

Migration Use Cases with the Migration Manager

Covers Tivoli Service Request Manager, CCMDB,
and Tivoli's process automation engine migrations

Learn about Migration Manager
implementation strategy

Experiment with a variety of
real-life scenarios

Review migration
troubleshooting tips



Bradley Downing
Burak Bolek
David A Gutierrez
John Dibble
Kylie Cameron
Sampath Sriramadhesikan
Scott Tyrrell
Thadeu de Russo e Carmo
Vasfi Gucer



International Technical Support Organization

Migration Use Cases with the Migration Manager

January 2011

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

First Edition (January 2011)

This edition applies to IBM Tivoli Change and Configuration Management Database V7.2.0 and V7.2.1, IBM Tivoli Service Request Manager V7.2.0 and V7.2.1, and Tivoli’s process automation engine V7.1.1.6 and V7.1.1.7

© Copyright International Business Machines Corporation 2011. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	ix
Trademarks	x
Preface	xi
The team who wrote this book	xi
Now you can become a published author, too!	xv
Comments welcome	xv
Stay connected to IBM Redbooks	xv
Chapter 1. Migration strategy	1
1.1 Requirements	3
1.2 Choice of tools to support migration	3
1.2.1 Data loading tools	4
1.2.2 Sequence of data loads	6
1.3 Content strategy	7
1.3.1 Dependencies among configuration content	8
1.3.2 Best practice content	9
1.3.3 Integrity of configuration content	10
1.3.4 Content validation	11
1.3.5 Content and source control systems	13
1.4 Migration approach	15
1.4.1 Snapshot mode characteristics	16
1.4.2 Change mode characteristics	16
1.4.3 Hybrid migration strategy	17
1.5 Migration planning	18
1.5.1 Contents of the migration plan	19
1.5.2 Working with migration time frames	24
1.6 Migration Manager reference material	26
Chapter 2. Migrating data dictionary	29
2.1 Requirements	30
2.2 Solution	30
2.3 Configuration applications	31
2.4 Object structures	32
2.5 Migration groups	35
2.6 Package definition	37
2.7 Deployment	47
2.8 Deployment considerations	48
2.8.1 Organization and clearing accounts	48

2.8.2 Crossover and table domains	48
Chapter 3. Migrating security configuration data	51
3.1 Migrating a new security group	52
3.1.1 Requirements	52
3.1.2 Solution	52
3.1.3 Configuration applications	53
3.1.4 Object structures	55
3.1.5 Migration groups	56
3.1.6 Package definition	57
3.1.7 Deployment	59
3.1.8 Deployment considerations	59
3.2 Migrating the conditional user interface	61
3.2.1 Requirements	61
3.2.2 Solution	61
3.2.3 Configuration applications	62
3.2.4 Object structures	65
3.2.5 Migration groups	65
3.2.6 Package definitions	66
3.2.7 Deployment	67
3.2.8 Deployment considerations	68
3.3 Migrating global data restrictions	68
3.3.1 Requirements	69
3.3.2 Solution	69
3.3.3 Configuration applications	69
3.3.4 Object structures	71
3.3.5 Migration groups	71
3.3.6 Package definitions	73
3.3.7 Deployment	75
3.3.8 Deployment considerations	75
3.4 Migrating access definitions and LDAP information	76
3.4.1 Requirements	76
3.4.2 Solution	76
3.4.3 Configuration applications	77
3.4.4 Object structures	77
3.4.5 Migration groups	78
3.4.6 Package definitions	79
3.4.7 Deployment	81
3.4.8 Deployment considerations	81
3.5 Considerations of security migrations	82
Chapter 4. Migrating escalations	83
4.1 Requirements	84

4.2	Solution	84
4.3	Configuration applications	84
4.3.1	Actions	84
4.3.2	People	85
4.3.3	Person groups	85
4.3.4	Roles	85
4.3.5	Communication templates	85
4.3.6	Escalations	85
4.3.7	Other applications	86
4.4	Object structures	87
4.5	Migration groups	89
4.6	Package definitions	92
4.6.1	Defining SQL criteria for an escalation with an action	92
4.6.2	Defining SQL criteria for an escalation with a notification	95
4.7	Deployment	98
4.7.1	Escalations and cron tasks	99
Chapter 5. Migrating applications and Start Centers		109
5.1	Basic application changes and signature options	110
5.1.1	Requirements	110
5.1.2	Solution	112
5.1.3	Configuration applications used for this solution	112
5.1.4	Object structures	112
5.1.5	Migration groups	113
5.1.6	Package definition	114
5.1.7	Deployment	116
5.1.8	Deployment considerations	116
5.2	Complex application changes, including menus, lookups, and global data restrictions	117
5.2.1	Requirements	117
5.2.2	Solution	119
5.2.3	Configuration applications used for this solution	119
5.2.4	Object structures	120
5.2.5	Migration groups	123
5.2.6	Package definitions	126
5.2.7	Deployment	129
5.2.8	Deployment considerations	130
5.3	Migrating a Start Center and a query	131
5.3.1	Requirements	131
5.3.2	Solution	132
5.3.3	Configuration applications used for the solution	132
5.3.4	Object structures	132
5.3.5	Migration groups	134

5.3.6	Package definition	135
5.3.7	Deployment	137
5.3.8	Deployment considerations	138
Chapter 6. Migrating workflows		139
6.1	Migrating specific workflow modifications	140
6.1.1	Requirements	140
6.1.2	Solution	140
6.1.3	Configuration applications used for this solution	141
6.1.4	Object structures	144
6.1.5	Migration groups	144
6.1.6	Package definitions	145
6.1.7	Deployment	146
6.1.8	Deployment considerations	147
6.2	Migrating workflows and all their related content	147
6.2.1	Requirements	147
6.2.2	Solution	148
6.2.3	Configuration applications used for this solution	148
6.2.4	Object structures	148
6.2.5	Migration groups	149
6.2.6	Package definitions	150
6.2.7	Deployment	152
6.2.8	Deployment considerations	152
Chapter 7. Migrating classifications		153
7.1	Scenario	154
7.1.1	Requirements	154
7.1.2	Solution	155
7.2	Configuration applications	155
7.3	Object structure	157
7.4	Migration group	158
7.5	Package definition	159
7.6	Deployment	160
7.7	Deployment considerations	161
Chapter 8. Migrating service offering content		163
8.1	Service offering scenario	165
8.2	Requirements	165
8.3	Solution	165
8.4	Configuration applications	168
8.5	Object structures	170
8.5.1	PMSC_JOBPLAN	170
8.5.2	PMSC_OFFERING	171
8.5.3	PMSC_DOCINFO	171

8.6 Migration groups	172
8.7 Package definitions	173
8.7.1 Defining the SQL definitions	175
8.8 Deployment	176
8.8.1 Deployment considerations	176
Chapter 9. Migrating a service catalog	177
9.1 Requirements	178
9.2 Solution	178
9.3 Configuration applications	180
9.3.1 Security groups	181
9.3.2 Catalogs	181
9.4 Object structures	181
9.5 Migration groups	187
9.6 Package definitions	188
9.6.1 PMSC72_OFFERING	189
9.6.2 PMSC72_CATALOG	193
9.7 Deployment	193
9.7.1 Deployment considerations	194
Chapter 10. Common topics	197
10.1 Comparing Integration Framework and Migration Manager	198
10.1.1 Selecting Integration Framework or Migration Manager	199
10.2 Multiple language considerations	201
10.3 Snapshot package versus Change package	204
10.4 Embedded URLs	208
10.5 Clustered environment considerations	209
10.6 Change tracking and ad hoc reporting	210
10.6.1 When to use Change packages	210
10.6.2 Using Change packages to track development activities	211
10.6.3 Configuring change roles	211
10.6.4 Change package definitions	213
10.6.5 Tracking change events	214
10.6.6 Creating ad hoc reports of change events	216
10.7 Admin mode	221
10.8 Migrating hierarchical data	221
10.9 Setting up logging for the Migration Manager	229
10.10 Start Center visibility into configurations	235
Chapter 11. Troubleshooting	239
11.1 Common migration package failures	240
11.1.1 Package deployment fails due to installation differences	240
11.1.2 Package deployment fails due to incorrect Source designation	243
11.1.3 Package deployment fails due to inconsistent manifest data with	

package file name	243
11.1.4 Package deployment fails due to improperly combined object structures	244
11.1.5 Security configuration migration errors	244
11.1.6 Security group and group reassign errors	245
11.1.7 Impact of invalid or deprecated MAXVARS	246
11.1.8 Attached documents and document types	247
11.2 Methods to solve migration package failures	248
11.2.1 Use of the Logging application	248
11.2.2 Modifying the migration package XML	252
11.3 Techniques to prevent migration package failures	254
11.3.1 Start with identical environments	254
11.3.2 Ensure that you understand migration group and object structure relationships	254
11.3.3 Making separate packages work in a dependent fashion	255
11.3.4 Making use of the SQL Where clause for package groups	255
Appendix A. Additional material	257
Locating the Web material	257
Using the Web material	257
System requirements for downloading the Web material	258
How to use the Web material	258
Related publications	259
Online resources	259
How to get IBM Redbooks publications	260
Help from IBM	260
Index	263

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](http://www.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:

Foundations™	Maximo®	Service Request Manager®
Global Business Services®	Redbooks®	Tivoli®
IBM®	Redbooks (logo)  ®	WebSphere®

The following terms are trademarks of other companies:

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Snapshot, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

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

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

Preface

The Migration Manager enables you to migrate configuration content from one production environment to another. The typical use is to migrate configuration content from a development environment to a test environment and then on to production for the Tivoli® process automation engine and its applications, such as IBM® Tivoli Change and Configuration Management Database (CCMDB) and IBM Tivoli Service Request Manager®. The goal of migration is to ensure that your production environment fully meets the needs of your users.

This IBM Redbooks® publication covers the most common migration use cases with the Migration Manager. Of course, these use cases are only a small subset of the possible migration scenarios that can be performed by the Migration Manager, but they were chosen to be representative of the capabilities of the Migration Manager.

In addition to these use cases, the book presents a migration strategy and a comprehensive chapter about troubleshooting possible migration problems when using the Migration Manager.

We strongly suggest that you read Chapter 1, “Migration strategy” on page 1 first before reading the other chapters. This chapter will give you a good foundation for all of the migration scenarios covered in the book.

This book will be a reference for IT Specialists and IT Architects working on migrating configuration content from one production environment to another using the Migration Manager.

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, Raleigh Center.



Bradley Downing currently works for IBM in the Tivoli Advanced Technology Group - ISM Lab Services. He began his career with the Maximo® product 15 years ago with a small integrator in California. In 2001, he joined MRO Software, and came to IBM through an acquisition in 2006. During that time, he has specialized in working with clients in a variety of industries, with emphasis on reporting, Mobile Maximo, the use of the Integration Framework, and recently with the Migration Manager module. He holds an MBA in Business Management.



Burak Bolek works in IS Bank as a lead developer and administrator. He has three years of experience with Maximo. He also has worked on various client implementations as a Java™ developer. He is certified in Tivoli's process automation engine and the Information Technology Infrastructure Library (ITIL®).



David A Gutierrez is a technical consultant for IBM Global Business Services® and an IBM-certified Deployment Professional in Maximo V7.x. He has six years of experience implementing, deploying, configuring, migrating, and administering package solutions, in both private and public sectors, inside and outside of IBM. David has been an IT professional for over 25 years, providing systems and networking technical support to clients. David graduated from the University of Lima, Peru, with a B.S. degree in Industrial Engineering, and he currently lives in San Diego, California.



John Dibble joined IBM Tivoli from MRO, where he led the merger of the MainControl IT Asset Management product with Maximo Enterprise Asset Management (EAM), resulting in the Maximo 6 release. Following the IBM acquisition of MRO, he transferred to the IBM Global Technology Services group to apply the new technology. Currently, he performs as the lead architect for Tivoli's process automation engine, CCMDB, Tivoli Service Request Manager, and Tivoli Asset Management for IT solutions.



Kylie Cameron works for IBM Global Business Services as a client-facing Maximo Consultant based in Sydney, Australia. She has a Bachelor of Information and Communication Technology degree majoring in Business Information Systems and E-commerce. Kylie is a certified Maximo Consultant with over three years of project experience across the Utilities and Pharmaceutical industries with roles including Inventory Management and Procurement Lead, Environment Manager, Release Manager, and Training Lead.



Sampath Sriramadhesikan is an architect in IBM Software Group, working at the IBM Littleton campus in Massachusetts. He is responsible for leading the design and development of the key components of Tivoli's process automation engine and the Maximo product portfolio, with a specific focus on content management and promotion capabilities. He has over 17 years of experience in the software development industry in various roles including product development, product management, and teaching. He has an MS in computer science from Mysore University and an MBA from Northeastern University.

Scott Tyrrell is a Maximo System Administrator at a Fortune 500 Corporation and currently supports day-to-day operational issues and application development for the product suite of Tivoli Service Request Manager, Tivoli Asset Manager for IT, and CCMDB with Service Provider. A former software developer, Scott is ITIL-certified and has worked for several years in ITIL process development and implementation. He has a BS in Electrical Engineering from Christian Brothers University and an MBA from the University of Arkansas at Little Rock.



Thadeu de Russo e Carmo works for the IBM Software Group as a Staff Software Engineer in the IBM Brazil Software Lab based in Sao Paulo, Brazil. He has a BS in Computer Science and is a graduate student at Universidade de Sao Paulo (USP) in the Distributed Systems area. He has four years of IBM Maximo Asset Management development and has been involved in the development of many add-ons and industry solutions, such as IBM Maximo for Nuclear Power Plants, IBM Maximo for Life Sciences, and IBM Maximo Mobile, as the lead developer. With over six years of experience in Java Enterprise development, he is a Java-certified developer and architect, and he also holds the Foundations™ of Tivoli process automation engine certification.



Vasfi Gucer is a Project Leader at the International Technical Support Organization, Austin Center. He has been with the ITSO since January 1999. He has more than 12 years of experience in the areas of systems management, networking hardware, and software on mainframe and distributed platforms. He has worked on various IBM Tivoli client projects as a Systems Architect in the U.S. He writes extensively and teaches IBM classes worldwide on Tivoli software. Vasfi is also an IBM Certified Senior IT Specialist, PMP, and ITIL Expert.

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

Tamikia Barrow, David Bennin, Emma Jacobs, Stephen Smith, Margaret Ticknor, Debbie Willmschen

International Technical Support Organization

Scott Dickerson, Tom Ellingwood, Adel Fahmy, Stephen Ridgill
IBM USA

Marieli Alves de Lucena
IBM Brazil

Thomas Gehrke
IBM Germany

Andrzej Wieclaw
Framsteg AB

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

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 email 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

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>



Migration strategy

In this chapter, we show how the Migration Manager works together with other tools to successfully migrate Tivoli system configurations across the client's environments. Typically, these environments are represented as: development (unit testing), test (integration testing), stage (final integration and performance testing), and production.

The Migration Manager is a component of Tivoli's process automation engine. It is a set of tools used to promote system configurations from a development environment to upper environments, such as stage, user acceptance testing, and production.

System configuration is a set of metadata that enables application functionality and controls application behavior in a production environment. For example, the user interface for the Service Request application is enabled and rendered through application presentation metadata.

The Migration Manager is heavily used in migrating system configurations prior to the production environment going live. However, it is also used in migrating changes to system configurations after the production environment is live. We will discuss the role that the Migration Manager plays in both scenarios.

A production environment that produces system configurations to be migrated is usually called a *Source environment*. A production environment that consumes system configurations is usually called a *Target environment*.

This chapter has the following sections:

- ▶ 1.1, “Requirements” on page 3
- ▶ 1.2, “Choice of tools to support migration” on page 3
- ▶ 1.3, “Content strategy” on page 7
- ▶ 1.4, “Migration approach” on page 15
- ▶ 1.5, “Migration planning” on page 18
- ▶ 1.6, “Migration Manager reference material” on page 26

1.1 Requirements

Clients typically have the following requirements for the migration of system configurations from development to production:

- ▶ Migration must be *repeatable* and *efficient* across environments that constitute the system landscape.
- ▶ Migration must be *durable* in that it can be exploited before the production environment goes live as well as after the production environment goes live.
- ▶ Migration must be *stable* in that all types of system configuration can be identified, collected, packaged, distributed, and deployed in the same manner.
- ▶ Migration must leave the Target environment usable and intact at all times. The integrity of the system configurations must not be compromised.
- ▶ Migration must minimize or eliminate the system downtime of the Target environments.

Test data, such as data that is used to drive unit testing in a development environment, is not to be promoted. Source code in the form of Java sources or compiled Java class files can be managed separately using traditional source control systems.

1.2 Choice of tools to support migration

To meet these requirements, the delivery architect needs to evaluate how Tivoli is intended to operate within the enterprise. Tivoli will exchange data records with the enterprise. It is necessary to define how this data will be maintained during migration and post-production, which affects the way that systems are built from one production environment to the next. For example, person records might never be migrated. Instead, each production environment will synchronize person records from a directory. Alternatively, person records might be always loaded into a production environment from an existing system, which has an impact on the migration plan, scope, and effort.

From a tools perspective, the Migration Manager will be primarily responsible for the migration of system configuration metadata that is generated as a result of configuration and customization activity in the development environment. The Migration Manager is also capable of migrating application content.

However, the following criteria must be applied to determine whether such application content needs to be migrated using the Migration Manager:

- ▶ Is the application static? Static data never changes. It is set up one time and subsequently used in the same static form across all production environments.
- ▶ Will the application content be created in a development environment? Application content that is created in a development environment might require to be promoted. For example, a hierarchy of classifications might support the Job Plan application.
- ▶ Can the application content be characterized as master or transactional data? Examples of master data include person, item master, and company records. Examples of transactional data include service requests, incidents, purchase orders, and invoices. These types of data are not set up in a development environment except as test data.

If application content is static, created in a development environment, and not master or transactional data, this content is a candidate to migrate to Target environments using the Migration Manager.

1.2.1 Data loading tools

From the point of view of preparing Target environments, such as test or production, through data loads, a number of IBM tools exist in addition to the Migration Manager. These data loading tools are geared to deliver specific data loading capabilities:

- ▶ Tivoli process automation engine Integration Framework
- ▶ Tivoli Integration Composer
- ▶ Tivoli Directory Integrator

In certain cases, a business application might offer an application-specific data loading capability. For example, the Application Designer provides an export and import feature supporting application and system presentations.

The choice of tool depends on the specific data loading requirement. Table 1-1 lists particular tools and their use.

Table 1-1 Tools and use cases

Type of data loading	Tivoli process automation engine Migration Manager	Tivoli process automation engine Integration Framework	Tivoli Integration Composer	Tivoli Directory Integrator
Batch loading of data from discovery tools	No	No	Yes	No
Batch loading of master and transactional data or existing data	No	Yes	No	No
Messaging-based integration with external systems	No	Yes	No	Yes
Promotion of system configurations between Tivoli's process automation engine-based production environments	Yes	No	No	No

Based on the tools and their uses, it is clear that the Migration Manager is not the only tool that is used to load data into upper environments. Selecting the appropriate tool based on your data loading needs helps ensure success in meeting production go-live time frames.

For a more in-depth discussion of data loading tools and capabilities, refer to the following best practices publication:

<http://www.ibm.com/developerworks/wikis/download/attachments/130515354/TpaeEcosystemDataIntegrationBestPractices.pdf?version=2>

1.2.2 Sequence of data loads

Most production environments are brought up the first time using a combination of the data loading tools. Figure 1-1 shows a typical initial data loading effort that includes migration with the Migration Manager.

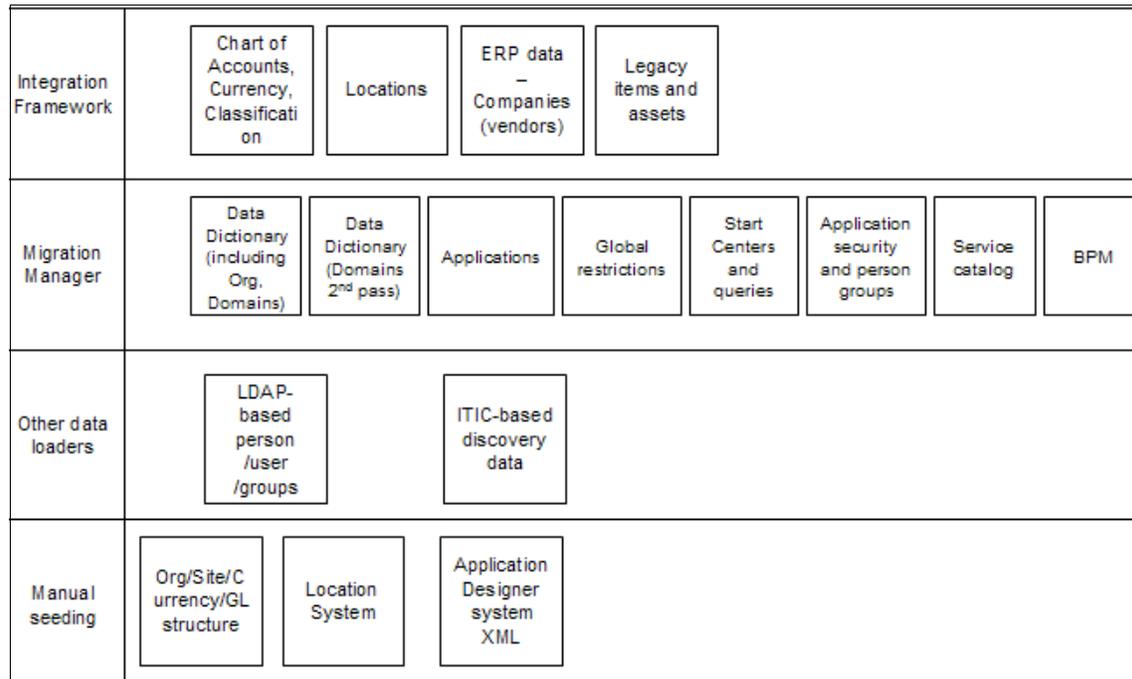


Figure 1-1 Typical sequence of data loads and migration

The following steps show a more detailed description of the typical load/migration sequence:

1. Manually seed the organization names and currency.
2. Manually set up the GL account structure (segments).
3. Manually create the default chart of accounts for each defined organization.
4. Manually define the location system.
5. Using Integration Framework object structures, load the physical location hierarchy, including regions.
6. Run Integrity Checker, and look for any errors.
7. Correct the errors manually or by using Integrity Checker in repair mode.

8. After the manual steps have been executed, the enterprise LDAP directory server is synchronized into the product, if the client uses directories. Person, user, and security groups can be loaded.
9. If Lightweight Directory Access Protocol (LDAP) is not part of your environment, migrate person, person groups, and security groups configurations, including employee and supervisory (management reports) information.
10. Migrate the data dictionary, including the domains.
11. Migrate the crossover and table domains to complete the data dictionary migration.
12. If the production environment includes Tivoli Service Request Manager, run the service catalog object structure migration. At this point, the structural migration is complete.
13. Load the financial chart of accounts.
14. Load the currencies and exchange rates.
15. Migrate all of the application configurations, including menus and lookups.
16. Migrate application security configurations, including signature options.
17. Migrate the Start Centers and queries.
18. Migrate the Tivoli Service Request Manager service catalog templates.
19. Migrate the Tivoli Service Request Manager service catalog single service configuration.
20. Migrate the workflows and their associated roles, actions, and communication templates.
21. Run Integrity Checker, and look for, and correct, any errors.

This sequence captures the essence of the data loading and migration steps. A number of tools are exploited when executing the steps in the sequence. Experience with these tools accelerates the preparation of the production environment. Documenting this sequence during the planning stage is vital to the success of the overall migration effort. Ad hoc tasks that are performed without prior knowledge of the tools, applications, and content slow down the migration effort and lead to a reduction in customer satisfaction.

1.3 Content strategy

Migration is a consequence of configuration activity. Configuration activity creates content in the production database using various Configuration applications. Part of this content might have been provided by IBM with the

product, and other content might have been created in the client's development environment. Visibility into and control of these configurations are extremely important to ensure the consequent migration effort. Figure 1-2 shows the interaction of configurations and migrations. Content is the underlying artifact that is being managed.

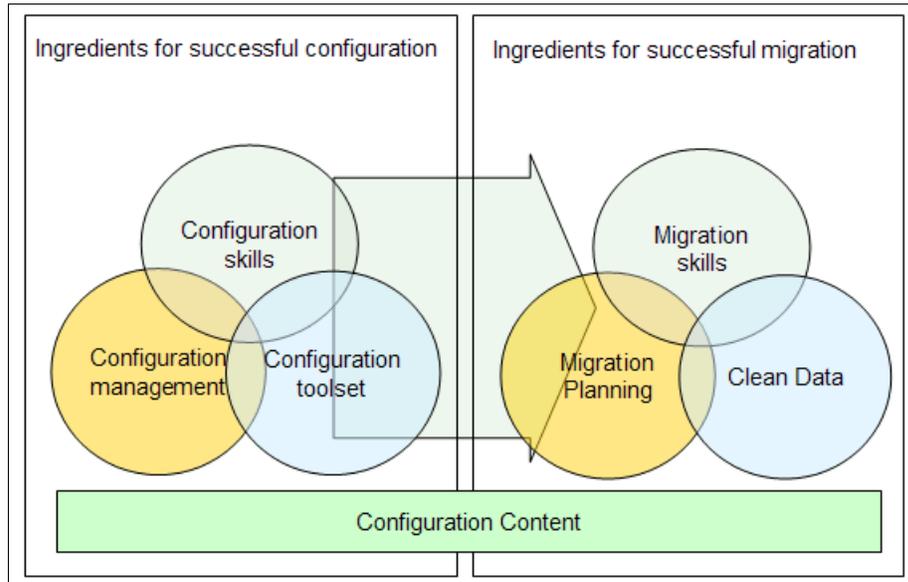


Figure 1-2 Interaction of configuration and migration

From the configuration perspective, configuration skills, configuration management, and the proper use of the configuration tool set are key to complete an efficient configuration in the client's development environment. Better configuration management practices lead to better migration planning. Consistent use of the Configuration applications and tools results in cleaner data that, in turn, accelerates the migration effort.

1.3.1 Dependencies among configuration content

Few product configurations are completely self-contained. Most product configurations have dependencies on other configurations. Product validations enforce these dependencies when configuration content is loaded into a production database. The dependencies among configurations and the validations that enforce them impact the migration effort. Figure 1-3 on page 9 shows a number of dependencies among the configurations that drive a significant amount of product functionality.

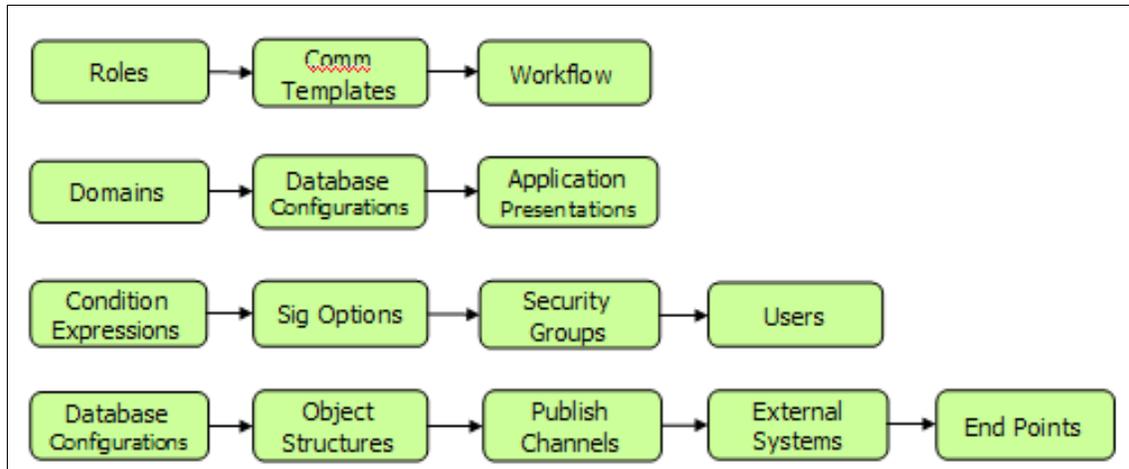


Figure 1-3 Dependencies among configurations

In Figure 1-3, a workflow process configuration depends on communication templates and roles. Application presentations rely on business objects whose attributes, in turn, are associated with value lists in the form of domains. Security groups are associated with signature options, which, in turn, might be conditionally applied to the security group based on conditional expressions. External systems that are identified using the Integration Framework are associated with Publish Channels, which, in turn, are supported by object structures and business objects.

Given the deep dependencies among configurations, sequencing the migration is extremely critical.

1.3.2 Best practice content

It is a Tivoli direction to provide best practice content with the product to help you get the product up and running quickly and to enable you to start using the product before engaging in specific configurations and customizations. One example is the predefined Tivoli Service Request Manager where an installation can include creating a IBMPMSC organization, PMSC site, and PMSC set of pre-configured offerings that cover the typical IT Infrastructure Library® (ITIL) situations.

Recent clients have loaded these predefined offerings only to opt for fully customized offerings, which raises the issue of deciding what to do with this content when it comes to migration. Obviously, you have a few choices: create migration packages that exclude the predefined offerings, use currently supplied tools to add these offerings to the client's organization, or delete them.

Example: This example is usually part of the installation process and will therefore be on the Target, unless specifically dropped at the installation stage.

Another example is the use of Intellectual Capital (iCAP) to provide best practice business process content (workflow-based) that adheres to ITIL guidelines. Because this iCAP is loaded at installation on all environments, it is only the subsequent client modifications that need to be promoted through the environment.

1.3.3 Integrity of configuration content

This section discusses the need to establish the integrity of configurations in the production database using the IBM-provided tool.

The Integrity Checker is the primary tool that is used to determine the integrity of many configurations. The Integrity Checker tool is shipped with the product and can be launched from the console command line. It produces a detailed log that includes warnings and errors pertaining to data integrity. It is important to continuously monitor the integrity of the underlying production database by periodically running the Integrity Checker in all of the production environments.

A best practice is to capture the Integrity Checker reported errors in a spreadsheet form, review the error messages, and identify the resolution of those errors. Preparing this spreadsheet enables you to make an assessment of the impact that the error will have on migration. Figure 1-4 shows a sample spreadsheet that has collected error messages and also offers resolutions.

Error message	Reason for error	Impacts migration?	TACTICAL FIX 1 Migration Remediation	TACTICAL FIX 2 Integrity Checker Remediation
BMXAA0443E - ERROR -- BMXAA0465E - The following Maximo indexes are missing from the database: Mon Nov 03 08:39:34 CST 2008 PLUSGCFW_NDX4 Mon Nov 03 08:39:34 CST 2008 PLUSGREGULAT_NDX4 Mon Nov 03 08:39:34 CST 2008	MAXSYSINDEXES table contains two entries, but underlying database does not	YES, if you attempt to migrate the PLUS* tables, the missing indexes will cause package deployment failures.	Do not carry PLUS* tables in your migration packages; use SQL conditions to exclude such tables for DMMAXOBJECTCFG	Correct error as per Integrity Checker troubleshooting/correction steps.
BMXAA0443E - ERROR -- BMXAA0467E - The database has a different UNIQUE property for the following maximo indexes: Mon Nov 03 08:39:34 CST 2008 Table MXIN_INTER_TRANS Keys TRANSID Mon Nov 03 08:39:34 CST 2008 Table MXOUT_INTER_TRANS Keys TRANSID Mon Nov 03 08:39:34 CST 2008	The uniqueness of the index in MAXSYSINDEXES is different from the uniqueness defined at the database level.	MAYBE, only in an update scenario where the same indexes already exist in target	For these two errors, use SQL conditions to exclude MXIN_INTER_TRANS and MXOUT_INTER_TRANS tables for DMMAXOBJECTCFG	Correct error as per Integrity Checker troubleshooting/correction steps.
BMXAA0443E - ERROR -- BMXAA0470E - Non-sequential primary key sequence on table(s): Mon Nov 03 08:40:09 CST 2008 DMPACKAGEDEF Mon Nov 03 08:40:09 CST 2008 MAXPROCCOLS Mon Nov 03 08:40:09 CST 2008	PRIMARYKEYCOLSEQ values in MAXATTRIBUTE table for the primary key columns are not in sequence	YES, if you attempt to migrate the MAXOBJECTCFG data for these two tables, it is highly likely package deployment failures will occur	Correct the error in both source and target environments such that the primary key col seq values are in sequence (1, 2, 3, etc)	Correct error as per Integrity Checker troubleshooting/correction steps.

Figure 1-4 Assessment of Integrity Checker errors

In Figure 1-4, the Error Message column reproduces the Integrity Checker error as retrieved from the Integrity Checker log file. The Reason for Error column reproduces the Integrity Checker product documentation by correlating the error message number with the Integrity Checker error tables. The Impacts Migration column makes a determination if the particular error will affect the Migration Manager activities. The TACTICAL FIX 1 column determines if the Migration Manager can be controlled to avoid migration failures due to the error. The TACTICAL FIX 2 column determines if the resolution is to fix the integrity error directly in the underlying database and then perform the migration. In most cases, it is necessary to fix the integrity error in the database and then perform the migration.

To understand and use the Integrity Checker tool, refer to the detailed product documentation at this website:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.srm.doc_721/reference/6_to_7_upgrade.pdf

1.3.4 Content validation

This section discusses the importance and impact of validation to the migration effort.

The configuration contents of a migration package are deployed into a production environment using the Migration Manager. The deployment consists of a well-defined automated procedure of extracting the contents of the migration package and processing those contents in a specific order. Depending on the type of content, the procedure is temporarily stopped to allow an administrator to perform manual tasks and the procedure is resumed when those manual tasks are complete. All content being deployed into the underlying production database is subject to validation. Such validation ensures that the integrity of the production database is preserved at all times. This validation cannot be turned off. Figure 1-5 on page 12 shows the role of validation in deployment.

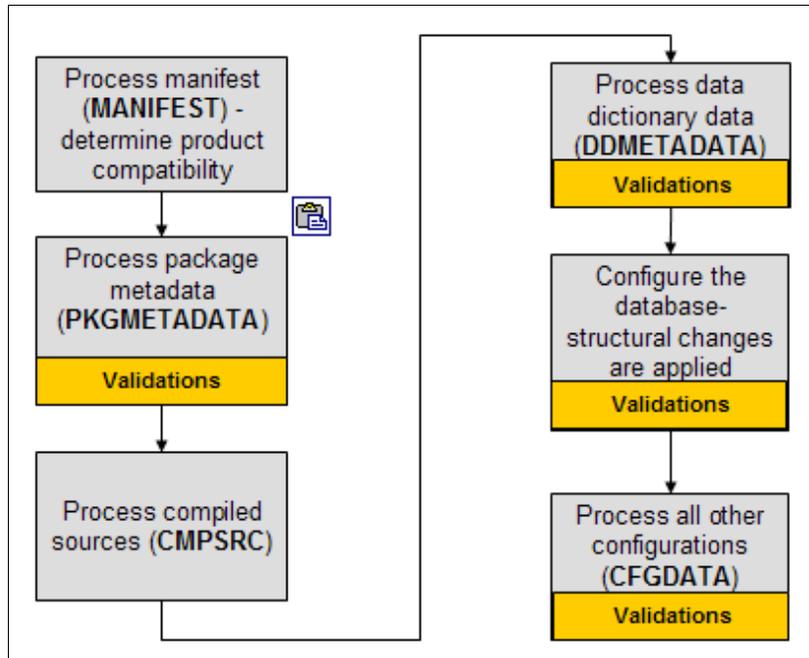


Figure 1-5 Role of validation in package deployment

In Figure 1-5, key validations are applied to the data dictionary configurations, the database configuration process, and all other configurations, such as person records, security group records, workflow processes, and so on.

When validations fail, the Migration Manager reports a deployment error. A deployment error represents the inability of the Configuration application to successfully apply the data validation rules to the particular configuration being deployed. It is the responsibility of the Migration Manager to report deployment errors resulting from validation failures.

The root cause of these validation errors can be one or a combination of the following reasons:

- ▶ Incomplete packaging of the configuration content
- ▶ Incorrect sequencing of migration
- ▶ Bad data in the Source production environment
- ▶ Limitation of the Migration Manager
- ▶ Defect in the product

In each case, the appropriate remedial measures are taken to ensure recovery from the error and continuation of the migration effort.

Incorrect sequencing or incomplete packaging can be addressed by improving product skills. Bad data is typically the result of loading data directly into the database using database commands. This practice must be avoided at all costs. Bad data can also be the result of database scripts supplied with the product. Part of the bad data can be corrected through the use of Integrity Checker. If bad data was shipped with the product, IBM treats this situation as a defect. The Migration Manager limitations can be addressed by adopting other tools to load the configuration or the related data needed. Product defects are addressed by IBM using standard procedures. Understanding the nature of the deployment error helps immensely in resolving the error.

1.3.5 Content and source control systems

The role of source control systems is a topic often raised by practitioners and clients when managing configurations and performing migrations. Tivoli's process automation engine does not offer any built-in versioning capability. Configurations are simply saved to the designated tables in the underlying production database, overwriting any previous values. How can versions of a configuration be maintained? Does the Migration Manager have a role to play?

The Migration Manager does not enable or mandate a particular source control model. However, from a change management perspective, implementing a source control model or extending an existing source control model can yield benefits to practitioners and clients that extend beyond the migration effort.

In this section, we describe two source control models:

- ▶ Manage versions of configurations with source control
- ▶ Publish migration packages through source control

Manage versions of configurations with source control

Part of the configuration content in the product is shared among team members. Multiple edits might be performed in a short period of time on this content. When the configuration content is stored in the production database, the changes made by the last team member saving the configuration are alone written into the database. This process can be a significant change management challenge in shared product development environments. If there is a need to revert to a previous revision of the configuration content, the only way to revert to a previous version is through the implementation of a source control model.

In this source control model, chosen configuration content is exported as a file from the production environment and checked into a source control system. Practitioners or developers requiring a change to that content must check out that file, import it into the production environment, and implement the change using the appropriate application. After it is complete, the configuration is

exported again and placed back into source control. Figure 1-6 illustrates this model.

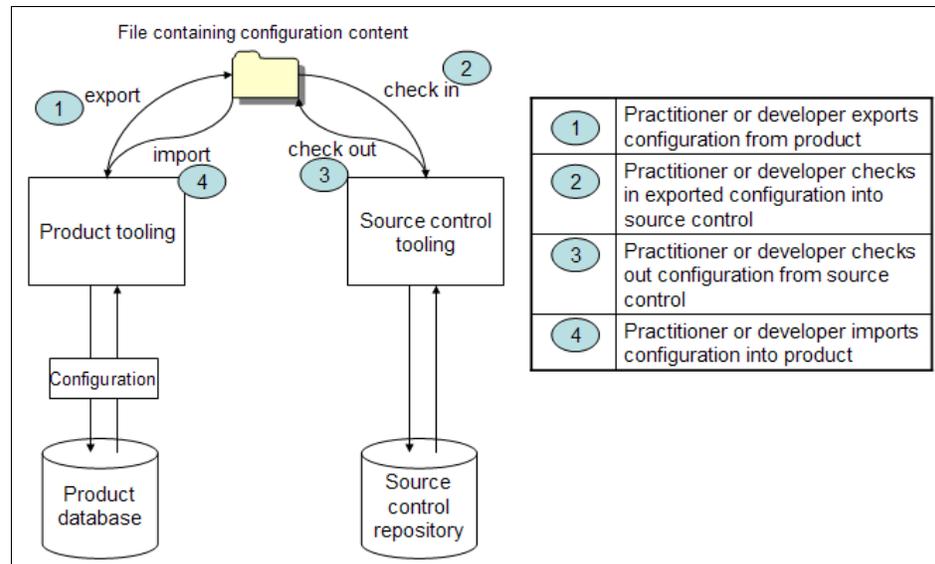


Figure 1-6 Source control model to manage versions

Although cumbersome, implementing this model ensures that changes are traceable and that the authors of those changes are accountable during the implementation cycle. This approach of managing revisions of a configuration must be considered as part of the overall content strategy.

This model is most commonly applied to shared Application Designer configurations, such as `library.xml`, `lookups.xml`, and `menus.xml`. These three configuration files are shared configurations that are accessed through and managed with the Application Designer.

Publish migration packages through source control

Certain clients choose also to check in to source control systems entire migration packages in the form of compressed files as provided by the Migration Manager. The rationale behind this approach is that the source control system alone holds the authorized set of migration packages that must be used to perform the migration. Direct distribution of packages from a Source environment to a Target environment via file system or database distribution is not allowed. Figure 1-7 on page 15 illustrates this model.

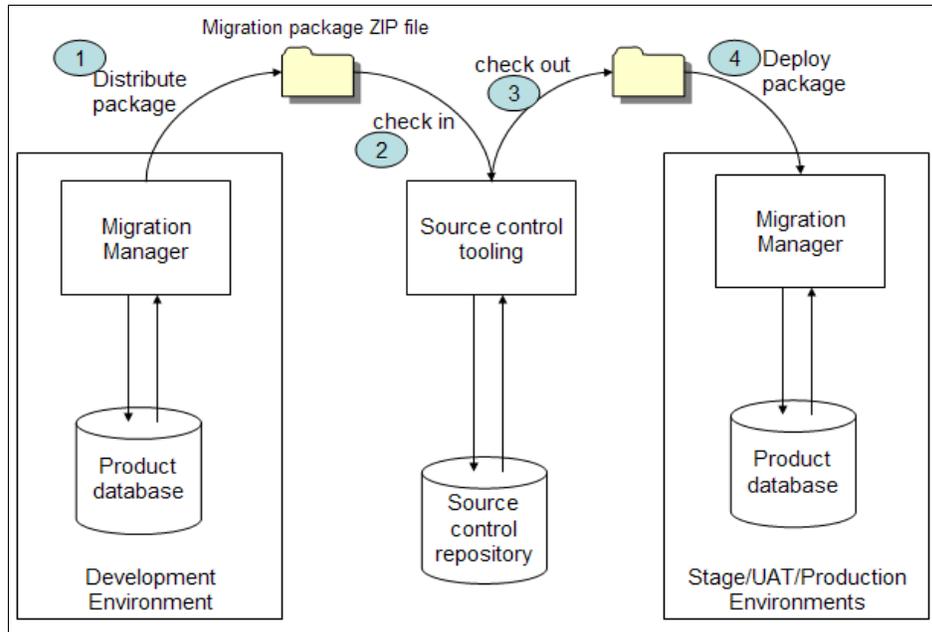


Figure 1-7 Source control model to publish migration packages

We recommend flexibility with respect to this practice. The overall implementation time line and resource constraints must be taken into account before implementing this approach. The practice is more easily implemented with clients who already have an established process of distributing content through source control systems.

