of the file is stored. To replace the file content with something else, provide a temporary file location in h_HookParam4
- h_HookParam10: contains the length of the in-memory copy of the file (in hex format)
D.1.8 ServicesCreateMember

This method is called when a new member is created (8.1 Theme or later only).

**Document context**
Contains the parameters only. Do NOT save this document.

**Parameters**
- `h_HookParam1`: DN name of user or group to be created.
- `h_HookParam2`: Access level to be created (possible values: "Owner", "Manager", "Editor", "Author", "Reader")
- `h_HookParam3`: set to "1" if the DN is a person, otherwise it is a group
- `h_HookParam4`: set to "1" if the DN is a local user (place specific), otherwise it is a directory user (that is, from an LDAP directory)
- `h_HookParam5`: display name used for this user
- `h_HookParam6`: password to set for this user (for local user only)
- `h_HookParam7`: e-mail address of user (for local users only)
- `h_HookParam8`: First Name of the user (for local users only)
- `h_HookParam9`: Last Name of the user (for local users only)
- `h_HookParam10`: Phone number of user (for local users only)
- `h_HookParam11`: Description (for local users only)
- `h_HookParam12`: "1" if it is an update for an existing user (that is, changing roles)
- `h_HookParam13`: Subject of notification mail to be sent for the invitation (set to empty if none should be sent)
- `h_HookParam14`: Body of notification mail to be sent for the invitation (set to empty if none should be sent)

D.1.9 ServicesCreateVersion

This method is called when a Version document is restored.

**Document context**
The version of the document being created.

**Parameters**
- `h_HookParam1`: Comment used to create this version (writable).

D.1.10 ServicesDeleteDocument

This method is called when a document is being deleted.

**Document context**
The document being deleted.

**Parameters**
None
D.1.11 ServicesDeleteMember

This method is called when a member is deleted from a place (for Quickr themes 8.1 or later).

**Document context**
Document containing only the parameters, Do NOT save this document.

**Parameters**
- h_HookParam1: DN name of the user to be deleted
- h_HookParam2: "1" if the DN is a person, otherwise it is a group
- h_HookParam3: "1" if it is a local user or group, otherwise it comes from a directory.

D.1.12 ServicesLockDocument

This method is called when a document is locked (or checked out).

**Document context**
The published document being locked.

**Parameters**
None

D.1.13 ServicesMoveDocument

This method is called when a document is moved to a different location.

**Document context**
The document being moved.

**Parameters**
h_HookParam1: destination path where document should be copied.

D.1.14 ServicesMoveFolder

This method is called when a folder is moved to a different location.

**Document context**
The folder being moved.

**Parameters**
h_HookParam1: destination path where document should be copied.
D.1.15 ServicesRejectDraft

This method is called when a workflow document is rejected.

**Document context**
The draft or published document being modified.

**Parameters**
None

D.1.16 ServicesRenameDocument

This method is called when the filename is being changed to a different name.

**Document context**
The document being renamed.

**Parameters**
- h_Name: original Name
- h_HookParam1: new Name

D.1.17 ServicesRestoreVersion

This method is called when a Version is restored.

**Document context**
The version document to revert.

**Parameters**
- h_PublishedVersionUNID: contains UNID of the published document being replaced with this version

D.1.18 ServicesSubmitDraft

This method is called when a workflow document is submitted for approval.

**Document context**
The document being submitted for approval.

**Parameters**
None
D.1.19 ServicesUnlockDocument

This method is called when a document is unlocked.

Document context
The document to unlock.

Parameters
None

D.1.20 ServicesUpdateDraftContents

This method is called when a draft document's file is updated. Depending on the file size, it will either pass in temporary file location or an in-memory copy.

Document context
The draft or published document being modified.

Parameters

- h_HookParam1: temporary file location where the file being uploaded is kept. Replace this file with another one if so desired. If the value is empty, h_HookParam2 and h_HookParam3 will contain the in-memory copy of the file being uploaded.
- h_HookParam2: if value is other than "0", it contains the address pointer (in hex) where the "in-memory-copy" of the file is stored. To replace the file content with something else, provide a temporary file location in h_HookParam1
- h_HookParam3: contains the length of the in-memory copy of the file (in hex format)

D.1.21 ServicesUpdateDraftOrDocument

This method is called when either a draft or published document's title, description or document type is modified.

Document context
The draft or published document being modified.

Parameters

- h_HookParam2: Title of the document
- h_HookParam3: A HuPath formatted location of the associated Form (document type) of this document
D.1.22  ServicesUpdateFolder

Update folder information (that is, changing folder name and description).

**Document context**
Folder document note.

**Parameters**
- h_HookParam1: folder title
- h_HookParam2: folder description

D.1.23  h_CreateRoom

**Description**
Creates a room as a subroom of the current room with the specified aesthetic set and of the specified room type.

**Arguments**
- h_Name: The name to be assigned to the new room.
- h_AreaType: The name of the template (without any extension) to be used as the template for this room. The template file resides in the "AreaTypes" directory.

**Notes**
The command creates a subroom of the specified name and type. The manager of the new subroom is the user that creates the room. Only managers of the parent room may create a subroom. Another user attempting to execute this command will receive an authentication error.

D.1.24  h_ChangeACL

**Description**
Creates, removes, or modifies new access to a room for a list of users.

**Arguments**
- h_SetEntryNames: Text list of names to have access set for this room. Use the unqualified name "h_Anybody" to change default access to the room.
- h_SetEntryTypes: Parallel text list of types for each entry name provided. Must be one of:
  - h_Person
  - h_Group
- h_SetAccessLevels: Parallel text list of access levels to be assigned to each name in the list. Each entry is one of:
  - h_NoneAccessLevel - remove user(s) from the ACL
  - h_ReadPagesAccessLevel - allow user(s) to read pages
  - h_AddPagesAccessLevel - allow user(s) to add pages
  - h_ManagerAccessLevel - give the user(s) manager access
Notes
This command properly sets the access for the members listed in "h_SetEntryNames" using the following rules:

1. if "h_SetEntryNames" contains "h_Anybody" then default access is changed to "h_SetAccessLevel"

2. For other names in "h_SetEntryNames":
   - if "h_SetAccessLevel" is set to "h_NoneAccessLevel" then any names are removed
   - if "h_SetAccessLevel" is any other valid access level, then the names are set to that access level. This could mean:
     • adding a new entry, or
     • modifying an existing entry

3. If this is being done on the top-level room then:
   - Granting manager access also grants manager access to the members database
   - Revoking manager access (by specifying a lower access level), removes all access to the members database (unless this is the members group itself).