Of the two source control models presented, the need to manage the versions of key configurations is more critical to the success of the development effort. Inability to successfully manage shared configurations can lead to a breakdown of the development effort.

1.4 Migration approach

In this section, we describe the recommended migration approach. The Migration Manager offers two modes of operation:

- ▶ Snapshot
- ▶ Change

1.4.1 Snapshot mode characteristics

The Snapshot mode is literally a view of the system at the point of creating a package. This mode is often used because the Migration Manager was used late in the program and thereby, the Snapshot sees all changes made since the installation. The disadvantage with Snapshot is that either you migrate everything (without any filtering) or the practitioner must define filters to select only the changes made. By using naming conventions, such as a client prefix on all structural and other database changes, it is possible to write these Where clauses and limit the migration package size to close to the size that is created by a corresponding Change mode package, for a given set of changes.

Consider the following characteristics of Snapshot packages:

- ▶ The Snapshot package provides the ability to filter (one-to-many) and migrate complete configuration content ensuring Source and Target synchronization.
- ▶ Snapshot processing actions can be specified that determine whether the configuration content is fully replaced or merged into the Target production environment.
- ▶ The entire migration requirement is clearly defined, identified, and documented, to the exact names of the content elements.
- ▶ The Source environment contains a considerable number and amount of configurations that took significant time to develop, a bulk migration is appropriate, and all of the previous characteristics are true or desired.

1.4.2 Change mode characteristics

The Change mode, after it is enabled, incorporates event listeners that support tracking changes made to business objects up until the time when it is disabled. Change mode tracks who does what and when. Content is extracted only based on the tracked changes.

Consider the following characteristics for Change packages:

- ▶ Useful when a Source environment and a Target environment already exist and have already been synchronized at the time that this type of package is activated (for example, when the last migration is done prior to production, as described previously).
- ▶ Useful when active packages do not overlap (listening to the same objects), development efforts do not overlap either (each development effort uses its own Change packages), and all of the previous characteristics are true or desired.

- ▶ When too many development efforts overlap and it is difficult to define non-overlapping packages, a single Change package is useful to be able to track the changes and define the migration requirements. This use is a hybrid use of Change mode where the Change package is used merely as a tracking device. The actual migration is performed using Snapshot packages.

Most of the scenarios in this document are designed and implemented as Snapshot packages.

For more details, refer to 10.6, “Change tracking and ad hoc reporting” on page 210.

1.4.3 Hybrid migration strategy

A *hybrid migration strategy* is to use Change and Snapshot packages at the same time. The purpose is to capture change information automatically, thereby reducing the need for practitioners and developers to laboriously enter the individual configurations that they create into a document. Implementing this hybrid migration strategy carries a minimal cost (two package definitions, instead of one) but yields significant benefit during the development and migration effort.

With this hybrid migration strategy, both package definitions identify the same types of content. The Change package is defined exclusively to track the same configuration content that will eventually be packaged with the Snapshot package. An ad hoc report is configured to display the changes that are tracked by the particular Change package.

Figure 1-8 on page 18 shows life cycle migration modes.

Tip: The recommended approach is to use both the Snapshot and Change modes (hybrid migration strategy). Use the Snapshot mode for the development through the production cycle, and then switch to Change mode (on the development system) for post-production development and the promotion of fix packs and minor updates.

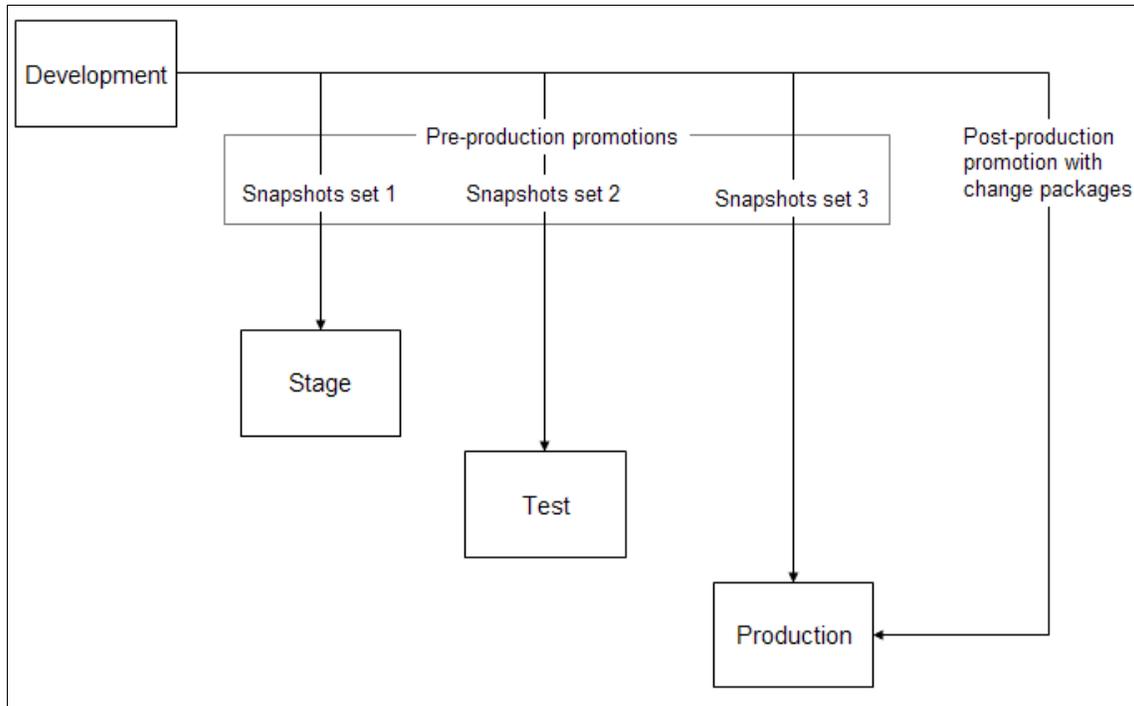


Figure 1-8 Migration modes

Naming conventions: The concept of using naming conventions for structural and other configurations can also be used in conjunction with queries that look for and track the state of the configuration.

For more details, refer to 10.6, “Change tracking and ad hoc reporting” on page 210.

1.5 Migration planning

A migration plan is the formulation of the set of actions that will promote configuration content from the development environment to upper environments, such as stage, test, production, or training. The migration plan manifests in the form of a controlled living document that serves as input to the Migration Manager tasks to be performed by a skilled resource. However, it is not limited to the Migration Manager tasks alone.

The plan also enumerates manual activities that are performed outside of the Migration Manager functionality:

- ▶ Creation of configurations using the corresponding Configuration application (for example, organization)
- ▶ Execution of SQL scripts (for example, loading a set of person records)
- ▶ Execution of Integration Framework-based data loads (for example, loading a set of GL accounts or loading a set of company records from an existing system)
- ▶ Execution of build process for Java classes (for example, building the product's deployable EAR file)

The critical part of the plan is the sequence in which all of these tasks are performed and the unambiguous identification of task owners. The goal is to eliminate as many unknowns as possible and streamline the preparation and delivery of the upper environments to their users.

The plan facilitates the migration prior to the production rollout, as well as periodic migrations after the production rollout has completed.

Creating and maintaining a migration plan document offer the following benefits:

- ▶ Scope of migration clearly established
- ▶ Required tools and skills clearly identified
- ▶ Owners of tasks clearly identified
- ▶ Migration progress monitored
- ▶ Status updates provided to stakeholders

This IBM Redbooks publication provides guidance to the content of a migration plan document, and you can also use the samples provided with this book. To download these samples, refer to Appendix A, "Additional material" on page 257.

1.5.1 Contents of the migration plan

The migration plan document benefits from a spreadsheet format. The spreadsheet is created and placed in a shared environment. Certain practitioners even place the spreadsheet under revision control so that separate versions of the spreadsheet can be maintained and retrieved. Placing the spreadsheet under revision control enables stakeholders to compare multiple versions of the spreadsheet and ensure the authorized access and updates to the controlled document.

There might be many variations of the spreadsheet based on the practitioners' approach, requirements of clients, and policies of IT-based organizations.

The migration plan typically includes these areas:

- ▶ Tasks
- ▶ Task sequence
- ▶ Owners
- ▶ Tools
- ▶ Dates

The following sections describe a reasonable and reusable spreadsheet-based migration plan.

The spreadsheet contains the following worksheets:

- ▶ Migration plan revision history
- ▶ Migration environments, products, and versions
- ▶ Migration tasks

Migration plan revision history

This worksheet records changes to the migration plan and the authors of each of those changes. Table 1-2 shows the columns that record revision history.

Table 1-2 Record revision history

Revision	Date	Author	Description	Reviewed by	Approved by

This worksheet provides these benefits:

- ▶ Provide an audit trail of changes to the plan
- ▶ Ensure accountability through review and approval information

Migration environments, products, and versions

This worksheet records each migration environment that is part of the IT landscape. At a minimum, this worksheet includes the development, test, and production environments. For each environment, this worksheet shows the list of installed products. This information is easily obtained from the production environment's web user interface (System Info menu item in the user's Start Center). In addition, the type of the environment is recorded; that is, a particular environment is either a Source or Target from the migration perspective.

Table 1-3 provides example information regarding the environment, products, and revisions.

Table 1-3 Columns that record the environment, products, and revision information

Environment type	Products and versions	Product URL	Migration package distribution	Package file location

Development	SRM Problem Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 SRM Incident Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 SRM Service Request Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 IBM Tivoli Common Process Components 7.2.0.01 Build 20100109D DB Build V7201-02 Base Services 7.1.1.6-LA20100312-093 7 Build 20091208-1415 DB Build V7116-173 HFDB Build HF7116-04	http://hostname:9999/maximo	FILE	H:\share\migration\packages
-------------	--	-----------------------------	------	-----------------------------

This worksheet provides these benefits:

- ▶ Provides at-a-glance information about the IT environment to stakeholders
- ▶ Avoids confusion among implementers regarding Sources and Targets
- ▶ Saves time when gathering information regarding deployment issues
- ▶ Avoids process breakdown when transitioning implementation resources

Migration tasks

This worksheet is the most critical element of the plan. This worksheet records a number of facets pertaining to the migration effort:

- ▶ Manual data entry into the production environment
- ▶ Import of existing data into the production environment using Integration Framework
- ▶ Deployment of migration packages into the production environment using the Migration Manager
- ▶ Necessary application server restarts

- Special situations where the standard order of migration has to be changed to accommodate dependent data

Table 1-4 shows sample content for this worksheet, based on the recommended approach. The line items in the worksheet are representative only. Depending on the extent of configuration, the number of manual steps, the need to import data using Integration Framework, and the number of migration packages required to complete the migration, the effort can vary. The worksheet is extremely beneficial in reducing the uncertainty that is inherent in data migrations and accelerating the migration activities.

Table 1-4 Migration tasks

Sequence	Summary	Description	Method	Owner
1	Integrity Checker	Run Integrity Checker on Target database in report mode; correct any errors	Manual	Product implementer
2	Correct Integrity Checker errors and warnings	Use Integrity Checker with repair mode or manually correct with SQL tools	Manual	Product implementer
3	Database backup	Back up the Target production database	Manual	DBA
4	Prepare EAR	Build the product EAR file	Manual	Product implementer
5	Deploy EAR	Deploy the product EAR and restart the application server	Manual	Product implementer
6	Currencies and exchange rates	Load currencies and exchange rates	Manual	Product implementers
7	Org/currency	Seed the ORG names and currency	Manual	Product implementer
8	GL structure	Set up the GL account structure	Manual	Product implementer
9	Chart of accounts	Create a chart of accounts for each organization	Manual	Product implementer
10	Default chart of accounts	Create a default chart of accounts for each organization	Manual	Product implementer
11	Chart of accounts	Load additional accounts via Integration Framework	Integration Framework	Migration Manager

12	System of locations	Define and load systems and locations	Manual or Integration Framework	Product implementer
13	Run Integrity Checker	Run Integrity Checker against the development database in report mode; correct any errors	Manual	Product implementer
14	Correct Integrity Checker errors and warnings	Use Integrity Checker with repair mode or manually correct with SQL tools	Manual	Product implementer, developers
15	Data dictionary (Phase 1)	Migrate conditional expressions	Package	Migration Manager
16	Data dictionary (Phase 2)	Migrate all domains except the table and crossover	Package	Migration Manager
17	Data dictionary (Phase 3)	Migrate data dictionary configurations, including objects and attributes	Package	Migration Manager
18	Data dictionary (Phase 4)	Migrate table and crossover domains	Package	Migration Manager
19	Applications	Migrate Application Designer system XMLs	Manual	Application Designer
20	Applications	Migrate all applications and menus except system XMLs; server restart needed if production environment is clustered	Package	Migration Manager
21	Persons, users, and security groups	If Lightweight Directory Access Protocol (LDAP) is synchronized for an LDAP-enabled client, go to Chapter 3, "Migrating security configuration data" on page 51	SYSTEM	Product implementer
14	Persons, users, and security groups	If LDAP is not enabled, migrate persons, users, and security groups	Package	Migration Manager
17	Global data restrictions	See Chapter 3, "Migrating security configuration data" on page 51	Package	Migration Manager
18	Application Security package	Application Security package for signature options and queries	Package	Migration Manager

19	Start Center	Start Center and Queries package; each individual user must update their Start Center to apply changes	Package	Migration Manager
20	Integrations (Phase 1)	Migrate integration object structures	Package	Migration Manager
21	Integrations (Phase 2)	Migrate all other integration configurations	Package	Migration Manager
22	Service Catalog	Service Catalog Object Structure package	Package	Migration Manager
23	Business process management (BPM) (Phase 1)	BPM (workflow) package to migrate the workflows and their associated role, actions, and communication templates, except application actions that launch workflows	Package	Migration Manager
24	BPM (Phase 2)	Migrate application actions that launch workflows	Package	Migration Manager
25	Post-migration task	Activate communication templates and workflows	Manual	Product implementers
26	Classifications	Migrate classifications	Package	Migration Manager
27	Service Catalog Templates	Service Catalog Templates package	Package	Migration Manager

After a plan document has been created, we recommend a meeting among the stakeholders to walk through the plan. This meeting disseminates the critical information to all participants and identifies missing or incorrect information in the plan document. Subsequent periodic meetings provide the continuous monitoring of the migration process.

1.5.2 Working with migration time frames

The migration effort is always measured by clients in terms of the time that is consumed, *which needs to be adequately scoped out as part of the project plan.*

In the pre-production phase of a project, the development of configurations can take several weeks and involve multiple practitioners and developers. It is extremely important for the migration planning and activities to remain in lockstep with the development activities during this phase. The most common mistake made is to defer the migration activities until after the development is complete. Most clients plan for a few days during which time they expect migration activities

to complete and the production environment to be ready for user access. Client expectations of migration cannot be met when development and migration activities are separated.

Best practice: In the pre-production phase of a project, the migration planning document must be complete at the same time that the development and configuration activities for the project complete.

In the post-production phase, certain configuration and development activity is ongoing. This activity is the result of feedback from users or application managers. A smaller scale migration is undertaken to promote this maintenance or incremental configurations to the existing production environment. Most clients expect this migration to begin and complete within specific maintenance windows. The most commonly encountered maintenance windows are two, four, or six hours. The maintenance windows are typically opened at the onset of a weekend when user activity is minimal and can be stopped without impacting the business. It is critical that the maintenance activities complete within the maintenance window that is specified by the clients. Developing a maintenance window migration plan is extremely useful. This plan can be a simpler version of the pre-production plan that is outlined in earlier sections.

A number of factors affect the duration of all types of migrations that might result in the implementation exceeding the expected migration time frames. The following factors will consume a considerable amount of time, not directly involving the Migration Manager or Integration Framework:

▶ Application server restarts

Production environments typically use multiple Java virtual machines (JVMs) executing in clustered application servers. There might be additional JVMs outside of the cluster supporting cron tasks, integration, and reporting functions. Shutting down and restarting all of the JVMs can consume a significant amount of time that reduces the migration time frame.

▶ Post-migration tasks

After a migration completes, manual tasks are typically performed, such as the activation of workflow processes, escalations, cron tasks, and so on. These tasks are done through the product's application user interface (UI) and consume a certain portion of the migration time frame.

▶ Running tools, such as Integrity Checker

Tools, including the Integrity Checker, might be executed periodically. Depending on the size of the production database, the tool can take a longer or shorter time to complete, yet consume a certain portion of the migration time frame.

► Other factors

DBAs might be required to review structural changes to the underlying production database and, if required, to make modifications to database objects, such as indexes, table spaces, and so on. The intent is to ensure the optimal performance of the database. These activities will consume a certain portion of the migration time frame.

Figure 1-9 shows a breakdown of the various types of tasks as a percentage of the total migration effort. This chart helps to establish the fact that the Migration Manager is not the only tool that is used to achieve total migration.

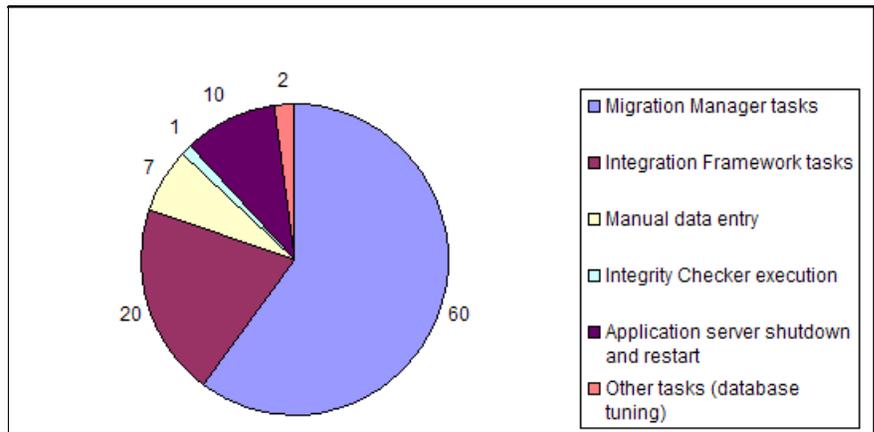


Figure 1-9 Migration effort breakdown

Keeping these factors in mind, managing the migration effort is as much managing the client's expectations with multiple constraints as performing the actual migration tasks using the Migration Manager. Up-front discussion and agreement with the clients can go a long way in ensuring a successful and timely migration.

1.6 Migration Manager reference material

The following reference material is available for the Migration Manager. These links are mostly the Technotes or webdocs that are published on IBM Tivoli Support websites about the Migration Manager:

► Migration Manager V7.1.1.5 features:

http://www-01.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&dc=DB560&dc=DB520&uid=swg21393047&loc=en_US&cs=utf-8&lang=en

This document describes usability features that were introduced with Base Services V7.1.1.5.

- ▶ Migration Manager V7.1.1.5 ticket templates:

http://www-01.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&dc=DB560&dc=DB520&uid=swg21393063&loc=en_US&cs=utf-8&lang=en

This document describes the support to migrate ticket templates.

- ▶ Migration Manager V7.1.1.5 classifications:

http://www-01.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&dc=DB560&dc=DB520&uid=swg21393048&loc=en_US&cs=utf-8&lang=en

This document describes the support to migrate classifications.

- ▶ Log setup for the Migration Manager:

<http://www-01.ibm.com/support/docview.wss?uid=swg21397377>

This document describes the steps to configure logging of the Migration Manager functions.

- ▶ Start Center migration:

<http://www-01.ibm.com/support/docview.wss?uid=swg21427580>

This document provides additional information about Start Center migration.

- ▶ Change size limit when uploading large Migration Manager packages:

<http://www-01.ibm.com/support/docview.wss?uid=swg21408312>

This document describes a technique to upload large package files.

- ▶ Migration Manager preview V7.1.1.6:

<http://www-01.ibm.com/support/docview.wss?uid=swg21414562>

This document describes the Preview capability of the Migration Manager, which was introduced with Base Services 7.1.1.6.

- ▶ Run Integrity Checker before the Migration Manager:

<http://www-01.ibm.com/support/docview.wss?uid=swg21299691>

This document describes the need to execute the Integrity Checker tool before performing migrations.

- ▶ Oracle length semantics affects the Migration Manager:

<http://www-01.ibm.com/support/docview.wss?uid=swg21411696>

This document describes the impact of Oracle's character length semantics on migration.

- ▶ No support for IBM Maximo Enterprise Adapter (MEA) interface table migration:

<http://www-01.ibm.com/support/docview.wss?uid=swg21299697>

This document describes the limitation of the Migration Manager related to migrating interface table definitions.

- ▶ Migrating conditions:

<http://www-01.ibm.com/support/docview.wss?uid=swg21389955>

This document describes a technique of migrating conditional expressions.



Migrating data dictionary

The data dictionary contains the building blocks for the environment. It consists of a variety of information, including database objects, attributes, and their relationships, as well as user information, such as organization, site, and currency. A few examples of uses of data dictionary migration include a new object, and its attributes, relationships, and domains to use in conjunction with other migrated data to build a custom application, or the addition of a new international site along with its default language and currency.

This chapter has the following sections:

- ▶ 2.1, “Requirements” on page 30
- ▶ 2.2, “Solution” on page 30
- ▶ 2.3, “Configuration applications” on page 31
- ▶ 2.4, “Object structures” on page 32
- ▶ 2.5, “Migration groups” on page 35
- ▶ 2.6, “Package definition” on page 37
- ▶ 2.7, “Deployment” on page 47
- ▶ 2.8, “Deployment considerations” on page 48

2.1 Requirements

The data dictionary objects are defined in a development environment and subsequently migrated to a production environment. For this scenario, we use the preformatted Data Dictionary migration package. Not all of the migration objects in the package will be explicitly covered in this scenario, but we will describe the major objects.

A good practice in any new development environment is to set up Change migration packages. The Change package can be used to migrate those changes made since its activation. Even if not used to migrate data, Change packages can be used to track environmental changes for those objects that are supported by the Migration Manager.

Change packages were set up in this scenario prior to any new content creation to facilitate an accurate list of those configurations to migrate.

2.2 Solution

Multiple packages will be created to facilitate the migration of two sets of configurations.

The first set of configurations consists of the addition of the typical components that are promoted when a new site is added to an existing organization. These objects are migrated in the following order:

- ▶ Currency
- ▶ Sets
- ▶ Site

The second set of configurations consists of configurations that are typically migrated from the data dictionary to support adding a new tab with a table to an existing application and the conditions to filter the data based on existing values for particular records. We will migrate these objects in order:

- ▶ Lookup map
- ▶ Domains
- ▶ Objects
- ▶ Relationships
- ▶ Conditional expressions
- ▶ Domains

Domains are included twice in the lists. Due to the complexity of crossover and table domains, extra steps are needed to migrate them correctly. Refer to the 2.7, “Deployment” on page 47 for more information.

Because the data dictionary package moves information that is needed by other packages, no external object dependencies need to be satisfied as part of the migration. However, other packages are dependent on configurations that are moved as part of the data dictionary migration. Proper planning needs to be taken to ensure that all migrations occur in the correct order.

2.3 Configuration applications

The Database Configuration application manages objects, attributes, indexes, views, tables, columns, autokeys, relationships, services, lookup maps, messages, sequences, and GL account configuration. It can be accessed from the Start Center by navigating to **Go To → System Configuration → Platform Configuration → Database Configuration**.

The Domains application manages the domains. These domains can be associated with attributes in the Database Configuration. The domains can be accessed from the Start Center by navigating to **Go To → System Configuration → Platform Configuration → Domains**.

The Sets application manages item and company sets that are associated with organizations. The sets can be accessed from the Start Center by navigating to **Go To → Administration → Sets**.

The Currency Codes application manages currency codes that are associated with organizations. The currency codes can be accessed from the Start Center by navigating to **Go To → Financial → Currency Codes**.

The Organizations application manages organizations, sites, and addresses. The organizations, sites, and addresses can be accessed from the Start Center by navigating to **Go To → Administration → Organizations**.

The Conditional Expression Manager application manages conditional expressions. The conditional expressions can be used with several other configuration objects. The conditional expressions can be accessed from the Start Center by navigating to **Go To → Administration → Conditional Expression Manager**.

The Object Structures application manages object structures that are used by the Integration Framework to import data and by the Query-Based Report functionality to join tables for ad-hoc reporting. The object structures can be

accessed from the Start Center by navigating to **Go To** → **Integration** → **Object Structures**.

2.4 Object structures

There are 15 predefined object structures that are part of the Data Dictionary migration package.

The DMMAXSERVICE object structure supports services migration (Figure 2-1).

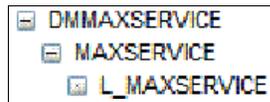


Figure 2-1 DMMAXSERVICE

The DMLANGUAGE object structure supports language migration (Figure 2-2).

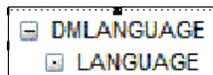


Figure 2-2 DMLANGUAGE

The DMMAXMESSAGES object structure supports messages migration (Figure 2-3).

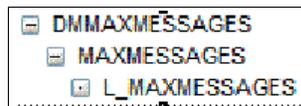


Figure 2-3 DMMAXMESSAGES

The DMMAXLOOKUPMAP object structure supports lookup migration (Figure 2-4).

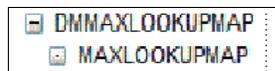


Figure 2-4 DMMAXLOOKUPMAP

The DMCURRENCY object structure supports currency migration (Figure 2-5 on page 33).

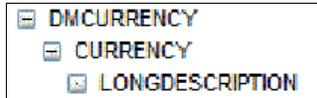


Figure 2-5 DMCURRENCY

The DMSETS object structure supports item and company set migration (Figure 2-6).

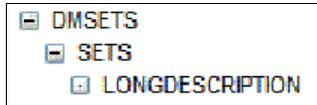


Figure 2-6 DMSETS

The DMORGANIZATION object structure supports organization, address, and site migration (Figure 2-7).

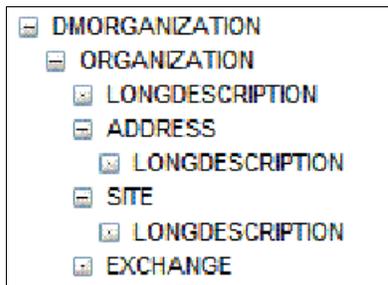


Figure 2-7 DMORGANIZATION

The DMMAXVARS object structure supports the maxvars migration (Figure 2-8).

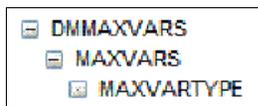


Figure 2-8 DMMAXVARS

The DMMAXDOMAIN object structure supports domain migration (Figure 2-9 on page 34).

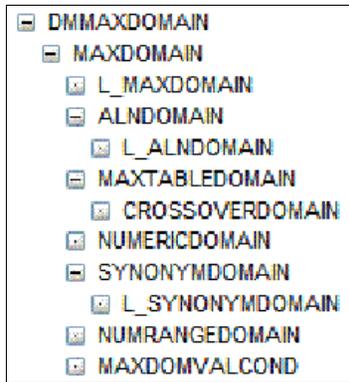


Figure 2-9 DMMAXDOMAIN

The DMMAXOBJECTCFG object structure supports table, view, column, attribute, and index migration (Figure 2-10).

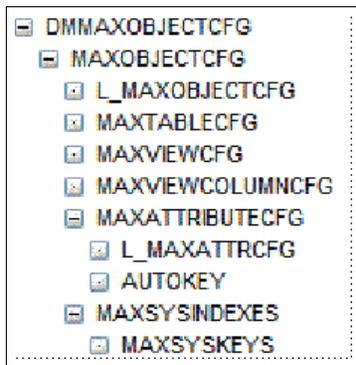


Figure 2-10 DMMAXOBJECTCFG

The DMMAXSEQUENCE object structure supports sequence migration (Figure 2-11).

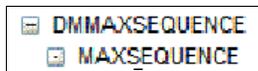


Figure 2-11 DMMAXSEQUENCE

The DMMAXRELATIONSHIP object structure supports relationship migration (Figure 2-12 on page 35).

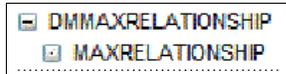


Figure 2-12 DMMAXRELATIONSHIP

The DMMAXINTOBJECT object structure supports integration object migration (Figure 2-13).



Figure 2-13 DMMAXINTOBJECT

The DMCONDITION object structure supports conditional expression migration (Figure 2-14).

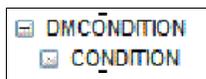


Figure 2-14 DMCONDITION

The DMGLCONFIGURE object structure supports general ledger component migration.

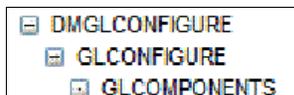


Figure 2-15 DMGLCONFIGURE

2.5 Migration groups

The object structures listed in the previous section are part of the DATA DICTIONARY migration group. Because the Data Dictionary migration group is the first group to be imported in any set of migration packages, the predefined package can be used, or a copy can be made and modified specifically to the configurations to migrate.

Figure 2-16 on page 36 shows the DATA DICTIONARY group.



Figure 2-16 Data Dictionary group

There are two methods for limiting the data that is brought over by the migration package. The first method is to duplicate the initial package and remove those object structures from the predefined package. The second method will be used in this scenario. We describe it in more detail in 2.6, “Package definition” on page 37. One of the features of the Migration Manager application is the ability to duplicate existing migration packages. This feature is helpful here, because multiple data dictionary packages will be used to migrate separate functionality. Because the Data Dictionary package has no dependencies, using the duplicate functionality and the modifications to the Where clause in the package saves time in the process.

The order of the object structures within the migration group is shown in Figure 2-17 on page 37.

Migration Object	Description	Migration Object Order
DMMAXSERVIC	System service migration object for Migration M	1
DMLANGUAGE	Language migration object for Migration Manag	2
DMMAXMESSA	System message migration object for Migration	3
DMCURRENCY	Currency migration object for Migration Manage	4
DMSETS	Set migration object for Migration Manager	5
DMORGANIZAT	Organization and site migration object for Migra	6
DMMAXVARS	MAXVARS migration object for Migration Mana	7
DMCONDITION	Condition expression migration object for Migra	8
DMMAXDOMAIN	Domain migration object for Migration Manager	9
DMMAXOBJECT	Data dictionary migration object for Migration Manager	10
DMMAXSEQUEN	Data dictionary sequence migration object for Migration Manager	11
DMMAXLOOKUP	Lookup map migration object for Migration Manag	12
DMMAXRELATION	Data dictionary relationship migration object for Migration Manager	13
DMMAXINTOBJ	Object structure migration object for Migration M	14
DMGLCONFIGU	GL configuration and component migration obje	15

Figure 2-17 Data Dictionary migration objects

Easier viewing: Figure 2-17 on page 37 has been modified to show all 15 objects in the object structure instead of our creating three separate screenshots showing the usual five objects each.

2.6 Package definition

A new Change migration package is created for the Data Dictionary using predefined Data Dictionary Migration package. We created this new Change migration package prior to building all of the configurations that are promoted from development to production as part of this scenario.

The Migration Manager application can be accessed from the Start Center by navigating to **Go To** → **System Configuration** → **Migration** → **Migration Manager**.

The process for building the Change package is fairly straightforward, and it is performed following product documentation. Figure 2-18 on page 38 shows the Package Definition tab of the new Change package.

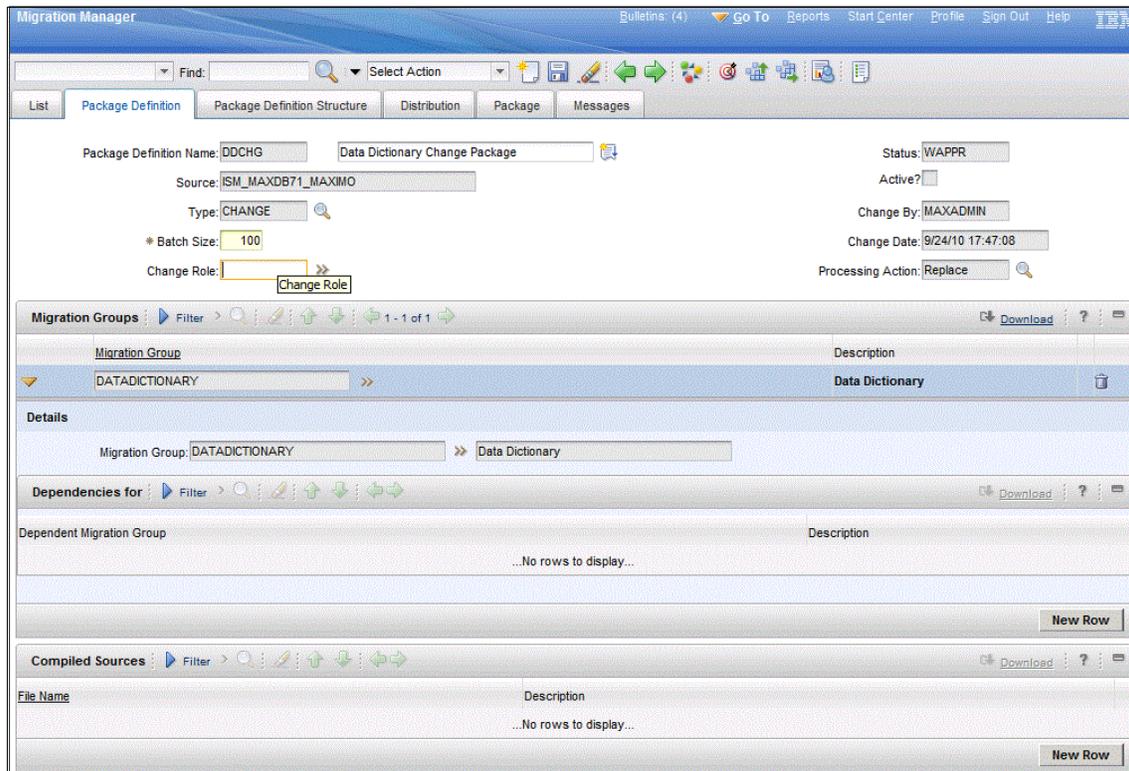


Figure 2-18 Data Dictionary Change Package

The package is approved using the Change Status toolbar or Select Action menu item and is activated using the Activate Package Definition Select Action menu item. After the new package is active, the Change package will now track data dictionary changes.

Accessing the Change package after building the configurations to migrate and selecting View Event Tracking Records shows the new configurations, as shown in Figure 2-19 on page 39.

View Event Tracking Records							
Root Event Tracking Records							
Object	Action	Primary Keys	Event Date	Event User	Object	Object Id	
CONDITION	ADD	CONDITIONNUM=SUPERVISOR	9/24/10 19:45:26	MAXADMIN	CONDITION	979	
CURRENCY	ADD	CURRENCYCODE=HKD	9/24/10 18:23:57	MAXADMIN	CURRENCY	18	
MAXDOMAIN	ADD	DOMAINID=SUPERVISOR	9/24/10 19:55:59	MAXADMIN	MAXDOMAIN	768	
MAXINTOBJECT	UPDATE	INTOBJECTNAME=REP_PERSON	9/24/10 20:19:42	MAXADMIN	MAXINTOBJECT	203	
MAXLOOKUPMAP	ADD	TARGET=EMPLOYEE,LOOKUPATTR=EMPLOYEE,TARGETATTR=EMPLOYEE,SOURCE=PERSON	9/24/10 20:09:46	MAXADMIN	MAXLOOKUPMAP	888,945	
MAXOBJECTCFG	UPDATE	OBJECTNAME=PERSON	9/24/10 19:10:35	MAXADMIN	MAXOBJECTCFG	26	
MAXOBJECTCFG	ADD	OBJECTNAME=EMPLOYEE	9/24/10 19:04:42	MAXADMIN	MAXOBJECTCFG	1,557	
ORGANIZATION	UPDATE	ORGID=TEST	9/24/10 18:40:24	MAXADMIN	ORGANIZATION	8	
SETS	ADD	SETID=ITEM1	9/24/10 18:28:35	MAXADMIN	SETS	12	
SETS	ADD	SETID=COMP1	9/24/10 18:30:54	MAXADMIN	SETS	13	
Child Event Tracking Records							
Object	Action	Primary Keys	Event Date	Event User	Parent Object	Parent Object Id	
MAXINTOBJDETAIL	ADD	INTOBJECTNAME=REP_PERSON,OBJECTID=6	9/24/10 20:19:42	MAXADMIN	MAXINTOBJECT	203	
MAXINTOBJCOLS	ADD	INTOBJECTNAME=REP_PERSON,OBJECTNAME=EMPLOYEE,NAME=HASLD,INTOBJFLDTYPE=EXCLUDE	9/24/10 20:19:42	MAXADMIN	MAXINTOBJDETAIL	919	

Figure 2-19 Data Dictionary Change Package Tracking Records

At this point, it is possible to select the items to promote, build, and distribute the package.

Three packages are built to migrate the data. The first package migrates the currency, organization, and sets. We separated this package from the other packages to illustrate that multiple users can build and migrate separate sections of the data dictionary as required by their individual projects. This package does not include the migration of a new organization or clearing account. We discuss the issues with migrating a new organization or clearing account in 2.8, “Deployment considerations” on page 48.

The second and third packages will be used to migrate the rest of the configurations. Two packages are required due to the inclusion of a crossover domain. Refer to 2.8, “Deployment considerations” on page 48 to review the discussion of the issue with Crossover/Table domain migration.

The first step is to create the packages. This creation follows the standard process that is defined in product documentation and is similar to the setup of the Change package with two exceptions: a package TYPE of SNAPSHOT and the definition of the SQL that is used to define the package's Where clauses. We discuss these exceptions in detail next:

1. In the first package, you can obtain the names of the organization to be updated and the sets and currency to be added in the event tracking of the initially created Change package or just documentation created for the project by the developer. These names are used in the SQL to define the CURRENCY, ORGANIZATION and SET values. All other objects not migrated were set to 1=2. Figure 2-20 through Figure 2-22 on page 41 illustrate the data that is migrated in this package.



Figure 2-20 New currency

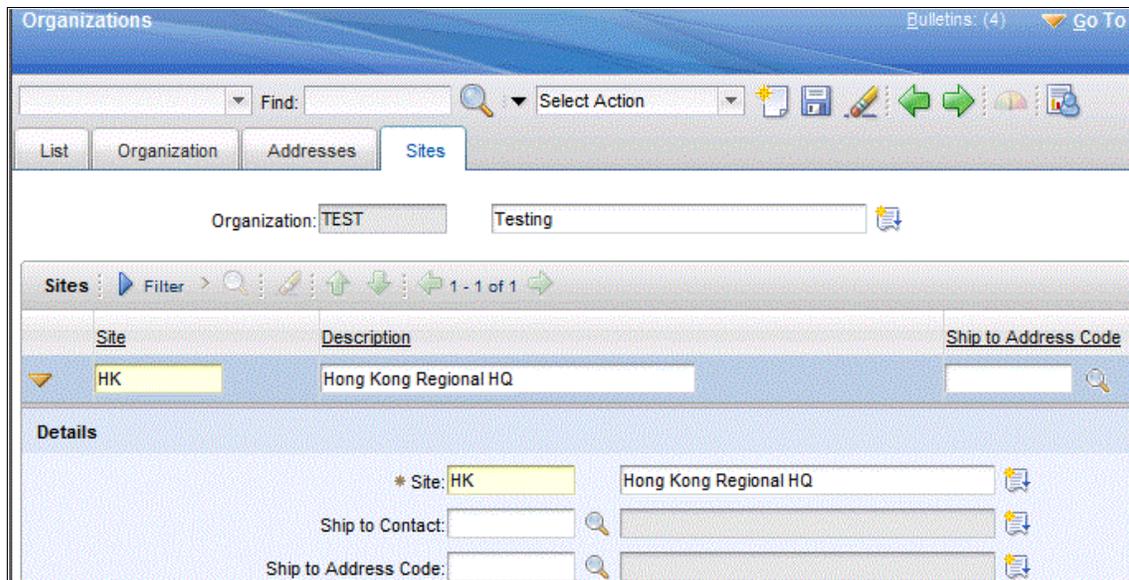


Figure 2-21 New site for existing organization

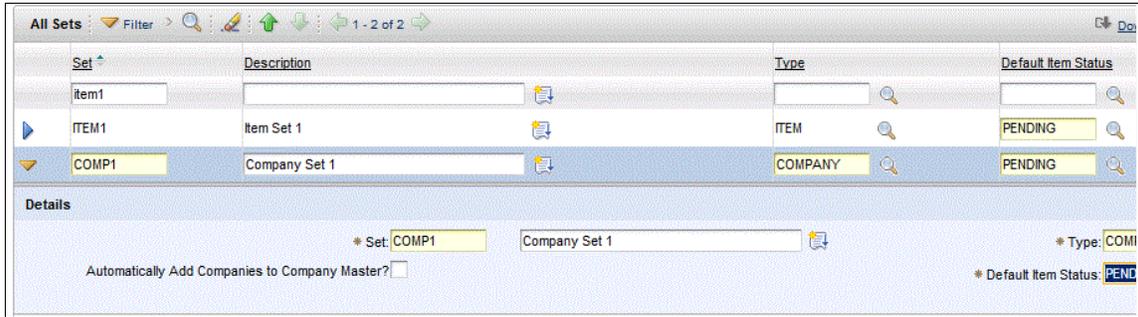


Figure 2-22 New item set and company set

- The names are used in the SQL to define the CURRENCY, ORGANIZATION, and SET values. All other objects that are not migrated are set to 1=2. The column names that are used in the Where clause for each object can be found either in the Change Package Event Tracking or in the field help for each update in its application. See Figure 2-23.

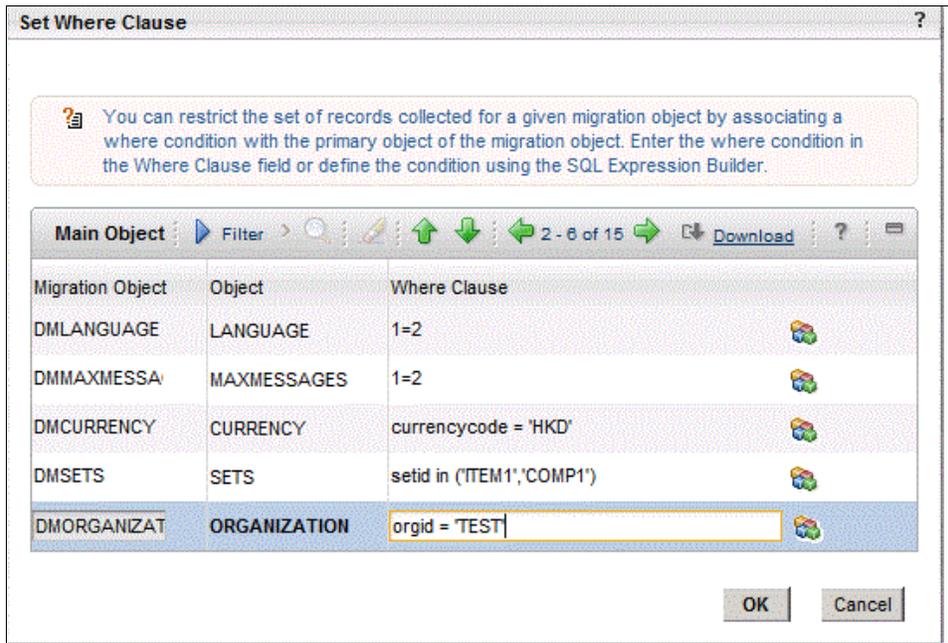


Figure 2-23 First Snapshot Package Where clause

- After the SQL criteria have been updated and all of the fields have been defined, the package is saved.

4. The other packages can be created from the recently created Snapshot package by using the Duplicate Package Definition option in the Select Action menu. This option duplicates all of the information in the existing package with the exception of the Package Definition Name and description. You must enter this information manually.
5. The SQL in the Where clauses for the second package will be updated to migrate all of the other objects with the exception of the DOMAIN, as described in 2.8.2, “Crossover and table domains” on page 48. Prior to the second package deployment, the change to the object structure is made and it is reverted prior to the third package deployment. Figure 2-24 through Figure 2-31 on page 46 illustrate the objects that are migrated in this package.

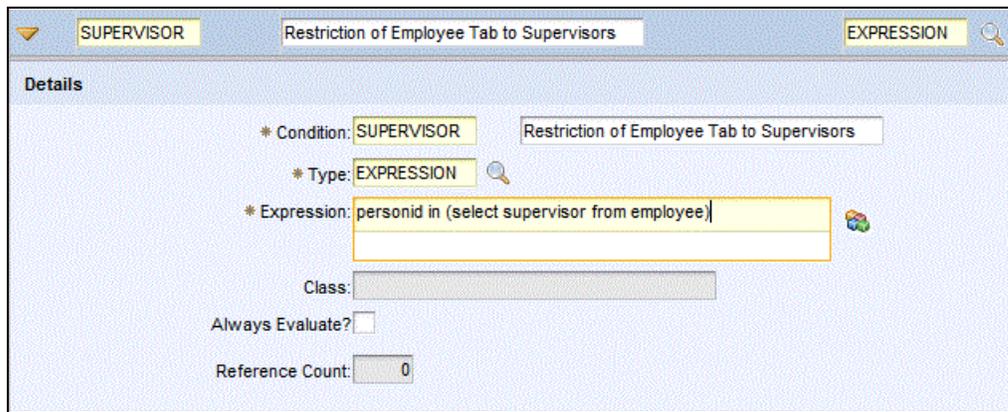


Figure 2-24 New conditional expression

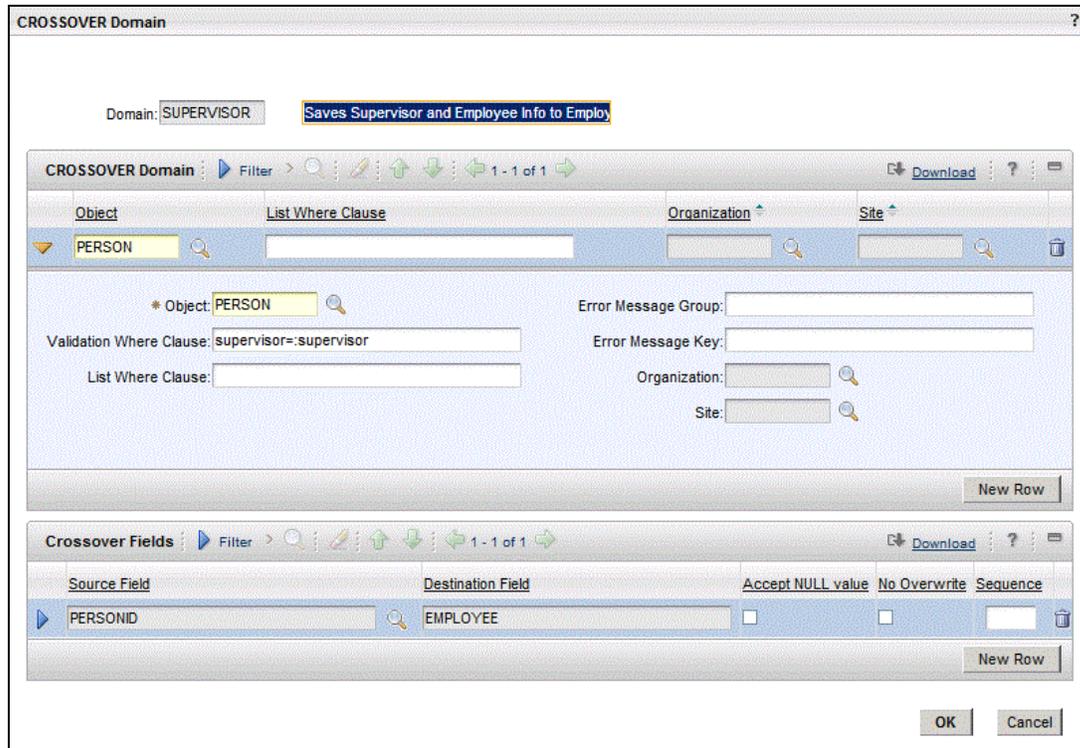


Figure 2-25 New CROSSOVER Domain

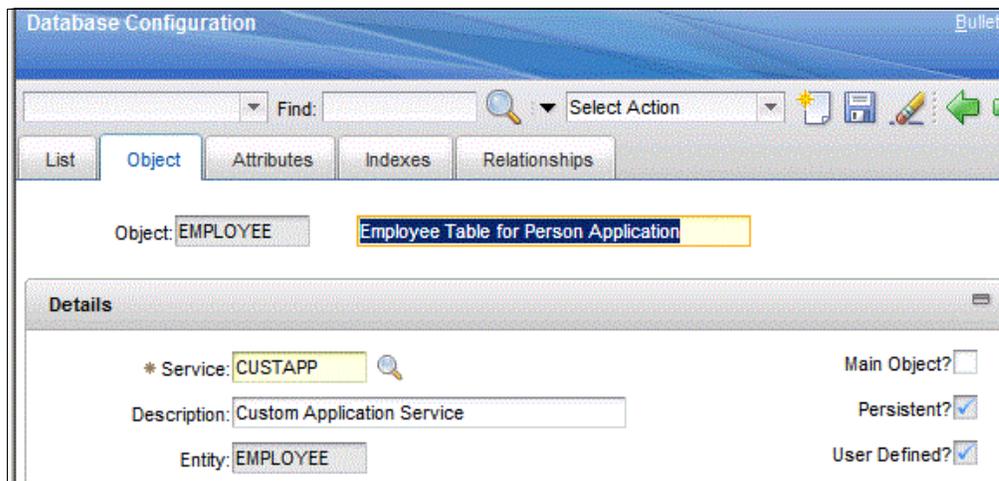


Figure 2-26 New EMPLOYEE object

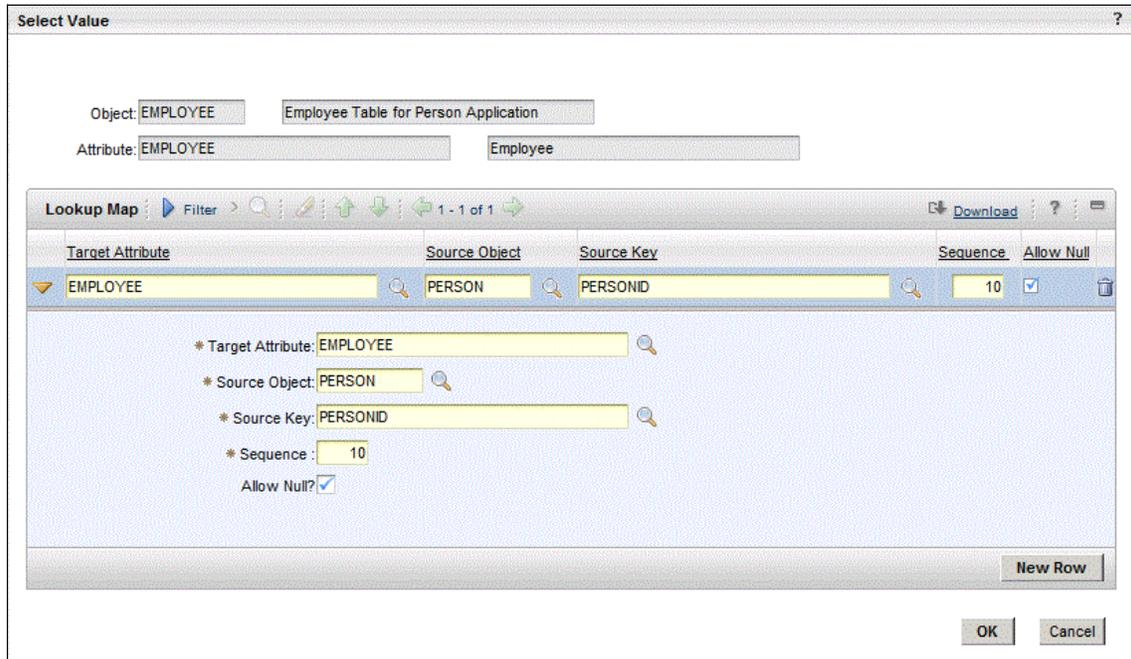


Figure 2-27 New Lookup Map

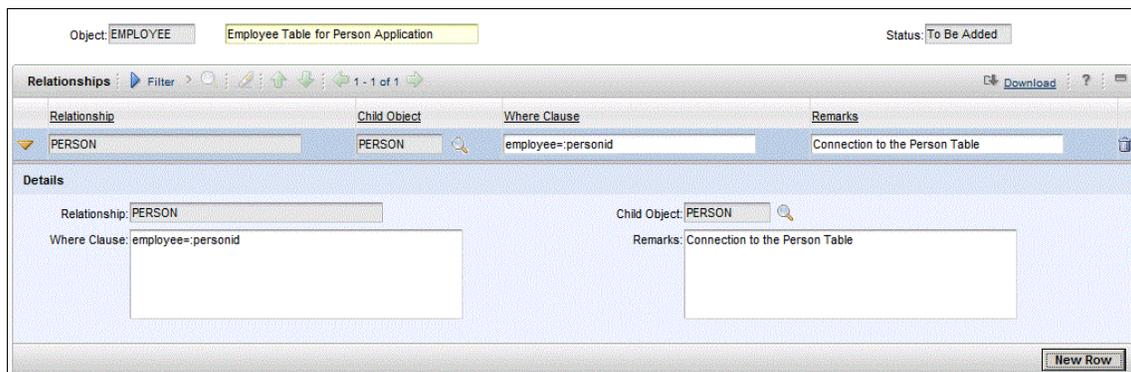


Figure 2-28 New Relationship for EMPLOYEE object

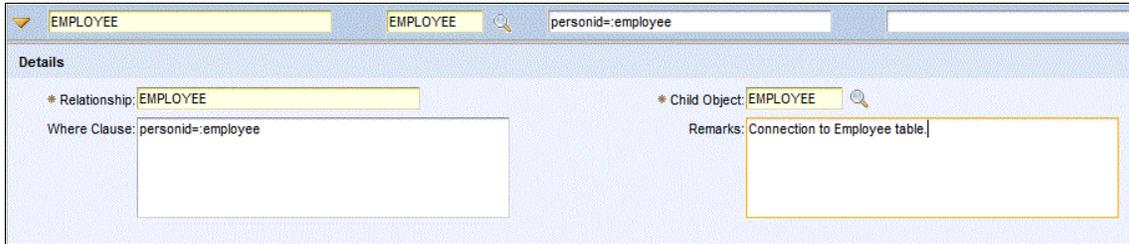


Figure 2-29 New Relationship for Person object

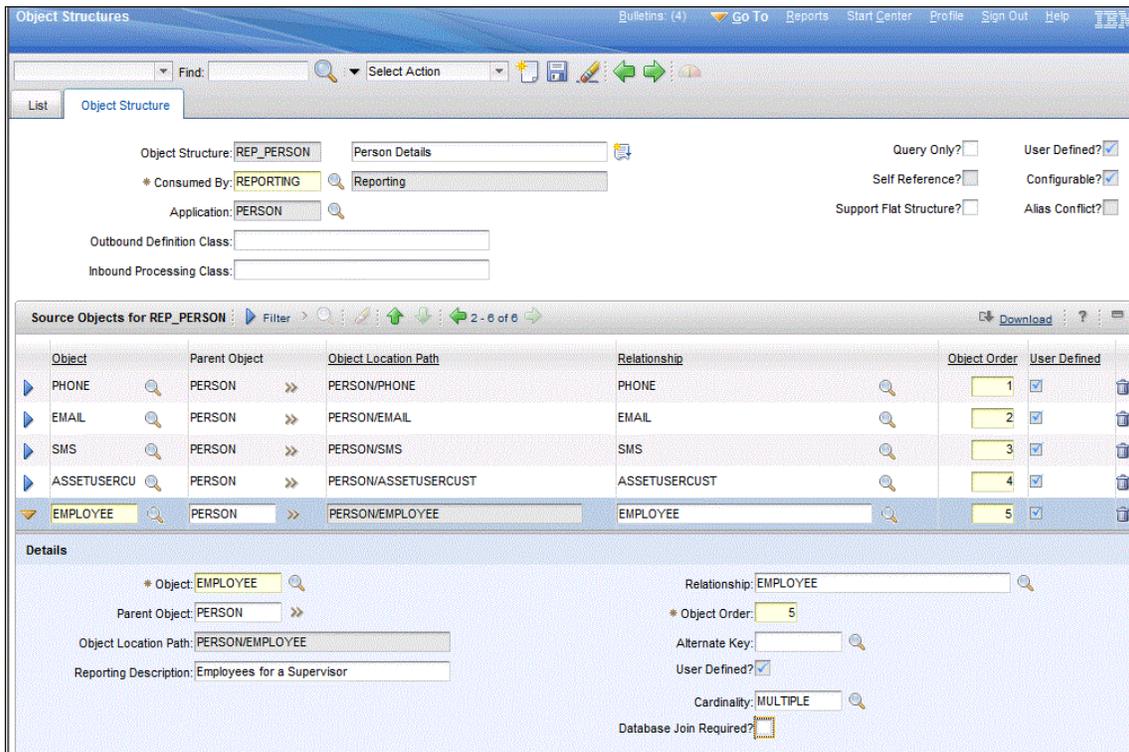


Figure 2-30 New Join to EMPLOYEE object in existing PERSON report object structure

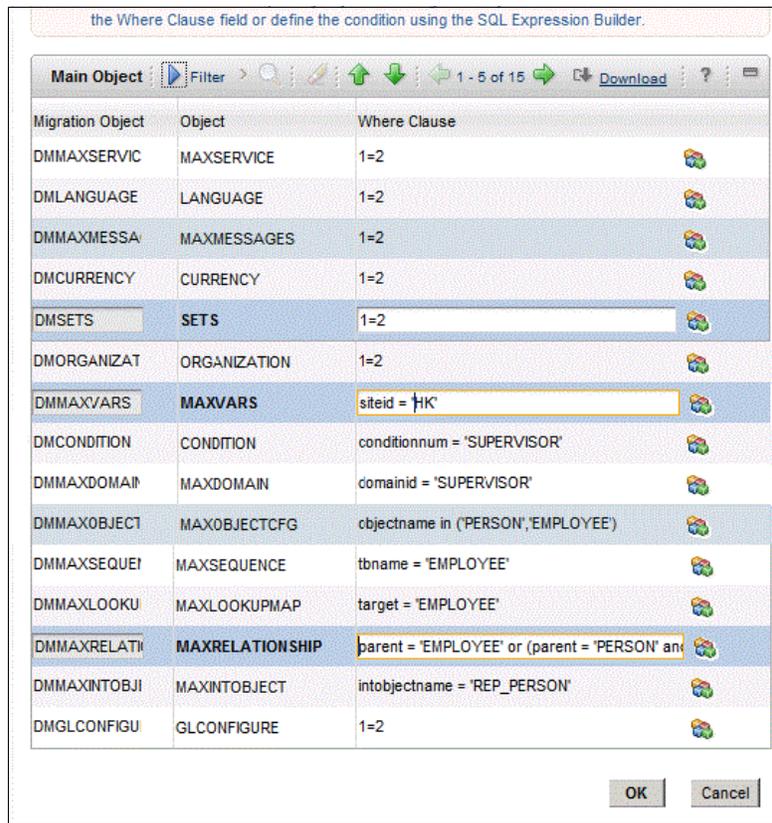


Figure 2-31 Second Data Dictionary Snapshot package

6. Now that the SQL criteria have been updated related to the second package, they can be saved.
7. After this package is saved, the third package is built from a duplicate of either the first or second package. Again, the Package Definition Name and description are not duplicated and must be entered manually.
8. The SQL criteria for the final package only contains the Where clause for the MAXOBJECTCFG. After the object structure exclusion is reverted, the linkage between the object and the domain is created, once deployed. All other objects will be set to 1=2 to exclude them from the migration. See Figure 2-32 on page 47.

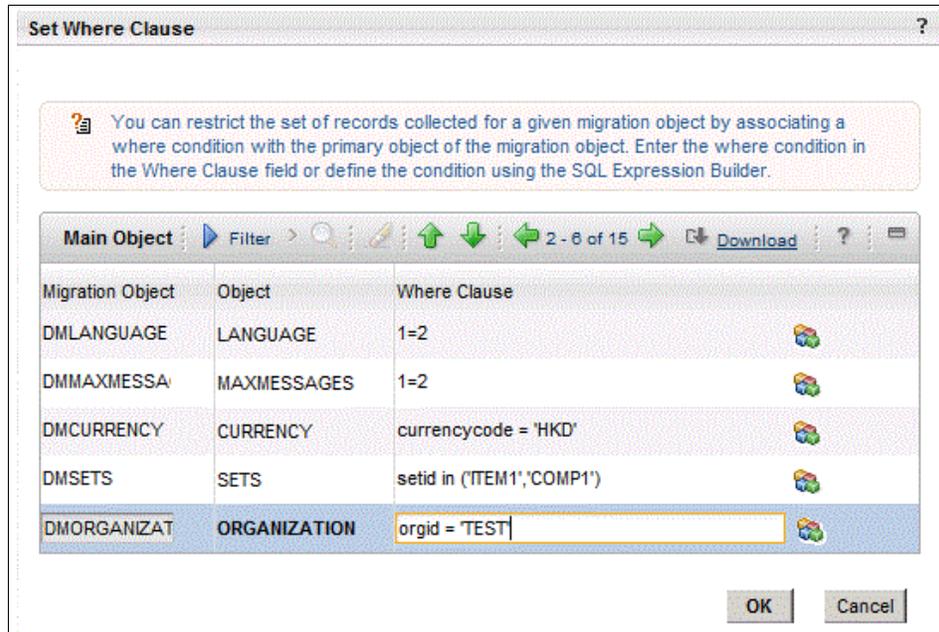


Figure 2-32 Third Data Dictionary Snapshot package

- Now that the SQL criteria for the final package have been updated, it can be saved.

2.7 Deployment

After the packages have been saved, they can be exported from the Source system. The crossover domain migration requires the extra steps that are defined in 2.8, “Deployment considerations” on page 48. Prior to the second package export, the DOMAINID must be excluded from the DMMAXOBJECTCFG object structure. And prior to the third package export, the DOMAINID must be included again in the DMMAXOBJECTCFG object structure.

Any data dictionary packages that make structural changes to the database require Admin mode or a restart to complete the migration. In this example, structural changes are made. After Admin mode is activated, the packages can be imported. You can obtain the exact steps for deployment in the product documentation.

Tip: When significant structural changes are made to the database, a best practice is to make a backup prior to the import of the new migration packages.

2.8 Deployment considerations

In the following sections, we will discuss the deployment considerations for this migration.

2.8.1 Organization and clearing accounts

The organization and the initial General Ledger (GL) account, which is identified as the clearing account in the organization, are not migrated. The GL account has to be associated with the organization, but the organization cannot be activated prior to the general ledger account being associated. The Migration Manager does not have the facility to activate the organization, so manual intervention is required to create the association and activate the organization. It might be more efficient in the case of these items to manually enter the values in the new environment.

2.8.2 Crossover and table domains

A crossover and table domain is one of the migrated objects. You must be extremely careful to associate this domain correctly with the related attribute. This process requires two packages: one package does not contain the domain and a second package contains the domain. We build and deploy these packages:

- ▶ Build, export, and import into the new environment the first package with no domain reference. The domain reference can be temporarily excluded from the object structure DMMAXOBJECTCFG in the Source environment by opening the object structure and excluding the DOMAINID field from the MAXATTRIBUTECFG using the Exclude/Include Fields pop-up dialog.
- ▶ Build, export, and import into the new environment the second package with the domain reference. The domain reference can be re-added by following the reverse of the previous steps that were used to exclude the DOMAINID field from the first package. This process creates the proper link between the attribute and the domain.

Figure 2-33 shows the exclusion or inclusion that is required for crossover and table domain migration.

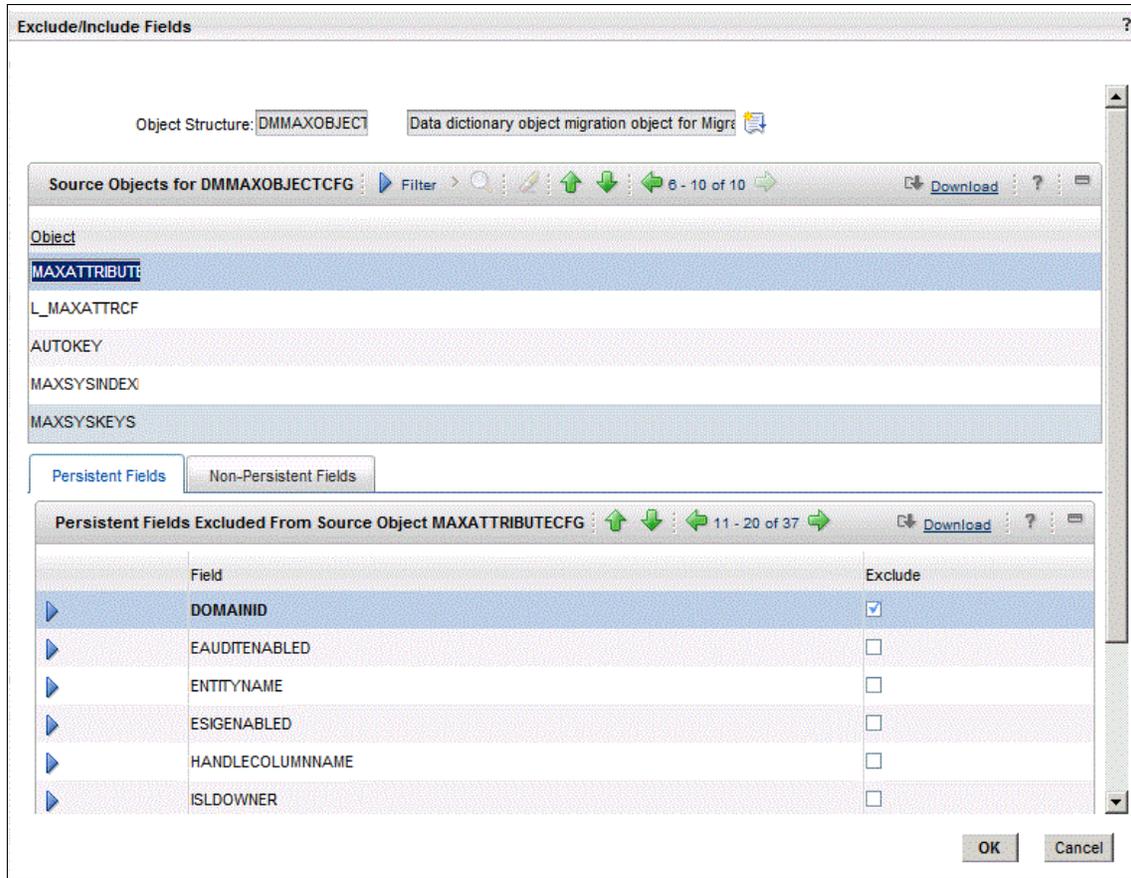


Figure 2-33 Exclusion or inclusion required for crossover and table domain migration



Migrating security configuration data

This chapter describes the use case scenarios for security configuration data migration. Administrators perform security configuration creating users, grouping them by security groups, and granting authorizations. We provide these scenarios:

- ▶ 3.1, “Migrating a new security group” on page 52
- ▶ 3.2, “Migrating the conditional user interface” on page 61
- ▶ 3.3, “Migrating global data restrictions” on page 68
- ▶ 3.4, “Migrating access definitions and LDAP information” on page 76
- ▶ 3.5, “Considerations of security migrations” on page 82

3.1 Migrating a new security group

System administrators configure the access that users can have to the system based on business requirements, for example:

- ▶ Only users who are members of the PLANNING security group can use the Job Plans and Routes applications.
- ▶ Users who are members of the MAINTENANCE security group can initiate and complete work orders.
- ▶ Only users who are members of the FINANCIAL security group can access the Chart of Accounts application.
- ▶ Users who are members of the MAINTENANCE security group cannot see closed or canceled work orders.

This scenario covers the migration of a new security group with new users, and the access definitions for existing application options. Additionally, this scenario includes the migration of conditions that are used for data restrictions.

To perform the tasks that we describe here, you must log in as an administrative user.

3.1.1 Requirements

The requirements for this migration consist of defining a new security group RBGROUP, with a new user RBUSER. This group has full access to all of the options of the Work Order Tracking application, but members of the group have a data restriction that blocks any changes in the work orders that are waiting on an approval status.

All of the configurations that we mention are created in the development environment and subsequently promoted to the test and production environments.

3.1.2 Solution

We define an approach to migrate a new security group and its related data in a single Snapshot migration package.

From a content perspective, the migration of a security group in this scenario also implies the migration of the following configurations in order:

- ▶ Conditions
- ▶ Security groups
- ▶ Users

Note: A security group might have a default Start Center template, a list of sites to which the group is authorized, and applications. It is not part of the requirements of this scenario to migrate sites, Start Center templates, and applications. We assume that these configurations have already been migrated or already exist in the Target environment. For more details, refer to 3.1.8, “Deployment considerations” on page 59.

3.1.3 Configuration applications

The following sections show the data that was created as an example for this migration scenario.

Conditional Expression Manager

Conditions are either SQL expressions or custom Java classes that are used, for example, to limit data access or editability. The restriction that the condition defines is only applied when the conditional expression evaluates to true. Conditions are created and managed using the Conditional Expression Manager application, which is accessible by selecting **Go To** → **Administration** → **Conditional Expression Manager**.

For this scenario, we created a condition with the values that are shown in Table 3-1.

Table 3-1 New condition values

Field	Value
Condition	RBOOKCOND
Description	IBM Redbooks Condition
Type	EXPRESSION
Expression	:status = 'WAPPR'
Always Evaluate	Selected

Security groups

You can create and manage security groups by using the Security Groups application, which is accessible by clicking **Go To** → **Security** → **Security Groups**.

For this scenario, we created a new group with the values that are shown in Table 3-2.

Table 3-2 New security group values

Field	Value
Group	RBGROUP
Description	IBM Redbooks Group

Under the Sites tab, the “Authorize Group for All Sites” check box is selected.

Under the Data Restrictions tab, we created an object restriction for the WORKORDER object. This data restriction, when evaluating to true, changes the editability of waiting on approval work orders to “read only”. Table 3-3 shows the values of this new data restriction.

Table 3-3 New object restriction

Field	Field
Object	WORKORDER
Type	READONLY
Condition	RBOOKCOND

Important: You can use the same condition for multiple data restrictions. In this example, when you set the Object field to WORKORDER, you are creating this association.

You can associate the same condition with other objects, if the condition can be applied to the other objects.

Under the Applications tab, the group received access to all of the options of the Work Order Tracking application.

Under the GL Components tab, we selected all of the component check boxes.

Users

You can create and manage users with the Users application, which is accessible by clicking **Go To** → **Security** → **Users**.

For this scenario, we created a new user with the values that are shown in Table 3-4.

Table 3-4 New user values

Field	Value
User	RBUSER1
User Name	RBUSER1
Type	TYPE 1

Important:

- ▶ The Person field is required, and the system asks if you want to create a new PERSON. It is not part of this scenario to migrate data that is created in the People application. We assumed that the person that you choose already exists in the Target environment and that its status is ACTIVE.
- ▶ The product has a predefined migration group called RESOURCES, which has the DMPERSON object structure.
- ▶ The password field might be required. If so, you need to enter a valid value, depending on your system configuration. We assumed that you are able to create a new user with all of the required data.
- ▶ We discuss details about referenced data and password migration in 3.1.8, “Deployment considerations” on page 59.

Under the Groups tab, we added the new group RBGROUP.

3.1.4 Object structures

The following object structures are delivered with the product to support the application security migration for this scenario.

Figure 3-1 shows the DMCONDITION object structure in the DATADictionary migration group supporting conditions.

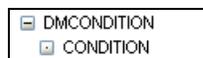


Figure 3-1 Condition object structure

Figure 3-2 shows the DMMAXUSER object structure in the APPSECURITY migration group supporting users.

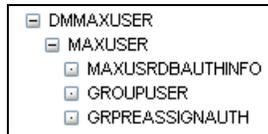


Figure 3-2 User object structure

Figure 3-3 shows the DMMAXGROUP object structure in the APPSECURITY migration group supporting the security groups.

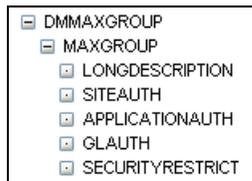


Figure 3-3 Security group object structure

3.1.5 Migration groups

The object structures that are listed in 3.1.4, “Object structures” on page 55 are located in two separate migration groups. The migration groups that are available in the product include dependencies on other migration groups, which are not required in the current scenario.

A suitable and efficient approach to migrate the data that you have created is to define a new migration group that contains only the object structures that have been identified to achieve the migration requirements of this scenario.

You can create the new migration group from the Migration Groups application, which is accessible by clicking **Go To** → **System Configuration** → **Migration** → **Migration Groups**.

Tip: You can duplicate the existing APPSECURITY group and remove all of the dependencies.

Table 3-5 on page 57 shows the values of the new migration group. All other fields have default values.

Table 3-5 New migration group

Field	Value
Migration Group	RBSECURITY01
Description	IBM Redbooks Security 01

Figure 3-4 shows the object structures that are in the new group and their order.

Migration Object	Description	Migration Object Order	Internal
DMCONDITION	Condition expression migration object for Migra	1	<input type="checkbox"/>
DMSIGOPTION	Signature option migration object for Migration I	2	<input type="checkbox"/>
DMMAXSERVSI	Service security migration object for Migration I	3	<input type="checkbox"/>
DMMAXGROUP	Security group migration object for Migration M	4	<input type="checkbox"/>
DMMAXUSER	User migration object for Migration Manager	5	<input type="checkbox"/>
DMSIGOPTFLA	Advanced signature option migration object for	6	<input type="checkbox"/>
DMCTRLGROUF	Control security migration object for Migration I	7	<input type="checkbox"/>

Figure 3-4 RBSECURITY01 object structures

3.1.6 Package definition

At this point, we have both the data set and the migration group defined, and the migration process can begin. The first step is to define the package using the Migration Manager application, which is accessible by clicking **Go To** → **System Configuration** → **Migration** → **Migration Manager**.

A single Snapshot package is enough to migrate the security configurations that we have made. The package needs to contain the migration group that you created in 3.1.5, “Migration groups” on page 56. Table 3-6 shows the values of this package. All other fields have default values.

Table 3-6 RBSECPKG01 migration package values

Field	Value
Package Definition Name	RBSECPKG01
Description	IBM Redbooks Security Pkg 01
Type	SNAPSHOT
Processing Action	AddChange

Under the Migration Groups tab, we have added the RBSECURITY01 migration group.

Defining SQL criteria

Figure 3-5 shows the queries that were created for each object structure that is part of the migration group.

Migration Object	Object	Where Clause
DMCONDITION	CONDITION	conditionnum in ('RBOOKCOND')
DMSIGOPTION	SIGOPTION	1=2
DMMAXSERVSI	MAXSERVSECURITY	1=2
DMMAXGROUP	MAXGROUP	groupname in ('RBGROUP')
DMMAXUSER	MAXUSER	status <> 'DELETED' and userid in(select useric
DMSIGOPTFLA	SIGOPTFLAG	1=2
DMCTRLGROUF	CTRLGROUP	1=2

Figure 3-5 Object structures queries

The complete Where clause for the DMMAXUSER migration object is:

```
status <> 'DELETED' and userid in(select userid from groupuser where
groupname in ('RBGROUP'))
```

Tip: Certain queries are defined as 1=2. You do not need to create a new migration group for other security migration-related scenarios. You only need to create other packages and define separate filters.

If an incorrect SQL Where clause is entered, the Migration Manager application will report the BMXAA7010E error. You must correct the SQL Where clause before the package definition can be saved.

Best practice: You must always exclude users whose status is DELETED from your migration package. If those users are part of the migration package, the system will report the BMXAA3837E error when you try to deploy the package to the Target environment.

You must also avoid migrating the MAXADMIN and MAXINTADM system users. If those users are part of the migration package, the system will report the BMXAA3829E error when you try to deploy the package to the Target environment.

3.1.7 Deployment

After you have saved and approved the package definition, you can proceed with the creation, distribution, and deployment.

3.1.8 Deployment considerations

For this scenario, you must observe the important points that we describe in this section in order for your security configurations to work in the Target environment in the same way that they worked in the Source environment.

Dependencies

The migration you have performed depends on data that must be in the Target environment:

- ▶ Admin mode needs to be on in the Target environment.
- ▶ You need to migrate new applications, new objects, and changes to existing objects prior to the deployment.

Best practice: Apply the same license keys in the Target environment as in the Source environment. Using diverse licenses will enable or disable separate sets of applications, which will result in deployment errors.

- ▶ If you use a default Start Center template for your security groups, you need to migrate or create manually the Start Center in the Target environment prior to deployment. You can find information about Start Center template migration in Chapter 5, “Migrating applications and Start Centers” on page 109.
- ▶ You need to migrate or create manually the sites to which your group is authorized in the Target environment prior to deployment.

- ▶ The group that we created in this example authorizes the use of GL account components. You need to migrate or create manually the GL account configuration in the Target environment prior to deployment.
- ▶ The user that we created in this example was related to an existing person. You need to migrate or create manually all related persons in the Target environment prior to deployment. You can find an approach to migrating person information in Chapter 4, “Migrating escalations” on page 83.

Application options: It is not part of the migration requirements of this scenario to migrate application options (sigoptions). This configuration uses the DMSIGOPTION object structure, which is available in the product and part of the migration requirements that are described in 3.2, “Migrating the conditional user interface” on page 61.

Handling passwords

The Migration Manager application does not migrate passwords for security reasons. All migrated users have their password reset and receive an email message with a temporary new password, which expires after the first login. For this reason, an email server must be configured in the Target environment before the migration of the security package, and all migrated users need to have a valid email as part of their Person record.

Note: In case the email server is not configured or users do not receive their temporary password, if your system is not configured with Lightweight Directory Access Protocol (LDAP) or an auto-generated password, a user with administrative privileges can define temporary passwords manually.

For an LDAP scenario, refer to 3.4, “Migrating access definitions and LDAP information” on page 76.

Attribute restrictions

You can use conditions to define data restrictions not only for objects, but also for attributes. Attribute data restrictions are migrated in the same way that object restrictions are migrated. The difference is that the restrictions apply for an attribute and not for the whole object.

Data that is not migrated

The following list shows security-related data that is not migrated:

- ▶ Password hints (both questions and answers)
- ▶ Login tracking
- ▶ Native database users
- ▶ Storeroom authorization

- ▶ Labor authorizations
- ▶ Limits and tolerances
- ▶ Status history
- ▶ Purchase GL

3.2 Migrating the conditional user interface

As part of the product configurations, users can redefine an existing screen, create new application options, and set conditions to show or hide user interface controls based on business requirements, for example:

- ▶ Hide the Failure Reporting tab of the Work Order Tracking application for users who are members of the ENGINEERS group and if the work order is for preventive maintenance.
- ▶ Hide the Actual Start and Actual Finish fields in the major tab of the Work Order Tracking application when displaying work orders that have not been completed.

This scenario covers the migration of a new application option, new conditional user interface definitions, a new security group, and new access definitions.

To perform the tasks that we describe here, you must sign in as an administrative user.

3.2.1 Requirements

For this scenario, our requirement is to hide the Actual Start and Actual Finish fields when the user who is accessing the work order is a member of RBGROUP1 and the work order is not completed. To configure this requirement, one approach is to define a new RBGROUP1 security group and a new option named RBOPTION1 for the Work Order Tracking application and to configure the conditional user interface to hide the fields using a condition based on the work order status.

All of the configurations that we mention are created in development environment and subsequently promoted to the test and production environments.

3.2.2 Solution

We define an approach so that a set of new application options, the conditional user interface, the new group, and the related data is migrated in a single Snapshot migration package.

The migration of the conditional user interface that is described in this scenario implies the migration of the following configurations in order:

- ▶ Conditions
- ▶ Sigoptions
- ▶ Security groups
- ▶ Control security

3.2.3 Configuration applications

The following sections show the configuration that was created as an example for this migration scenario.

Conditional Expression Manager

Conditions are either SQL expressions or custom Java classes used, for example, to limit data access or editability. The restriction that the condition defines is only applied when the conditional expression evaluates to true. Conditions are created and managed using the Conditional Expression Manager application, which is accessible by selecting **Go To** → **Administration** → **Conditional Expression Manager**.

For this scenario, we created a condition with the values that are shown in Table 3-7.

Table 3-7 New condition values

Field	Value
Condition	RBOOKCOND1
Description	IBM Redbooks Condition 1
Type	EXPRESSION
Expression	:status = 'COMP'
Always Evaluate	checked

Security groups

You can create and manage security groups using the Security Groups application, which is accessible by clicking **Go To** → **Security** → **Security Groups**.

For this scenario, we created a new group with the values that are shown in Table 3-8 on page 63.

Table 3-8 New security group values

Field	Value
Group	RBGROUP1
Description	IBM Redbooks Group 1

Under the Sites tab, we select the “Authorize Group for All Sites” check box.

Authorizing sites: If you do not want to authorize all sites, make sure that you have already migrated the sites. See 3.1.8, “Deployment considerations” on page 59 for more details.

Under the Applications tab, this group received access to all of the options of the Work Order Tracking application.

Under the GL Components tab, we selected all of the component check boxes.

Application Designer

You can create and manage applications with the Application Designer application, which is accessible by clicking **Go To** → **System Configuration** → **Platform Configuration** → **Application Designer**.

After you open the application WOTRACK, you can create a new application option by clicking **Select Action** → **Add/Modify Signature Option**.

For this scenario, we created the application option in Table 3-9 for the Work Order Tracking application (WOTRACK).

Table 3-9 New sigoptions

Option	Description	Advanced signature options
RBOPTION1	IBM Redbooks Opt1	None

To configure a conditional property for an user interface control, you need to select the control first and then click **Control Properties** in the Application Designer toolbar, as shown in Figure 3-6 on page 64.

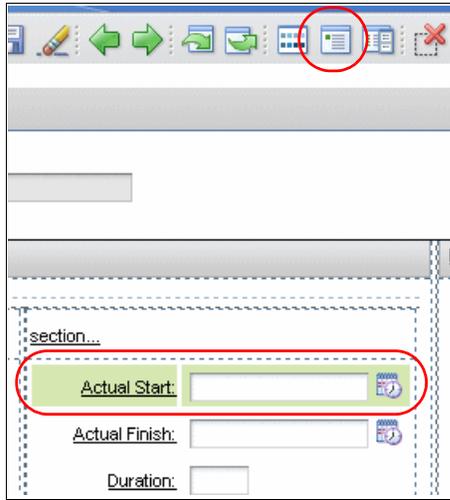


Figure 3-6 Control properties access

We needed to add conditional configuration to both the Actual Start and Actual Finish fields. Therefore, we selected the first field in the Work Order tab and clicked the Control Properties toolbar. You can control the conditional configuration by clicking Configure Conditional Properties in the dialog. The configuration that we created for this field is shown in Table 3-10.

Table 3-10 New control security condition for Actual Start field

Field or table	Value
Signature Option	RBOPTION1
Security Groups	RBGROUP1
Conditions for security group RBGROUP1	RBOOKCOND1
Property values when condition is true	display - true
Property values when condition is false	display - false

Then, we follow the same steps for the second field, using the same values that are shown in Table 3-10.

We returned to the Security Groups application to grant access to RBOPTION1 to group RBGROUP1. The access definitions need to be migrated as described in the migration requirements.

3.2.4 Object structures

The following object structures are delivered with the product to support the conditional user interface migration for this scenario.

The DMSIGOPTION object structure in the APPSECURITY migration group supports the application options, as shown in Figure 3-7.



Figure 3-7 DMSIGOPTION object structure

The DMCTRLGROUP object structure in the APPSECURITY migration group supports control security, as shown in Figure 3-8.

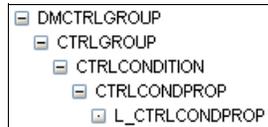


Figure 3-8 DMCTRLGROUP object structure

Refer to 3.1.4, “Object structures” on page 55 for the DMCONDITION and DMMAXGROUP object structures, which are required by this scenario.