4. Room references and the area index are updated as necessary to hide rooms to which the user has no access.

D.1.25 h_CreateFolder

Description
Creates a folder.

Arguments
- h_Name: The user visible name of the new folder.
- h_FolderStyle: The style of the new folder. A string of a decimal-formatted number:
  - 1 - Standard List - a chronological list of documents
  - 3 - Headline - a horizontal list of document titles with a preview below
  - 4 - Slide Show - view pages through previous/next actions
  - 5 - Response List - threaded discussion list
  - 7 - Ordered List - a list of documents that can be re-ordered
- h_FolderStorage: The name of the Notes folder associated with this folder.

D.1.26 h_CreateGroup

Description
Creates a group composed of the specified users.

Arguments
- h_Name: Common name of the new group.
- h_Members: Text list containing the common names of each member of the group.
Notes
Once properly created, the system maintains four fields for each group:

- **h_Name**: The user-visible common name of the group -- for example: ProjectTeam.
- **h_SystemName**: The fully qualified, hierarchical name of the group -- for example: CN=ProjectTeam/OU=Acme/O=Certifier.
- **h_Members**: A text list containing the hierarchical names of the group members. Note that this same item is used to submit a member list of common names. The group commands take the common names as input, then convert the names into hierarchical names. For security reasons, these names must remain fully qualified.
- **h_Type**: h_Group

### D.1.27 h_CreateOffice

**Description**
Creates a new Place (formerly known as an office). This means creating a directory for the Place, creating the top level room specified (and any subrooms specified by that room), and creating a manager for the Place.

**Arguments**

- **h_Name**: The name for the new Place. This name becomes the title of the Place main database and the "h_Name" of the "room settings" document stored in the main database for the Place.
- **h_AreaType**: The name of the template (without any extension) to be used as the top level place for this Place. The template file resides in the "AreaTypes" directory.
- **h_AeName**: The name of the aesthetic set to be initially applied to the place created. The aesthetic is applied to the top level room and any sub-rooms created as a result of this command. The aesthetic set referred to should be stored in the Place resources database (resources.nsf).
- **h_UserName**: The name of the person creating this Place who in turn becomes the manager of the Place.
- **h_Password**: The password to be assigned to the manager of the Place.
- **h_EmailAddress**: The e-mail address of the person creating the Place.

### D.1.28 h_CreateUsers

**Description**
Creates members in a Place.

**Arguments**

- **h_SetUserNames**: Text list holding each user name
- **h_SetPasswords**: Text list holding each password.
- **h_SetEmailAddresses**: Text list holding e-mail addresses.
- **h_SetNewLevels**: Text list holding the access level for each new user. If specified, this command will make sure that each user is added to the current room's ACL with the level specified. If specified, the next three parameters can also be used to simultaneously change existing entries. Note that unless this parameter is specified, the following three parameters will have no effect.
Appendix D. Reference hook methods for Lotus Quickr

- **h_SetEntryNames**: Text list of names to have access set for this room. Use the unqualified name "h_Anybody" to change default access to the room.

- **h_SetAccessLevels**: Parallel text list of access levels to be assigned to each name in the list. Each entry is one of:
  - **h_NoneAccessLevel**: remove user(s) from the ACL
  - **h_ReadPagesAccessLevel**: allow user(s) to read pages
  - **h_AddPagesAccessLevel**: allow user(s) to add pages
  - **h_ManagerAccessLevel**: give the user(s) manager access

**Notes**
The parallel text lists must all have the same count for the command to execute.

### D.1.29 h_DeleteGroups

**Description**
Deletes one or more groups from a Place.

**Arguments**
h_SetGroupNames: Text list of group names to delete. Names may be provided as common names.

**Notes**
Removes all listed group names from the system. The names provided may be in hierarchical or common format. SetGroupNames checks every ACL in the Place and removes the name where found.

### D.1.30 h_DeleteUsers

**Description**
Deletes the specified users from the system.

**Arguments**
h_SetUserNames: Text list containing the names of users to be deleted. Names may be provided in "common" format.

**Notes**
Users are removed from the system. For each user, the following is done:
- user is removed from the members database (contacts.nsf)
- user is removed from any groups
- user is removed from every ACL in the Place
D.1.31  h_SetAesthetic

Description
Sets aesthetics on a room by copying from a stored aesthetic set, copying from the parent's aesthetics, or resetting a room to use its parent's aesthetics.

Arguments
- h_AestheticCmd  The type of operation to perform. Value s may be one of:
  - h_Set - copy from named, stored aesthetic found in h_AestheticName
  - h_Copy - copy from parent
  - h_Reset - reset room to use parent's aesthetic settings
- h_AeName  The name of the aesthetic set to be applied to the room. The aesthetic is applied to the room and any sub-rooms created as a result of this command. The aesthetic set referred to is stored in the Resources database (Resources.nsf).

Notes
This command sets all the aesthetics at once: it is not used to set the individual parameters of an aesthetic set on a room.

The aesthetic set specified in the h_Set argument is sought in the configuration database (Haiku.nsf). The name sought is the h_SystemName, sought through the Haiku view. All items in the aesthetic set document that are prefixed with h_Aesthetic are copied into the current document. For this command to work, this document should be the room's settings document (h_RoomSettings).

D.1.32  h_UpdateGroup

Description
Changes the name or membership list of a group.