3.2.5 Migration groups

The object structures that are listed in 3.2.4, “Object structures” on page 65 are contained into two separate migration groups. The migration groups that are available in the product include dependencies on other migration groups, which are not required in the current scenario.

A suitable and efficient approach to migrate the configuration that you have created is to define a new migration group that contains only the object structures that are identified for the purpose of this scenario. The migration group equals the group that we created in 3.1.5, “Migration groups” on page 56.

3.2.6 Package definitions

At this point, you have both the data set and the migration group defined, and the migration process can begin. The first step is to define the package using the Migration Manager application, which is accessible by clicking **Go To** → **System Configuration** → **Migration** → **Migration Manager**.

A single Snapshot package is enough to migrate the conditional user interface configurations that we have created, and it needs to contain the migration group that we created in 3.2.5, “Migration groups” on page 65. Table 3-11 shows the values of this package. All of the other fields have default values.

Table 3-11 RBUIPKG01 migration package values

Field	Value
Package Definition Name	RBUIPKG01
Description	IBM Redbooks UI Security Pkg 01
Type	SNAPSHOT
Processing Action	AddChange

Under the Migration Groups tab, we have added the RBSECURITY01 migration group.

Defining SQL criteria

Figure 3-9 shows the queries that we used to filter the data that was created in this example.

Migration Object	Object	Where Clause
DMCONDITION	CONDITION	conditionnum in ('RBOOKCOND1')
DMSIGOPTION	SIGOPTION	app in ('WOTRACK') and optionname in ('RBOP
DMMAXSERVSI	MAXSERVSECURITY	1=2
DMMAXGROUP	MAXGROUP	groupname in ('RBGROUP1')
DMMAXUSER	MAXUSER	1=2
DMSIGOPTFLA	SIGOPTFLAG	1=2
DMCTRLGROUF	CTRLGROUP	app in ('WOTRACK') and groupname in ('RBGR

Figure 3-9 Object structure queries

The complete Where clause for the migration object DMSIGOPTION is:
app in ('WOTRACK') and optionname in ('RBOPTION1')

The complete Where clause for the migration object DMCTRLGROUP is:
app in ('WOTRACK') and groupname in ('RBGROUP1') and optionname in ('RBOPTION1')

Tip: Certain queries are defined as 1=2. You do not need to create a new migration group for other security migration-related scenarios. You only need to create other packages and define separate filters.

If an incorrect SQL Where clause is entered, the Migration Manager application will report the BMXAA7010E error. The SQL Where clause must be corrected before the package definition can be saved.

Be careful when trying to copy and paste the SQL Where clause of the examples. We prefer to type them manually to avoid any validation errors.

Because we created new options for an existing application in this example, we must filter them to avoid deployment problems. If you create a new application or clone an existing application, and you want to migrate all of the options of that application, you do not need to filter them individually. Merely filter them using the application name:

```
app in ('MYNEWAPP1', 'MYNEWAPP2')
```

It filters all of the options for both applications. For additional information, refer to 3.2.8, "Deployment considerations" on page 68.

DMSIGOPTION: For the DMSIGOPTION object structure query, in case you migrate the conditional user interface configuration of a new application, you can filter by the application name and group names:

```
app in ('MYNEWAPP1', 'MYNEWAPP2') and groupname in ('MYGRP1', 'MYGRP2')
```

It filters all of the control security for the groups and applications.

3.2.7 Deployment

After you have saved and approved the package definition, you can proceed with the creation, distribution, and deployment.

3.2.8 Deployment considerations

For this scenario, you need to observe the important points that we describe in this section to successfully have your UI security working in the same way as it works in the Source environment.

Reference: All of the deployment considerations in 3.1.8, “Deployment considerations” on page 59 also apply for this scenario.

New applications

If you create sigoptions for new applications, make sure that the applications exist in the Target environment prior to this migration. It is possible to migrate the sigoptions of a new application in the same package as the application. For additional information, refer to Chapter 5, “Migrating applications and Start Centers” on page 109.

New users

Even though we did not add new users to this scenario, it is supported using, for example, the approach that is described in 3.1, “Migrating a new security group” on page 52.

3.3 Migrating global data restrictions

Data restrictions can be created to a particular set of groups or to the whole system. They are in the object level or the application level. Global data restrictions are restrictions that apply to all of the groups in the system and are in the object level. They are created based on business requirements, for example:

- ▶ Canceled or closed work orders must be filtered out from any listing, independently of the user group or application that is used to list the records.
- ▶ Deleted users must be filtered out from any listing, independently of the user group or application that is used to list the records.

This scenario covers the migration of a new global data restriction and its related data. To perform the tasks that we describe here, you must sign in as an administrative user. For global data restriction in the application level, refer to Chapter 5, “Migrating applications and Start Centers” on page 109

3.3.1 Requirements

The requirements for this migration consist of defining a new global data restriction to filter out closed or canceled work orders from any list. The global data restriction uses a new condition, which is also part of this migration requirement.

Global data restrictions and conditions are defined in a development environment and subsequently promoted to the test and production environments.

3.3.2 Solution

We define an approach so that a new global restriction and its related condition are migrated in two Snapshot migration packages.

The migration of global data restrictions that is described in this scenario implies the migration of the object structure configurations in the first migration package.

The migration of global data restrictions that is described in this scenario implies the migrations of the conditions and security restrictions configurations in the second migration package.

Security restrictions: Security restrictions, when associated with a security group, are migrated along with the group, as detailed in 3.1, “Migrating a new security group” on page 52.

3.3.3 Configuration applications

The following sections show the data that was created as an example for this migration scenario.

Conditional Expression Manager

Conditions are either SQL expressions or custom Java classes that are used, for example, to limit data access or editability. The restriction that the condition defines is only applied when the conditional expression evaluates to true. Conditions are created and managed using the Conditional Expression Manager application, which is accessible by selecting **Go To** → **Administration** → **Conditional Expression Manager**.

For this scenario, we created a condition with the values that are shown in Table 3-12 on page 70.

Table 3-12 RBOOKCOND2

Field	Value
Condition	RBOOKCOND2
Description	IBM Redbooks Condition 2
Type	EXPRESSION
Object Name	WORKORDER
Expression	:status = 'CLOSE' or :status = 'CAN'
Always Evaluate	Selected

Security groups

You can create and manage global data restrictions using the Security Groups application, which is accessible by clicking **Go To** → **Security** → **Security Groups** and, then, by clicking **Select Action** → **Global Data Restriction**.

For this scenario, we created the object data restriction with the values that are shown in Table 3-13.

Table 3-13 Object data restriction

Field	Value
Object	WORKORDER
Type	HIDDEN
Reevaluate	Selected
Condition	RBOOKCOND2

Important: You can use the same condition for separate data restrictions. In this example, when you set the Object field with WORKORDER, you are creating this association.

You can associate the same condition with other objects, if the condition can be applied to the other objects.

3.3.4 Object structures

For this scenario, we need to create a new object structure for global data restrictions. You can create new object structures using the Object Structure application, which is accessible by clicking **Go To** → **System Configuration** → **Migration** → **Object structures**.

The configurations of the global data restrictions are currently in the SECURITYRESTRICT table, which is defined as one of the children in the DMMAXGROUP object structure. In that object structure, there is a relationship between the security group and the data restriction based on the group, which is not a requirement of this scenario.

Because there is no predefined object structure in the product that we can use, to achieve our migration requirements, we have created a new object structure with the values that are shown in Table 3-14. All of the other fields have default values.

Table 3-14 *DMSECURITYRESTRICT* object structure

Field	Value
Object Structure	DMSECURITYRESTRICT
Description	Global Restrictions object structure
Consumed by	MIGRATIONMGR

Under the Source Objects tab, we added the SECURITYRESTRICT object.

Refer to 3.1.4, “Object structures” on page 55 for the DMCONDITION object structure that this scenario needs.

3.3.5 Migration groups

A suitable and efficient approach to migrate the data that we have created is to define two new migration groups.

Table 3-15 on page 72 shows the values of the first migration group. All of the other fields have default values.

Table 3-15 RBGLOBALSE01 migration group

Field	Value
Migration Group	RBGLOBALSE01
Description	IBM Redbooks Global Secur Obj Structure

This group contains only the DMMAXINTOBJECT migration object. Figure 3-10 shows the migration tree of this group.

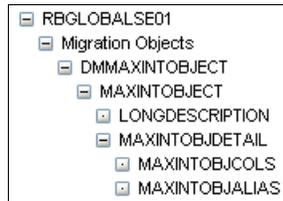


Figure 3-10 RBGLOBALSE01 migration group tree

Table 3-16 shows the values of the second migration group. All of the other fields have default values.

Table 3-16 RBGLOBALSE02 migration group

Field	Value
Migration Group	RBGLOBALSE02
Description	IBM Redbooks Global Secur Data

This group contains the DMCONDITION and DMSECURITYRESTRICT object structures, as shown in Figure 3-11.

Migration Object	Description	Migration Object Order	Internal
DMCONDITION >>	Condition expression migration object for Migra	1	<input type="checkbox"/>
DMSECURITYRE >>	Global Restrictions Object Structure	2	<input type="checkbox"/>

Figure 3-11 RBGLOBALSE02 migration group

Figure 3-12 on page 73 shows the migration tree of this group.

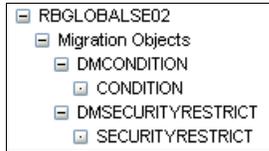


Figure 3-12 RBGLOBALSE02 migration group tree

3.3.6 Package definitions

At this point, we have both the data set and the migration group defined, and the migration process can begin. The first step is to define the package using the Migration Manager application, which is accessible by clicking **Go To** → **System Configuration** → **Migration** → **Migration Manager**.

Because we are migrating a new object structure, two Snapshot packages are needed to migrate the global data restrictions. The first package migrates the new object structure, and the second package migrates the security restrictions.

Alternative: You can choose to not migrate the new object structure and to create it directly in the Target environment. Refer to 3.3.8, “Deployment considerations” on page 75 for additional information.

Table 3-17 shows the values of the first package. All of the other fields have default values.

Table 3-17 RBGLOBALSEC01PKG migration package values

Field	Value
Package Definition Name	RBGLOBALSEC01PKG
Description	IBM Redbooks Global Secur object structure
Type	SNAPSHOT
Processing Action	AddChange

Under the Migration Groups tab, we have added the RBGLOBALSE01 migration group.

Table 3-18 on page 74 shows the values of the second package. All of the other fields have default values.

Table 3-18 RBGLOBALSEC02PKG migration package values

Field	Value
Package Definition Name	RBGLOBALSEC02PKG
Description	IBM Redbooks Global Secur Data
Type	SNAPSHOT
Processing Action	AddChange

Under the Migration Groups tab, we have added the RBGLOBALSE02 migration group.

Defining SQL criteria

Figure 3-13 shows the queries that we used to filter data for the RBGLOBALSEC01PKG package.

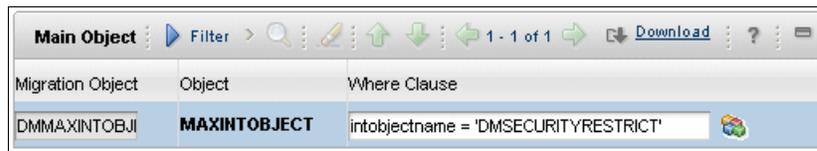


Figure 3-13 Queries for RBGLOBALSEC01PKG package

Figure 3-14 shows the queries that we have used to filter data for the RBGLOBALSEC02PKG package.

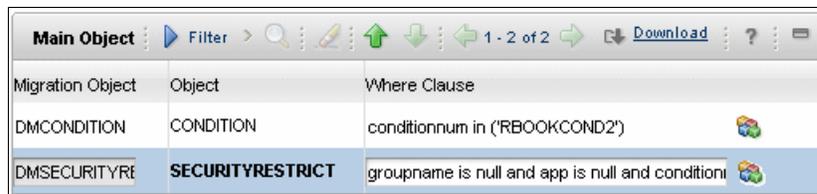


Figure 3-14 RBGLOBALSEC02PKG object structure query

The complete query to filter the DMSECURITYRESTRICT object structure is:
 groupname is null and app is null and conditionnum in ('RBOOKCOND2')

3.3.7 Deployment

After we have saved and approved the two package definitions, we can proceed with the creation, distribution, and deployment. We must migrate the packages separately and in this order:

- ▶ RBGLOBALSEC01PKG
- ▶ RBGLOBALSEC02PKG

3.3.8 Deployment considerations

For this scenario, you need to observe the important points that we describe in this section in order for your global security to work successfully in the Target environment in the same way that it works in the Source environment.

Admin mode

To avoid side effects to the users that are connected to the system, you need to turn on the Admin mode in the Target environment prior to the package deployment.

Migrating the new object structure

You need to migrate the new object structure that we created to the Target environment prior to the global security migration. During the creation of the packages, the system prompts you asking about data dictionary migration. You need to continue with the process because that prompt is a warning message that is displayed every time that a new custom object structure is migrated. It is assumed that no data dictionary changes are pending that you need to migrate.

You must deploy the package containing the new object structure prior to the package containing the global security configuration.

An alternative to the migration of a package with the new object structure is to manually create the object structure in the Target environment and, then, to migrate only the second package.

Migrating the global attribute restrictions

We did not describe the migration of global attribute restrictions. You can migrate them in the same way that we migrated the object restrictions, because they are stored in the same table. The only difference is the query that we defined filters based on the RBOOKCOND2 condition. If there is any attribute-level restriction referencing that condition, it will be migrated, too.

Conditional expression dependencies

Global security restrictions are related with expression conditions. If the conditions are not in the migration package as shown in this scenario, they need to be in the Target environment prior to the migration of the global restrictions package.

3.4 Migrating access definitions and LDAP information

You can perform authentications in the product by using the native authentication of the system or by using the application server security and LDAP.

When you configure your system to authenticate using application server security connected to an LDAP directory server, there is one cron task called Virtual Member Management that updates the system with the groups and users that are changed in the directory server.

After you have loaded the groups and users into the system, you can start to configure the access definitions to them. This scenario is extremely common when you have LDAP-based authentication.

To perform the tasks that we describe here, you must sign in as an administrative user.

3.4.1 Requirements

The requirements of this scenario consist of migrating only the access definitions for the application options.

Access definitions are defined in a development environment and subsequently promoted to the test and production environments.

Important: Security groups and users are not migrated using the Migration Manager application if LDAP synchronization is set up and application server security is enabled.

This process uses external tools and is not part of the scope of this book.

3.4.2 Solution

We define an approach so that a set of new access definitions and related conditions are migrated in two Snapshot migration packages.

The migration of access definitions that is described in this scenario implies the migration of the object structure configurations in the first migration package and the migration of the authorizations configurations in the second migration package.

3.4.3 Configuration applications

The following sections show the data that was created as an example for this migration scenario.

Security groups

For this scenario, we use the group that we created in “Security groups” on page 54, which, in this context, means a group that we loaded from the LDAP server directory.

Important: We assume that this group exists in the Target environment and has no access definitions. For additional information, refer to 3.4.8, “Deployment considerations” on page 81.

Under the Applications tab, the group received access to all of the options of the People application.

3.4.4 Object structures

For this scenario, we need to create a new object structure for the access definitions. You can create new object structures using the Object Structure application, which is accessible by clicking **Go To** → **System Configuration** → **Migration** → **Object structures**.

The data access configurations are currently in the table APPLICATIONAUTH, which is defined as one of the children in the object structure DMMAXGROUP. In that object structure, there is a relationship between the security group and the access definitions based on the group, which is not a requirement of this scenario.

Because there is no predefined object structure in the product that we can use, to achieve our migration requirements, we have created a new object structure with the values that are shown in Table 3-19 on page 78. All of the other fields have default values.

Table 3-19 DMAPPLICATIONAUTH object structure

Field	Value
Object Structure	DMAPPLICATIONAUTH
Description	Authorizations
Consumed by	MIGRATIONMGR

Under the Source Objects tab, we added the APPLICATIONAUTH object.

3.4.5 Migration groups

A suitable and efficient approach to migrate the data we that have created is to define two new migration groups.

Table 3-20 shows the values of the first migration group. All of the other fields have default values.

Table 3-20 RBAUTHORIZATIONS01 migration group

Field	Value
Migration Group	RBAUTHORIZATIONS01
Description	IBM Redbooks Authorizations Obj Structure

This group contains only the object structure DMMAXINTOBJECT.

Figure 3-15 shows the migration tree of this group.

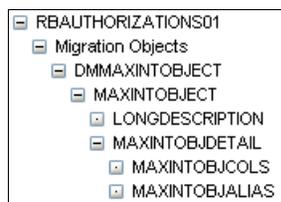


Figure 3-15 RBAUTHORIZATIONS01 migration group tree

Table 3-21 on page 79 shows the values of the second migration group. All other fields have default values.

Table 3-21 RBAUTHORIZATIONS02 migration group

Field	Value
Migration Group	RBAUTHORIZATIONS02
Description	IBM Redbooks Access Definitions Data

This group contains the object structure DMAPPLICATIONAUTH. Figure 3-16 shows the migration tree of this group.



Figure 3-16 RBAUTHORIZATIONS02 migration group tree

3.4.6 Package definitions

At this point, we have both the data set and migration group defined, and the migration process can begin. The first step is to define the package using the Migration Manager application, which is accessible by clicking **Go To** → **System Configuration** → **Migration** → **Migration Manager**.

Two Snapshot packages are required to migrate the data access. The first package migrates the new object structure that we have created, and the second package migrates access definitions and conditions.

Alternative: You can choose to not migrate the new object structure and instead to create it directly in the Target environment. For additional information, refer to 3.4.8, “Deployment considerations” on page 81.

Table 3-22 shows the values of the first package. All other fields have default values.

Table 3-22 RBAUTH01PKG migration package values

Field	Value
Package Definition Name	RBAUTH01PKG
Description	IBM Redbooks Auth object structure
Type	SNAPSHOT
Processing Action	AddChange

Under the Migration Groups tab, we have added the migration group RBAUTHORIZATIONS01.

Table 3-23 shows the values of the second package. All other fields have default values.

Table 3-23 RBAUTH02PKG migration package values

Field	Value
Package Definition Name	RBAUTH02PKG
Description	IBM Redbooks Auth data
Type	SNAPSHOT
Processing Action	AddChange

Under the Migration Groups tab, we have added the migration group RBAUTHORIZATIONS02.

Defining SQL criteria

Figure 3-17 shows the queries that we used to filter the data for the package RBAUTH01PKG.

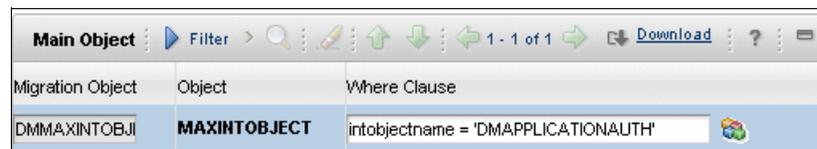


Figure 3-17 Queries for package RBAUTH01PKG

Figure 3-18 shows the queries that we used to filter data for package RBAUTH02PKG.



Figure 3-18 Queries for package RBAUTH02PKG

The query that is used to filter the DMAPPLICATIONAUTH migration object is:

app = 'PERSON' and groupname in ('RBGROUP')

3.4.7 Deployment

After you have saved and approved the two package definitions, you can proceed with the creation, distribution, and deployment. You must migrate the packages separately and in this order:

- ▶ RBAUTH01PKG
- ▶ RBAUTH02PKG

3.4.8 Deployment considerations

For this scenario, you need to observe the important points that we describe in this section to successfully have your global security working in the same way as it works in the Source environment.

Admin mode

To avoid side effects to users that are connected to the system, you need to turn on the Admin mode in the Target environment prior to the package deployment.

Migrating the new object structure

You need to migrate the new object structure that we have created to the Target environment prior to the access definitions. During the creation of the packages, the system prompts you asking about the data dictionary migration. Continue with the process, because this warning message is displayed every time that a new custom object structure is migrated. It is assumed that there are no data dictionary pending changes that you need to migrate.

The deployment of the package containing the new object structure must be done prior to the deployment of the package containing the data access configuration.

An alternative to migrating the package with the new object structure is to create the object structure manually in the Target environment.

Metadata assumptions

We assume that your product configuration has not modified the primary key of the APPLICATIONAUTH table or extended its object.

Application and security group dependencies

The application definitions, security groups, and application options for which you have defined access must exist in the Target environment.

Conditional expression dependencies

Access definitions are related with expression conditions. If the conditions are not in the migration package, they need to be in the Target environment prior to the migration of the access definitions package.

Considerations about migrating access definitions

This scenario covered the first migration of access definitions, which means that the data that you are migrating does not exist in the Target environment.

If you need to make changes in the Source environment and perform a new migration, you can use almost the same approach that we presented in this section, but consider a Replace processing action, or use a Change package. For additional information, refer to Chapter 10, “Common topics” on page 197.

3.5 Considerations of security migrations

This chapter presented several scenarios for security migration. These scenarios are presented individually and have no dependency on each other, but you can combine or extend them to achieve new migration scenarios.



Migrating escalations

Escalations are server-side functions that automate actions and notifications in any Tivoli process automation engine-based product. Escalations are used extensively by clients and practitioners to automate many aspects of the business that are supported by Tivoli's process automation engine-based products. For example, an escalation proactively monitors service desk compliance with the commitments of a service-level agreement (SLA). If the response time for a high priority service request (SR) is approaching the defined SLA threshold that is defined in a commitment, the escalation fires actions and notifications. The actions might increase the severity of the SR. The notifications might alert a service desk manager of approaching noncompliance.

This chapter describes the best practices in migrating escalations and has the following sections:

- ▶ 4.1, "Requirements" on page 84
- ▶ 4.2, "Solution" on page 84
- ▶ 4.3, "Configuration applications" on page 84
- ▶ 4.4, "Object structures" on page 87
- ▶ 4.5, "Migration groups" on page 89
- ▶ 4.6, "Package definitions" on page 92
- ▶ 4.7, "Deployment" on page 98
- ▶ 4.7.1, "Escalations and cron tasks" on page 99

4.1 Requirements

Escalations are defined in a development environment and subsequently promoted to test and production environments. A repeatable process is required to reliably (without any errors or failures) and accurately (without missing any required content) migrate escalation configurations and any related configurations.

4.2 Solution

We define an approach so that an escalation configuration and all related configurations are migrated in a single Snapshot migration package.

From a content perspective, the migration of an escalation primarily implies the migration of the following configurations in order:

- ▶ Actions
- ▶ Action groups
- ▶ Person
- ▶ Person group
- ▶ Role
- ▶ Communication template
- ▶ Escalations

4.3 Configuration applications

This section lists the applications supporting an escalation configuration. The section also describes dependencies among these applications that drive the migration of escalations.

4.3.1 Actions

You can access actions and action groups from the appropriate product Start Center by clicking **Go To** → **System Configuration** → **Platform Configuration** → **Actions**.

4.3.2 People

Whether or not a role or a communication template is associated with a person record, creation and management of the person record is done using the People application. You can access person records from the appropriate product Start Center by clicking **Go To → Administration → Resources → People**.

4.3.3 Person groups

The creation and management of the person group record is done using the Person Groups application. You can access person groups from the appropriate product Start Center by clicking **Go To → Administration → Resources → Person Groups**.

The configurations and governing applications identified so far constitute the primary dependencies for an escalation.

4.3.4 Roles

When a communication template is associated with a role, the role is defined using the Roles application. You can access roles from the appropriate product Start Center by clicking **Go To → System Configuration → Platform Configuration → Roles**

A role can be built from a person or person group record.

4.3.5 Communication templates

Notifications are managed using the Communication Templates application. You can access communication templates from the appropriate product Start Center by clicking **Go To → System Configuration → Platform Configuration → Communication Templates**.

A communication template, in turn, can define one or more recipients in the form of email address, role, person, or person group records.

4.3.6 Escalations

Escalations are created and managed using the Escalations application. You can access escalations from the appropriate product Start Center by clicking **Go To → System Configuration → Platform Configuration → Escalations**.

An escalation executes one or more actions and notifications. Therefore, every escalation is associated with a corresponding set of actions or notifications.

4.3.7 Other applications

In addition to these configurations, an escalation also depends on an additional set of configurations. These configurations are secondary in the sense that they might have been delivered with the product or might have been previously migrated:

- ▶ Cron tasks

An escalation always requires a cron task instance to execute in a periodic manner. Cron tasks are defined with the Cron Task Setup application. All escalation cron task instances are derived from the ESCALATION cron task that is delivered with the product. When an escalation is activated, a cron task instance is created if a cron task instance does not exist. The dynamic creation of a cron task instance implies that there are alternative approaches to migrating the cron task instance associated with an escalation.

- ▶ Organization and site

An escalation might optionally specify an association to an organization or site. Organizations and sites are defined with the Organizations application. For this migration scenario, it is assumed that the organizations and sites have been already migrated.

- ▶ Business object

An escalation always specifies an object that is the Target of the escalation actions and notifications. In fact, roles, communication templates, and escalations all depend on the existence of a business object. A business object is defined with the Database Configuration application. The majority of roles, communication templates, and escalations are defined for business objects that are delivered with the product. For this migration scenario, it is assumed that the business object definitions have been already migrated.

We have now established that an escalation is built not in isolation but in conjunction with configurations that are managed through other applications. Escalation migration must take into account all of these related configurations. Figure 4-1 on page 87 captures these dependencies.

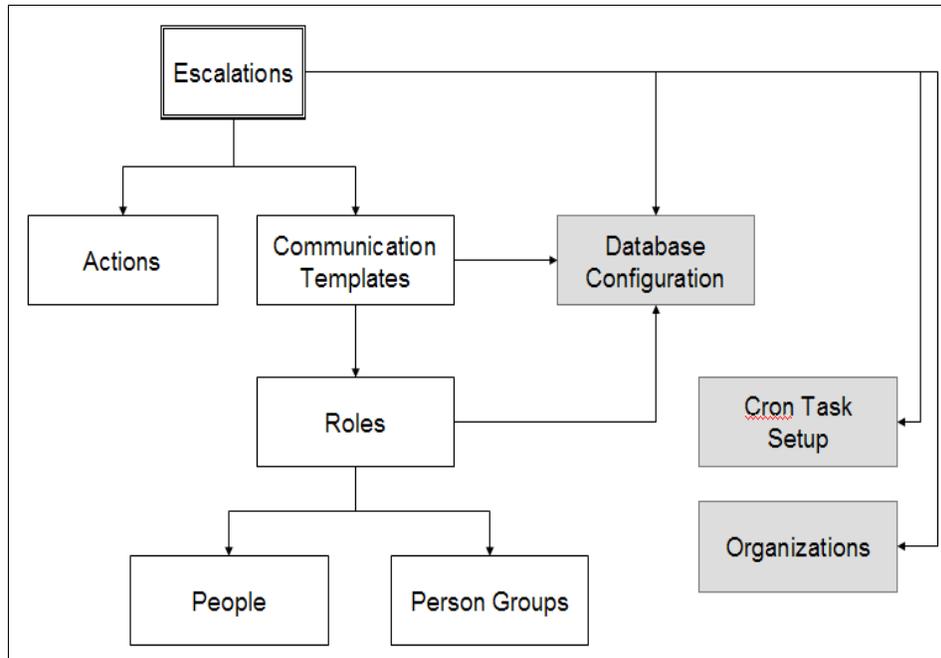


Figure 4-1 Escalations and related applications

4.4 Object structures

The following object structures are delivered with the product.

The DMACTION object structure, as part of the business process management (BPM) migration group, supports action group migration. Figure 4-2 shows the contents of the DMACTION object structure.

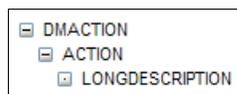


Figure 4-2 DMACTION object structure

The DMACTIONGROUP object structure, as part of the BPM migration group, supports action migration. Figure 4-3 on page 88 shows the contents of the DMACTIONGROUP object structure.

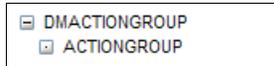


Figure 4-3 *DMACTIONGROUP* object structure

The DMPERSON object structure, as part of the RESOURCES migration group, supports person configuration migration. Figure 4-4 shows the contents of the DMPERSON object structure.

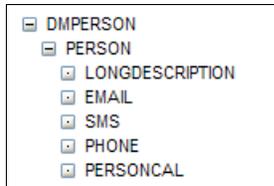


Figure 4-4 *DMPERSON* object structure

The DMPERSONGROUP object structure, as part of the RESOURCES migration group, supports person group migration. Figure 4-5 shows the contents of the DMPERSONGROUP object structure.

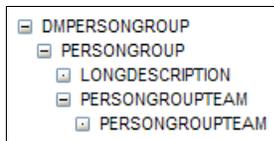


Figure 4-5 *DMPERSONGROUP* object structure

The DMROLE object structure, as part of the BPM migration group, supports role migration. Figure 4-6 shows the contents of the DMROLE object structure.

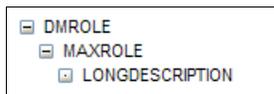


Figure 4-6 *DMROLE* object structure

The DMCOMMTEMPLATE object structure, as part of the BPM migration group, supports the communication template migration. Figure 4-7 on page 89 shows the contents of the DMCOMMTEMPLATE object structure.



Figure 4-7 DMCOMMTEMPLATE object structure

The DMESCALATION object structure, as part of the BPM migration group, supports the escalation migration. Figure 4-8 shows the contents of the DMESCALATION object structure.

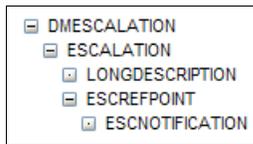


Figure 4-8 DMESCALATION object structure

4.5 Migration groups

The object structures that support escalation migration belong to two migration groups: BPM and RESOURCES.

Figure 4-9 shows the BPM migration group and its dependencies as delivered with the product.

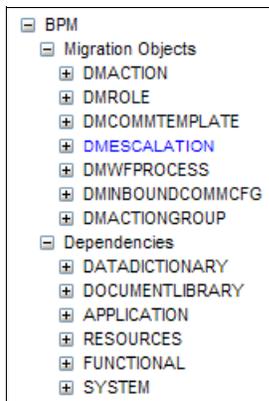


Figure 4-9 BPM migration group

Figure 4-10 shows the RESOURCES migration group and its dependencies as delivered with the product.

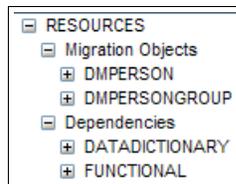


Figure 4-10 RESOURCES migration group

Using the BPM and RESOURCES migration groups is inefficient for the following reasons:

- ▶ Only a subset of the configurations is required to be migrated.
- ▶ Configurations pertaining to the dependencies listed with the two migration groups might have been already migrated.

Consequently, Figure 4-11 shows the object structures and migration groups pertaining to BPM that are not required for the escalation migration scenario.

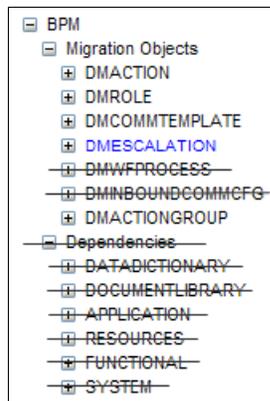


Figure 4-11 Discarded BPM object structures and dependencies

Note: The strike-through lines in Figure 4-11 on page 90 are only for explanatory purposes. The product does not provide a capability to remove object structures and migration groups in a strike-through manner.

Figure 4-12 on page 91 shows the object structures and migration groups pertaining to RESOURCES that are not required for this migration scenario.

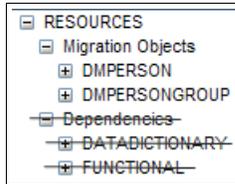


Figure 4-12 Discarded RESOURCES dependencies

For the escalation migration scenario, create a new migration group containing only the required object structures. The new migration group can be easily constructed by duplicating the existing BPM migration group and modifying its contents. Figure 4-13 shows the new MYESCALATION migration group and its contents.

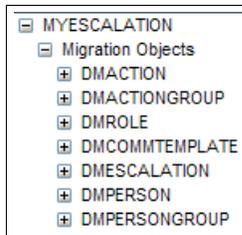


Figure 4-13 Object structures and migration group supporting escalation migration

Figure 4-14 shows the ordering of the object structures within this new migration group.

Migration Object	Description	Migration Object Order
DMACTION	Action migration object for Migration Manager	1
DMACTIONGR	Actiongroup migration object for Migration Man	2
DMPERSON	Person migration object for Migration Manager	3
DMPERSONGR	Person group migration object for Migration Ma	4
DMROLE	Role migration object for Migration Manager	5
DMCOMMTEMP	Communication template migration object for Mi	6
DMESCALATIO	Escalation migration object for Migration Manag	7

Figure 4-14 Object structure order

Note: Figure 4-14 on page 91 is used to depict the object structures and their ordering in the new migration group for explanatory purposes only. The product user interface might differ.

4.6 Package definitions

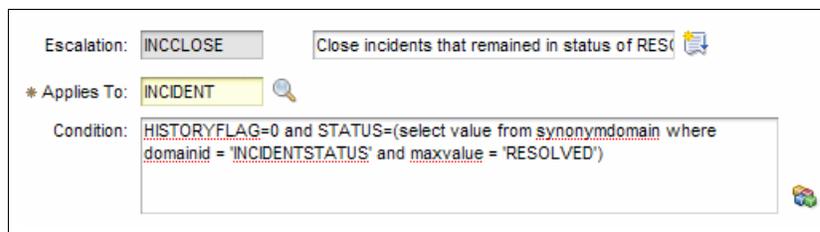
Having identified and organized the content, you can begin the migration. As a first step, a Snapshot migration package is defined. The definition includes the newly created migration group. SQL criteria are associated with the appropriate object structures that are members of the new migration group.

Defaults: Accept all of the default values that the Migration Manager application provides for the package definition.

The following section uses sample escalations that are delivered with the product to illustrate setting up SQL criteria.

4.6.1 Defining SQL criteria for an escalation with an action

The sample escalation INCCLOSE is used for this discussion. This escalation is intended to close Incident records in the product that have remained in the resolved state for more than ten days. Figure 4-15 shows the escalation header for the INCCLOSE escalation.



The screenshot shows a configuration window for an escalation. It contains the following fields:

- Escalation:** INCCLOSE
- Description:** Close incidents that remained in status of RES(
- * Applies To:** INCIDENT
- Condition:** HISTORYFLAG=0 and STATUS=(select value from synonymdomain where domainid = 'INCIDENTSTATUS' and maxvalue = 'RESOLVED')

Figure 4-15 INCCLOSE sample escalation

The INCCLOSE escalation has a single escalation point that runs an action, but no associated notifications. Figure 4-16 on page 93 shows the INCIDENT CLOSED action that is associated with the escalation.

Action	Description	Type
INCIDENT CLOS >>	TIDE Incident Closed Action	CHANGESTATUS

Figure 4-16 INCIDENT CLOSE action

This action is part of an action group. An *action group* enables multiple actions to be executed in sequence when an escalation runs.

Escalation: An escalation is always associated with an action group that, in turn, contains one or more individual actions.

Figure 4-17 shows the INCCLOSEGRP action group of which the INCIDENT CLOSED action is a member.

Select Action Group or add individual actions by clicking New Row. Action Group: INCCLOSEGRP >>

Action	Description
INCIDENT CLOS >>	TIDE Incident Closed Action 3

Figure 4-17 INCCLOSEGRP action group and INCIDENT CLOSED member action

The migration of the INCCLOSE escalation includes the migration of the associated action INCIDENT CLOSED and action group INCCLOSEGRP. Thus, SQL criteria can be set up for the object structures that are part of the migration package. Figure 4-18 on page 94 shows the specific SQL criteria setup for the escalation, action, and action group object structures.

Migration Object	Object	Where Clause
DMACTION	ACTION	action in (select action from ACTIONGROUP wh
DMACTIONGRC	ACTIONGROUP	action in (select action from ESCREFPOINT wh
DMPERSON	PERSON	1=0
DMPERSONGRI	PERSONGROUP	1=0
DMROLE	MAXROLE	1=0
DMCOMMTEMPI	COMMTEMPLATE	1=0
DMESCALATIO	ESCALATION	escalation in ('INCCLOSE')

Figure 4-18 SQL criteria to migrate INCCLOSE escalation

The complete SQL Where clause for the DMACTION object structure is:

action in (select action from ACTIONGROUP where action in (select action from ESCREFPOINT where escalation in ('INCCLOSE'))) or action in (select member from actiongroup where action in (select action from escrefpoint where escalation in ('INCCLOSE')))

Note: The SQL Where clause for the DMACTION object structure is designed to collect both the action groups and individual actions that are required for the escalation. The entries for action groups and individual actions both reside in the same ACTION table in the underlying production database.

The complete SQL Where clause for the DMACTIONGROUP object structure is:

action in (select action from ESCREFPOINT where escalation in ('INCCLOSE'))

The complete SQL Where clause for the DMESCALATION object structure is:

escalation in ('INCCLOSE')

Because no other related configuration is required to be collected for the migration of the INCCLOSE escalation, the other object structures in the migration package specify the SQL Where clause to be:

1=0

This SQL Where clause results in no configuration records being picked up for the DMPERSON, DMPERSONGROUP, DMROLE, and DMCOMMTEMPLATE object structures.

Note: Leaving the Where clause field empty implies all the configuration records of the primary business object of the object structure must be collected for migration. This result might not be the desired result. If no configuration record is to be extracted for an object structure, specify the SQL Where clause 1=0.

The SQL Where clauses are constructed in a manner so that only the name of the specific escalation needs to be specified. If a separate escalation will be migrated, the same migration package definition can be reused by replacing the INCCLOSE escalation identifier with the identifier of the desired escalation.

Tip: To the extent that the various object structures in a migration group are related to each other, specify the SQL clauses for these object structures in a manner so that they reference the primary object structure. Knowledge of the data models for the various applications is necessary to develop the SQL clauses.

Save the package definition. The remainder of the migration steps can now be performed.

4.6.2 Defining SQL criteria for an escalation with a notification

The sample escalation SLAREVIEW is used for this discussion. This escalation is intended to proactively send SLA review notifications to the designated individual in charge of reviewing an SLA prior to its expiration. Figure 4-19 shows the escalation header for the SLAREVIEW escalation.

Escalation: SLAREVIEW Notify SLA Administrator 30, 60, and 90 days

* Applies To: SLA

Condition: STATUS=(select value from synonymdomain where domainid = 'SLA STATUS' and maxvalue = 'ACTIVE') and reviewdate is not NULL

Figure 4-19 SLAREVIEW sample escalation

The SLAREVIEW escalation has three escalation points, each of which sends a notification, but no associated actions. All three escalation points send the same notification; however, each notification is sent out at a separate point in time. Figure 4-20 on page 96 shows the SLAREVIEW communication template that is associated with the escalation.

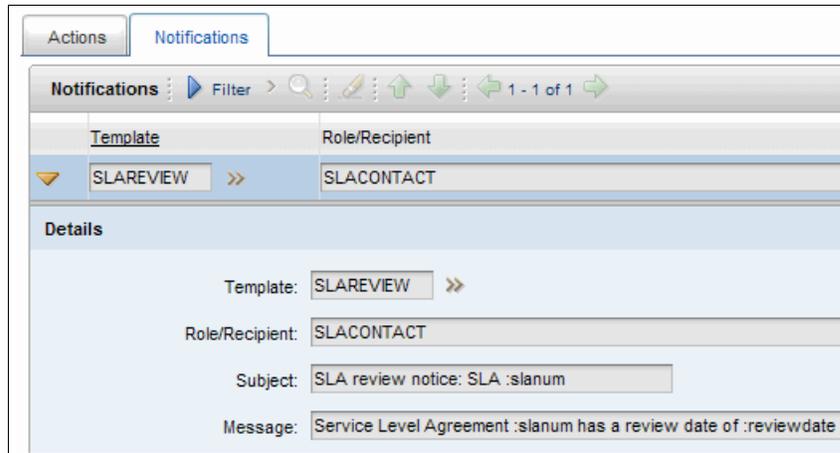


Figure 4-20 SLAREVIEW communication template

Use the Detail menu icon to the right of the Template field to open the SLAREVIEW communication template with the Communication Templates application. The Communication Template tab displays the content of the notification, such as the Subject and Message information. The Recipients tab lists a variety of recipient types. Opening the Roles section lists the roles to which this notification will be sent. Figure 4-21 shows the Roles section.



Figure 4-21 SLACONTACT role for SLAREVIEW communication template

Use the Detail menu icon to the right of the Role field to open the SLACONTACT role with the Roles application. The Role tab displays the content of the role. This role does not depend on a person or person group configuration. Figure 4-22 on page 97 shows the SLACONTACT role.

The screenshot shows a configuration form for the SLACONTACT role. The fields are as follows:

- Role:** SLACONTACT
- Search:** Contact person named on SLA
- * Type:** A set of data relat
- Object:** SLA
- * Value:** :SLACONTACT
- Parameter:** (empty)
- E-mail?:**
- Broadcast?:**

Figure 4-22 SLACONTACT role details

The migration of the SLAREVIEW escalation includes the migration of the associated communication template SLAREVIEW and the role SLACONTACT.

The SLAREVIEW escalation will be migrated with the same package definition that was defined earlier. There is no need to create a separate package definition. To reuse the existing package definition, simply alter the SQL criteria that are associated with the various object structures in the package definition.

Alternatives: Before migrating the SLAREVIEW escalation, a physical package must be created before altering the SQL criteria to ensure that the configurations pertaining to the INCCLOSE escalation are extracted as desired. If a physical package has not been created yet and the SQL criteria are altered, it will no longer be possible to extract the configurations pertaining to the INCCLOSE escalation. Alternatively, SQL criteria can be combined to collect the configurations pertaining to both INCCLOSE and SLAREVIEW escalations. The choice is driven by implementation requirements.

Thus, SQL criteria can be set up for the object structures that are part of the migration package. Figure 4-23 on page 98 shows the specific SQL criteria setup for the escalation, action, and action group object structures.

Migration Object	Object	Where Clause
DMACTION	ACTION	1=0
DMACTIONGRC	ACTIONGROUP	1=0
DMPERSON	PERSON	1=0
DMPERSONGR	PERSONGROUP	1=0
DMROLE	MAXROLE	maxrole in (select sendtovalue from commtmpl
DMCOMMTEMPI	COMMTEMPLATE	templateid in (select distinct templateid from es
DMESCALATIO	ESCALATION	escalation in ('SLAREVIEW')

Figure 4-23 SQL criteria to migrate SLAREVIEW escalation

The complete SQL Where clause for the DMROLE object structure is:

```
maxrole in (select sendtovalue from commtmplsendto where templateid in
(select distinct templateid from escnotification where escalation in
('SLAREVIEW')))
```

The complete SQL Where clause for the DMCOMMTEMPLATE object structure is:

```
templateid in (select distinct templateid from escnotification where
escalation in ('SLAREVIEW'))
```

The complete SQL Where clause for the DMESCALATION object structure is:

```
escalation in ('SLAREVIEW')
```

Tip: If an incorrect SQL Where clause is entered, the Migration Manager will report the BMXAA7010E error. You must correct the SQL Where clause before the package definition can be saved.

Save the package definition. The remainder of the migration steps can now be performed.

4.7 Deployment

Create, distribute, and deploy the package. The escalation configurations and their related configurations can be deployed into a live production environment. We discuss the deployment considerations in this section.

4.7.1 Escalations and cron tasks

Every escalation executes periodically exploiting an underlying cron task instance. All escalation cron task instances are built from the ESCALATION cron task that ships with the product.

When an escalation is activated for the first time, the corresponding cron task instance is also created and linked to the ESCALATION cron task. The cron task instance name includes the name of the corresponding escalation and is preceded by a three letter prefix of 'ESC'. A reference to the cron task instance name is retained in the INSTANCENAME column of the ESCALATION table. For example, the SLAREVIEW escalation will carry a reference to the ESCSLAREVIEW cron task instance. After an escalation is deactivated, it continues to retain the reference to the previously created cron task instance.

There are two approaches to resolving the dependency of an escalation on a cron task instance:

- ▶ Use the Escalation application to create the cron task instance.
- ▶ Migrate the cron task and its instances to support an escalation.

Use the Escalation application to create a cron task instance

This approach exploits the Escalation application's ability to create the required cron task that supports periodic escalation execution. This approach takes effect when the escalation is migrated for the first time, resulting in the insertion of various configuration records in the Target production environment. Follow these steps:

1. Exclude the INSTANCENAME attribute from the DMESCALATION object structure in both the Source and Target production environments:
 - a. Open the **Object Structures** application.
 - b. Find the DMESCALATION object structure in the List tab.
 - c. Open the DMESCALATION object structure in the Object Structure tab.
 - d. From the Select Action menu, execute the **Exclude/Include Fields** action.
 - e. In the Exclude/Include Fields pop-up dialog, ensure that the ESCALATION object is selected in the Source Objects for ESCALATION table window.
 - f. In the Persistent fields subtab, scroll until you locate the INSTANCE Field.
 - g. Select **Exclude**.
 - h. Click **OK** to save the changes and dismiss the Exclude/Include Fields pop-up dialog.
2. Deactivate the desired escalations in the Source production environment.

3. Define a migration package and associate the appropriate SQL criteria to the object structures in the migration group.
4. Approve, activate, and create the migration package.
5. Distribute the migration package to the Target production environment.
6. Deploy the migration package.
7. After the deployment is complete, open the migrated escalations in the Escalations application and activate each of them.

Upon activation, the Escalation application determines a cron task instance is required and creates the instance and links it back into the escalation. The escalation is now in effect in the Target production environment.

Updating an escalation: This approach is also valid for a migration scenario where the escalation is being updated rather than inserted. For an update migration scenario, the cron task instance is already associated with the escalation and this association is left intact during migration.

Migrating escalations with escalation cron task instances

This approach collects all of the cron task instances that are associated with the ESCALATION cron task and migrates them along with the escalations themselves. This approach requires you to perform additional steps that result in a longer migration effort. Follow these steps:

1. In the Source production environment, add the DMCRONTASKDEF object structure to the newly created migration group supporting escalation migration. Reorder the object structures in the migration group so that DMCRONTASKDEF becomes the first object structure. Figure 4-24 on page 101 shows the updated migration group.

Migration Object	Description	Migration Object Order
DMCrontaskI	Cron task migration object for Migration Manager	1
DMACTION	Action migration object for Migration Manager	2
DMACTIONGRC	Actiongroup migration object for Migration Manager	3
DMPERSON	Person migration object for Migration Manager	4
DMPERSONGR	Person group migration object for Migration Manager	5
DMROLE	Role migration object for Migration Manager	6
DMCOMMTEMPI	Communication template migration object for Migration Manager	7
DMESCALATIO	Escalation migration object for Migration Manager	8

Figure 4-24 Migration group containing DMCrontaskDef object structure

- Reuse the existing migration package definition, which now reflects the addition of the DMCrontaskDef object structure. Figure 4-25 shows the SQL Where clause that is associated with the DMCrontaskDef object structure.

Migration Object	Object	Where Clause
DMCrontaskI	CRONTASKDEF	crontaskname in ('ESCALATION')
DMACTION	ACTION	1=0
DMACTIONGRC	ACTIONGROUP	1=0
DMPERSON	PERSON	1=0
DMPERSONGR	PERSONGROUP	1=0
DMROLE	MAXROLE	maxrole in (select sendtovalue from commtmpl)
DMCOMMTEMPI	COMMTEMPLATE	templateid in (select distinct templateid from es)
DMESCALATIO	ESCALATION	escalation in ('SLAREVIEW')

Figure 4-25 SQL criteria to migrate ESCALATION cron task

- Perform the standard migration steps to create, distribute, and deploy the package.

Upon deployment, all cron task instances that are associated with the ESCALATION cron task are inserted or updated into the Target production database. A significant drawback of this approach is the inability to only take the specific cron task instance that is associated with the escalation being migrated. The Migration Manager provides the ability to specify SQL criteria only against the primary object of an object structure and not on subordinate objects.

Important: If you do not intend to migrate any cron tasks, ensure that you do not collect any cron task configurations by specifying the SQL Where clause 1=0 for the DMCRONTASKDEF object structure.

Best practice: If a business application provides the capability to create or associate configurations automatically, you need to exploit this capability in support of migration. This capability can save time and resources.

Snapshot processing actions and effect on escalations

A Snapshot migration package definition is always associated with a processing action. The processing action determines how the content extracted from the Source production environment is processed during deployment into the Target production environment. The default processing action associated with a Snapshot migration package definition is Replace. In the Source production environment, the processing action can be changed to AddChange prior to approving the package definition or during package creation. Each processing action potentially produces separate results during migration.

Figure 4-26 shows an escalation 1001 that was created in a Source production environment.

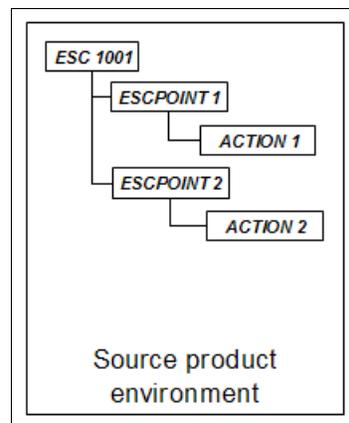


Figure 4-26 Escalation 1001 initially created in a Source environment

Escalation 1001 is migrated to the Target production environment. Figure 4-27 shows the result of the migration.

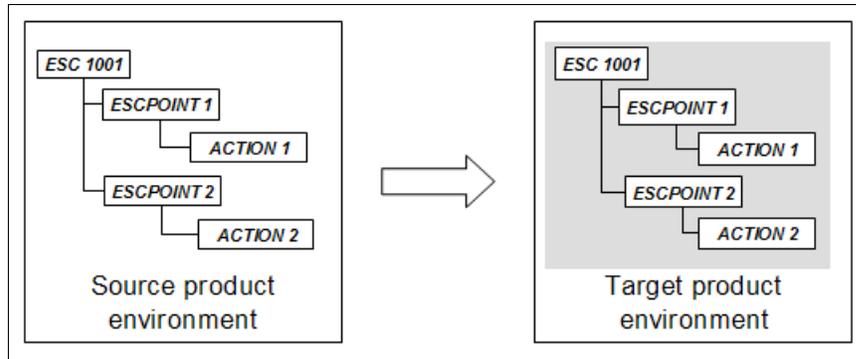


Figure 4-27 Escalation 1001 migrated to the Target environment

In the Target production environment, a new escalation point is added to escalation 1001. Figure 4-28 shows the updated escalation.

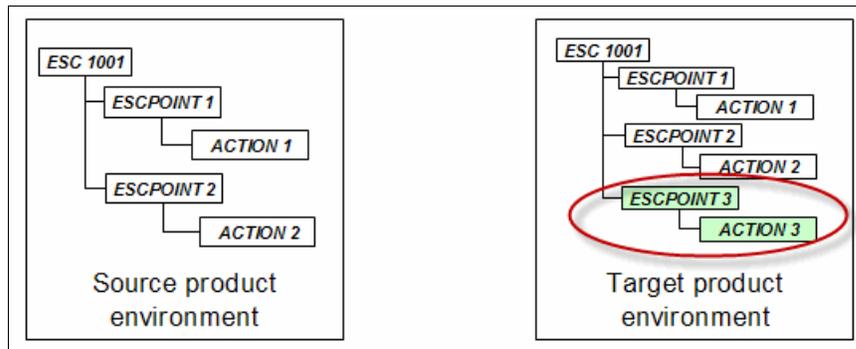


Figure 4-28 Escalation 1001 modified directly in the Target environment

In the Source production environment, a new escalation point is added to escalation 1001. Figure 4-29 on page 104 shows the updated escalation.

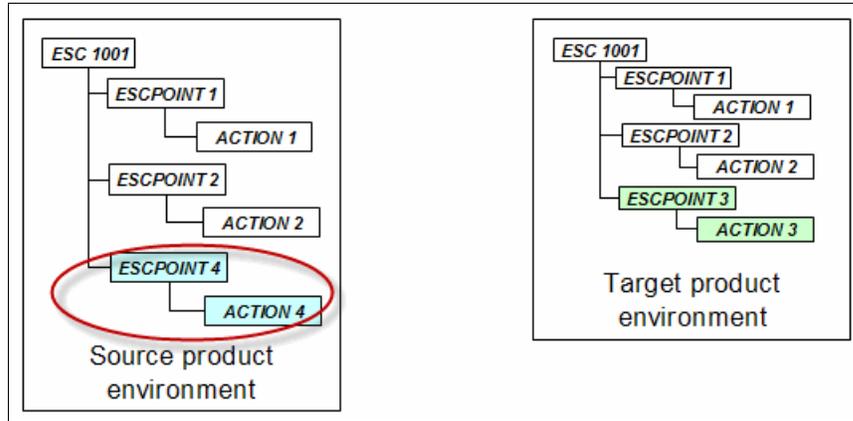


Figure 4-29 Escalation 1001 modified in the Source environment

If escalation 1001 is now migrated from Source to Target with a processing action Replace, escalation point ESCPOINT3 in the Target will be deleted. The purpose of the Replace processing action is to completely replace the configuration in the Target production environment with the content from the migration package. Figure 4-30 shows the effect of the Replace processing action.

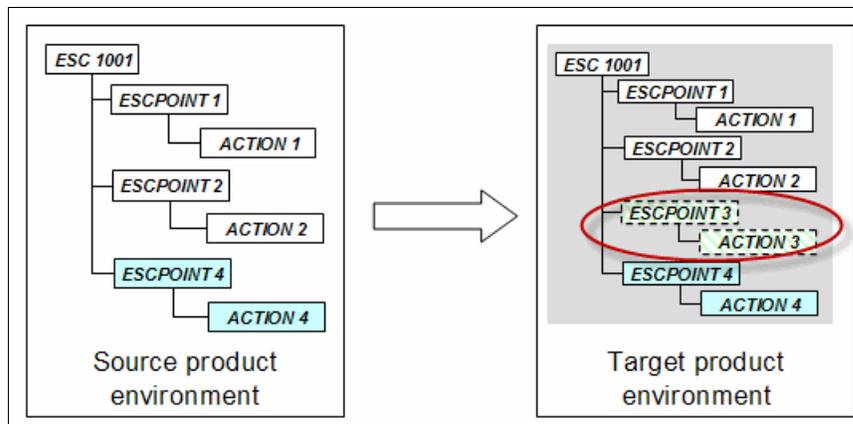


Figure 4-30 Modified escalation 1001 migrated with the Replace processing action

This effect is not the desired result. Changes made directly in a Target production environment, such as a production environment, need to be preserved. If escalation 1001 is migrated from the Source environment to the Target environment with processing action AddChange, escalation point ESCPOINT3 is preserved. The purpose of the AddChange processing action is to leave intact the elements of the configuration in the Target production environment. Figure 4-31 on page 105 shows the effect of the AddChange processing action.

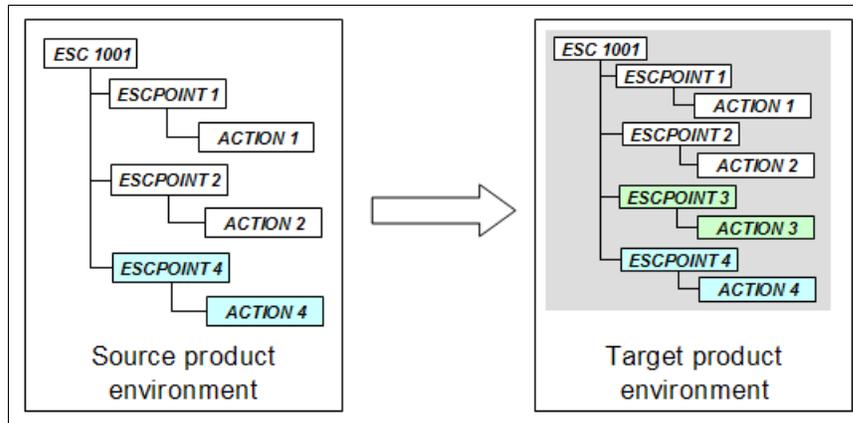


Figure 4-31 Modified escalation 1001 migrated with the AddChange processing action

Best practice: If you deploy into a live production environment, specify the processing action AddChange for Snapshot migration packages.

Collecting attached documents

Certain configurations might be associated with attached documents. For example, a specific communication template might always include an operations manual document as an attachment. When a notification is sent using this communication template, the document is delivered to the recipients along with the mail notification. During migration, the ability to include the attached document along with the configuration content is invaluable.

With the release of Maximo Base Services Fix Pack 7.1.1.6, the DMCOMMTEMPLATE object structure includes the DOCLINKS object that maintains the association between a communication template and its attached documents. This association ensures that the attached document files are automatically collected and included in the migration package.

All attached document files are associated with folder and file name information. This information is stored in the DOCINFO table. When migrating the communication template and its attached documents, the physical file is copied into the folder structure that is specified in the DOCINFO table. Ensure that the folder structure on the file system is the same for both Source and Target environments. If the folder structures differ, the file is saved but it cannot be viewed as an attached document from the Communication Templates application.

For more information about object structure support for attached documents, review this document:

<http://www-01.ibm.com/support/docview.wss?uid=swg24027858>

Best practice: During migration, exploit the ability to automatically collect an attached document for an object. Ensure that the attached document folder structure on the file system is the same in the Source and Target production environments. This approach is more efficient than organizing and collecting compiled sources for a migration package.

Escalations and workflow processes

Escalations can be designed to launch workflow processes through actions. When migrating escalations and actions that launch workflow processes, ensure that the workflow process is migrated before the escalation and action. Use two migration packages. The first package migrates the desired workflow process, and the second package migrates the escalation and action.

Escalations and Java classes

Actions and roles that are associated with an escalation can execute custom Java classes. Typically, Java classes are managed in a designated folder structure on a file system in a product development or build environment. A source control system is used to manage multiple versions of the Java class and to control access to the Java source code.

Java classes can be collected in a migration package in the form of compiled sources. When the package is deployed, the Migration Manager prompts the user to download the Java classes in the migration package to a folder from which the Java deployment Enterprise Archive (EAR) can be built. The Migration Manager does not support any build process or deployment of an EAR file into an application server.

An alternative approach is to directly obtain the desired Java classes from the source control system and execute the build steps.

Pitfalls of migrating an active escalation

The migration of an active escalation without using either of the approaches outlined earlier will result in an error-free deployment; however, the escalation cannot be successfully activated. The Escalations application will report error BMXAA4214E. Figure 4-32 on page 107 shows the error message.

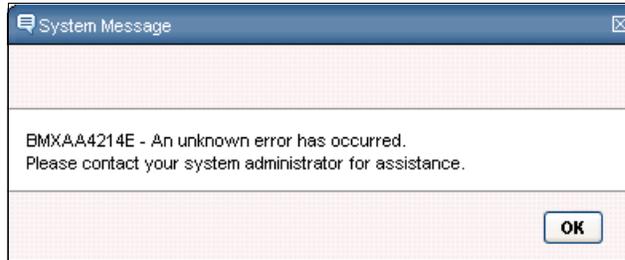


Figure 4-32 Error encountered when activating an escalation following the migration

The product log file or system console presents a Java stack trace:

```
java.lang.NullPointerException
    at
psdi.app.escalation.Escalation.activateCronTaskInstance(Escalation.java:140)
    at
psdi.app.escalation.Escalation.actDeactEscalation(Escalation.java:104)
    at
psdi.webclient.beans.escalation.EscalationAppBean.ADESC(EscalationAppBean.java:48)
```

This error occurs because the escalation configuration includes a reference to the cron task instance but the cron task itself has not been migrated. When the escalation is activated, the Escalation application attempts to activate this non-existent cron task instance.

Following the best practice in migrating escalations avoids this pitfall.



Migrating applications and Start Centers

The product installs a rich set of standard application user interfaces (UI) that enables users to perform most of the tasks that are needed to accomplish their jobs. However, sometimes modifications are necessary to simplify the UI experience or to comply to business process rules.

This chapter provides use cases for determining the migration content related to application user interfaces (UIs), its related menu items when applicable, and the tasks involved in achieving the goal of promoting it to another environment.

Each case makes an effort to cover the most common scenarios, and, in certain situations, the scope of the modifications involves the use of other tools to migrate in addition to the Migration Manager.

This chapter has the following sections:

- ▶ 5.1, “Basic application changes and signature options” on page 110
- ▶ 5.2, “Complex application changes, including menus, lookups, and global data restrictions” on page 117
- ▶ 5.3, “Migrating a Start Center and a query” on page 131

5.1 Basic application changes and signature options

When an application is launched, a rendered screen presentation of the application UI is displayed for user interaction. In this use case scenario, we describe the migration of screen presentation modifications in the application UI, using *only* the Application Designer tool, which is accessible by clicking **Go To** → **System Configuration** → **Platform Configuration** → **Application Designer**.

5.1.1 Requirements

This use case covers the migration of the Work Order Tracking application changes. To be able to analyze the migration requirements, we first review the hypothetical business needs leading to the change requirements. For example, the application UI needed the following simple modifications:

- ▶ The Failure Reporting screen is not needed and needs to be removed.
- ▶ The Work Type field needs to be called Type.
- ▶ The Customer field is always required.
- ▶ The License field is not used and needs to be removed from the Work Order screen and from the More Search Fields in the Advanced Search menu.
- ▶ Add the Report Date in the list screen between the Description and Location, and make it sortable.
- ▶ Initially sort the searching results list by the Report Date field.
- ▶ The signature option for Apply SLA must not be visible to security.
- ▶ When clicking the Create KPI tool bar button when displaying a listing result, the system must warn that the action will affect all listed records.
- ▶ Make the Run Report menu option the first item in the Select Action menu.
- ▶ Disable the Change Status button in the list screen and make it available only in the Work Order screen.
- ▶ Remove the tool bar buttons for Initiate, Approve, Complete, and Close work order.
- ▶ Make the Where clause option first in the Advanced Search menu.

To fulfill these needs, make the following changes to the application UI based on the business needs:

1. Using the Application Designer, search the application Work Order Tracking and open it for modifications. The name of the application is WOTRACK.

2. Using the workspace in Application Designer, we made the following changes to the WOTRACK application to meet the requirements:
 - a. In the List tab screen, add a new table column control for the existing object attribute Report Date, between the Description and Location columns. You can change the layout of the control in the screen using editing techniques, such as click and drag, or cut and paste.
 - b. Edit the properties of the Report Date control just added, clearing the Sortable? attribute and making the Input Mode attribute Read only.

Note: The most commonly modified controls are text boxes, check boxes, sections, static text, help grid, table, and push buttons.

- c. Remove the unwanted control for the License field from the Main tab screen by deleting the control.
- d. Modify the signature option APPLYSLA advanced signature options, selecting the second option Warning appears when this action is selected.
- e. Modify the visibility of the key value APPLYSLA of the Select Action menu option, by clearing the Visible? option.
- f. Modify the visibility of the key value STATUS in the Toolbar Menu options, by making the tool bar button only available in the MAIN tab.
- g. Modify the position of the key value SEARCHWHER in the Search Menu options, by changing the value of sub position to a number lower than the first key value in the list.

Note: This book does not cover the details of how to make the modifications for each solution, because it depends on the details of every specific requirement. For more information about how to use the Application Designer tool, see the *IBM Maximo Asset Management 7.1*, *IBM Tivoli Asset Management for IT 7.1*, *IBM Tivoli Change and Configuration Management Database 7.1.1*, and *IBM Tivoli Service Request Manager 7.1 Application Developer Guide* at this website:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tamit.doc_7.1/pdf/mam71_app_dev_guide.pdf

These changes are considered minor in the sense that only the Application Designer tool, which is provided by the product, is used to fulfill the change requirements.

5.1.2 Solution

After reviewing the modifications to the Work Order Tracking application, an approach is identified so that all the related configuration content is migrated using a single Snapshot migration package.

From a content perspective, the migration of the modified application requires the migration of the application's configuration.

5.1.3 Configuration applications used for this solution

Application configurations are accessed by clicking **Go To → System Configuration → Platform Configuration → Application Designer**.

5.1.4 Object structures

The following standard object structures that are provided with the product are used to support the migration of the solution that is defined in this use case scenario:

- ▶ DMMAXAPPS
- ▶ DMMAXMENU
- ▶ DMSIGOPTFLAG

Figure 5-1 shows the object structure details for these three objects. You can access migration object structures by clicking **Go To → System Configuration → Migration → Object Structures**.

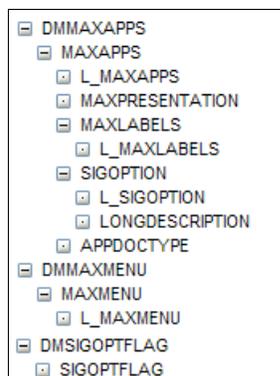


Figure 5-1 Object structures for our case scenario

Note: For more information about migration object structures, refer to the *Migration Manager Guide* at this website:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.mam.doc_7.1/pdf/mam71_migration_mgr_guide.pdf

5.1.5 Migration groups

After the change requirements have been clearly identified, the next step is to define and create a group of object structures that are the logical containers of the configuration needed to migrate.

The object structures for this migration are in two standard migration groups called APPLICATION and APPSECURITY. These groups are shipped with the product to support application and security changes.

The configuration content for the advanced signature options defined in the Application Designer is not in the standard APPLICATION group; however, it is in the APPSECURITY group.

These migration groups with their dependencies are not useful for scenarios in which subsets of configurations need to be selectively migrated and therefore must be left as is. Figure 5-2 displays the object structure list in these migration groups.

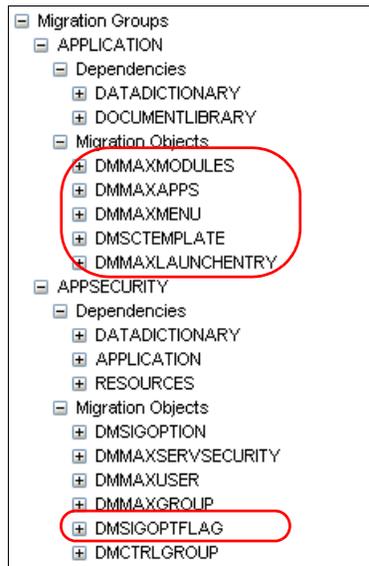


Figure 5-2 APPLICATION and APPSECURITY migration groups

A new application migration group that includes the object structures needed to create a migration package for our use case scenario is necessary. The new group is based on the two groups mentioned previously and contains the highlighted object structures as illustrated in Figure 5-2 on page 113. Using the Migration Groups application, create a new Migration Group by *duplicating* the APPLICATION, removing the dependent migration groups.

Now, we need to include the object structure that contains advanced signature option configuration in this new group. Add the object DMSIGOPTFLAG to the new group after the DMMAXLAUNCHENTRY object, and then, save. Figure 5-3 shows the new group.

Migration Object	Description	Migration Object Order
DMMAXMODUL >>	Application module migration object for Migratio	1
DMMAXAPPS >>	Application migration object for Migration Mana	2
DMMAXMENU >>	Application menu migration object for Migration	3
DMSCTEMPLAT >>	Start Center template migration object for Migra	4
DMMAXLAUNCI >>	Launch entry migration object for Migration Mar	5
DMSIGOPTFLA(>>	Advanced signature option migration object for	6

Dependent Migration Group	Description
...No rows to display...	

Figure 5-3 New migration group MYAPPLICATION

Tip: Migration Groups are the shells that are used to define the specific content (or objects) to migrate, and they can be reused many times for multiple migration scenarios.

5.1.6 Package definition

After the object structures have been identified and organized, the migration process can begin. Create a new Snapshot package, which includes the newly defined migration group, to use for defining the specific content to migrate to the Target environment.

The next step is to define the SQL criteria within each of the object structures in the group to select the content. The Work Order Tracking application's name, WOTRACK, is used to create the criteria in the SQL conditions.

Table 5-1 lists the Where clause SQL command statement conditions for each object. These conditions result in one or many records of configuration content for each migration object.

Tip: A migration package might contain no records for a specific migration object. The Where clause of 1=2 will not generate an error and is commonly used when the object does not contain relevant configuration information in the specific object.

Table 5-1 Set Where clause definitions for the basic application package

Migration object	Object	Where clause
DMMAXMODULES	MAXMODULES	1=2
DMMAXAPPS	MAXAPPS	app in ('WOTRACK')
DMMAXMENU	MAXMENU	moduleapp in ('WOTRACK')
DMSCTEMPLATE	SCTEMPLATE	1=2
DMMAXLAUNCHENTRY	MAXLAUNCHENTRY	1=2
DMSIGOPTFLAG	SIGOPTFLAG	app in ('WOTRACK')

In this case, we migrate only configuration content that is stored in the MAXAPPS, MAXMENU, and SIGOPTFLAG objects. Therefore, the conditions for each object generate the precise content to be stored in the package.

Tip: When more than one application is changed and promoted for migration, you can add more application names separated by commas, as shown in the following example:

```
app in ('WOTRACK', 'CHANGE', 'JOBPLAN')
```

The Where clauses for DMMAXMODULES, DMSCTEMPLATE, and DMLAUNCHENTRY are defined with a logical false statement (1=2), which generates no results. That is the intention in this particular case.

Enter the SQL conditions by clicking the Set Where clause icon for the MYAPPLICATION migration group, as illustrated in Figure 5-4 on page 116.

Migration Object	Object	Where Clause
DMMAXMODUL	MAXMODULES	1=2
DMMAXAPPS	MAXAPPS	app in ('WOTRACK')
DMMAXMENU	MAXMENU	moduleapp='WOTRACK'
DMSCTEMPLAT	SCTEMPLATE	1=2
DMMAXLAUNCI	MAXLAUNCHENTRY	1=2
DMSIGOPTFLA	SIGOPTFLAG	app in ('WOTRACK')

Figure 5-4 Where clause conditions for WOBASICAPP migration package

5.1.7 Deployment

After the package definition is approved and activated, you can create and distribute the package. Then, deploy the package to the Target environment. Turn Admin mode on before deploying the package. The Migration Manager requires Admin mode when deploying a package that contains the object MAXAPPS.

Resource: This document does not repeat the standard steps that are involved in deploying a package. For detailed information, refer to the *Migration Manager Guide*:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.mam.doc_7.1/pdf/mam71_migration_mgr_guide.pdf

5.1.8 Deployment considerations

This use case scenario shows how to migrate simple changes that have been made on any application; however, keep in mind the considerations that we discuss in the following sections.

Language considerations

Language considerations are described in detail in 10.2, “Multiple language considerations” on page 201.

5.2 Complex application changes, including menus, lookups, and global data restrictions

In this new use case scenario, we describe the migration of more complicated configuration content that involves the use of other applications, before and after using the Application Designer tool.

A new menu item is created, new library lookups are built, and security restrictions are applied to data attributes.

This sequence of events in the configuration cycle causes migration dependencies that are analyzed during the migration requirements and solutions. Then, the migration dependencies are addressed during the migration package definition and deployment considerations.

5.2.1 Requirements

For this use case, we migrate modifications that have been made for the Assets application. To be able to analyze the migration requirements, we first review the hypothetical change requirements in which is a need to record test results for serialized assets, keep track of these tests and send notifications when a test failed:

- ▶ A new table is created with a one-to-many relationship from the assets table to this new Test Results table.
- ▶ This new table will store information, such as the test result, date, type of test, description of the test, and a defect code if the test failed, for any asset that is serialized.
- ▶ The test results are recorded using a new application UI that is available only to the testing team.
- ▶ The test type and results will be validated against predefined values, and the date field will default to current date.
- ▶ The test result cannot be deleted or duplicated.
- ▶ Searching and listing options are available in the List tab.
- ▶ All information is required with the exception of the defect code. A defect code is only required if the test failed.
- ▶ A new tab is added to the Asset application to display the test results associated to the asset, and it is accessible only to the test team.

To fulfill this need, we make the following minimum change configurations to accomplish the business needs under this scenario:

1. Using the Domains application, two new ALN domains are created to use for the validation values of the results and type of test. The names of these domains are MYTESTRESULT and MYTESTTYPE.
2. Using the Domains application, a new Table domain is created for looking up values in the linked asset field of the new table. The name of the domain is MYASSETTB.
3. Using the Database Configuration application, a new table object is created with all the attributes required, new domain associations, and a new relationship to the ASSET table, which is used to display the asset serial number and other object definitions. The name of the new table object is MYTESTS.
4. Using the Application Designer application, a new table ID reference to the table domain created in prior steps is created in the LOOKUPS system library XML.
5. Using the Application Designer application, a new application of type Power App is created with all of the configuration data that is required for the new table object. The name of the new application is MYTESTS.
6. Using the Application Designer application, the existing Assets application is configured with a new tab and controls needed to display the test results records linked to the asset. A new signature option is created to configure the security restrictions for this new tab. The name of the existing application is ASSET.
7. Using the Security Groups application, two new security groups are created for signature security access configuration to the new Test Results application and the new tab in the Assets application. The names of the two new security groups are TESTMGR and TESTTECH.
8. Using the Conditional Expression Manager application, a new condition is created that evaluates the test result attribute when the value is FAILED. The name of the new conditional expression is MYFAILED.
9. Using the Security Groups application, a new global data restriction is created to evaluate the test result field during data entry, based on the condition that was created in the previous step, and to dynamically change the input mode to Required of the test defect code field. The attribute restriction is for the application name MYTEST.

The configuration content that is generated from at least five separate applications needs to be migrated, and the order in which the configuration is created is extremely important.

5.2.2 Solution

After reviewing the modifications for this case scenario, we identified an approach so that all the related configuration content is migrated using more than one Snapshot migration package. These packages are deployed on the Target environment in a specific sequence.

From a content perspective, the migration of the modified application requires the migration of the following configurations in order:

- ▶ Domains
- ▶ Database table, sequence, and relationships
- ▶ Conditional expressions
- ▶ System library lookups
- ▶ Applications, menus, and signature options
- ▶ Global security restrictions
- ▶ Security groups

5.2.3 Configuration applications used for this solution

Next, we show the configuration applications that were used for this solution and how you can access these applications:

- ▶ Domains
To access the Domains application, click **Go To → System Configuration → Platform Configuration → Domains**.
- ▶ Database table, sequence, and relationships
To launch the Database Configuration application, click **Go To → System Configuration → Platform Configuration → Database Configuration**.
- ▶ Conditional expressions
To access the Conditional Expression Manager application, click **Go To → Administration → Conditional Expression Manager**.
- ▶ System library lookups
To launch the Application Designer application, click **Go To → System Configuration → Platform Configuration → Application Designer**. From the Application Designer, select **Select Action → Export System XML**.
- ▶ Applications, menus, and signature options
Applications, menus, and signature options can be configured from the Application Designer application, which is accessed by clicking **Go To → System Configuration → Platform Configuration → Application Designer**.

- ▶ Global security restrictions
To configure global security restrictions, click **Go To** → **System Configuration** → **Security Groups**, and select **Select Action** → **Global Data Restrictions**.
- ▶ Security groups
To launch the Security Groups application, click **Go To** → **System Configuration** → **Security Groups**.
- ▶ Object structures
The Object Structures application can be accessed by clicking **Go To** → **System Configuration** → **Migration** → **Object Structures**.

5.2.4 Object structures

Several standard object structures are identified for the migration of the configuration content for this scenario. However, the configuration content of the global data restrictions and the system library lookups requires special attention.

Tip: Always try using the standard objects and groups shipped with the product, because they reflect the logical business object structures and associations that have been built by the developers.

Standard object structures

The following standard object structures that are provided with the product are used to support the migration requirements:

- ▶ DMMAXOBJECTCFG
- ▶ DMMAXDOMAIN
- ▶ DMMAXRELATIONSHIP
- ▶ DMCONDITION
- ▶ DMMAXAPPS
- ▶ DMMAXMENU
- ▶ DMMAXGROUP

Global data restrictions content related to the application

This configuration content is currently in the table SECURITYRESTRICT, which is defined in the standard object structure DMMAXGROUP, and its key relationship is a security group. When the data restriction is global, it is not related to a security group.

We cannot migrate this content using the standard object structure, because our use case configured a global data restriction, which is linked to an application name and not to a security group.

DMMAAXGROUP: You can obtain more information about the standard migration object structure DMMAAXGROUP in Chapter 3, “Migrating security configuration data” on page 51.

Therefore, to migrate the global security restriction, you need to create a new migration object structure. The new object is a duplicate of the DMMAXAPPS object, and the table SECURITYRESTRICT is added to relate its content by application name.

Create a new migration object structure by *duplicating* DMMAXAPPS with the following characteristics:

Object structure	MYMAXAPPS
Description	Application with data security restriction

Add a new source object at the end of the list of source objects for MYMAXAPPS with the following characteristics:

Object	SECURITYRESTRICT
Parent object	MAXAPPS
Relationship	SECURITYRESTRICT
Object order	6 (or default)

Save the new object structure.

Figure 5-5 shows the new MYMAXAPPS object structure list.

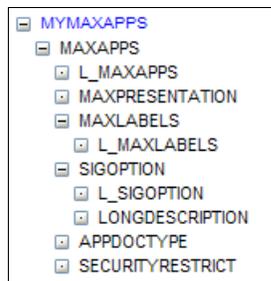


Figure 5-5 New migration object structure MYMAXAPPS

The new object structure created here requires the migration of its configuration content, and it is in the standard object structure DMMAXINTOOBJECT.

System library lookups object structure

The configuration of system lookups is done using the Application Designer, and it can also be migrated using the same method in the Target environment. However, in this scenario, we create a migration object structure to include in our migration group.

This configuration is stored in the MAXPRESENTATION table in the form of XML content. This table also contains the XML representations of every application, and the table is in the MYMAXAPPS object structure, as shown in Figure 5-5 on page 121. Because there is no application built for the LOOKUPS library, its content is identified by the application name in the presentation table.

Create a new migration object structure called MYMAXPRESENTATION with all of the default values and the following characteristics:

Object structure	MYMAXPRESENTATION
Description	XML presentations
Consumed by	MIGRATIONMGR

Add a new source object at the end of the list of source objects for MYMAXPRESENTATION with the following characteristics, leaving all of the default values:

Object	MAXPRESENTATION
Object order	1 (or default)

Save the new object structure.

Figure 5-6 shows the new MYMAXPRESENTATION object structure.

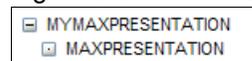


Figure 5-6 MYMAXPRESENTATION object structure

The new object structure created here also requires the migration of its configuration content, which is migrated using the standard object structure DMMAXINTOBJECT.

Migration object structures

The following revised list shows the object structures that are used to support the migration requirements, in no particular order:

- ▶ DMMAXOBJECTCFG
- ▶ DMMAXINTOBJECT
- ▶ DMMAXDOMAIN
- ▶ DMMAXSEQUENCE
- ▶ DMMAXRELATIONSHIP

- ▶ DMCONDITION
- ▶ DMMAXMENU
- ▶ DMMAXGROUP
- ▶ MYMAXAPPS
- ▶ MYMAXPRESENTATION

Note: For more information about migration object structures, refer to the *Migration Manager Guide*:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.mam.doc_7.1/pdf/mam71_migration_mgr_guide.pdf

5.2.5 Migration groups

After reviewing the change requirements and identifying the object structures, the next step is to define the groups that will be used to define the package content.

In our first scenario, 5.1, “Basic application changes and signature options” on page 110, we only used one group and one package, because the changes were limited to one configuration tool and were independent of other configurations.

This time, we are migrating separate groups of configurations in a sequence that reflects the order in which the configurations were performed in the Source environment. Defining the configuration into three separate areas of the product configuration cycle, such as database configuration, application, and security, allows us to better control the migration process, is easy to understand, and also allows reusability.

Reuse: Migration groups are the shells that are used to define the specific content (or objects) to migrate, and they can be reused many times for multiple migration scenarios.

We will use three groups: the data dictionary group, application group, and application security group.

Data dictionary group

The first group that we use is the standard DATADictionary group that is provided by the product. The needed objects in this group are highlighted in Figure 5-7 on page 124.

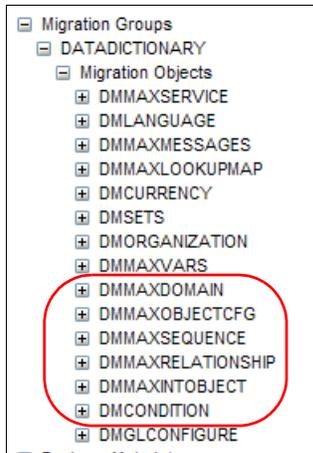


Figure 5-7 DATADICTONARY migration group

Application group

This new migration group is created from standard groups as it was in our basic changes scenario in 5.1.5, “Migration groups” on page 113. However, this situation differs in regard to the conditional expressions and the system lookups.

The CONDITION table holds the conditional expressions, and in our scenario, we need to migrate a new expression that is built to evaluate the new table. The Migration Manager creates the new table using DMMAXOBJECTCFG, which is part of the first group, as shown in Figure 5-7. Also, DMCONDITION is part of that group, too; however, the group’s migration order is reversed, meaning that, the new table is created after the condition. If we try to migrate the condition with that group, it will fail, because the new table has not been created when it processes the condition. Figure 5-8 on page 125 displays the migration object configuration and the object order.

Migration Object	Description	Migration Object Order	Inte
▶ DMMAXSERVIC >>	System service migration object for Migration M	1	<input checked="" type="checkbox"/>
▶ DMLANGUAGE >>	Language migration object for Migration Manag	2	<input checked="" type="checkbox"/>
▶ DMMAXMESSA >>	System message migration object for Migration	3	<input checked="" type="checkbox"/>
▶ DMCURRENCY >>	Currency migration object for Migration Manage	4	<input checked="" type="checkbox"/>
▶ DMSETS >>	Set migration object for Migration Manager	5	<input checked="" type="checkbox"/>
▶ DMORGANIZAT >>	Organization and site migration object for Migra	6	<input checked="" type="checkbox"/>
▶ DMMAXVARS >>	MAXVARS migration object for Migration Mana	7	<input checked="" type="checkbox"/>
▶ DMCONDITION >>	Condition expression migration object for Migra	8	<input checked="" type="checkbox"/>
▶ DMMAXDOMAIN >>	Domain migration object for Migration Manager	9	<input checked="" type="checkbox"/>
▶ DMMAXOBJECT >>	Data dictionary object migration object for Migr	10	<input checked="" type="checkbox"/>

Figure 5-8 DATADictionary migration objects

To take care of the issue, we migrate the conditional expressions in the applications group and not in the data dictionary group. We also include the new object structure that was created for migrating the system lookups library with this new group.

Create the new group by duplicating the standard APPLICATIONS group first, leaving all defaults, and then, follow these steps:

1. Remove all dependency groups.
2. Add the object structure DMCONDITION to the list of objects and rearrange the object order so that the added object is first, shifting the other objects one place.
3. Remove the object DMMAXAPPS. Take note of this object order before deleting it.
4. Add the object structure MYMAXAPPS that was created in 5.2.4, “Object structures” on page 120. Make the object order the same as it was for DMMAXAPPS.
5. Add the object structure MYMAXPRESENTATION to the list of objects. Leave it at the end of the list.
6. Save the group.

Figure 5-9 on page 126 shows the newly created group object list and the order of the migration objects.

Migration Object	Description	Migration Object Order	Inte
DMCONDITION	Condition expression migration object for Migra	1	<input type="checkbox"/>
DMMAXMODUL	Application module migration object for Migratio	2	<input type="checkbox"/>
MYMAXAPPS	Application with data security restriction	3	<input type="checkbox"/>
DMMAXMENU	Application menu migration object for Migration	4	<input type="checkbox"/>
DMSCTEMPLAT	Start Center template migration object for Migra	5	<input type="checkbox"/>
DMMAXLAUNCI	Launch entry migration object for Migration Mar	6	<input type="checkbox"/>
MYMAXPRESEN	System XML migration object for Migration Man	7	<input type="checkbox"/>

Figure 5-9 MYNEWAPP migration objects

Application security group

Our third group is a new group that is created based on a standard group, APPSECURITY, which is provided by the product for security group migration. Create a new group by duplicating the standard APPSECURITY group, leaving all of the defaults, except remove all dependency groups and then save the new group.

Resource: We review more application security use case scenarios in Chapter 3, “Migrating security configuration data” on page 51.

Now that we have defined our migration groups for this use case scenario, we are ready to start the migration process.

5.2.6 Package definitions

For this use case scenario, the migration approach is to deploy three separate Snapshot packages. Each package is created with one of the migration groups that was defined in 5.2.5, “Migration groups” on page 123, and each package is configured with SQL statement conditional criteria that reflect the migration requirements.