Arguments
- h_SystemName: Old name for the group. This item is required even if the name is not changing. The name must be in Notes hierarchical format.
- h_Name: New name for the group (or the same name if not changing). Expected in common format (not hierarchical).
- h_Members: Text list containing the list of members for this group. The names in the list can be in common OR hierarchical format (the system will convert all names to hierarchical format during the processing of this command).

Notes
This command updates a group. It makes sure all listed h_Members are in hierarchical format and it keeps the group in sync with the "mirror" group in the server's NAB. The mirror group is needed to perform group expansion when the replicator attempts to access databases. If the group's name has changed, this command checks every ACL in the Place and updates the name.
D.1.33  h_UpdateUser

**Description**
Changes a user's information.

**Arguments**
- h_SystemName: The fully qualified hierarchical name of the user. This is required in all cases - it is used to lookup the user in the member list.
- h_UserName: The name the user logs on with.
- h_FirstName: The first name of the user.
- h_LastName: The last name of the user.
- h_EmailAddress: The e-mail address of the user
- h_SetPassword: The password to be assigned to the user.
- h_PhoneNumber: The phone number of the user

D.1.34  h_DeleteFolder

**Description**
Deletes the current folder.

**Arguments**
None.

D.1.35  h_DeleteOffice

**Description**
Deletes one or more Places.

**Arguments**
- h_HaikusToDelete: Textlist of places to be deleted.

D.1.36  h_DeleteRoom

**Description**
Deletes the current room.

**Arguments**
None
D.1.37  h_DeletePages

**Description**
Deletes pages indicated.

**Arguments**
h_SetDeleteList: A textlist containing the UNIDs of the pages which you want to delete

D.1.38  h_MoveRoom

**Description**
Moves a room within the room hierarchy.

**Arguments**
h_NewRoomName: The nsf name of the room's new parent

D.1.39  h_ReorderRoomArea

**Description**
Use this command to change the room name, set the mail delivery into a specific room, show or hide the Calendar or tutorial, and show or hide security information for reader and authors. This command can also be used to reorder the table of contents.

**Arguments**
- h_Name: the new name of the room
- The following items can have a 0 or 1 as value. 1 means show, 0 means hide:
  - h_SetCalendar - show calendar in this room
  - h_SetTutorial - show tutorial folder in this room
  - h_SetSecurity - show or hide security page to readers and authors
  - h_MailDb - values h_None or nsf name of the db you want to deliver the mail into
- h_ReorderPositionList: list of position numbers
- h_ReorderUnidList: list of corresponding UNIDs (unique document IDs)

**Notes**
The setting of mail delivery can only be done from the top-level room.

D.1.40  h_SendMail

**Description**
Sends an e-mail to the indicated recipients.

**Arguments**
- h_SetPublishEmailAddresses: A textlist of e-mail addresses
- h_SetPublishEmailSubject: The subject line of the e-mail
- h_SetPublishEmailMessage: The message body of the e-mail
D.1.41 h_SpellCheck

**Description**
Spell checks the indicated text.

**Arguments**
- **h_PageText**: The text that was entered in the rich text editor

D.1.42 h_CopyMoveFolder

**Description**
Moves a folder to another room.

**Arguments**
- **h_DestRoomNsfName**: The nsf name of where this folder is to be moved
- **h_Move**: 1 (accepts only 1 for now)

D.1.43 h_ServerBasics

**Description**
This command is used to set the following server-wide settings:
- SameTime enabled/disabled
- Agents enabled/disabled
- UploadFileSize

**Arguments**
- **h_QuickPlaceSameTime**: 1 | 0
- **h_QuickPlaceEnableAgents**: 1 | 0
- **h_QuickPlaceMaxUploadFileSizeInKB**:

D.1.44 h_GetSkinGroupName

**Description**
Gets the Skin Group Name (Theme name) to use when processing the current request.

**Arguments**
- **h_SetSkinGroup**: system name of the skin group to use
D.1.45  h_DeleteSkinGroup

Description
Deletes the current skin group (Theme).

Arguments
None

Notes
The skin group is the current note passed to the QDK event handler.

D.1.46  h_PublishSkinGroup

Description
Publish the current skin group (Theme).

Arguments
- h_Name: name of the Theme
- h_SystemName: system name of the theme
- h_Description: description of the Theme
- h_ThumbImage: image to use when displaying this Theme

12.15.1  h_PublishPlaceType

Description
Creates a new PlaceType.

Arguments
- h_Name: the name of the PlaceType.
- h_SourcePlaceName: the name of the Place that the PlaceType is based on.
- h_SetRefresh: if 1 and the PlaceType exists, the PlaceType will be recreated from the source Place.
- h_PlaceTypeDescription: description of the PlaceType
- h_PlaceTypeImage: image to use when displaying this PlaceType
- h_PlaceInfoURL: URL for additional information about this PlaceType

D.1.47  h_RefreshPlaceTypesGallery

Description
Updates the list of PlaceTypes to match the PlaceTypes available in the LotusQuickr\AreaTypes directory.

Arguments
None
D.1.48  h_DeletePlaceType

Description
Deletes a PlaceType.

Arguments
h_Name: the name of the PlaceType to delete.

D.1.49  h_RefreshPlaceFromPlaceType

Description
Copies any new or updated objects (for example, Rooms, Folder, Forms, Pages, Skins, and so forth) from the PlaceType to the Place.

Arguments
None

Notes
Objects that have been updated in the Place prior to Refreshing from the PlaceType will not be overwritten.

12.15.2  h_SetNewUsersAllowed

Description
When an external directory is hooked up to the Lotus Quickr server, this command controls whether local member entries can be created.

Arguments
h_NewUsersAllowed: The setting value - if 1 then New Users will be allowed, a value of 0 means New Users disallowed.

Notes
When calling this command the user must have Lotus Quickr Server administrator privileges.

D.1.50  h_RunAgent

Description
Run a placebot manually.

Arguments
h_SystemName: The UNID of the Domino agent design document.
D.1.51  h_PreProcessForm

**Description**
Preprocess new Lotus Quickr Pages as required based on which Form they use.

**Arguments**
None

**Notes**
This command performs any required processing on new Lotus Quickr Pages after the user selects which form to use, in the "New..." (form index) scene.