Follow the process in the next sections to create the new packages with the exact configuration content to migrate.

Data dictionary package

Create a new package definition that you can name MYTESTS-DD. This package is the Snapshot type and contains the migration group DATADictionary.

The next step is to define the SQL criteria within each of the object structures in the group. Using the information that was gathered from 5.2.1, “Requirements” on page 117, we use the names of the objects (new or changed) to construct the SQL conditions.

Table 5-2 lists the Where clause SQL command statement conditions for each object. These conditions result in one or many records of configuration content for each migration object.

Table 5-2 Where clause definitions for MYTESTS-DD package

Migration object	Object	Where clause
DMMAXSERVICE	MAXSERVICE	1=2
DMLANGUAGE	LANGUAGE	1=2
DMMAXMESSAGES	MAXMESSAGES	1=2
DMCURRENCY	CURRENCY	1=2
DMSETS	SETS	1=2
DMORGANIZATION	ORGANIZATION	1=2
DMMAXVARS	MAXVARS	1=2
DMCONDITION	CONDITION	1=2
DMMAXDOMAIN	MAXDOMAIN	domainid in ('MYASSETTB', 'MYTESTRESULT', 'MYTESTTYPE')
DMMAXOBJECTCFG	MAXOBJECTCFG	objectname in ('MYTESTS')
DMMAXSEQUENCE	MAXSEQUENCE	tbname in ('MYTESTS')
DMMAXLOOKUPMAP	MAXLOOKUPMAP	1=2
DMMAXRELATIONSHIP	MAXRELATIONSHIP	parent in ('MYTESTS') or child in ('MYTESTS')
DMMAXINTOBJECT	MAXINTOBJECT	intobjectname in ('MYMAXAPPS','MYMAXPRESENTATION')
DMGLCONFIGURE	GLCONFIGURE	1=2

The conditions for each object generate the precise content for this package. The Where clauses for the objects that are defined with a logical false statement (1=2) do not generate any results, because there is no relevant content to migrate in this particular case.

Click the Set Where Clause icon to enter the SQL conditions, as shown in the Table 5-2 on page 127, and then, save the package definition.

Application package

Now, create a new package definition that you can name MYTESTS-APP. This package is the Snapshot type and contains the new migration group MYNEWAPP, which was created for this scenario.

The next step is to define the SQL criteria within each of the object structures in the group. Using the information that was gathered from 5.2.1, “Requirements” on page 117, we use the names of the objects (new or changed) to construct the SQL conditions.

Table 5-3 lists the Where clause SQL command statement conditions for each object. These conditions result in one or many records of configuration content for each migration object.

Table 5-3 Where clause definitions for MYTESTS-APP package

Migration object	Object	Where clause
DMCONDITION	CONDITION	conditionnum in ('MYFAILED')
DMMAXMODULES	MAXMODULES	1=2
MYMAXAPPS	MAXAPPS	app in ('MYTESTS', 'ASSET')
DMMAXMENU	MAXMENU	moduleapp in ('MYTESTS') or keyvalue in ('MYTESTS')
DMSCTEMPLATE	SCTEMPLATE	1=2
DMMAXLAUNCHENTRY	MAXLAUNCHENTRY	1=2
MYMAXPRESENTATION	MAXPRESENTATION	app in ('LOOKUPS')

The conditions for each object generate the precise content for this package. The Where clauses for the objects that are defined with a logical false statement (1=2) do not generate any results, because there is no relevant content to migrate in this particular case.

Click the Set Where Clause icon to enter the SQL conditions, as shown in Table 5-3 on page 128, and then, save the package definition.

Security package

Now, create a new package definition that you can name MYTESTS-SEC. This package is the Snapshot type and contains the new migration group MYAPPSECURITY, which was created for this scenario.

The next step is to define the SQL criteria within each of the object structures in the group. Using the information that was gathered from 5.2.1, “Requirements” on page 117, we use the names of the objects (new or changed) to construct the SQL conditions.

Table 5-4 lists the Where clause SQL command statement conditions for each object. These conditions result in one or many records of configuration content for each migration object.

Table 5-4 Where clause definitions for MYTESTS-SEC package

Migration object	Object	Where clause
DMSIGOPTION	SIGOPTION	1=2
DMMAXSERVSECURITY	MAXSERVSECURITY	1=2
DMMAXGROUP	MAXGROUP	groupname in ('TESTMGR', 'TESTTECH')
DMMAXUSER	MAXUSER	1=2
DMSIGOPTFLAG	SIGOPTFLAG	1=2
DMCTRLGROUP	CTRLGROUP	1=2

The conditions for each object generate the precise content for this package. The Where clauses for the objects defined with a logical false statement (1=2) do not generate any results, because there is no relevant content to migrate in this particular case.

Click the Set Where Clause icon to enter the SQL conditions, as shown in Table 5-4, and then, save the package definition.

5.2.7 Deployment

After all of the package definitions are created, approved, and activated, create and distribute each package to the Target. After the packages are available from the Target environment, it is time to deploy them.

For the purposes of this book, we assume that all of the required change and release protocols (if any) have been put in place. We also assume that the users

in the Target environment have been notified that the system will not be available during deployment time.

For our use case scenario, follow these deployment steps:

1. Turn Admin mode on from the Migration Manager application.
Because the configuration content modifies the data dictionary and touches the application presentations for the UI (the MAXAPPS object is in the package), the system requires this mode of operation during deployment. We also recommend that you have the minimum amount of application infrastructure traffic by having no users connected to the environment.
2. After verifying that the system is now in Admin mode operation, load and deploy the Data Dictionary package, MYTESTS-DD. If an error occurs, stop and follow troubleshooting procedures until the error has been corrected. Continue to the next step only if there are no errors.
3. After deploying the first package without any errors, verify that the content has been migrated, and do not proceed until the content has been verified.
4. Load and deploy the Application package, MYTESTS-APP. Again, if errors occur, stop and follow troubleshooting procedures. Continue to the next step only if there are no errors.
5. After deploying the second package without any errors, verify that the expected content has been migrated and do not proceed until the content has been verified.
6. Load and deploy the Security package, MYTESTS-SEC. Again, if errors occur, stop and follow the troubleshooting procedures.
7. After all of the content is validated, turn Admin mode off in the system to allow users to log in.

Refer to Chapter 11, “Troubleshooting” on page 239 for more information about troubleshooting.

5.2.8 Deployment considerations

In the following section, we discuss deployment considerations for this migration.

Admin mode

When deploying packages containing structural changes, such as in the data dictionary or application presentation, the system expects the Admin mode operation be turned on. In a single Java virtual machine (JVM) server environment, there is no problem; however, when working on a multiple server environment or a cluster, all of the servers in the cluster must switch to this mode.

During this mode, the application does not allow any users to log in without the administrator's permission. Always remember to switch off this mode of operation after completing the deployment.

5.3 Migrating a Start Center and a query

Start Centers are the launching screen portal that is presented to the user when logging in to the product. There are several Start Centers to which a user can have access, based on the security configuration. The layout and content of the Start Center is stored in XML presentation form, in a similar manner to the application UI.

In this case scenario, we review the migration of a new Start Center template, plus the application query that has been configured for a result set.

5.3.1 Requirements

Previously, we reviewed a case scenario where a new application for test results was created. As part of the same requirement, a new Start Center is requested to be migrated:

- ▶ A new test portal Start Center is created for test managers and technicians and is linked to the identified security groups for these roles.
- ▶ The Start Center contains a favorite application portlet with links to the new application Test Results and to the Assets application.
- ▶ The Start Center contains the usual bulletin board and task portlets.
- ▶ The Start Center contains a result set portlet that contains a new query that was created in the new application Test Results and that filters the test table to records with the result of Failed.

The following minimum change configurations are made to fulfill the requirement:

1. Using the Test Results application that was created in 5.2, "Complex application changes, including menus, lookups, and global data restrictions" on page 117), create a new query to filter Failed results. The clause name for the query is identified as FAILEDTEST.
2. Using the Start Center application, create a new template with a Narrow-Wide layout with the Favorite Applications portlet on the left and the Bulletin Board, Inbox Assignment, and Result Set portlets on the right. The Result Set portlet needs to point to the new query in the Test Results application. The name of the template was identified as Template-20100927205646.

In this case scenario, we created the configuration using the query capabilities of existing applications and the Start Center template configuration. The *sequence in which the configuration was created* is extremely important to consider during migration.

5.3.2 Solution

After reviewing the changes in our scenario, our migration approach identifies the use of one package that uses the application group, as shown in the basic use case scenario in 5.1, “Basic application changes and signature options” on page 110.

However, after carefully reviewing the configuration object content, we see the need to create a new migration object for the migration of the Query object. This object was not shipped with the product. Creating the new object structure will require its migration using the Data Dictionary migration group in a separate package.

5.3.3 Configuration applications used for the solution

Next, we show the Configuration applications that were used for this solution and how you can access these applications.

Queries for the Test Results application

To create queries, click **Go To** → **Assets** → **Test Results**, and then, click **Save Query**.

Start Center templates

You can modify or create new Start Center templates by clicking **Start Center** → **Create New/Modify Template**.

Object Structures

You can access the Object Structures application by clicking **Go To** → **System Configuration** → **Migration** → **Object Structures**.

5.3.4 Object structures

A standard object structure is identified for the migration of the configuration content for the Start Center template. However, the configuration content of the Queries requires special attention.

Tip: Always try using the standard objects and groups that are shipped with the product, because they reflect the logical business object structures and associations that have been built by the developers.

Standard object structures

We use the following standard object structures in the migration group:

- ▶ DMSCTEMPLATE
- ▶ DMMAXINTOBJECT

Query object structure

Queries are linked to applications and user IDs. The content is stored in the table named QUERY.

Create a new migration object structure called MYQUERY, with all of the default values and the following characteristics:

Object structure	MYQUERY
Description	Query object structure
Consumed by	MIGRATIONMGR

Add a new Source object to the list of Source objects for MYQUERY with the following characteristics, leaving all of the default values:

Object	QUERY
Object order	1 (or default)

Save the new object structure.

Figure 5-10 shows the new MYQUERY object structure.

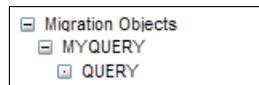


Figure 5-10 MYQUERY object structure

The new object structure that is created here also requires the migration of its configuration content, which is migrated using the standard object structure DMMAXINTOBJECT.

Migration object structures

The following revised list shows the object structures that are used to support the migration requirements, in no particular order:

- ▶ DMSCTEMPLATE

- ▶ DMMAXINTOBJECT
- ▶ MYQUERY

Resource: For more information about migration object structures, refer to the *Migration Manager Guide*:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.mam.doc_7.1/pdf/mam71_migration_mgr_guide.pdf

5.3.5 Migration groups

The next step is defining the migration groups to use in the package definition. After the object structures are identified and because of the newly created object structure, we use two groups. The first group migrated the data dictionary content, in this case, the new object structure definition. Then, the Start Center content is migrated in an Application group with a slight modification to include the query content.

Tip: After you migrate the new object structure, the subsequent migration of Start Center templates that include queries for result sets will not require the data dictionary group. Only one group, which is for Start Center templates and queries, will be necessary.

We need the following groups in our case scenario.

Data dictionary group

The first group is the standard DATADictionary group. The only object structure that we need to configure in our deployment is the DMMAXINTOBJECT.

Application group

As we have seen in previous scenarios, we must modify the standard migration group to remove the dependency groups and to add a missing, but related in content, object structure.

The Start Center template content is already in this standard group, in the DMSCTEMPLATE object structure. We are adding the query content in this group.

Using the Migration Object Structures application, create the new group by duplicating the standard APPLICATIONS group. The group name is MYAPPSC. We need to perform these configuration actions:

1. Remove all dependency groups.
2. Add the object structure MYQUERY, which was created in 5.3.4, “Object structures” on page 132. Leave this object at the end of the list.
3. Save the group.

Figure 5-11 shows the newly created group object list and the order of the migration objects.

Migration Object	Description	Migration Object Order	Intern
DMMAXMODUL	Application module migration object for Migratio	1	<input type="checkbox"/>
DMMAXAPPS	Application migration object for Migration Mana	2	<input type="checkbox"/>
DMMAXMENU	Application menu migration object for Migration	3	<input type="checkbox"/>
DMSCTEMPLAT	Start Center template migration object for Migra	4	<input type="checkbox"/>
DMMAXLAUNCI	Launch entry migration object for Migration Mar	5	<input type="checkbox"/>
MYQUERY	Query object for Migration	6	<input type="checkbox"/>

Figure 5-11 New migration group MYAPPSC

5.3.6 Package definition

The migration approach for this use case scenario is to deploy two separate Snapshot packages. The first package migrates the data dictionary content, and the second package migrates the application content.

Data dictionary package

Create a new package definition that you can name MYSC-DD. This package is the Snapshot type and contains the migration group DATADictionary.

The next step is to define the SQL criteria within each of the object structures in the group. Using the information that was gathered from 5.3.1, “Requirements” on page 131, we use the names of the objects to construct the SQL conditions.

Table 5-5 lists the Where clause SQL command statement conditions for each object. These conditions result in one or many records of configuration content for each migration object.

Table 5-5 Where clause definitions for MYSC-DD package

Migration object	Object	Where clause
DMMAXSERVICE	MAXSERVICE	1=2
DMLANGUAGE	LANGUAGE	1=2
DMMAXMESSAGES	MAXMESSAGES	1=2
DMCURRENCY	CURRENCY	1=2
DMSETS	SETS	1=2
DMORGANIZATION	ORGANIZATION	1=2
DMMAXVARS	MAXVARS	1=2
DMCONDITION	CONDITION	1=2
DMMAXDOMAIN	MAXDOMAIN	1=2
DMMAXOBJECTCFG	MAXOBJECTCFG	1=2
DMMAXSEQUENCE	MAXSEQUENCE	1=2
DMMAXLOOKUPMAP	MAXLOOKUPMAP	1=2
DMMAXRELATIONSHIP	MAXRELATIONSHIP	1=2
DMMAXINTOBJECT	MAXINTOBJECT	intobjectname in ('MYQUERY')
DMGLCONFIGURE	GLCONFIGURE	1=2

The conditions for each object generate the precise content for this package. The Where clauses for the objects defined with a logical false statement (1=2) do not generate any results, because there is no relevant content to migrate in this particular case.

Click the Set Where Clause icon to enter the SQL conditions, as shown in the table Table 5-5, and then, save the package definition.

Application package

Next, we create a new Snapshot package definition that you can name MYSC-APP. This package contains the new migration group MYAPPSC, which we created for this scenario.

Define the SQL criteria within each of the object structures in the group. Using the information that was gathered from 5.3.1, "Requirements" on page 131, we use the names of the objects to construct the SQL conditions.

Table 5-6 lists the Where clause SQL command statement conditions for each object. These conditions result in one or many records of configuration content for each migration object.

Table 5-6 Where clause definitions for MYSC-APP package

Migration object	Object	Where Clause
DMMAXMODULES	MAXMODULES	1=2
MYMAXAPPS	MAXAPPS	1=2
DMMAXMENU	MAXMENU	1=2
DMSCTEMPLATE	SCTEMPLATE	name in ('Template-20100927205646')
DMMAXLAUNCHENTRY	MAXLAUNCHENTRY	1=2
MYQUERY	QUERY	clausename='FAILEDTESTS' and app='MYTESTS'

The conditions for each object generate the precise content for this package. The Where clauses for the objects defined with a logical false statement (1=2) do not generate any results, because there is no relevant content to migrate in this particular case.

Click the Set Where Clause icon to enter the SQL conditions, as shown in Table 5-6, and then save, the package definition.

5.3.7 Deployment

After the two package definitions are created, approved, and activated, you create and distribute each package to the Target. Make the packages available to the Target environment, and perform all of the change management preparation. It is now time to deploy.

For our use case scenario, follow these deployment steps:

1. Turn Admin mode on from the Migration Manager application. This task is always a requirement when the deployment involves database configuration or screen presentation updates.
2. After verifying that the system is now in Admin mode operation, load and deploy the Data Dictionary package, MYSC-DD. If an error occurs, stop and follow troubleshooting procedures until the error has been corrected. Continue to the next step only if there are no errors.

3. After deploying the first package without errors, verify that the content has been migrated, and do not proceed until the content has been verified.
4. Load and deploy the Application package (MYSC-APP) next. Again, if errors occur, stop and follow troubleshooting procedures. Continue to the next step only if there are no errors.
5. After all of the content is validated, turn Admin mode off in the system to allow the users to log in.

Refer to Chapter 11, “Troubleshooting” on page 239 for more information about troubleshooting.

5.3.8 Deployment considerations

When approving, creating, or deploying packages that contain user-defined migration objects, the Migration Manager displays a system message, as shown in Figure 5-12.

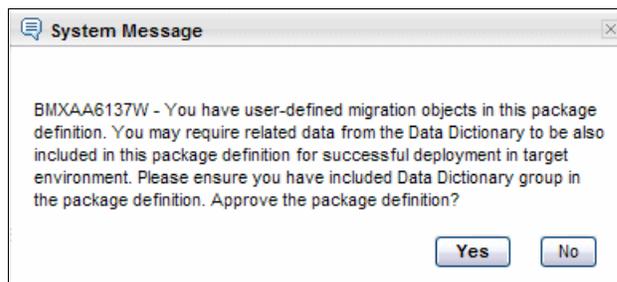


Figure 5-12 System Message BMXAA6137W

Always click **Yes** to proceed.



Migrating workflows

Workflow represents a defined business process that is layered on a *Maximo business object* (MBO). It is used extensively to automate many aspects of the business. For example, you can use a workflow process to automate the approval routing of a user-submitted purchase requisition or change request.

In its simplest form, a workflow process can consist of a conditional action, based on a submitted record criteria, requiring no user interaction. More complex workflow processes can be split into sub-processes and used to manage highly interactive change management approval processes, such as multiple assignments, actions (for example, status updates), and notifications via email or pager systems, using predefined communication templates.

This chapter has the following sections:

- ▶ 6.1, “Migrating specific workflow modifications” on page 140
- ▶ 6.2, “Migrating workflows and all their related content” on page 147

6.1 Migrating specific workflow modifications

A workflow is typically defined in a development environment and subsequently promoted to test and production environments. A repeatable process is required to reliably (without any errors or failures) and accurately (without missing any required content) migrate the workflow processes.

6.1.1 Requirements

In this use case scenario, we use the Source workflow PRSTATUS. The requirement is to migrate a duplicate of this workflow with modifications to the action lines, including communication templates that advise the requestor of the status of the request.

6.1.2 Solution

The original workflow might already exist (PRSTATUS) in the Target environment, and a new workflow that is based on the existing workflow needs to be migrated. Therefore, only the new workflow process and the components that have been added will be migrated. Any existing component that has not been changed is not migrated.

We defined an approach so that the workflow configuration and the related new configurations are migrated in a single Snapshot migration package.

From a content perspective, the migration of a workflow also implies the migration of the workflow configurations in the following order:

- ▶ Actions
- ▶ Action groups (assumes Data Dictionary is migrated)
- ▶ Roles
- ▶ Communication templates
- ▶ Workflow process

We use the standard business process management (BPM) workflow group as the base to define the content to migrate. This group also includes the INBOUNDCOMMCONFIG and ESCALATIONS object structures. These object structures are not required to support this focused migration use case. They will remain in the group definition, but they will be filtered out using SQL query filters.

In addition, a workflow specifies interaction with the users by using assignments and communication templates that are targeted at persons or role players, which are defined as either a Person or Person Group. For the purpose of this document, we also assumed that these dependencies and other dependencies

have been migrated or loaded into the Target system, as described in Chapter 1, “Migration strategy” on page 1.

6.1.3 Configuration applications used for this solution

This section describes the applications that are used in making the changes to support the example.

You can access the Workflow Designer application by navigating to **Go To** → **System Configuration** → **Workflow Designer**.

From Workflow Designer, open the PRSTATUS workflow, duplicate it, and name it REDBOOK_WF. Save it as Version 1.

The goal is to add a notification to the action in the connecting line between the SUB_APPR and CLOSE tasks, as illustrated in Figure 6-1.

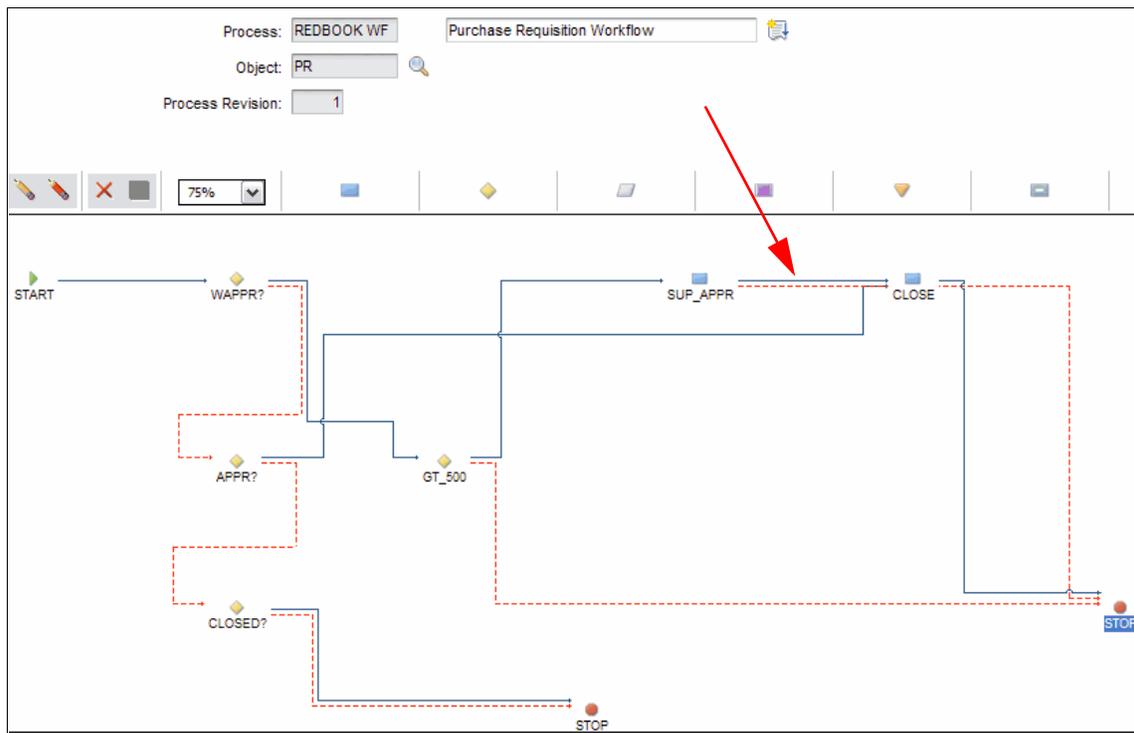


Figure 6-1 Workflow Designer canvas

The notification will include the new communication template RB_REQ_APP, which is shown in Figure 6-2.

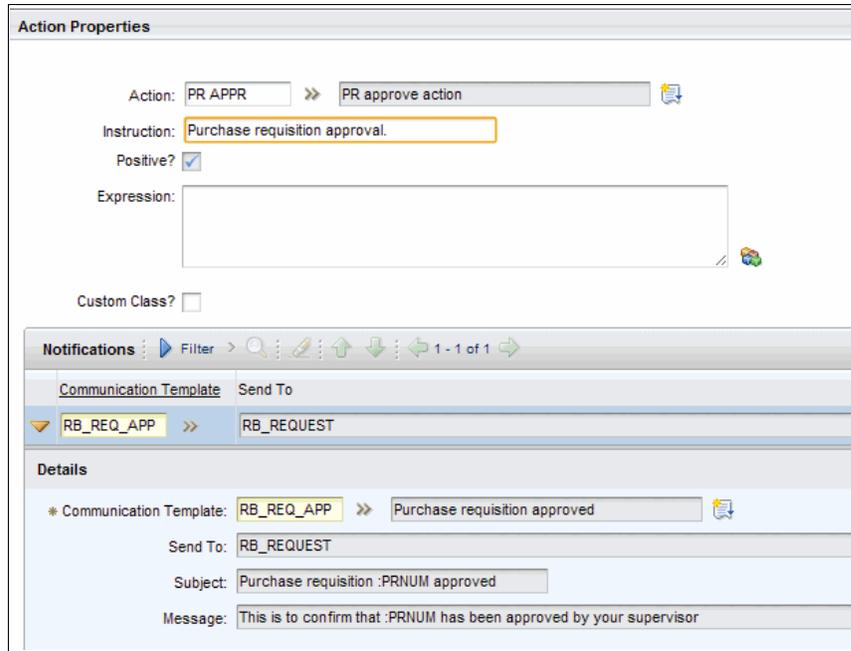


Figure 6-2 Connecting line action properties

The naming convention with the prefix RB_ will be applied to support the selective migration process that we will describe later. This naming convention is extremely important and supports the selective migration process that is described further in this chapter.

Notifications are managed using the Communication Templates application and can be accessed by navigating to **Go To → System Configuration → Platform Configuration → Communication Templates**.

A communication template can define one or more recipients in the form of email address, role, person, or person group. For this exercise, a communication template, RB_REQ_APP is added to the Action PR_APPR in the connector line between the two task nodes. In addition, a role will be related to the Purchase Requisition requestor. See Figure 6-3 on page 143.

Template:	RB_REQ_APP	Purchase requisition approved	
* Applies To:	PR		
* Accessible From:	ALL		
Comm Log Entry?	<input checked="" type="checkbox"/>		
<hr/>			
To:	RB_REQUEST		
cc:			
bcc:			
* Send From:	maxadmin@us.ibm.com		
Reply To:			
Subject:	Purchase requisition :PRNUM approved		
Message:	This is to confirm that :PRNUM has been approved by your supervisor		

Figure 6-3 Communication template

When the communication template is associated with a role, the role has to be defined using the Roles application, which can be accessed by clicking **Go To** → **System Configuration** → **Platform Configuration** → **Roles**. You can create the role based on a previously defined person or person group record or a role that is associated with a person record. For this example, the use of a record-related role player (PR.Requestedby) allows the use of the previously loaded PR records on a role-based basis. The requestor of the purchase requisition is the role player in this case, as shown in Figure 6-4.

Role:	RB_REQUEST	Requester of requisition	
* Type:	A set of data relat		
* Object:	PR		
* Value:	:requestedby		
Parameter:			
E-mail?	<input type="checkbox"/>		
Broadcast?	<input checked="" type="checkbox"/>		

Figure 6-4 Roles

6.1.4 Object structures

The following standard object structures, which are shown in Figure 6-5, are required to support this use case migration.

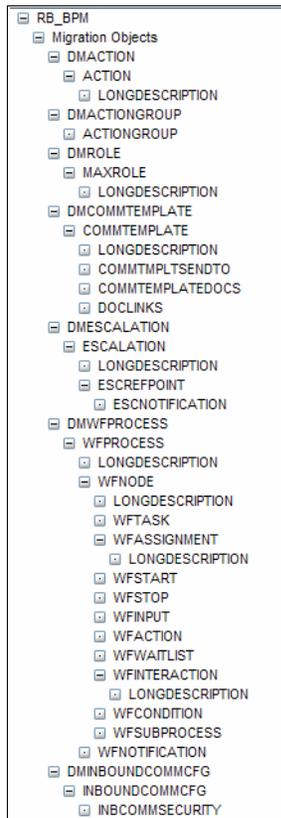


Figure 6-5 RB_BPM object structures

6.1.5 Migration groups

The object structures are in one migration group, named RB_BPM. For the workflow migration scenario, using the predefined migration groups is inefficient, because those groups include dependencies on other groups that are provided to ensure that the deployment will not fail. We are working on a scenario that implies the Target already contains the workflow elements and that precautions have been taken to migrate any dependency objects prior to this migration.

Create the new migration group by duplicating the existing BPM group. The dependencies are simply deleted, and the order of the objects is the same.

Figure 6-6 shows the resulting migration group RB_BPM that includes only the object structures that are required to support this workflow migration.

Migration Group: **RB_BPM** **Redbook Business Process Manager Group** Change By: MAXADM
 * Migration Group Order: 29 Change Date: 10/11/10 :
 Internal?

Migration Object	Description	Migration Object Order	Inter
DMACTION	Action migration object for Migration Manager	1	<input type="checkbox"/>
DMACTIONGRO	Actiongroup migration object for Migration Man:	2	<input type="checkbox"/>
DMROLE	Role migration object for Migration Manager	3	<input type="checkbox"/>
DMCOMMTEMPL	Communication template migration object for Mi	4	<input type="checkbox"/>
DMESCALATIOI	Escalation migration object for Migration Manag	5	<input type="checkbox"/>
DMWFPROCES	Workflow process definition migration object fc	6	<input type="checkbox"/>
DMINBOUNDCO	E-mail listener definition migration object for Mig	7	<input type="checkbox"/>

Figure 6-6 Migration objects for the RB_BPM group

6.1.6 Package definitions

At this point, the components that are needed to assemble and organize the configuration content is complete. The next step in this process is to define the Snapshot migration package using the Migration Manager application. A single package definition is sufficient to migrate all of the configuration content that is identified for this use case. The tasks to create a migration package definition are not described here, because they are well documented through the product documentation. The new RB_BPM migration group is the only group in this package definition.

We name the new migration package RB_BPM, also. After the RB_BPM package definition is saved, SQL filter criteria can be associated with each of the object structures in the group, as needed, to identify the subset of the configurations requiring migration.

The following example illustrates how an workflow must be reviewed to determine the appropriate SQL criteria:

1. The modified connector's action requires a communication template, RB_REQ_APP, which was added with the corresponding new role RB_REQUEST.
2. In the Workflow Designer application, the Workflow field specifies the process name of the workflow that was created called REDBOOK_WF, which will be included in the SQL clause. When specifying the workflow name, we also specify the status ACTIVE, because a workflow can have many revisions, but only one version is active.
3. The SQL criteria for the various object structures are specified using the "Set Where Clause" function of the Migration Manager application. The Where clauses specified identify the particular configuration records to export. Where no export is required, the clause 1=0 or 1=2 is specified.
4. You can now save the package definition, because you have entered the SQL criteria necessary to migrate the workflow, as shown in Figure 6-7.

Migration Object	Object	Where Clause	
DMACTION	ACTION	1=2	
DMACTIONGRO	ACTIONGROUP	1=2	
DMROLE	MAXROLE	maxrole = 'RB_REQUEST'	
DMCOMMTEMPL	COMMTEMPLATE	templateid = 'RB_REQ_APP'	
DMESCALATIOI	ESCALATION	1=2	
DMWFPROCES	WFPROCESS	processname = 'RB_BPM'	
DMINBOUNDCO	INBOUNDCOMMCFG	1=2	

Figure 6-7 Package Where clause SQL queries

6.1.7 Deployment

After the package definition has been saved and approved, the physical package can be created, distributed, and deployed using the Migration Manager application. The deployment of workflows and their associated configurations can be performed in a live environment without the need to restart the application server or turn on Admin mode.

6.1.8 Deployment considerations

Before migrating Workflows, it is important to review the following migration options.

Version

It is common practice to develop and test multiple versions of a workflow process. However, you will migrate the chosen active version, and the version will reset to “1” in the Target environment.

Activation

Although a manual activation step is necessary, this step also gives you the opportunity to verify the contents of the workflow and to only activate the workflow at the appropriate point in time in the Target environment.

6.2 Migrating workflows and all their related content

In this scenario, we describe the migration of an existing workflow that has been changed and that contains sub-processes that are called from a main process. We demonstrate how to include the sub-processes and all of the workflow components in the same migration package.

6.2.1 Requirements

For this example, the main process ISMACCEPT and all of its sub-processes have been changed extensively. There have been many actions, roles, and communication templates changed and added to the process to meet the change requirements.

However, there has been no documentation of what exactly was changed, and the Migration Manager Change package was not used to track these configuration changes. The only known fact is that the workflow process ISMACCEPT has been reconfigured and needs to be migrated.

A repeatable process is required to reliably (without any errors or failures) and accurately (without missing any required content) migrate workflow process changes.

6.2.2 Solution

After reviewing the migration requirements for this case scenario, we identified an approach so that all the related configuration content is migrated using a Snapshot migration package. The package is configured to use the standard migration group for business process management (BPM), which is installed by Tivoli's process automation engine.

From a content perspective, the migration of modified workflow and related content requires the migration of the following configuration objects:

- ▶ Roles
- ▶ Actions
- ▶ Communication templates
- ▶ Workflow processes and sub-processes

6.2.3 Configuration applications used for this solution

Next, we show the configuration applications that are used for this solution and how you can access these applications:

- ▶ Roles
To access the Roles application, click **Go To → System Configuration → Platform Configuration → Roles**.
- ▶ Actions
To access the Actions application, click **Go To → System Configuration → Platform Configuration → Actions**.
- ▶ Communication templates
To access the Communication Templates application, click **Go To → System Configuration → Platform Configuration → Communication Templates**.
- ▶ Workflows
To access the Workflows application, click **Go To → System Configuration → Platform Configuration → Workflow Designer**.

6.2.4 Object structures

The following standard object structures that are provided with the product are used to support the migration requirements:

- ▶ DMACTION
- ▶ DMACTIONGROUP
- ▶ DMROLE

- ▶ DMCOMMTEMPLATE
- ▶ DMWFPROCESS

Resource: For more information about migration object structures, refer to the *Migration Manager Guide*:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.mam.doc_7.1/pdf/mam71_migration_mgr_guide.pdf

6.2.5 Migration groups

After reviewing the change requirements and identifying the object structures, the next step is to define the groups to define the package content.

The standard group BPM, which ships with the product, contains these migration objects. We will create a new group that is based on this group for our scenario.

Create the new group by duplicating the standard BPM group first, leaving all of the defaults, and then, follow these steps:

1. Name the group MYBPM with a description Business Process Modules without dependencies.
2. Remove all dependency groups.
3. Save the group.

Not removing any of the existing objects from the duplicated BPM group allows for future use of the new group to migrate other similar configurations, for example, Escalations.

By removing the dependency groups, we are not including all of the configuration content in these groups. However, dependencies must be always part of the migration process. In this use case scenario, we work on an existing workflow. Therefore, we assume that the configuration on which the workflow elements might depend is already in the Target environment. If, for example, a new table attribute was created or a new person group for a role was added, these objects must be migrated before you attempt to migrate our workflow.

Figure 6-8 on page 150 shows the new migration group.

Migration Group: <input type="text" value="MYBPM"/>	<input type="text" value="BPM without dependency groups"/>	Change By: <input type="text" value="MAXADMIN"/>
* Migration Group Order: <input type="text" value="209"/>	Change Date: <input type="text" value="10/8/10 13:55:44"/>	
Internal? <input type="checkbox"/>		

Migration Objects for MYBPM			
Migration Object	Description	Migration Object Order	Internal
DMACTION	Action migration object for Migration Manager	1	<input type="checkbox"/>
DMACTIONGRO	Actiongroup migration object for Migration Man:	2	<input type="checkbox"/>
DMROLE	Role migration object for Migration Manager	3	<input type="checkbox"/>
DMCOMMTEMPL	Communication template migration object for Mi:	4	<input type="checkbox"/>
DMESCALATIOI	Escalation migration object for Migration Manag	5	<input type="checkbox"/>
DMWFPROCES	Workflow process definition migration object fc	6	<input type="checkbox"/>
DMINBOUNDCO	E-mail listener definition migration object for Mig	7	<input type="checkbox"/>

Dependency	
Dependent Migration Group	Description
...No rows to display...	

Figure 6-8 New MYBPM migration group

6.2.6 Package definitions

For this use case scenario, the migration approach is to deploy one Snapshot package.

Create a new package definition that you can name MYTESTS-WF. This package is the Snapshot type and contains the migration group MYBPM.

The next step is to define the SQL criteria within each of the object structures in the group. The only information that we need for this criteria is the name of the main workflow, ISMACCEPT. The conditions for other related workflow dependencies, such as roles, actions, and templates, will use in-line SQL statements that are built to retrieve the information within each object.

Table 6-1 on page 151 lists the Where clause SQL command statement conditions for each object. These conditions result in one or many records of configuration content for each migration object.

Table 6-1 Where clause conditions for MYTESTS-WF package

Migration object	Object	Where clause
DMACTION	ACTION	action in (select action from ACTIONGROUP where action in (select action from WFACTION where processname in ('ISMACCEPT') or processname in (select subprocessname from wsubprocess where processname in ('ISMACCEPT')))) or action in (select member from actiongroup where action in (select action from WFACTION where processname in ('ISMACCEPT') or processname in (select subprocessname from wsubprocess where processname in ('ISMACCEPT'))))
DMACTIONGROUP	ACTIONGROUP	action in (select action from WFACTION where processname in ('ISMACCEPT') or processname in (select subprocessname from wsubprocess where processname in ('ISMACCEPT')))
DMROLE	MAXROLE	maxrole in (select sendtovalue from COMMTEMPLTSENDTO where type='ROLE' and templateid in (select templateid from COMMTEMPLATE where templateid in (select templateid from WFNOTIFICATION where processname in ('ISMACCEPT') or processname in (select subprocessname from wsubprocess where processname in ('ISMACCEPT')))))
DMCOMMTEMPLATE	COMMTEMPLATE	templateid in (select templateid from WFNOTIFICATION where processname in ('ISMACCEPT') or processname in (select subprocessname from wsubprocess where processname in ('ISMACCEPT')))
DMESCALATION	ESCALATION	1=2
DMWFPROCESS	WFPROCESS	(processname in ('ISMACCEPT') and active=1) or processname in (select subprocessname from wsubprocess where processname in ('ISMACCEPT'))
DMINBOUNDCOMMCFG	INBOUNDCOMMCFG	1=2

The conditions for each object generate the precise content for this package. The Where clauses for the objects defined with a logical false statement (1=2) will

generate no results, because escalation or inbound communication content will not be migrated in this particular case.

6.2.7 Deployment

After all package definitions are saved, approved, and activated, the next step is to create and distribute the package to the Target environment. After the package is available to the Target environment, it is time to deploy.

Load and deploy the MYTESTS-WF package in the Target environment. The deployment does not require turning Admin Mode on; however, it is a good idea to turn Admin mode on as part of the change management process.

6.2.8 Deployment considerations

Because you are updating an existing package, there might be transactions under this workflow in the Target environment. The records that are affected will continue using the old version of the workflow.

The new workflow version will not be activated in this case. After the records using the old workflow are out of the workflow process, the new workflow can be activated.

The updated workflow will be deployed as inactive and disabled, and its subprocesses will not be enabled. To complete the deployment, you must disable and deactivate the old versions of the workflows.



Migrating classifications

Business data grows daily. To ease reporting and understand data better, you must classify data according to your business needs. This chapter provides information about migrating classification data via Migration Manager, and it contains the following sections:

- ▶ 7.1, “Scenario” on page 154
- ▶ 7.2, “Configuration applications” on page 155
- ▶ 7.3, “Object structure” on page 157
- ▶ 7.4, “Migration group” on page 158
- ▶ 7.5, “Package definition” on page 159
- ▶ 7.6, “Deployment” on page 160
- ▶ 7.7, “Deployment considerations” on page 161

7.1 Scenario

Classifications are hierarchical data that allows you to group and classify data. Users can report and understand the problem distribution easier with classification data. You can use classifications in many applications. Also, you can define separate attributes for each classification. Attributes are the characteristics of the data. Attributes help improve data consistency and searching for data. For example, consider that there are two classifications. One classification is a service classification `NET.NETWORKSERVICE`, and the other classification is an application server classification named `APP.J2EE.WEBSHERE.WEBSHERESERVER`. You can see the examples of the classification attributes in Table 7-1.

Table 7-1 Example of classification attributes

NET.NETWORKSERVICE classification attributes	APP.J2EE.WEBSHERE.WEBSHERESERVER classification attributes
Service name	Status
Context IP	Product name
Bidirectional (BIDI) flag	Product version

You can search configuration items (CIs) via their attributes. For example, you can search CIs with an Active status and Version 6.0.2.1, according to your classifications.

There might be separate domains attached to these attributes to provide data consistency. You can refer to Chapter 2, “Migrating data dictionary” on page 29 for more information.

You must create classifications to group data according to your grouping needs.

7.1.1 Requirements

Classification data is hierarchical data. Classifications can have a maximum of one parent, but they might have infinite child classifications. Classifications must be transferred from the development environment to the production environment sequentially. Classifications that will be promoted must be migrated without an interruption.

7.1.2 Solution

You cannot use the predefined DMCLASSIFICATION object structure to migrate classification data, because classification data has a child and parent hierarchy, and this object structure is not enough for a hierarchical data structure. Child data and parent data might have been created in random order in the Source environment, which can lead to the random export order of classifications. If the child classification is exported before the parent classification, it will cause an error, because the migration process is unable to find the parent classification in the database while processing the child classification.

To prevent this situation, create a new object structure to migrate the data successfully.

Make the new object structure recursive. It must begin from the top-level classification. When the Migration Manager processes the top level, it then processes the next level until there is no lower level of classification.

Classification data can have an organization, site, or person as the owner group and a service group attached to itself. Migrate these groups before migrating classification data. Otherwise, you can have validation errors while migrating classifications.

7.2 Configuration applications

You can define, manage, and delete classifications in the Classifications application. Reach the Classifications application by navigating to **Go To** → **Administration** → **Classifications**.

Figure 7-1 on page 156 shows an example of a classification hierarchy.

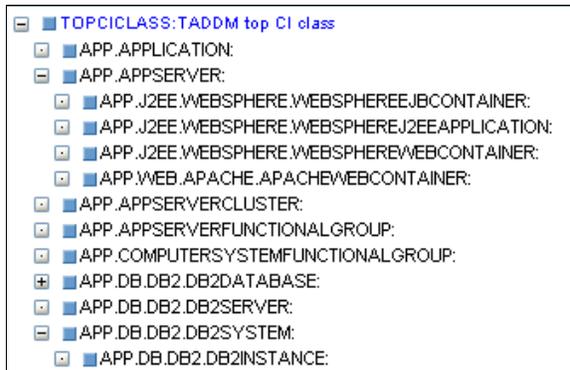


Figure 7-1 Example of a classification hierarchy

Before migrating classifications, be sure to migrate organizations, person groups, domains, measure units, and service groups if you have used these objects in classification or attribute definitions.

Figure 7-2 illustrates the supporting objects.

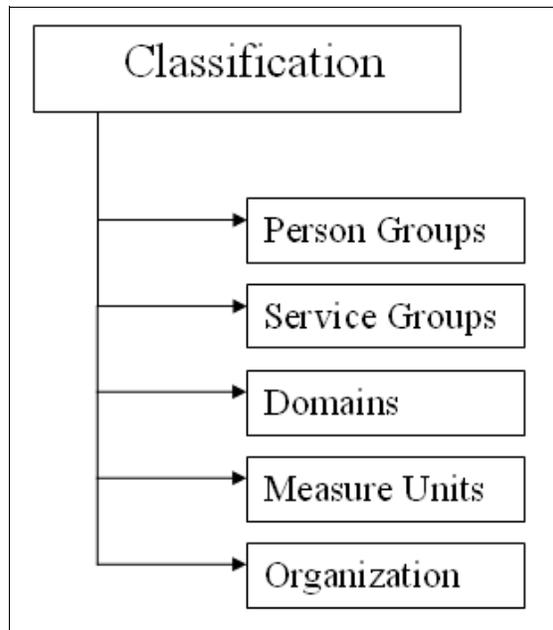


Figure 7-2 Supporting objects of classification data

7.3 Object structure

You need to duplicate and change the DMCLASSIFICATION object structure that shipped with the product to migrate classification data.

You can see the predefined object structure named DMCLASSIFICATION in Figure 7-3.

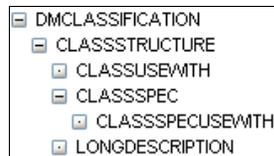


Figure 7-3 DMCLASSIFICATION object structure

To duplicate the object structure, navigate to **Go To** → **System Configuration** → **Migration** → **Object Structures**.

Select the DMCLASSIFICATION object structure, and select Duplicate Object Structure in the Select Action menu, as shown in Figure 7-4.

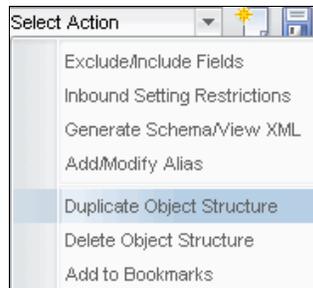


Figure 7-4 Duplicate Object Structure option in the Select Action menu

When you duplicate the object structure, give it a name, such as MYCLASSIFICATION, and then, select the Self Reference option to make the object structure recursive, as shown in Figure 7-5.

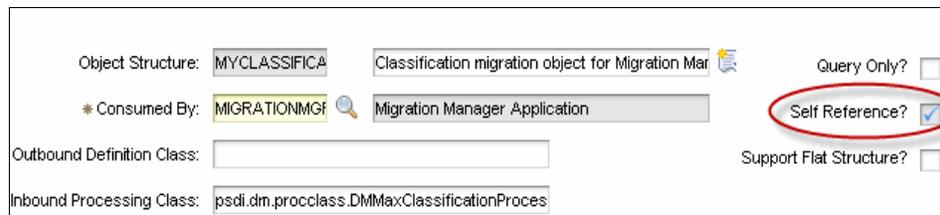


Figure 7-5 Self Reference option in the object structure

When you select the check box, you see that the relationship field next to the first object in the object structure becomes editable. Type CHILDREN to the relationship field next to the first object, as shown in Figure 7-6.

Object	Parent Object	Object Location Path	Relationship
CLASSSTRUCT		CLASSSTRUCTURE	CHILDREN
CLASSUSEWITH	CLASSSTRUCT	CLASSSTRUCTURE/CLASSUSEWITH	CLASSUSEWITH
CLASSSPEC	CLASSSTRUCT	CLASSSTRUCTURE/CLASSSPEC	CLASSSPEC
CLASSSPECUSEWITH	CLASSSPEC	CLASSSTRUCTURE/CLASSSPEC/CLASSSPEC	CLASSSPECUSEWITH
LONGDESCRIP1	CLASSSTRUCT	CLASSSTRUCTURE/LONGDESCRIPTION	LONGDESC

Figure 7-6 Relationship name

Now, you have the correct object structure to create a migration group.

Relationships: The child relationship is a relationship from the classtructure object to the same object. It is used to find the child records for a given classtructure. The Where clause of the relationship is parent=:classtructureid.

Figure 7-7 shows the new object structure.

```

MYCLASSIFICATION
├── CLASSSTRUCTURE
│   ├── CLASSUSEWITH
│   ├── CLASSSPEC
│   ├── CLASSSPECUSEWITH
│   └── LONGDESCRIPTION

```

Figure 7-7 MYCLASSIFICATION object structure

7.4 Migration group

To migrate classification data, create a new migration group that includes the new object structure:

1. Navigate to **Go To** → **System Configuration** → **Migration** → **Migration Groups**.
2. Create a new migration group.
3. Give it a name, such as MYCLASSIFICATION.
4. Add the MYCLASSIFICATION object structure to the migration group.

The migration group has only one migration object, as shown in Figure 7-8.

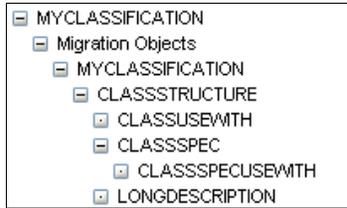


Figure 7-8 Migration group of classification data

This migration group does not depend on other groups. You can directly migrate this group without any group order.

7.5 Package definition

To start the migration process, define the migration package. Create a single migration package and add the MYCLASSIFICATIONGROUP migration group to the package. The remainder of the package definition is described in the *Migration Manager Guide*:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ta.mit.doc_7.1/pdf/mam71_migration_mgr_guide.pdf

After defining the package, create the SQL statement to define the classification data that you want to migrate.

Then, create a statement, similar to the statement that is shown in Figure 7-9.

SQL Expression Builder

Expression: `classificationid='TOPICCLASS'`

Custom Class?

Figure 7-9 Classification migration SQL statement

Warning: You must write the top-level classification to the SQL to migrate all classifications successfully. The Migration Manager will migrate the child classifications automatically.

The Migration Manager validates the SQL statement. There will be an error message prior to saving the package if any SQL statement fails on validation.

With this method, classification data will be migrated from the top level to the bottom level consecutively. The Migration Manager will get all of the classifications recursively. You can see the Where clause statement window in Figure 7-10.

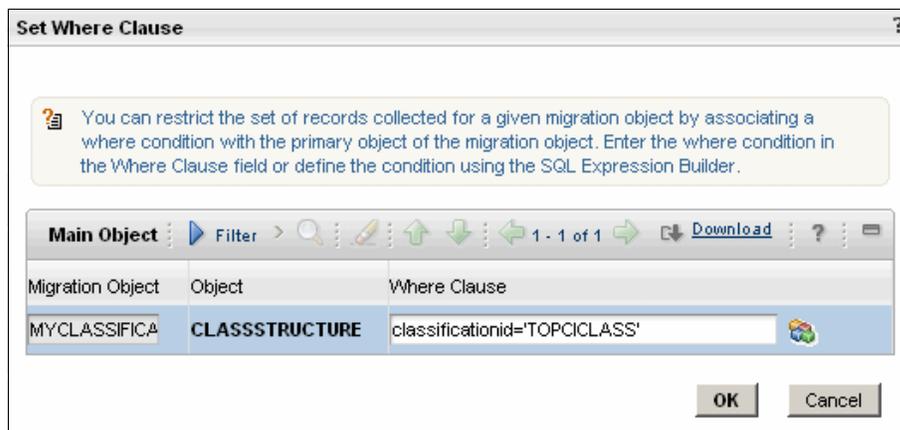


Figure 7-10 Where clause window

Now, you can save the package and activate it.

7.6 Deployment

After creating the package, you must distribute the package to get it to the Target environment. You can distribute the package via file-type distribution or database-type distribution. For more information, refer to Chapter 10, “Common topics” on page 197.

To migrate the data successfully, migrate the myclassification object structure to the Target environment.

Tip: When you migrate classifications, you *do not* need to *switch Admin mode on or restart the server*.

After the deployment to the Target environment, users can start to use the promoted classifications in the new environment.

7.7 Deployment considerations

To migrate new classifications successfully to the Target environment, you and the users must never create classifications manually in the Target environment. All of the data must be created first in the Source environment and then migrated to the Target environment. Otherwise, you can have index problems when migrating the classification data to the Target environment.

Snapshot migration: You must always use Snapshot migration for classification migration. The Migration Manager cannot handle the hierarchical structure when migrating classifications with the Change package.

Before deployment, be sure that the object structure includes the HIERARCHYPATH non-persistent field in the CLASSSTRUCTURE object. Otherwise, you get the BMXAA5592E - Inbound data does not have a valid hierarchy path error. The Migration Manager relies on this attribute to determine if inserts or updates need to be performed.

The Migration Manager does not check whether the processing action is Replace or AddChange in a classification migration when using Snapshot migration. If the Migration Manager finds the same HIERARCHYPATH in the Target environment, it updates the record.



Migrating service offering content

This chapter provides you with an overview of the process steps, issues, and concerns that we experienced when migrating the IBM Service Offering content in the IBM Tivoli Service Request Manager product.

Because the service offerings are technically content-type data, you must move this data from one environment to another environment using the Integration Framework. However, IBM provides specialized knowledge to the Integrated Service Management (ISM) marketplace by embedding it in the tool with specialized object structures to migrate the IBM-delivered content.

This chapter presents scenarios in which both the Source and Target contain the IBM content. If your Target environment does not contain the same product installations as your Source environment with respect to the content packages, refer to Chapter 11, “Troubleshooting” on page 239 for further guidance about how to manage these situations.

We provide only two scenarios with the understanding that the process provided will enable you to interpret what is required for other content packages.

This chapter has the following sections:

- ▶ 8.1, “Service offering scenario” on page 165
- ▶ 8.2, “Requirements” on page 165

- ▶ 8.3, “Solution” on page 165
- ▶ 8.4, “Configuration applications” on page 168
- ▶ 8.5, “Object structures” on page 170
- ▶ 8.6, “Migration groups” on page 172
- ▶ 8.7, “Package definitions” on page 173
- ▶ 8.8, “Deployment” on page 176

8.1 Service offering scenario

The Service Offering content that IBM delivers with the Tivoli Service Request Manager uses an extensive number of objects in the database. Migration of these object structures requires attention regarding which service offerings you need to migrate. The various migration groups that make up the predefined migration package template include a large number of various configurations, as well as content-type object structures. Managing this migration requires you to input the proper data and to include (or exclude) it by managing the Where clauses in the package definition.

8.2 Requirements

Migrate the additional and modified service offerings in a Tivoli Service Request Manager implementation using the sample migration packages that are provided as templates for the various service offerings that are provided or developed.

8.3 Solution

You can access the template migration package by navigating to the Migration Package application by selecting **Go To** → **System Configuration** → **Migration** → **Migration Manager**, as depicted in Figure 8-1 on page 166.

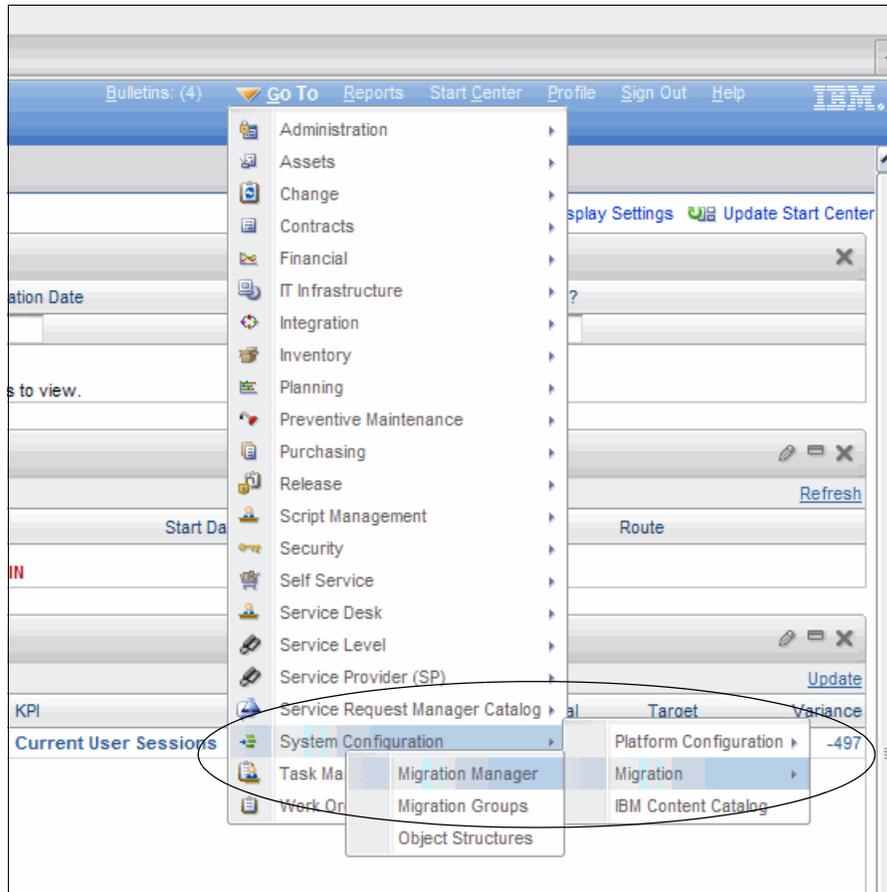


Figure 8-1 Menu navigation to the Migration Manager application

For this scenario, we need the PMSC72_ServiceTemplate package. Because the name of this package can differ from your exact environment, you can search for it, as demonstrated in Figure 8-2 on page 167.

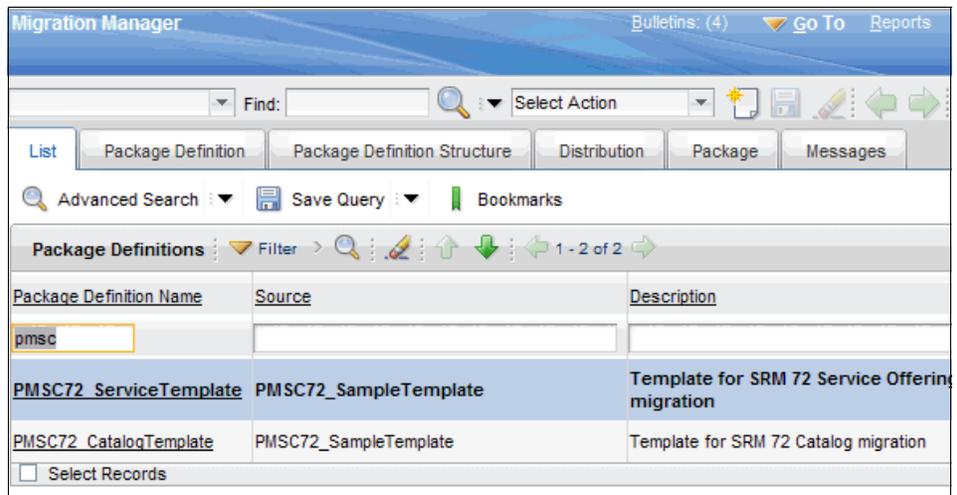


Figure 8-2 Service Template Offering

When you examine the Migration Groups structure, you can see that there are many things to consider about this scenario, as illustrated in Figure 8-3.

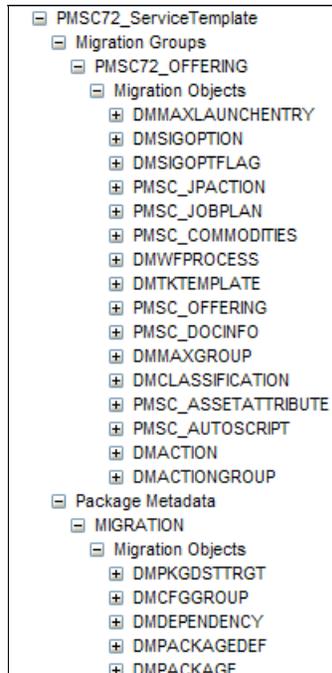


Figure 8-3 Migration group for service offering template

8.4 Configuration applications

In order to understand the Migration Group, Table 8-1 illustrates the various applications that make up a service offering.

Table 8-1 Configuration applications

Application name	Object structure
Launch in Context	<ul style="list-style-type: none"> [-] DMMAXLAUNCHENTRY <ul style="list-style-type: none"> [-] MAXLAUNCHENTRY <ul style="list-style-type: none"> [-] LONGDESCRIPTION [-] MAXLECONTEXT
Job Plans	<ul style="list-style-type: none"> [-] PMSC_JOBPLAN <ul style="list-style-type: none"> [-] JOBPLAN <ul style="list-style-type: none"> [-] JOBTASK <ul style="list-style-type: none"> [-] JOBTASKSPEC [-] LONGDESCRIPTION [-] JOBITEM [-] JOBPLANCLASS [-] JOBPLANSPEC [-] JOBSERVICE [-] JOBLABOR [-] JOBMATERIAL [-] JOBT TOOL [-] JPASSETSPLINK [-] JPTASKRELATION
View Documents	<ul style="list-style-type: none"> [-] PMSC_DOCINFO <ul style="list-style-type: none"> [-] DOCINFO <ul style="list-style-type: none"> [-] DOCLINKS
Offerings	<ul style="list-style-type: none"> [-] PMSC_OFFERING <ul style="list-style-type: none"> [-] PMSCOFFERING <ul style="list-style-type: none"> [-] PMSCOFFERINGAUTH [-] PMSCITEMSPEC [-] PMSCOFFDIALOG [-] IMLIB [-] LONGDESCRIPTION
Actions Note: There are two Object Structures for this application.	<ul style="list-style-type: none"> [-] PMSC_JPACTION <ul style="list-style-type: none"> [-] ACTION <ul style="list-style-type: none"> [-] LONGDESCRIPTION [-] DMACTION <ul style="list-style-type: none"> [-] ACTION <ul style="list-style-type: none"> [-] LONGDESCRIPTION [-] DMACTIONGROUP <ul style="list-style-type: none"> [-] ACTIONGROUP

Application name	Object structure
Classifications Classifications (SP)	<div style="border: 1px solid black; padding: 5px;"> <ul style="list-style-type: none"> [-] DMCLASSIFICATION <ul style="list-style-type: none"> [-] CLASSSTRUCTURE <ul style="list-style-type: none"> [-] CLASSUSEWITH [-] CLASSSPEC [-] LONGDESCRIPTION </div>
Workflow Designer	<div style="border: 1px solid black; padding: 5px;"> <ul style="list-style-type: none"> [-] DMWFPROCESS <ul style="list-style-type: none"> [-] WFPROCESS <ul style="list-style-type: none"> [-] LONGDESCRIPTION [-] WFNODE <ul style="list-style-type: none"> [-] LONGDESCRIPTION [-] WFTASK [-] WFASSIGNMENT <ul style="list-style-type: none"> [-] LONGDESCRIPTION [-] WFSTART [-] WFSTOP [-] WFINPUT [-] WFACTION [-] WFWAITLIST [-] WFINTERACTION <ul style="list-style-type: none"> [-] LONGDESCRIPTION [-] WFCONDITION [-] WFSUBPROCESS [-] WFNOTIFICATION </div>
Ticket Templates	<div style="border: 1px solid black; padding: 5px;"> <ul style="list-style-type: none"> [-] DMTKTEMPLATE <ul style="list-style-type: none"> [-] TKTEMPLATE <ul style="list-style-type: none"> [-] TKTEMPLATESPEC [-] TKTEMPLACTIVITY [-] TKTEMPLACTYSPEC [-] LONGDESCRIPTION </div>
Service Groups	<div style="border: 1px solid black; padding: 5px;"> <ul style="list-style-type: none"> [-] PMSC_OFFERING <ul style="list-style-type: none"> [-] PMSCOFFERING <ul style="list-style-type: none"> [-] PMSCOFFERINGAUTH [-] PMSCITEMSPEC [-] PMSCOFFDIALOG [-] IMGLIB [-] LONGDESCRIPTION </div>
Automation Scripts	<div style="border: 1px solid black; padding: 5px;"> <ul style="list-style-type: none"> [-] PMSC_AUTOSCRIPT <ul style="list-style-type: none"> [-] AUTOSCRIPT </div>

You can use the following navigation links to access these applications:

- ▶ **Launch In Context:** Select **Go To** → **System Configuration** → **Platform Configuration** → **Launch in Context**.
- ▶ **Job Plans:** Select **Go To** → **Planning** → **Job Plans**.

- ▶ **View Documents:** Select **Go To** → **Administration** → **View Documents**.
- ▶ **Offerings:** Select **Go To** → **Service Request Manager Catalog** → **Offerings**.
- ▶ **Actions:** Select **Go To** → **System Configuration** → **Platform Configuration** → **Actions**.
- ▶ **Classifications (or Classifications (SP)):** Select **Go To** → **Administration** → **Classifications**.
- ▶ **Workflow Designer:** Select **Go To** → **System Configuration** → **Platform Configuration** → **Workflow Designer**.
- ▶ **Ticket Templates:** Select **Go To** → **Service Desk** → **Ticket Templates**.
- ▶ **Service Groups:** Select **Go To** → **Service Request Manager Catalog** → **Service Groups**.
- ▶ **Automation Scripts:** Select **Go To** → **Script Management** → **Automation Scripts**.

8.5 Object structures

This section deals with only those object structures that are unique to the Tivoli Service Request Manager (that are not distributed with Tivoli's process automation engine). These object structures are unique to migrating service offering content:

- ▶ PMSC_JOBPLAN
- ▶ PMSC_OFFERING
- ▶ PMSC_DOCINFO

8.5.1 PMSC_JOBPLAN

This object structure has classes for inbound and outbound processing that are unique to the service offering content. These classes are not used for other object structures or purposes. Figure 8-4 demonstrates the identified classes.

Outbound Definition Class:	com.ibm.ism.pmsc.migration.PMSC_JOBPLAN_
Inbound Processing Class:	com.ibm.ism.pmsc.migration.PMSC_JOBPLAN_

Figure 8-4 Custom Tivoli Service Request Manager classes for Job Plan migration

Additionally, IBM delivers a new relationship for the Tivoli Service Request Manager product offering to enable you to migrate the service offerings properly. Figure 8-5 shows this relationship.

Object	Parent Object	Object Location Path	Relationship
JOBPLAN		JOBPLAN	PMSC_NESTEDJOBPLAN

Figure 8-5 PMSC nested job plan relationship

We encourage you to review the definition separately using the Database Configuration application.

8.5.2 PMSC_OFFERING

The PMSC_Offering object structure requires that you understand that additional classes and relationships are used separately from the standard product. See Figure 8-6.

Outbound Definition Class:	com.ibm.ism.pmsc.migration.PMSC_OffAndCat
Inbound Processing Class:	com.ibm.ism.pmsc.migration.PMSC_OffAndCat

Figure 8-6 PMSC_OFFERING specialized classes

The relationships for this object structure are listed in Figure 8-7.

Object	Parent Object	Object Location Path	Relationship
PMSCOFFERING		PMSCOFFERING	
PMSCOFFERING	PMSCOFFERING	PMSCOFFERING/PMSCOFFERINGAUTH	PMSCOFFERINGAUTH
PMSCITEMSPEC	PMSCOFFERING	PMSCOFFERING/PMSCITEMSPEC	PMSCITEMSPEC
PMSCOFFDIALOG	PMSCOFFERING	PMSCOFFERING/PMSCOFFDIALOG	PMSCOFFDIALOG
IMGLIB	PMSCOFFERING	PMSCOFFERING/IMGLIB	IMGLIB

Figure 8-7 PMSCOFFERING object structure relationships

8.5.3 PMSC_DOCINFO

The Tivoli Service Request Manager Service Offering content provides you with the following Classes for the PMSC_DOCINFO object structure, as shown in Figure 8-8 on page 172.

Outbound Definition Class:	<code>com.ibm.ism.pmsc.migration.PMSC_Doclnfo_Ou</code>
Inbound Processing Class:	<code>com.ibm.ism.pmsc.migration.PMSC_Doclnfo_Inf</code>

Figure 8-8 *PMSC_DOCINFO specialized classes*

These classes enable the product to properly process content-type data with the migration of configuration data.

Important: This object structure is tightly coupled with the PMSCOFFERING object. If you try to use it for docinfo objects that are not owned by PMSCOFFERING, it will fail in the creation phase. The Where clause that is provided with the product is:

```
docinfofid in (select docinfofid from doclinks where  
ownertable='PMSCOFFERING' and ownerid in (select itemid from  
pmscoffering where itemnum='SERVICENAME'))
```

8.6 Migration groups

The migration group for the service offering is the PMSC_OFFERING migration group, which is provided for you to duplicate and modify only as necessary. Note the object structure order, as shown in Figure 8-9 on page 173.

Object	Description	Migration Object Order
AUNC >>	Launch entry migration object for Migration Ma	5
TION >>	Signature option migration object for Migration I	10
TFLAI >>	Advanced signature option migration object for	15
ITOSC >>	PMSC Automation Scripts	20
SSETA >>	Asset attribute migration object	25
SIFICA >>	Classifications migration object	30
ACTIC >>	Action migration object for Migration Manager	35
BPLA >>	Jobplan migration object for Migration Manager	40
MMOI >>	Commodities migration object	45
N >>	Action migration object for Migration Manager	50
NGRC >>	Action Group migration object for Migration Ma	55
OCES: >>	Workflow process definition migration object fo	60
PLAT >>	Ticket Template migration object	65
ROUP >>	Security group migration object	70
FFERIN >>	SRM 72 Service Offering migration object	75

Figure 8-9 Service offering migration group object order

Note that the order is important, but also that there are gaps in the numbering order to allow you to easily add any object structures that you require.

8.7 Package definitions

This section addresses the majority of the work for you to perform. The SQL definitions that IBM supplies require you to modify the definition to migrate the client-modified and client-added offerings.

Access the package

Navigate to the Migration Manager application (**Go To** → **System Configuration** → **Migration** → **Migration Manager**) and search for PMSC72_ServiceTemplate. When found, click the **Package Definition** tab, and then, click the icon, as displayed in Figure 8-10 on page 174.



Figure 8-10 Migration package definition

Figure 8-11 displays one of the SQL criteria statements for the various packages.

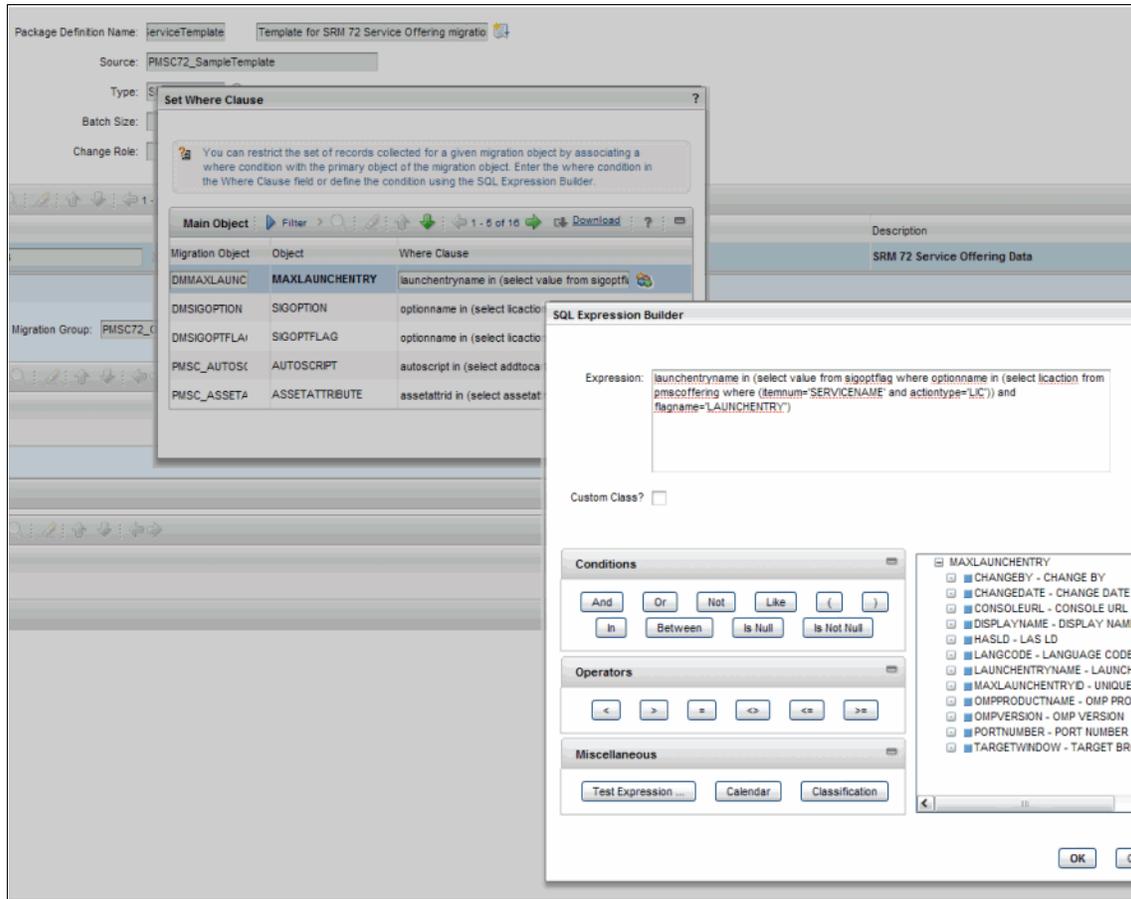


Figure 8-11 Set Where clause dialog box with MAXLAUNCHENTRY selected

8.7.1 Defining the SQL definitions

Access the Expression Builder by clicking the expression builder icon on the far right side of each row. Now, manipulate the SQL *for each migration object* in the group (Example 8-1).

Example 8-1 Sample caption

```
launchentryname in (select value from sigoptflag where optionname in  
(select licaction from pmscoffering where (itemnum='SERVICENAME' and  
actiontype='LIC')) and flagname='LAUNCHENTRY')
```

In this code, replace 'SERVICENAME' with the individual itemnum values or another construct that satisfies your record selection criteria.

The SQL syntax in Example 8-2 enables you to select several offerings to migrate.

Example 8-2 Sample caption

```
launchentryname in (select value from sigoptflag where optionname in  
(select licaction from pmscoffering where (itemnum in  
( 'SERVICENAME1', 'SERVICENAME2', 'SERVICENAME3', 'SERVICENAME4') and  
actiontype='LIC')) and flagname='LAUNCHENTRY')
```

In this fashion, you choose to migrate four service offerings. This criteria *must match* the other object structures' criteria to ensure the correct migration.

Example 8-3 demonstrates how to migrate all of the offerings for this specific migration object.

Example 8-3 Sample caption

```
launchentryname in (select value from sigoptflag where optionname in  
(select licaction from pmscoffering where (actiontype='LIC')) and  
flagname='LAUNCHENTRY')
```

Criteria must match: We emphasize that *the selection criteria must match* from one object structure to another object structure to ensure that the correct record sets are migrated. The syntax of the SQL and the various artifacts that are used to create the record set might vary, but the *criteria* must be the same.

8.8 Deployment

After creating the package, you must distribute the package to get it to the Target environment. Use file distribution. *The Migration Manager Guide* explains this procedure.

Migration: The migration of service offering does not require *admin mode* or *server restart*.

After the deployment to the Target environment, users can start to use the service offering in the Target environment.

8.8.1 Deployment considerations

The primary considerations for this particular migration package are that you must ascertain that there are changes in the migration groups' artifacts that the content requires in order for the Migration Manager to migrate the content properly. You must also ascertain that the changes have been made to the content that you need to move from one environment to the next environment.

If either of these conditions is not true, do not migrate the content, because the content is installed by the IBM product installer when the system is first set up.

Because this material is content-based material, you and your implementation team must carefully consider if the content is required for unit testing and integration testing in order for production to assimilate the changes.

If business processes require the content to work properly, this chapter is important for you.



Migrating a service catalog

This chapter describes the best practices that are associated with migrating a service catalog.

The *service catalog* consists of numerous standard database objects, which together form a core component of the Tivoli Service Request Manager. Service catalogs are made up of service offerings, grouped logically. A typical example of a service catalog is an IT service catalog, which might include server hardware and email account creation service offerings.

The Tivoli Service Request Manager solution is delivered with a standard pre-formatted package definition template that is complete with configurable Where clauses to include and exclude data in the service catalog migration.

This chapter has the following sections:

- ▶ 9.1, “Requirements” on page 178
- ▶ 9.2, “Solution” on page 178
- ▶ 9.3, “Configuration applications” on page 180
- ▶ 9.4, “Object structures” on page 181
- ▶ 9.5, “Migration groups” on page 187
- ▶ 9.6, “Package definitions” on page 188
- ▶ 9.7, “Deployment” on page 193
- ▶ 9.7.1, “Deployment considerations” on page 194

9.1 Requirements

Service catalogs are defined in a development environment and subsequently promoted to test and production environments. As standard, Tivoli Service Request Manager is delivered with the two migration groups that are required to migrate service catalogs. In Chapter 8, “Migrating service offering content” on page 163, we explored service offering migration. The service catalog consists of service offerings; therefore, we can use the same PMSC72_OFFERING migration group, in addition to the PMSC72_CATALOG migration group.

The predefined packages migrate data that is relevant to the following concepts:

- ▶ Launch in context
- ▶ Signature options
- ▶ Actions (job plan)
- ▶ Job plans
- ▶ Commodities
- ▶ Workflow processes
- ▶ Ticket templates
- ▶ Offerings
- ▶ Document information
- ▶ Security groups
- ▶ Classifications
- ▶ Asset attributes
- ▶ Automated scripts
- ▶ Actions
- ▶ Action groups
- ▶ Security groups
- ▶ Service catalog

9.2 Solution

The PMSC72_CatalogTemplate Package Definition does not require any changes to the object structures or migration objects. You are required to configure to the predefined Where clauses that are associated with the object structures to specify the service catalog (CATALOGNAME) to migrate.

Access the template Migration Package by navigating to the Migration Package application by selecting **Go To** → **System Configuration** → **Migration** → **Migration Package**, as depicted in Figure 9-1 on page 179.

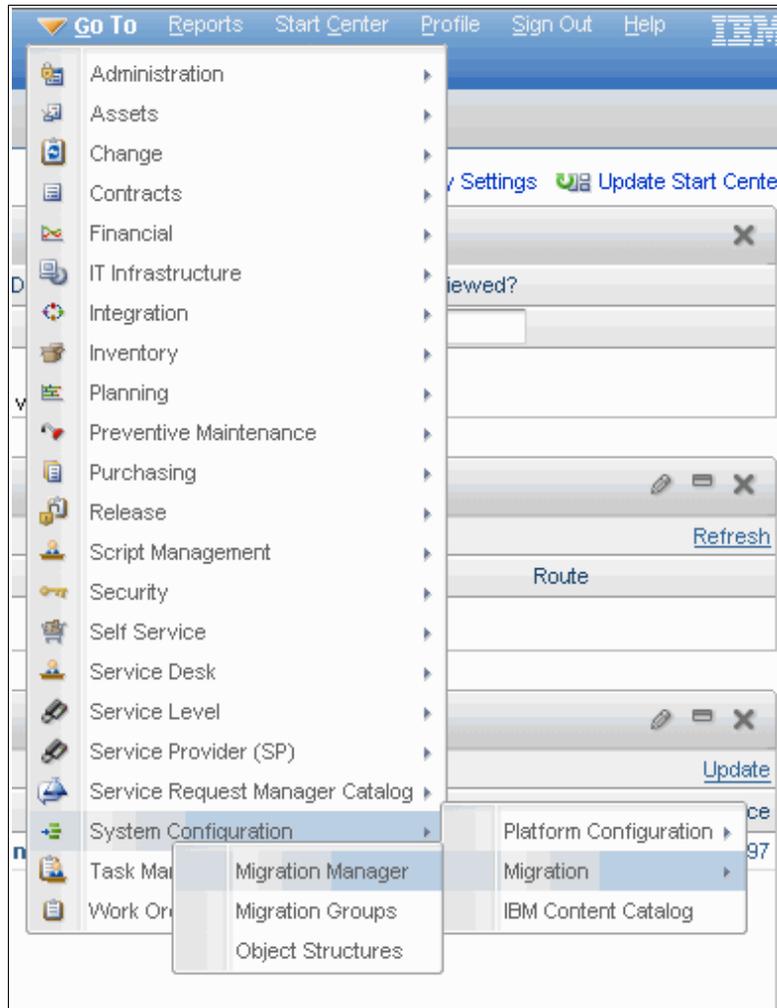


Figure 9-1 Menu navigation to the Migration package application

Use the migration package template that ships with the product, as depicted in Figure 9-2 on page 180, to migrate a service catalog.

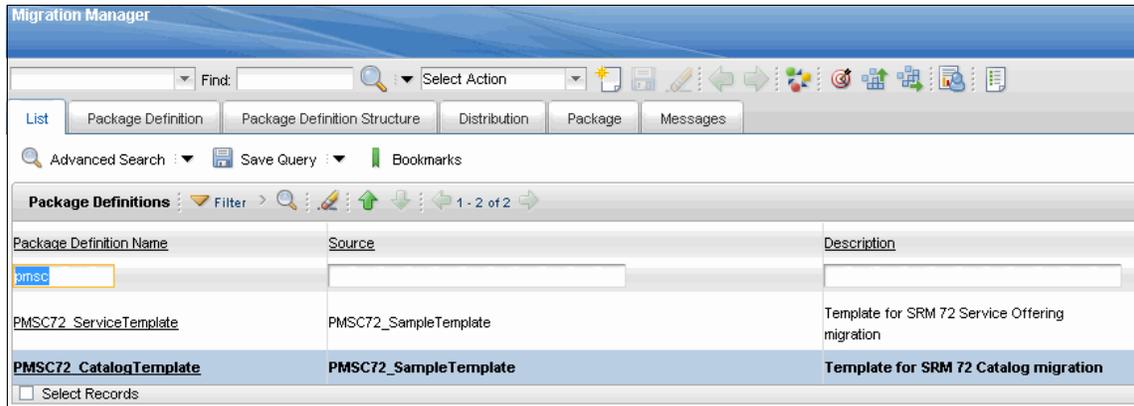


Figure 9-2 Service catalog package definition

The PMSC72_CatalogTemplate consists of the PMSC72_CATALOG and PMSC72_OFFERING migration groups, as illustrated in Figure 9-3.

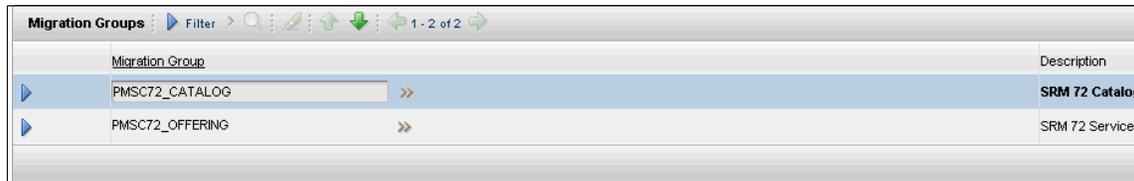


Figure 9-3 PMSC72_CatalogTemplate migration groups

9.3 Configuration applications

In the previous scenario of this chapter, service offering migration, we explored the configuration applications that are associated with an individual service offering (PMSC72_OFFERING migration object). Similarly, the Services Catalog uses the configuration applications that are used to define a service offering with the addition of the Security Groups and Catalogs applications. Table 9-1 on page 181 illustrates the additional configuration applications and associated object structures.

Table 9-1 Configuration applications

Application name	Object structure
Security Groups	<ul style="list-style-type: none"> [-] DMMAXGROUP <ul style="list-style-type: none"> [-] MAXGROUP <ul style="list-style-type: none"> [-] LONGDESCRIPTION [-] SITEAUTH [-] APPLICATIONAUTH [-] GLAUTH [-] SECURITYRESTRICT
Catalogs	<ul style="list-style-type: none"> [-] PMSC72_CATALOG <ul style="list-style-type: none"> [-] PMSCCATALOG <ul style="list-style-type: none"> [-] PMSCCATALOGAUTH [-] PMSCCATALOGOFFMAP [-] IMGLIB

9.3.1 Security groups

The Security Groups application is accessed from the appropriate product Start Center by selecting **Go To** → **Security** → **Security Groups**.

9.3.2 Catalogs

The Catalogs application is accessed from the appropriate product Start Center by selecting **Go To** → **Service Request Manager Catalog** → **Catalogs**.

9.4 Object structures

The following object structures are delivered with the PMSC72_CatalogTemplate package definition that ships with the product.

The DMMAXGROUP object structure, as part of the PMSC72_CATALOG migration group, supports the service catalog migration. Figure 9-4 on page 182 shows the contents of the DMMAXGROUP object structure.

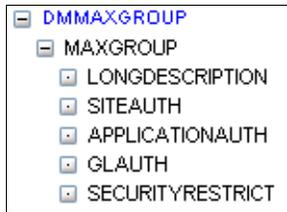


Figure 9-4 *DMMAXGROUP* object structure

The PMSCCATALOG object structure, as part of the PMSC72_CATALOG migration group, supports the service catalog migration. Figure 9-5 shows the contents of the PMSCCATALOG object structure.

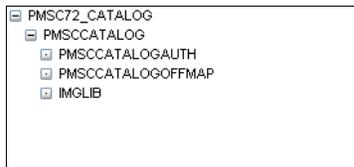


Figure 9-5 *PMSCCATALOG* object structure

The DMMAXLAUNCHENTRY object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-6 shows the contents of the DMMAXLAUNCHENTRY object structure.

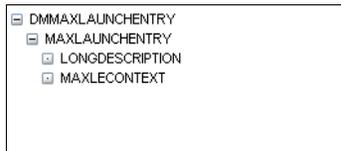


Figure 9-6 *DMMAXLAUNCHENTRY* object structure

The DMSIGOPTION object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 7 shows the contents of the DMSIGOPTION object structure.

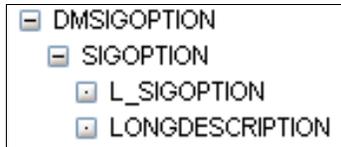


Figure 9-7 *DMSIGOPTION* object structure

The DMSIGOPTFLAG object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-8 shows the contents of the DMSIGOPTFLAG object structure.