For example, if the user creates a Microsoft Office Form using a Word .DOC file, this file along with any image, HTML or style sheet files created when the .DOC file was imported are copied to the new Lotus Quickr page.

D.1.52  h_DeleteAgent

**Description**
Delete a placebot.

**Arguments**
h_SystemName: The UNID of the Domino agent design document.

**Notes**
If this is a form placebot, the corresponding form is updated so that it no longer references this placebot.

This operation will delete the Domino agent design document.

D.1.53  h_EditAgent

**Description**
Edits an existing placebot.

**Arguments**
The arguments are the same as for h_CreatePlaceBot command. Refer to that command.
D.1.54 h_CheckMemberName

Description
This command will check the input list of names against those members of a Lotus Quickr Place for a conflict.

Arguments
h_SetUserNames: the text list of names to be checked against the place's membership.

Notes
When calling this command the user must have Manager privileges for the place being checked.

D.1.55 h_CreateAgent

Description
Creates a new placebot.

Arguments
- h_Name: The placebot's name assigned by the user
- h_AgentDescription: Description of the placebot assigned by the user
- h_AgentTrigger: Type of placebot:
  - Form = h_AgentTriggerForm
  - Scheduled = h_AgentTriggerSchedule
- h_AgentBaseClass: Java placebots Only: Specifies the class that derives from AgentBase
- h_AgentFormUNID: Form placebots Only: Specifies the UNID of the form
- h_AgentDisable: Scheduled placebots Only: 0 = placebot is disabled
- h_AgentFrequency: Scheduled placebots Only:
  - _Monthly = Once a month
  - _Weekly = Once a week
  - _Daily = Once a day
  - _Hourly = More than Once a day
- h_AgentMonthlyRunDay: Scheduled placebots Only: For Monthly schedule, the day (1 - 31) of the month to run on
- h_AgentWeeklyRunDay: Scheduled placebots Only: For Weekly schedule, the day (1 - 7) of the week to run on
- h_AgentStartTime: Scheduled placebots Only: Time of day to start placebot run (hh:mm)
- h_AgentEndTime: Scheduled placebots Only: Time of day to end placebot run (hh:mm)
- h_AgentStartDate: Scheduled placebots Only: Start placebot run on this date
- h_AgentStopDate: Scheduled placebots Only: Stop placebot run on this date
- h_AgentDailySkipWeekends: Scheduled placebots Only: For Weekly schedule, run on weekends?
- h_AgentHourlySkipWeekends: Scheduled placebots Only: For More than Once Daily schedule, run on weekends?
- **h-AgentHourlyFrequency**: Scheduled placebots Only: For More than Once Daily schedule, specifies number of minutes between runs
- **h-AgentSelection**: Scheduled placebots Only: Process modified pages only or all pages.
  - **h_Modified**: Only modified or new pages
  - **h_All**: All pages
- **h-AgentQueryFolderName**: Scheduled placebots Only: Process only pages specified by this folder name.

### D.1.56 hExternalDirectory

**Description**
Sets (or clears) the external directory for the Lotus Quickr server to use.

**Arguments**
- **h_UserDirName**: The name of the directory, blank for Local or No Directory
- **h_UserDirType**: The type of directory being hooked up. Valid entries are:
  - **h_UserDirTypeLocal**: no directory hooked
  - **h_UserDirTypeLDAP**: LDAP
  - **h_UserDirTypeNAB**: Domino NAB
  - **h_UserDirTypeNT**: NT Domain
- **h_UserDirDisplayName**: This should be set to "dir". If there is an LDAP directory connected, then there are optional additional settings.
- **h_LDAPPort**: The port for the LDAP server, default is "389"
- **h_LDAPBaseDn**: The Base Search hierarchy for all subsequent LDAP searches by Lotus Quickr, default is blank.

**Note**: Some LDAP directories do not allow Base searches on blank.

- **h_LDAPSearchWithCredentials**: Whether to perform LDAP Lookups with or without credentials.
  - A value of "1" will perform the search with the credentials provided.
  - A value of "0" will perform Anonymous lookups.
- **h_LDAPUserName**: The User Name to use if h_LDAPSearchWithCredentials is "1"
- **h_LDAPPassword**: The Password to use if h_LDAPSearchWithCredentials is "1"
- **h_LDAPNarrowToQuickPlace**: If Lotus Quickr Place names map directly to the hierarchy of your LDAP directory, setting this to "1" will limit the scope of LDAP Lookups to the current Lotus Quickr Place Name.

**Notes**
When calling this command the user must have Lotus Quickr Server administrator privileges.
D.1.57  h_CopyAgent

Description
Copies a placebot.

Arguments
- h_ObjToCopy: The UNID of the agent to copy.
- h_DestRoomNsfName: The name of the room (database) to copy to.

Notes
When a form placebot is copied to another database, the corresponding form is NOT also copied to that database.

D.1.58  h_MoveCopyPages

Description
Used to Move or Copy pages (within a room or across rooms).

Arguments
- h_Move: set to ‘1’ if you want to move pages, else set to ‘0’
- h_HandleResponses: set to “1” if you want to move or copy responses along with the page.
  If the value of this argument is NOT “1” only the page will be copied or moved (all its responses, if any, will not be touched).
- h_AllDocs: set to “1” if you want ALL the documents in the specified folder to be copied or moved.
  (value of h_HandleResponses will be ignored).
- h_DestRoomNsfName: the nsf name of the destination room where the pages are to be copied or moved.
- h_DestFolderUNID: the UNID of the destination folder to where the pages are being copied or moved.
  Please set this value to ‘h_Index’ if the destination is the Index or to ‘h_Toc’ if the destination is the Table of Contents
- h_SetDeleteList: text list of the UNIDs of the pages that are to be moved or copied.
D.1.59 h_CopyForm

**Description**
Copies the specified form.

**Arguments**
- h_DestRoomNsfName: Destination Room where the form is to be copied (may be the same room)
- h_ObjToCopy: UNID of the form to be copied

**Notes**
This command copies a form; the user issuing the command request must be a manager of both rooms.

D.1.60 h_EditIndex

**Description**
Customizes the Index. To customize the index columns see h_EditFolder.

**Arguments**
- h_MgrsSeeIndex: set to “1” if you want only managers to see the Index.

D.1.61 h_EditFolder

**Description**
Edits the folder.

**Arguments**
- h_Name: new name (if applicable) of the folder
  - If order of the documents needs to be changed – provide the following two items:
  - h_ReorderUnidList: UNID text list
  - h_ReorderPositionList: corresponding position numbers of the documents pointed to by the UNID list supplied in h_ReorderUnidList
  - If the folder columns need to be modified (you need to add/delete or change the order of the columns) provide the following information:
    - h_ColumnsChanged “1” if you are changing the column information of this folder
    - h_DefaultSortItemName: item name of the default sort column
    - h_SelectedFIdSysNames: text list of the item names (columns) in the order you want them to appear in the folder

**Notes**
The item names you define on the form can be obtained from the page once you have created a page using the form. If you are importing an HTML form the item name is the name you gave to the control. (Item names beginning with h_ and h_Doc are reserved for Lotus Quickr use).
Additionally, the following predefined item names may be used:

- h_Name: to display the name of the page
- h_Modified: to display the modified date of the page
- h_Author: to display the author of the page
- h_Created: to display the creation date of the page
- h_DocAuthor: to display the creator of the page
- h_DocCreated: to display the creation date of the page
- h_DocModified: to display the modified date of the page
- h_DocSize: to display the size of the page
- h_DateControl: if you have a date control on your form and want to display the results of that control in the column
- h_NamePopup: if you have a name pop-up on your form and want to display the results of that control in the column
- h_TimeControl: if you have a time control on your form and want to display the results of that control in the column

D.1.62 h_ConfigureMailSettings

**Description**
Sets the e-mail Domain and SMTP Server for a standalone Lotus Quickr Server.

**Arguments**
- h_SetMailDomain: The e-mail Domain to be set.
- h_SetSMTPServer: The SMTP Server to be set.

**Notes**
The e-mail Domain is set in the following places:
- the NOTES.INI variable '$h_MailDomain'.
- the server's Global Domain document in the fields 'SMTPDomainSuffixes' and 'LocalPrimaryInternetDomain'.

The SMTP Server is set in the following places:
- the server's Server document in the field 'MailRelayHost'.

Appendix D. Reference hook methods for Lotus Quickr 287
12.15.3 h_LookupMembers

**Description**
Search for external directory entries to add to the Lotus Quickr Place.

**Arguments**
- h_LookupString: string to use as the search criteria, may contain * wildcard
- h_Start: start index to use
- h_Count: max number of entries to retrieve
- h_DistinguishedNames: Textlist of the unique names of the entries (unique as they appear in this user directory)
- h_CommonNames: Textlist of the common names of the entries
- h_QPUserNames: Textlist of the user name to use inside this place
- h_Emails: Textlist of the e-mail addresses of the entries
- h_IsGroups: Textlist indicating whether an entry is a Group (= ‘1’) or a Person (= ‘0’)

D.1.63 h_ReorderItems

**Description**
Reorder forms, fields, or pages in a Lotus Quickr Place.

**Arguments**
- h_ReorderUnidList: UNID text list
- h_ReorderPositionList: corresponding position numbers of the documents pointed to by the UNID list supplied in h_ReorderUnidList

D.1.64 h_DeleteAttachments

**Description**
Deletes the specified file attachments from the current document.

**Arguments**
- h_SetDeleteAttachments: Text list containing the names of file attachments to be deleted. This field can contain the name of a single file attachment to delete from the server, or a list of file names to delete.
Additional material

This book refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks publication Web server. Point your Web browser at:

ftp://www.redbooks.ibm.com/redbooks/SG247747

Alternatively, you can go to the IBM Redbooks publication Web site at:

ibm.com/redbooks

Select the Additional materials and open the directory that corresponds with the IBM Redbooks publication form number, SG247747.

Using the Web material

The additional Web material that accompanies this book includes the following files:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDM v7 Templates.zip</td>
<td>The original DDM v. 7 Templates. Use these as a clean baseline for comparing customizations to an existing implementation.</td>
</tr>
<tr>
<td>DDM-security_site_map.zip</td>
<td>This is the template to install and run the Site Map and Security report described in Appendix A, “Domino Document Manager Site Map and Security Analysis Tool” on page 253.</td>
</tr>
</tbody>
</table>
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

Online resources

These Web sites are also relevant as further information sources:

- IBM Product Page for Lotus Domino Document Manager
  http://www-01.ibm.com/software/lotus/products/dominodocumentmgr/
- IBM Developerworks technical resources page for Domino Document Manager

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads
ibm.com/support

IBM Global Services
ibm.com/services
Index

A
adding
  expandable tag view form 243
  site map functionality 227
tag cloud 244
tag field 240
tagging functionality 238
administration/configuration events 74
advanced customization 56
advanced/team document management 54
Alphaworks XMLDiff 141
analyzing customizations 137
API (application program interface) 118
approval workflow 112, 250
archiving 114–115
arguments 278, 285

B
backup and recovery/high availability 36
basic hierarchy
  Domino Document Manager (DDM) 90
  basic hierarchy concepts of Lotus Quickr 92
  basic to moderate customization 54
  basic to moderate workflow 55
binary comparison tools 138
binder 128
binder category 49–50, 65, 91, 184, 217
binder task bar menu customization 130
binder task bar subform 130
binder type 47–48, 68, 72, 129–130
blog functionality in Quickr 16
bookmarking 114
bridging the gap between basic content services and ECM 40
building your plan
  analyze and document your current environment 61
  business goals 8, 63, 79
  business process 24, 35, 37, 46, 54, 62–63, 74, 79–80, 118, 260, 262
  business process management and optimization 37
  business processes
    support by Domino Document Manager 80
  business stakeholders 63, 65, 260

C/C++ code 247
capturing a Quickr event 247
categorization concept 91
categorization example 91
caveats and functionality gaps 249
changing hardware/operating system platforms 84
changing the branch levels into rooms and setting security 207
changing your Enterprise e-mail System 83
check in/check out 110–111
clean sales taxonomy 205
collaboration tools 114–115
collaborative authoring and versioning 251
combinations 52
common areas of customizations 72
comparing 2 DXL files 174
comparing DXL exports from the command line 156
comparison of the document management features between DDM and Lotus Quickr 113
comparison UI 150
ComparisonReportDetail.xslt 161
ComparisonReportShort.xslt 160
completed installation creates a menu entry 147
compliant archival and discovery 37
configuration 192
configuration within the migration tool 202
configuring DXLMagic 145
confirming your business goals and key business decisions 79
connectors 18, 118
containers available to both DDM and Quickr 217
containers available to DDM an Quickr 96
content federation 39
content library 10, 13, 84
content manager
  collaboration 29–30
  Collaboration Edition 30–31, 42
  service 30–31
content security 36
continuum of content and collaboration 7, 35
contract document type subform 220
contract form 217, 231, 233
  basic fields 232
contract type 223
contract.html 232
conversion/working View 184
creating hooks for Quickr events 247
creating rooms 196
creating the contract form 231
creating the correspondence form 230
creating the document profiles 198
creating the folders and rooms in Quickr 199
creating the tagging view 242
custom agents 74, 135
custom document type 20, 72, 76
custom document types 249
customer scenarios 52
customers keeping Domino Document Manager Implementation 55
customers willing to move to IBM Lotus Quickr 55
customization 47, 71, 115–116
  compare process 168
  correspondence form 231
DXLMagic 165
event handling 74
Quickr form 230
Quickr theme 227
reports 165
subform applied to all documents within a file cabinet 69

D
DAOS 105
data relations 95
data types mapping 109
database
  security settings 103
  size 102, 106
  structure 101
  structure comparison 106
databases 190
date 221
date field 230
DDM
  See Domino Document Manager (DDM) 215
default hierarchy 53
default security 53
description 272–276, 280, 282–283, 286, 288
design element 262
  review 8, 31, 123
design element additions and modifications 72
design template name 127
Desktop Enabler 64, 70, 114, 117, 216, 250, 260–261
Diff Tools in XML editors 141
Dim doc 208, 210, 247
directory level 97
discovery model 28, 83
div ID 229, 242
DNS name 192
DOC file 282
DocEvents Script Library 225
document context 247, 266–272
document database 102–103, 126, 128
  replica ID 103, 107
document events 74
document life cycle 63
document management features
  DDM 109
  Quickr 111
document management functionality 109
document model 108
document naming 110, 112
document spread
  databases 107
  folders 107
document store 52, 80
  implementation 53
  scenario 52–53
document title 109–110, 193, 247, 273
  and binder types 72
  approval/review settings 110
document types and simple forms 211
documentation workflow 149
documents 133
Does your organization need ECM? 36
dojo.addOn nLoad 234
Domino agent 115, 247, 252
Domino Attachment Object Store (DAOS) 116
Domino Directory 54–55, 70, 96–97, 221, 262
  complexity level 46
  customizations 220
  data 93, 99, 135, 179, 182
  database structure 101
  document 64, 67, 84, 179, 261–262
    background processing 76
  environment 76, 81, 83, 85, 90, 101, 131, 196
  event 115, 224
  export view 183
  features 114
  functionality 6, 63, 71, 116, 135
  hierarchy 65–66
  hierarchy example 90
  implementation 46, 62, 65, 80, 100, 249, 260–261
  interfaces 117
  library 64, 67, 90, 92, 178, 215, 217, 260–261
  installation 183
    structure 67, 217
  mapped to Quickr 219
migration
  assessment 71
  plan 82–83
  planning process 83
  strategy 84
Migration Tool 178, 181
  model 178, 183
  security 96
  security levels and roles 97
  security model 67
  sitemap 189
  support 65
  system 127, 181
Domino Document Manager Hierarchy 49
Domino.Doc API programming 76
Domino.Doc document model 108
Downloading DXLMagic 145
DXL Magic Installer Welcome window 146
DXL Peek / DXL Explorer 140

E
ECM services for Lotus Quickr software 42
enhanced view 244
Enter a name and description 191
enterprise content 10, 28, 33, 39
  active management 44
  comprehensive picture 39
enterprise content management (ECM) 38–39
In your future? 33
infrastructure 36–37
offering 33, 42
offerings from IBM 42
service 33, 41, 83
additional information 42
capabilities 42
solution 6, 28, 37–38, 64, 83–84
system 29, 36–37, 83
tool 83
enterprise search 36
entitlement to replacement product 4
events 115–116
events in the LibEvents and DocEvents script libraries 75
Example of adding tagging functionality 239
Example of how a Quickr room will appear when you have imported documents using the migrator tool 211
Example of publishing with a link to an ECM System 29
executing the report generation 160
existing DDM implementation
functional requirements 65
Export Profile
second tab 212
export profile 179, 182, 188, 192, 198, 200
Process button 188
export progress display 152
exporting a design from the command line 155
exporting databases to DXL from the command line 153
exporting DDM Taxonomy 194
exporting DXL 173
exporting the documents from Domino Document Manager 198
extend ECM control over content 41

F
fain seamless access to business documents 41
favorites 114
feature comparison
DDM and Lotus Quickr 116
general migration considerations 188
file cabinet 48, 50, 65, 70, 91, 94–95, 124, 126, 183, 189, 196, 217, 219, 254, 261
91
access control lists 254
administration document 103
administrations setting 103
administrator 103
and/or binder level 97
binder definitions 102
data size 107
database 126, 135
database properties dialog 127
document databases 107
level 50, 102, 218
name 102, 107
primary purpose 53
profile document 68, 75, 91
security settings 51
template 72, 127, 217, 220, 262
Template field 132
file room 91
file selection dialog 151
filename 209, 270
FileNet Content Manager Collaboration Edition 29
license 29
software 40, 42
FileNet Content Manager Collaboration Edition 30
FileNet P8 30, 33–34, 39, 43, 83–84, 86
platform 43
storage management engine 44
FileNet Services for Lotus Quickr 30
flow of finding the right element 169
flow of the documentation 150
folder-level security 93, 100
form types used to import the data using the migration tool 212
formal documentation 62
forms and subforms 74
formula that performs the actual lookup used to populate the picklist 224
FTP site 255, 258
available download 258
function initCustomEdit 234, 236
function initCustomRead 234, 236
functional overview 62

G
generating additional Reports from DXL or the Compare Results 157
generating Reports from the compare results from the command line 159
getting at the priority values from a view 235
getting started with ECM services 42
growing need for managing content 34

H
h_LookupMembers 288
h_PublishPlaceType 280
h_SetNewUsersAllowed 281
Hierarchical and Structural Considerations between DDM and Quickr 89
hierarchy 48
Binder level 50
comparison 93
dependency 93
File Cabinet level 50
level 90–91, 93
Library level 50
Quickr 92
structure 90
HTML form 113, 133, 286

I
IBM Content Manager 4, 7, 10, 30, 33, 40, 57, 83, 118
IBM Lotus Quickr 31
IBM Content Manager (formerly DB2 Content Manager) 43
IBM Domino (ID) 255
IBM FileNet
  Content Manager 7, 10, 29, 40, 42, 61
  Content Manager Collaboration Edition 42
  P8 43
  Service 41
IBM FileNet Content Manager
  Collaboration Edition 42
  P8 43
IBM FileNet Content Manager P8 43
IBM Redbook 5, 7
illustration of a typical DDM on-server database structure 101
implementation details 65
import documents 200
importing the documents into Quickr 211
Influencing document profiles 209
inheritance from the template name 128
in-memory copy 267, 271
installing the Migration tools 190
integration 114–115
  DXLMagic in your own code 173
  Lotus Quickr with Lotus Connections and Lotus Sametime 82
  other applications 76
interface comparison 118
Introducing an ECM Solution 83
Introducing DXL Magic as a tool for comparison 143
Introducing Lotus Connections or Lotus Sametime 82
Introduction 3

J
Java designer 251
Javascript 229–230

K
key components 182
key dates 4
key question 61, 65, 260
  complete summary 77

L
LDAP directory 96–97, 268, 284
left navigator 68–69
  library administration 68, 72
  select document types 68, 72
  system profile 69
levels at which you can set security in DDM 97
LibEvents Script Library 224
library 91, 124
library and file cabinet events 224
library database 68–69, 102–103, 124, 128
life cycle management 110–111
lookup database design 223
Lotus Connections 82, 117
Lotus Notes 41, 43, 250
  integration points 83
Lotus Notes or Outlook Email 19
Lotus Quickr xiv, 4, 36, 55, 82, 118, 187, 216, 219,
Index

Domino Document Manager Edition 177
Lotus Sametime 18, 23, 82–83, 114–115
integrated functionality 24
Lotus Symphony 10, 15, 21, 115, 118
Lotus Quickr connector 21

M
manage content to achieve compliance 41
manager field 221
manager picklist 237
manipulating the hierarchy and security 194, 204
manually configuring DXLMagic 147
mapping considerations 217
mapping example 95
mapping example between DDM and Quickr 95
mapping possibilities 96
maximize user acceptance 38
meet diverse organizational needs 39
metadata 111–112
Microsoft Office 10, 15, 30, 41, 112–113
and Lotus Symphony 20
Microsoft Outlook 18, 20, 41, 64, 83, 115, 118, 261
Quickr Connectors 20
migration clean up 212
Migration Tool 8, 99, 109, 177–178, 187, 191
configuration 202
functional and technical overview 187
operational scenarios 187
minimal customizations 53
minimize the burden on IT 38
moderate security 54
moderate to complex hierarchy 54, 56
moderate to complex security 56
moderate to complex workflow 56
multiple user interfaces 115–116

N
navigating to a file using Windows Explorer and selecting
from the menu of options 23
navigators 73
new NotesSession 208, 210, 247
next step 3, 5, 192, 194
no workflow 53
nonfunctional requirements 65
non-static content 94
extent hierarchy model 94
static content 94
Notes Client 70, 72, 103, 112, 117, 255
Notes client 54, 64, 69, 103, 107, 117–118, 133, 140, 145, 255, 260
interface 118
NotesPeek 140
nsf name 278

O
ODMA (Open Document Management API) 76
offline 114, 116
oField.options 234, 236
operating system 43, 70, 181
organize content across the enterprise 41
ot gain 28
oTagField.value 241, 245
other features 114
other IT project considerations 82
other scenarios 57
outlining the mapping strategy 218
Outlook 20
out-of-the-box DDM template 262
Outsourcing your IT Function 84
overview 138, 216
overview of the Lotus Quickr Migrator 178
overview of the migration process 179
overview of the migration tool Lotus Quickr Migrator
Domino Document Manager Edition 177

P
parameters 266–272
performance issues using hook event placebots 252
personal file 6, 10, 25, 35
personal file sharing 25
picklist 109
value 222, 235
place type 10, 129, 132, 190
placebot 116, 248, 281, 283
placebot hook 247–248
placebot hook for document checkout 247
PlaceType 215–216, 280
placetype 252
populating the contract type picklist 233
populating the manager picklist 237
populating the priority picklist 235
possible transition paths from Domino Document Manager 6
preparing the Quickr places 202
principles of end-to-end ECM 38
priority 222
priority field 222
priority matrix 45–47
priority picklist 235–236
priority value 235
process the current step 194, 200
profiles
where they are stored 198
proposed timeline 4
putting it all together 162

Q
Quickr 9–10, 79–80, 109
Quickr + FileNet 29
Quickr Blog feature 11
Quickr connector through Windows 15
Quickr connectors 15, 18, 41–42, 83, 112, 115, 250
file store 28
Quickr container 199, 211
Quickr Containers 199
Quickr form 109, 132, 230
Quickr interfaces 118
Quickr Library 215–216
Quickr migration 8, 215, 218, 251
effort 80
tool 219
Quickr Migrator Tool
components 178
Quickr place 10–11, 92, 95, 135, 181, 188, 190, 227, 237
drop documents 22
resulting creation 202
simple Forms 211
whole library 193
Quickr preview View 185
Quickr preview view 185
Quickr security 97
Quickr security levels and roles 98
Quickr structure 105
Quickr teamplace 218
Quickr Web interface
    additional abilities 118

R
Redbooks Web site 291
    Contact us xvii
registering the placebot hook 248
remove unwanted levels within the hierarchy 197
removing intermediate levels 205
removing levels we do not want to include in this migration 204
removing the uncategorized levels 197
renaming levels 206–207
report screen selection 158
review and approval capability 63, 249, 260
reviewing business goals 63
reviewing implementations 124
role of DDM in supporting the Business Process 62
room 18, 93
RSS feed 16–17
running transformations/Reports 174

S
Sametime 23
saving to a Quickr place 21
scalability 36
Scenario 2
    Exploring more complex migration options 201
scenario example 208–209, 216
scenario I
    Straight forward migration 188
scenarios - Which might apply to you? 45
scheduled placebots 283
script library 115, 126, 134, 208, 217, 224
script type 229, 234
search events 75
search.nsf 105
searching 114–115, 251
security 36, 51, 96
security comparison 99
security dependencies 94
security levels and roles 99
security mapping 99
security mapping table 100
security model 8, 178, 251
dealing with this in a sample migration 196
Migration Tool 177
within an ECM 36
set security 200
setting for spawning new databases 104
setting room security 196, 207
setting security 200
setting security rights 197
setting up DXLMagic 144
setting up Quickr 190
setting up the Quickr Places 202
short overview with changed design elements 161
showing how the categorization is listed in the profile 92
site administration
database 102, 124
database customizations 123–124
site profile 69
SMTP server 287
spawning 103
specific location for the directory and path where the attachments will be stored 203
specify a different working database for each of the export profiles 203
standard team place 12
static content versus non-static content 94
status display 181
steps involved in a typical migration 181
storage cost reduction 36
subform 130, 132, 220, 231
viewed from the Domino Designer 220
subroom 272
summary
    key questions 65, 71, 76
summary questionnaire 259
system profile (library configuration) 70
system security architecture - LDAP (SSL) 70, 262
systems architecture 70

T
tag a document associated with the contract form 240
tag cloud 27, 51, 239, 244
team places 10
Teamstudio Delta 138
templates 24
text list 272–273
title of a checked out document 226, 248
total size 70, 261
transition options 5
transition Plan 4

U
UML diagram 63
UNID 270, 281
UNID list 286, 288
unique document IDs (UNIDS) 278
uniting collaboration with content management 41
user directories 96–97, 99, 288
   Domino Document Manager 89–90
   Lotus Quickr 92–93
user experience 64
user Interface
   between Dom Doc and Quickr 117
user interface 10, 12, 47–48, 64, 71, 114–115, 117, 125, 132
   specific folder type 12
using DXLMagic 149
using programmatic options to influence migration taxonomy 207
using the comparison user interface 150
using the PostCheckOut DocEvent in the ITSO Library 226
using the Quickr Connectors with Microsoft Outlook 20

V
verifying security has been set correctly 201
verifying that rooms and folders have been created 199
versioning 110–111
versioning and filenaming settings 193
versioning strategy 64, 260
view customizations 129
views and folders 73
visual comparison tools 138
volumetrics 70

W
Web 2.0
   service 238
   technology 118
   thought process 51
   world 80
Web browser
   folder content 118
web browser
   search results 75
Web interface 29–30, 115, 118
web interface
   products reply 118
Web UI 29, 48, 93, 112, 132, 135
   page name 113
What changes to look for 142
What is IBM Lotus Quickr? 10
What's in this IBM Redbook 7
Where does your organizational fit into the matrix in terms of hierarchy, security, customization? 80
Why use DDM Events? 225
wiki
   functionality within Quickr 17
   wiki template 17
Windows Explorer 10, 15, 22, 64–65, 112, 114, 219, 250, 260
workflow 51
   type 112, 250
workflow (review and approval) 249
workflow process 38, 51–52
workflows supported 250
workflows supported in Quickr 112
working databases and views into the DDM taxonomy 183

X
XML comparison tools 141
XML schema of comparison results 166
Self Assessment and Strategy Guide for Migrating from Domino Document Manager

This IBM Redbooks publication is intended to help your organization plan and prepare for a migration away from Lotus Domino Document Manager (formerly known as Domino.Doc). This IBM Redbooks publication focuses primarily on analyzing, planning, and preparing with only a small portion devoted to the actual process of migrating to another solution. While there is no exact equivalent to Domino Document Manager (DDM), we discuss IBM Lotus Quickr in detail and illustrate how your organization can achieve many of the same functional goals using Lotus Quickr services for Domino. We introduce the tool for migrating, namely the Lotus Quickr Migrator: Domino Document Manager Edition, and discuss how you can use this for migrating content.

Moving beyond Domino Document Manager requires some new thinking about your organizations' document management and content management needs. In many cases, Lotus Quickr may offer a collaborative solution which can provide significant benefit to how your organization creates, shares and stores content. In other cases, this may present an opportunity for you to re-evaluate if you need a larger scale, more robust enterprise Content Management solution.

This IBM Redbooks Publication is intended to help you in making a well informed decision about the best way forward. We help you to better understand and analyze your current Domino Document Manager environment and then guide you in determining the best options for moving forward. With proper planning and analysis up front, this book will help you recognize and overcome the potential obstacles and challenges, and to help you explore the opportunities opened up by alternate content management solutions.