Figure 9-8 DMSIGOPTFLAG object structure

The PMSC_JPACTION object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-9 shows the contents of the PMSC_JPACTION object structure.

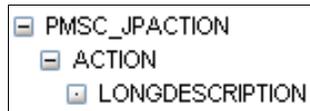


Figure 9-9 PMSC_JPACTION object structure

The PMSC_JOBPLAN object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-10 shows the contents of the PMSC_JOBPLAN object structure.

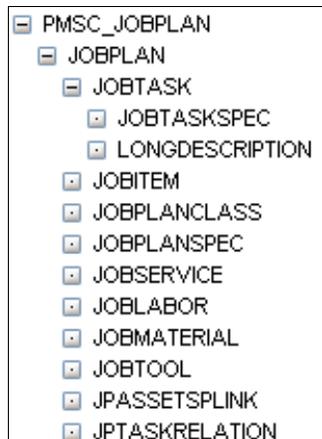


Figure 9-10 PMSC_JOBPLAN object structure

The PMSC_COMMODITIES object structure, as part of the PMSC72_OFFERING migration group, supports service catalog migration. Figure 9-11 on page 184 shows the contents of the PMSC_COMMODITIES object structure.

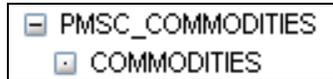


Figure 9-11 *PMSC_COMMODITIES* object structure

The DMWFPROCESS object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-12 shows the contents of the DMWFPROCESS object structure.

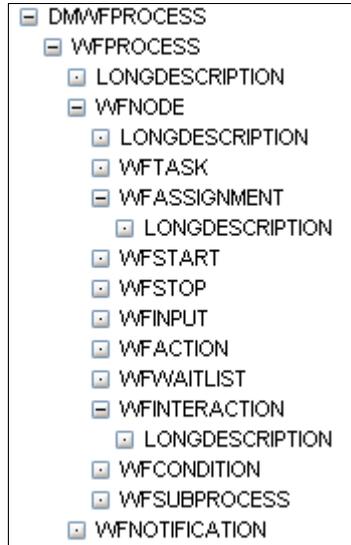


Figure 9-12 *DMWFPROCESS* object structure

The DMTKTEMPLATE object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-13 shows the contents of the DMTKTEMPLATE object structure.

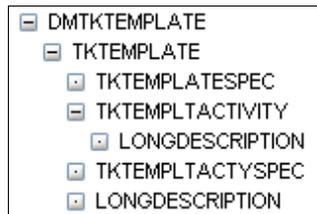


Figure 9-13 *DMTKTEMPLATE* object structure

The PMSC_OFFERING object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-14 shows the contents of the PMSC_OFFERING object structure.

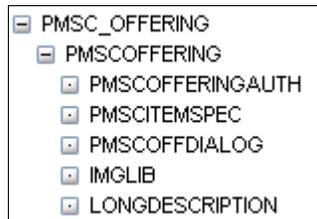


Figure 9-14 PMSC_OFFERING object structure

The PMSC_DOCINFO object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-15 shows the contents of the PMSC_DOCINFO object structure.

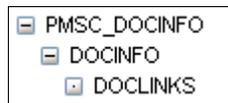


Figure 9-15 PMSC_DOCINFO object structure

The DMMAXGROUP object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-16 shows the contents of the DMMAXGROUP object structure.

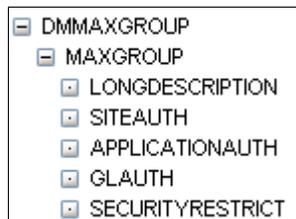


Figure 9-16 DMMAXGROUP object structure

The DMCLASSIFICATION object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-17 on page 186 shows the contents of the DMCLASSIFICATION object structure.

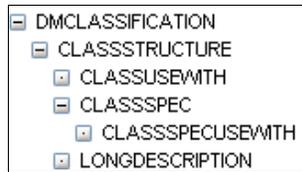


Figure 9-17 DMCLASSIFICATION object structure

The PMSC_ASSETATTRIBUTE object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-18 shows the contents of the PMSC_ASSETATTRIBUTE object structure.

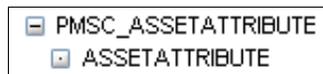


Figure 9-18 PMSC_ASSETATTRIBUTE object structure

The PMSC_AUTOSCRIPPT object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-19 shows the contents of the PMSC_AUTOSCRIPPT object structure.

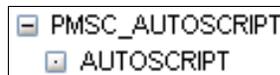


Figure 9-19 PMSC_AUTOSCRIPPT object structure

The DMACTION object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-20 shows the contents of the DMACTION object structure.

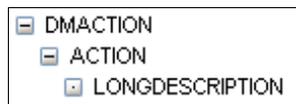


Figure 9-20 DMACTION object structure

The DMACTIONGROUP object structure, as part of the PMSC72_OFFERING migration group, supports the service catalog migration. Figure 9-21 shows the contents of the DMACTIONGROUP object structure.

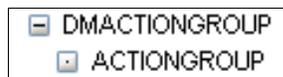


Figure 9-21 DMACTIONGROUP object structure

You are not required to create additional object structures or modify the existing object structures to migrate a service catalog.

In the previous scenario, we explored the special features of the PMSC_OFFERING object structure. These features are also applicable to the service catalog migration scenario.

9.5 Migration groups

This section outlines the migration groups that ship with the product and that migrate a designated service catalog. No modifications are required to the following listed migration groups to perform this migration scenario.

The object structures supporting service catalog migration belong to two migration groups:

- ▶ PMSC72_OFFERING
- ▶ PMSC72_CATALOG

Figure 9-22 on page 188 shows the predefined PMSC72_OFFERING and PMSC72_CATALOG migration groups and their dependencies.

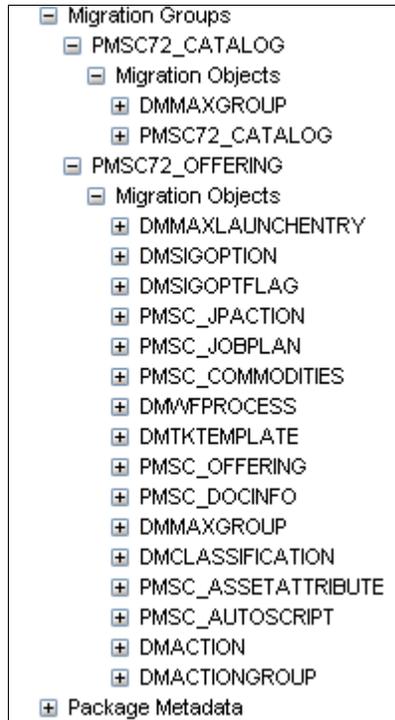


Figure 9-22 PMSC72_OFFERING and PMSC72_CATALOG migration group structure

9.6 Package definitions

This section addresses the majority of the work for you to perform. The Where clauses that IBM supplies require you to modify the value to migrate the designated service catalog.

Access the package

Navigate to the Migration Manager application (**Go To** → **System Configuration** → **Migration** → **Migration Manager**), and then, search for and select **PMSC72_CatalogTemplate**.

The PMSC72_CatalogTemplate consists of two predefined migration objects: PMSC72_OFFERING and PMSC72_CATALOG.

9.6.1 PMSC72_OFFERING

The PMSC72_OFFERING migration object has been delivered with standard Where clauses for each object structure in the migration object. You must edit these Where clauses to specify the name of the catalog to migrate. You insert the Catalog Name, as it appears in the database, in each Where clause where "CATALOGNAME" appears. For example, to migrate the SERVICE CATALOG1 service catalog, Where clause A (Example 9-1) is replaced with Where clause B (Example 9-2).

Example 9-1 Where clause A

```
launchentryname in (select value from sigoptflag where optionname in
(select licaction from pmscoffering where actiontype='LIC' and itemnum
in (select offeringnum from pmsccatalogoffmap where
itemnum='CATALOGNAME')) and flagname='LAUNCHENTRY')
```

Example 9-2 Where clause B

```
launchentryname in (select value from sigoptflag where optionname in
(select licaction from pmscoffering where actiontype='LIC' and itemnum
in (select offeringnum from pmsccatalogoffmap where itemnum='SERVICE
CATALOG1')) and flagname='LAUNCHENTRY')
```

Table 9-2 on page 190 lists the object structures that make up the PMSC72_OFFERING migration object and the associated Where clauses in the order in which they are migrated.

Table 9-2 Predefined PMSC72_OFFERING Where clauses

Migration order	Object structure	Where clause
	DMMAXLAUNCH	launchentryname in (select value from sigoptflag where optionname in (select licaction from pmscoffering where actiontype='LIC' and itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')) and flagname='LAUNCHENTRY')
10	DMSIGOPTION	optionname in (select licaction from pmscoffering where actiontype='LIC' and itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME'))
15	DMSIGOPTIONFLAG	optionname in (select licaction from pmscoffering where actiontype='LIC' and itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')) and flagname='LAUNCHENTRY'
20	PMSC_AUTOSCRIP	autoscript in (select addtocartscript from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')) or autoscript in (select offsubmitscript from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')) or autoscript in (select prepopulationscript from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')) or autoscript in (select validationattr from pmscoffdialog where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME'))
25	PMSC_ASSETATTRIBUTE	assetattrid in (select assetattrid from pmscitemspec where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME'))

Migration order	Object structure	Where clause
30	DMCLASSIFICATION	<p> classtructureid in (select classtructureid from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')) or classtructureid in (select classtructureid from pmsccatalogoffmap where itemnum='CATALOGNAME') or classtructureid in (select classtructureid from tktemplate where templateid in (select templateid from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')))) or classtructureid in (select classtructureid from tktemplactivity where templateid in (select templateid from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')))) or classtructureid in (select classtructureid from jobplan where jpnun in (select jpnun from tktemplactivity where templateid in (select templateid from tktemplate where templateid in (select templateid from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')))))) or classtructureid in (select classtructureid from jobtask where jpnun in (select jpnun from tktemplactivity where templateid in (select templateid from tktemplate where templateid in (select templateid from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')))))))) </p>
35	PMSC_JPACTION	<p> action in (select flowaction from jobplan where jpnun in (select jpnun from tktemplactivity where templateid in (select templateid from tktemplate where templateid in (select templateid from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')))))) or action in (select flowaction from jobtask where jpnun in (select jpnun from tktemplactivity where templateid in (select templateid from tktemplate where templateid in (select templateid from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')))))))) </p>
40	PMSC_JOBPLAN	<p> jpnun in (select jpnun from tktemplactivity where templateid in (select templateid from tktemplate where templateid in (select templateid from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')))) </p>

Migration order	Object structure	Where clause
45	PMSC_COMMODITIES	commodity in (select commodity from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')) or commodity in (select commoditygroup from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME'))
50	DMACTION	action in (select action from wfaction where processname in ((select mgrapprprocess from pmscofferingext where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')) union (select srapprprocess from pmscofferingext where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')))) or action in (select member from actiongroup where action in (select action from wfaction where processname in ((select mgrapprprocess from pmscofferingext where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')) union (select srapprprocess from pmscofferingext where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME'))))
55	DMACTIONGROUP	action in (select action from wfaction where processname in ((select mgrapprprocess from pmscofferingext where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')) union (select srapprprocess from pmscofferingext where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME'))))
60	DMWFPROCESS	processname in (select srapprprocess from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')) or processname in (select mgrapprprocess from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME'))
65	DKTICKETTEMPLATE	templateid in (select templateid from pmscoffering where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME'))
70	DMMAXGROUP	groupname in (select groupname from pmscofferingauth where itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME'))
75	PMSC_OFFERING	itemnum in (select offeringnum from pmsccatalogoffmap where itemnum='CATALOGNAME')

Migration order	Object structure	Where clause
80	PMSC_DOCINFO	docinfo in (select docinfo from doclinks where ownertable='PMSCOFFERING' and ownerid in (select itemid from pmsoffering where itemnum in (select offeringnum from pmscatalogoffmap where itemnum='CATALOGNAME'))))

Explanation: The migration order values increase in increments of five to allow for additional object structures to be added between the predefined object structures in the future.

9.6.2 PMSC72_CATALOG

The PMSC72_CATALOG migration object has been delivered with standard Where clauses for each object structure in the migration object. Similarly to the PMSC72_OFFERING migration group, you must edit these Where clauses to specify the name of the service catalog to migrate. Insert the catalog name, as it appears in the database, in each Where clause.

Table 9-3 lists the object structures that make up the PMSC72_CATALOG migration object and the associated Where clause in the order in which they are migrated.

Table 9-3 Predefined PMSC72_CATALOG Where clauses

Migration order	Migration object	Where clause
	DMMAXGROUP	groupname in (select groupname from pmscatalogauth where itemnum='CATALOGNAME')
10	PMSCCATALOG	itemnum='CATALOGNAME'

9.7 Deployment

You need to understand the deployment considerations that are associated with this scenario prior to deploying the package to the Target environment, in particular, the migration prerequisites.

After the package definition is approved and activated, you can create and distribute the package. Then, deploy the package to the Target environment.

Resource: This document does not repeat the standard steps that are involved in deploying a package. For detailed information, refer to the *Migration Manager Guide*:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.mam.doc_7.1/pdf/mam71_migration_mgr_guide.pdf

After deployment to the Target environment, the users can start to use the service catalog and its related service offerings in the Target environment.

9.7.1 Deployment considerations

You must accommodate for the following specified deployment considerations prior to the deployment of the PMSC72_CatalogOffering, or else the package migration will fail.

Migration prerequisites

The PMSC72_CatalogTemplate migrates the additional records that are directly associated with the service catalog. The package definition that ships with the product is not an all encompassing package to migrate everything in Maximo. Therefore, it is important to ensure that the following objects are migrated manually or by using alternative migration packages:

- ▶ **Job plans:** The PMSC72_OFFERING migration object migrates the job plans in their entirety (with the exclusion of nested job plans); however, it does not create the referenced labor, items, services, and tools in their respective database objects in the Target environment.
- ▶ **Nested job plans:** Where a job plan has been specified against a job plan task (nested job plan), the nested job plan will not be migrated as part of the PMSC72_CatalogTemplate, only the job plan that is associated directly with the ticket template that is applied to the service catalog will be migrated in the package. The nested job plan must be migrated in advance.
- ▶ **Metadata:** Data dictionary objects are not included in the PMSC72_CatalogTemplate. Use the migration groups that ship with the product to migrate metadata prior to the service catalog migration.
- ▶ **Response plans:** Response plans are not included in the predefined service catalog migration package. Response plans that are associated with service groups and service offerings must be migrated prior to deploying the PMSC72_CatalogTemplate to the Target environment.

- ▶ **Parent classifications:** The PMSC_OFFERING migration object migrates the individual classification that is referenced by a service offering; however, it does not migrate the entire classification hierarchy that is associated with the referenced classification. Parent classifications are required to be created prior to their associated child classifications.
- ▶ **Circular relationships:** A situation can occur where actions reference workflows, and workflows reference actions. The PMSC72_OFFERING will migrate the first level of actions that are associated with relevant workflows. You must first migrate workflows that are referenced by actions, and the actions that are referenced by workflows prior to migrating service catalogs.

Where clause inclusions and exclusions

Where clauses include and exclude data from the migration package. The PMSC72_CatalogTemplate only migrates objects where they are relevant to the 'CATALOGNAME' that is specified in each of the Where clauses that is associated with each of the migration objects. For example, if a classification is not specified against the job plan, an offering of a catalog, or against the catalog itself, the migration group will not migrate any of the objects in the DMCLASSIFICATION object structure.

Duplicating the PMSC72_OFFERING migration group

The PMSC72_CatalogTemplate has been developed with the PMSC72_OFFERING migration group set with a migration order that is lower than the migration order of the PMSC72_CATALOG migration group. In order for the package to migrate properly, the PMSC72_OFFERING package must be processed prior to the PMSC72_CATALOG.

It might be necessary to modify the predefined PMSC72_OFFERING migration group to meet specific migration needs. However, because the migration group is also used in the PMSC72_ServiceTemplate, we advise that you duplicate it. After it has been duplicated, you need to be aware that when added to the PMSC72_CatalogTemplate, the auto-number on the migration groups will set the duplicated PMSC72_OFFERING to a higher number than the PMSC72_CATALOG. Therefore, the PMSC72_CATALOG will process before the duplicated PMSC72_OFFERING migration group. In this instance, the package will fail if the prerequisite data for the predefined original PMSC72_OFFERING migration group does not already exist in the Target environment.

To avoid this issue, after the PMSC72_OFFERING migration group is duplicated, the Migration Group Order field on the PMSC72_CATALOG migration group must be set to a number *greater* than that of the duplicated PMSC72_OFFERING migration group. Figure 9-23 on page 196 depicts the Migration Group Order field automatically populated on the duplication of the PMSC72_OFFERING migration group.

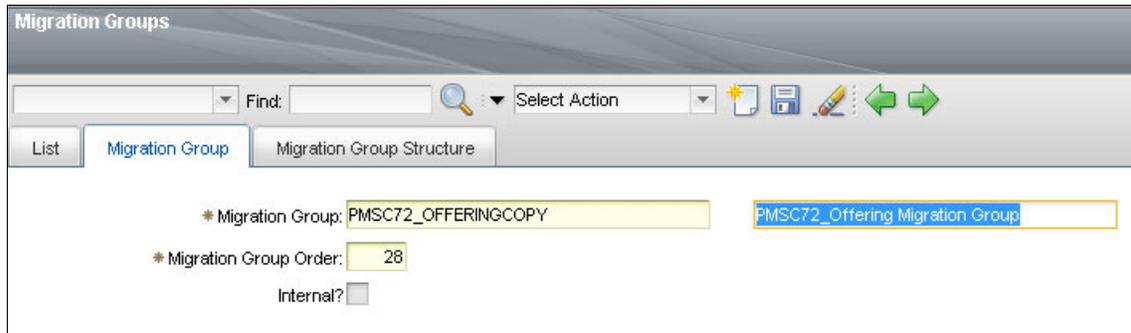


Figure 9-23 Migration Group Order field in the Migration Groups application



Common topics

The Migration Manager has many built-in capabilities, most of which are used in almost every migration scenario. Also, there are certain considerations of which you must be aware. This chapter provides an overview of the following common topics:

- ▶ 10.1, “Comparing Integration Framework and Migration Manager” on page 198
- ▶ 10.2, “Multiple language considerations” on page 201
- ▶ 10.3, “Snapshot package versus Change package” on page 204
- ▶ 10.4, “Embedded URLs” on page 208
- ▶ 10.5, “Clustered environment considerations” on page 209
- ▶ 10.6, “Change tracking and ad hoc reporting” on page 210
- ▶ 10.7, “Admin mode” on page 221
- ▶ 10.8, “Migrating hierarchical data” on page 221
- ▶ 10.9, “Setting up logging for the Migration Manager” on page 229
- ▶ 10.10, “Start Center visibility into configurations” on page 235

10.1 Comparing Integration Framework and Migration Manager

Integration Framework and the Migration Manager are two key components of Tivoli's process automation engine. These two components have a variety of export and import capabilities. Each component has been designed for separate purposes. Table 10-1 compares the key features of the two components.

Table 10-1 Comparing Integration Framework and Migration Manager

	Integration Framework	Migration Manager
Purpose	<ul style="list-style-type: none"> ▶ Transactional integration to external applications ▶ Export and import to and from external applications or existing applications ▶ Service invocations 	Package-based configuration export and import between similar production environments
Process/discipline	Enterprise application integration and service-oriented architecture	Change management
Primary content support	Master and transactional data	Metadata (configuration data)
Other content	Any business object can potentially be targeted for integration	Any business object can potentially be targeted for migration
Unit of work	Transaction, file, or message (can be a single document or record)	Package (multiple records)
Typical user	Practitioner or implementer with programming skills; experience with application integration; familiarity with ERP systems	Practitioner or implementer with product configuration skills and familiarity with change management
Framework	Integration Framework built from the ground up	Migration framework built from the ground up; uses several integration constructs
Key feature differences	Extensive mapping and transformation capabilities that are based on Java, XSL, or rules	No mapping or transformation capabilities
	Database event-based single transactions from a given application	Database event-based change tracking and aggregation across multiple applications (change package type)

	Integration Framework	Migration Manager
	Asynchronous processing using Java Message Service (JMS) queues	No queues
	Tivoli's process automation engine-attached document can be extracted and processed with the parent record	Tivoli's process automation engine-attached document can be extracted and processed with the parent record; arbitrary attachments (Java class or image files) can be placed in a package
		Automation of database configuration during package deployment
	Expose web services (Web Services Description Language (WSDL) or Representational State Transfer (REST)-based)	
	Invoke web services	
	Extensive error correction and resubmission capability at the transaction level (Message Reprocessing application)	Limited error correction capability at the package level (package might have to be recreated in the Source environment)
	Single object structure processing (data exported for multiple files to be sequenced manually for import)	Multiple object structures are grouped into a single migration group; every object structure in a group is sequenced, precluding the need for manual sequencing; migration groups are also sequenced
	XML file containing data can be edited and associated with action attributes that determine the action to be performed with the data elements	Package contains multiple XML files, which are not intended to be edited; action attributes are automatically set up by the Migration Manager

10.1.1 Selecting Integration Framework or Migration Manager

In this section, we review a few key scenarios and identify the appropriate component to use to execute the required tasks:

- ▶ **Scenario 1:** A developer has implemented an Information Technology Infrastructure Library (ITIL)-based incident management workflow using Tivoli's process automation engine Workflow Designer and related applications. This workflow must be promoted to the production environment. What tool do I use?

The Migration Manager is the tool of choice to package up workflow and related configuration data. A single package can contain your workflow, as well as any actions, roles, and communication templates together with any custom code that a client has to build into a Maximo EAR.

- ▶ **Scenario 2:** An existing ticket management application needs to be integrated with the Tivoli Service Request Manager product. Historical tickets need to be loaded into the Tivoli product. Also, the existing ticket management application will be in use until Tivoli Service Request Manager goes into production.

Integration Framework (MEA) is the tool of choice. Existing data, such as tickets, can be loaded into Tivoli Service Request Manager using a number of methods, including flat files, interface tables, or XML. Periodically, data can be exported out of the existing tool in the form of a file and dropped into a designated folder that Integration Framework can automatically process into Tivoli Service Request Manager.

- ▶ **Scenario 3:** My client has implemented several SAP R/3 modules, including Materials Management. There is a requirement to integrate specific SAP modules to Maximo Asset Management to achieve the total integration of the purchasing and inventory processes.

Integration Framework (MEA) and the SAP R/3 Adapter are the tools of choice. The SAP R/3 Adapter offers pre-built integration points to SAP R/3 from Maximo. In addition, custom integration points can be built.

- ▶ **Scenario 4:** The client has a development environment in which a team of developers configures the IBM Tivoli Change and Configuration Management Database (CCMDB) product. Configuration includes the addition of new objects (tables), attributes (columns), and domains. Also, the client wants to add new tabs and dialogs to the existing configuration item (CI) application. All of these configurations have to be migrated to a User-Acceptance Test (UAT) environment before being promoted into production.

The Migration Manager is the tool of choice. It was designed specifically to cater to a controlled promotion of configurations to production from development through UAT. The tool also automates the structural changes to the underlying database that are required as a result of adding new objects and attributes and therefore provides a seamless deployment of a package containing variable content.

- ▶ **Scenario 5:** The client needs to migrate *foundation* data, which is also known as *implementation data*, from development to production to avoid having to re-enter data, such as locations and classifications, in the production environment.

Typically, foundation data consists of discrete sets of data that do not have multiple or deep relationships with other data. The Migration Manager can be

used to migrate such foundation data. If an object structure does not exist that supports this type of data, the practitioner or implementer needs to put together an object structure and, if necessary, author Java-based processing code. This type of an object structure can be used from the Integration Framework, as well. With its queue-based processing, error correction, and resubmission capability, using the Integration Framework to load foundation data is more efficient than using the Migration Manager.

10.2 Multiple language considerations

Tivoli's process automation engine-based products can support multiple languages from a single environment and database. The language configuration for a production environment is set up during product installation time and stored in the underlying production database. Every production environment is associated with a base language, and additional languages can be enabled as needed by installing appropriate language packs. As each additional language is installed, a set of translated content is added into the production database into dedicated multiple language tables.

The enablement of multiple languages is configured for the attribute of a business object. If a particular attribute has been enabled for multi-language, a dedicated multi-language table is created to hold the multi-language content for the targeted attribute. The base table always holds the base language content. A number of attributes that ship with the product are already enabled for multi-language. These attributes allow key functionality in the product to be available and presented in the local language that is chosen by the user when additional languages have been installed and enabled.

For more details about the product's multiple language capabilities and configuration, refer to the following documentation:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.srm.doc_7.1/reference/mam71_sys_admin_guide.pdf

The Migration Manager requires that the base language in the Source and Target production environments is the same. The Migration Manager issues an error message, BMXAA5651E, and prevents the deployment of packages when the base language of a Target environment differs from that of the Source environment.

The multi-language configurations of the production environment need to be supported from the migration perspective. Two types of tables must be supported when migrating multi-language content:

- ▶ LONGDESCRIPTION

► L_ Multi-language table

Each Migration Manager object structure that is supplied with the product has been designed to include the multi-language table if the multi-language table was also shipped with the product. Figure 10-1 on page 202 shows the DMMAXAPPS object structure that contains both the LONGDESCRIPTION and the multi-language tables.

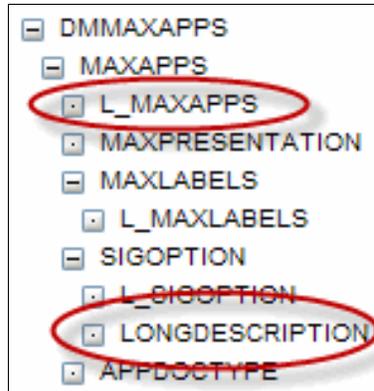


Figure 10-1 DMMAXAPPS object

When configurations are migrated from production environments that have additional languages enabled, the base language content is extracted by the Migration Manager to be part of the base table and the additional language content is extracted as part of the L_ multi-language table. Thus, the Migration Manager automatically exports multi-language-enabled content when a package is created.

Multi-language content: The Migration Manager will extract all of the available multi-language content from the L_ table. There is no capability to extract multi-language content for only certain languages.

This approach to the Migration Manager object structures is especially important when migrating configurations that constitute the most visible parts of a business application. The various visible elements of an application are configured using the Application Designer. When application presentations are migrated, the localized content of various elements of the application presentation must also be migrated. Table 10-2 enumerates these presentation elements and the corresponding Migration Manager object structures that are enabled to capture multi-language content.

Table 10-2 Presentation elements and the corresponding Migration Manager object structures

User interface element	Migration Manager object structure	Migration Group
Presentation XML	DMMAXAPPS	APPLICATION
Labels	Part of DMMAXAPPS	APPLICATION
All menus	DMMAXMENU	APPLICATION
Signature options	DMSIGOPTION	APPSECURITY
Conditional user interface	DMCTRLGROUP	APPSECURITY
Domains	DMMAXDOMAIN	DATADICTIONARY

Consider the following conditions before performing the migration:

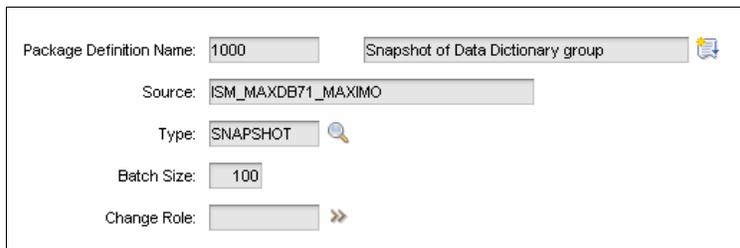
- ▶ If additional languages were enabled in the Source environment, the information that is stored in the LANGUAGE table must be migrated to the Target environment using the DMLANGUAGE object structure before migrating any other configurations. The DMLANGUAGE object structure is part of the DATADICTIONARY migration group. If a migration is performed from a Source environment to a Target environment that has fewer or other languages enabled, the migration will fail.
- ▶ If a multi-language table was configured on-site, determine if there is a corresponding Migration Manager object structure that needs to be updated to include the multi-language table. The multi-language table must be added as a child of the base table for which it has been configured.
- ▶ If the long description capability was configured for a business object, determine if there is a corresponding Migration Manager object structure that needs to be updated to include the LONGDESCRIPTION table. The LONGDESCRIPTION table must be added as a child of the base table for which it has been configured.

Important: If you modify an existing Migration Manager object structure to add a multi-language table in a Source environment, you must modify the object structure of the Target environment with the same change.

10.3 Snapshot package versus Change package

There are two types of packages in the Migration Manager application: the Snapshot migration package and the Change migration package.

The Snapshot migration package contains configuration data content that exists in the Source environment at the time that the package was created. Use the Snapshot package when you have significant changes in the environment. For example, use this package when you create a new environment or create a new set of applications and workflows. You can use the SQL Where clause to filter the content that you want to migrate in a Snapshot migration. Figure 10-2 on page 204 shows an example of a Snapshot package definition.



The screenshot displays a form for defining a Snapshot package. The fields are as follows:

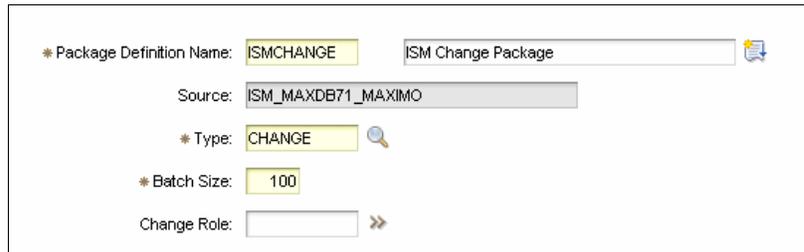
- Package Definition Name: 1000
- Snapshot of Data Dictionary group
- Source: ISM_MAXDB71_MAXIMO
- Type: SNAPSHOT
- Batch Size: 100
- Change Role: >>

Figure 10-2 Definition of a Snapshot package

Tip: Use the 1=0 SQL statement to migrate no content in the selected migration group SQL.

To use Snapshot packages, the entire migration requirement must be clearly defined, identified, and documented, to the exact name of the content elements. The configuration of Snapshot packages requires a considerable amount of configuration on the Source environment.

Change migration packages usually contain less content data compared to the Snapshot packages. Change packages contain the configuration data that has been created since the package was activated. Figure 10-3 shows an example of a Change package definition.



The screenshot shows a form for defining a Change package. It includes the following fields:

- Package Definition Name:** A text input field containing "ISMCHANGE" and a dropdown menu showing "ISM Change Package".
- Source:** A text input field containing "ISM_MAXDB71_MAXIMO".
- Type:** A dropdown menu showing "CHANGE".
- Batch Size:** A text input field containing "100".
- Change Role:** A text input field with a right-pointing arrow icon.

Figure 10-3 Definition of a Change package

Type: After you have defined the type field, it becomes read-only and cannot be changed.

In Change package definitions, the Change Role field is editable. This field allows you to track a specific role's changes.

Change packages are useful when there is already a Source environment and a Target environment that are in synch at the time that this type of package is activated. Separate Change packages must not overlap. For example, separate Change packages must not listen to the same objects. It is useful if each developer has its own Change package.

Change packages: Change packages cannot track changes when Admin mode is turned on. Admin mode shuts off all of the event broadcasts. If a Change package is active, you will get a warning when you turn Admin mode on.

Certain object structures, which are created to be used with a Snapshot package, cannot be used for Change packages. The Migration Manager cannot handle certain data in a Change package. For example, hierarchical data cannot be migrated with a Change package. The Change package has no way to know the hierarchical order, which causes validation errors. Classifications, locations, and assets are examples of hierarchical data. Overlapped packages, which have the same content, can cause an error for the same reason. Always preview the deployment changes before deployment to see if the content is valid.

DELETE statements: Snapshot packages cannot track DELETE statements, because they do not keep historical data. If you want to keep track of deleted configuration content, use a Change package instead of a Snapshot package.

To see and manage the changes in a Change package, perform the following steps:

1. Navigate to the Migration Manager application via **Go To → System Configuration → Migration → Migration Manager**.
2. Select the related Change package, and then, select **View Event Tracking Records** in the Select Action menu, as shown in Figure 10-4 on page 206.

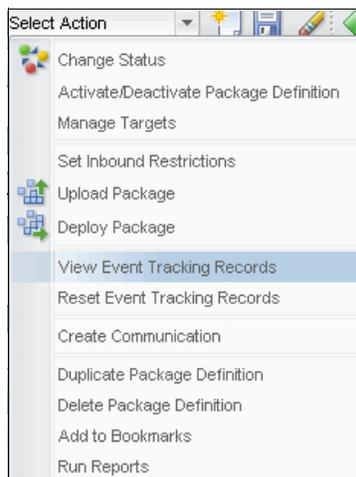


Figure 10-4 View Event Tracking Records option

3. Figure 10-5 shows the event tracking records for a Change package. The figure shows which objects and related objects are altered. If there are certain records that you do not want to migrate, click the Trash Can icon to the right of the line. Child event tracking records are deleted automatically.

View Event Tracking Records						
Root Event Tracking Records						
Object	Action	Primary Keys	Event Date	Event User	Object	Object Id
CLASSTRUC	DELETE	CLASSTRUCID=2524	24.09.2010 16:30:34	MAXADMIN	CLASSTRUC	866.406
Child Event Tracking Records						
Object	Action	Primary Keys	Event Date	Event User	Parent Object	Parent Object Id
CLASSTRUC	UPDATE	CLASSTRUCID=2522	24.09.2010 16:30:34	MAXADMIN	CLASSTRUC	866.406
CLASSSPEC	DELETE	CLASSTRUCID=2524,ASSETATTRID=C16AT,TRIBUTE,SECTION=,ORGID=,SITEID=	24.09.2010 16:30:34	MAXADMIN	CLASSTRUC	866.406
CLASSUSEWTH	DELETE	CLASSTRUCID=2524,OBJECTNAME=CI	24.09.2010 16:30:34	MAXADMIN	CLASSTRUC	866.406
CLASSSPECUSEWTH	DELETE	CLASSTRUCID=2524,ASSETATTRID=C16AT,TRIBUTE,SECTION=,ORGID=,SITEID=,OBJECTNAME=CI	24.09.2010 16:30:34	MAXADMIN	CLASSSPEC	17.721.985

Figure 10-5 Events recorded in a Change package

- If you want to delete all of the event tracking records, select **Reset Event Tracking Records** in the Select Action menu, as illustrated in Figure 10-6 on page 207.

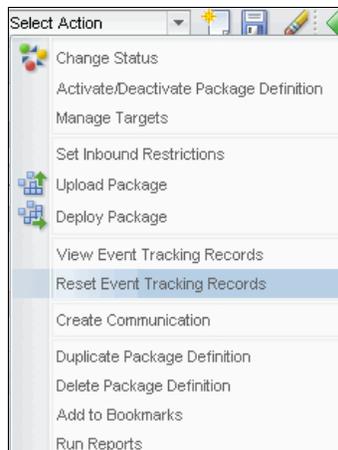


Figure 10-6 Reset Event Tracking Records option

- When you select this option, a confirmation dialog is displayed. Click **Yes** if you are sure that you want to delete the tracked events.

10.4 Embedded URLs

Certain configuration data can include embedded URLs. For example, you want to send an email with a link to the system to the users. You can use the Communication Templates application. Figure 10-7 on page 208 illustrates an example of a communication template with an embedded URL.

Template: ESCINCSURVE Escalation for Tracked Survey.

* Applies To: INCIDENT

Accessible From: ALL

Comm Log Entry?

Details

To: AFFECTEDBY, REPORTEDBY

cc:

bcc:

* Send From: maxadmin@mro.com

Reply To:

Subject: :CLASS :TICKETID has been closed. Please giv

Message: Report Date - :REPORTDATE Classification - :CLASSSTRUCTURE_DESCRIPTION Summary - :D
http://maximoserver/maximo/ui/maximo.jsp?event=loadapp&value=surveyreq&uniqueid=2&recordkey=:TICKETID

Figure 10-7 Example of a communication template with an embedded URL

URLs that are used in communication templates, endpoints, and launch-in-contexts generally differ in separate environments.

You must correct these URLs manually after migrating these types of objects. Communication templates are INACTIVE after migration. Check the URLs in the communication templates, and then, change their status to ACTIVE. Other objects, such as launch-in-context and endpoints are stateless. Be careful before using these objects in the Target environment.

10.5 Clustered environment considerations

Clustered environments might behave differently in migration scenarios. There are many scenarios in which clustered environments behave differently.

Packages can contain structural and non-structural content. Structural content and part of the non-structural content are cached by the Java virtual machine (JVM) to provide performance. The following examples are types of cached content:

- ▶ Table, view, and column structures
- ▶ Application presentations
- ▶ Domains

In a single server environment, you will have no problem migrating these types of content. You can use this content as soon as the migration completes.

In a clustered environment, each member JVM server connects to the same back-end database to read and load this content into its cache. When deploying a new package, cached content must be refreshed on each clustered server.

Restart each JVM in the cluster to ensure that the changes in the migration package are reflected on each JVM. Otherwise, the users that are using the other JVMs will not be affected by the changes. In clustered environments, *changes must be scheduled (using an outage window)* to be able to restart all JVMs. If a full outage window cannot be scheduled, you can use a *partial* or *rolling* outage approach in which JVM servers are restarted in phases, and one or more servers are always available. This approach requires the use of a load balancer appliance.

Tip: During the deployment of packages in full outage windows, stop all of the JVMs in the cluster, except the deployment JVM that is used through the end of the entire migration process. After you complete the migration process, be sure that the Admin mode is off and that the JVMs are started.

Warning: The first users that log in to each JVM will experience slow response time until the server has built its cache. Afterward, the performance improves.

10.6 Change tracking and ad hoc reporting

A common misconception is that the Migration Manager is considered an afterthought and that it is a tool to use *only after* configuration has been completed.

In most scenarios, a Snapshot package is used to promote configuration changes. However, the developer who performs the migration is not always the original developer, or it might be a multiple developer project. As we see in the examples covered in this publication, the migration requirements must be clearly defined, identified, and documented, to the exact names of the configuration elements, to be able to use the Snapshot package effectively.

This section covers how the Change package in the Migration Manager is an important element in the development cycle, and how it can be used effectively.

10.6.1 When to use Change packages

Change packages are useful when the Source environment and Target environment are synchronized and the development efforts have not started. But to use the Change package effectively, the Migration Manager must be part of the development planning to harness its capabilities during the development cycle.

The simplest and ideal scenario is when the development planning has determined that Change packages will capture the configuration activities in a development environment and then deploy them to a Target environment in an organized way.

Roles have been assigned, and package content definitions have been created. Most importantly, the developers' roles do not overlap, which means that no two developers work on the same configuration content, even if separate Change packages are listening to the same migration objects.

If roles are not necessary (for example, there is only one developer), it is better to define only one Change Package that contains all of the migration objects that have been identified as part of the migration requirements. Deploying this type of package performs all of the tracked configuration events in one deployment.

You can also define several packages that each track only one specific migration group, without overlapping the migration objects. The deployment of Change packages of this type requires taking into account the dependency of groups in the sequence of the deployment. It might also require coordinated efforts when partially completed development needs to be promoted to create the package,

and then, you reset the package and activate the package to continue further development.

10.6.2 Using Change packages to track development activities

The development environment is a shared environment with multiple developers working in a coordinated manner to configure the product to meet product solution requirements in a specific time frame.

In this scenario and typically, development activities begin rapidly. The planning and incorporation of the Migration Manager Change packages often does not happen at the beginning.

In most Migration Manager implementations, the Snapshot migration package functionality is used to promote configuration. The use of SQL Where clause statements to target specific configuration content is critical for the promotion of specific content. The challenge is to define this content, identifying the names of configuration key values.

The records that need to be migrated must be identified either by means of a naming convention (for example, object key values beginning with prefix 'CUST') or must be clearly documented in the form of spreadsheets or other tabular type documents. Without this information, it takes considerable time and effort to proceed with migration.

The Change package provides the ability to capture database events (add, update, or delete) that affect specific configuration content records. This functionality can be used as a tracking device only, without creating, distributing, and deploying. Then, a Snapshot package can be created based on the changes that have been tracked.

10.6.3 Configuring change roles

Change package functionality provides the ability to track changes made by specific users through the use of *roles*, which is particularly useful in multiple developer teams. Each developer is assigned a user ID to log in and perform configuration.

A role can be assigned to a person or to a person group. It is up to the team to decide the type of group to use.

For this example, we create a DEVELOPER role of type Person Group, as shown in Figure 10-8 on page 212, based on the Person Group called DEVELOP, which is shown in Figure 10-9 on page 212. There are two person IDs assigned to this

group, and these IDs correspond to the user IDs of the two developers. In our example, the users are WILSON and LWAYNE.

Role: DEVELOPER Developer Group Role

* Type: Person Group

Object: []

* Value: DEVELOP

Parameter: []

E-mail?

Broadcast?

Figure 10-8 DEVELOPER role

Important: In Figure 10-8, if the broadcast flag is left unselected, the Migration Manager will not record the events triggered by all members of the group. Only the events of the designated default person of the group will be recorded.

Person Group: DEVELOP Developer Group

People Filter > 1 - 2 of 2

Person	Name	Use for Organization	Use for Site
WILSON	Mike Wilson	<input type="checkbox"/>	<input type="checkbox"/>
LWAYNE	Lisa Wayne	<input type="checkbox"/>	<input type="checkbox"/>

Figure 10-9 DEVELOP person group

You can use person type roles in Change packages, as well. The advantage of a person group is the ability to add or remove users in a dynamic environment.

The two roles that are defined here are for two developers who will configure changes in applications that affect the data dictionary and the application framework.

In this example, their work does not overlap. WILSON will change the objects in the data dictionary group, and LWAYNE will make the configuration changes on the application side. This scenario is merely an example of how you can organize

the development. Both developers can perform configurations in both groups. The important point here is to be able to track these events.

10.6.4 Change package definitions

After the roles have been defined, it is time to create the Change packages. These packages will track the changes to specific configuration objects that have been made by the users that are defined in the role that is assigned to the package.

In this example, a Change package definition is created with the standard DATADictionary migration group, to track changes to any object structure that is included in this group. This definition is associated with the DEVELOPER role.

The screenshot shows a configuration window for a Change package definition. The fields are as follows:

- Package Definition Name: MYCHANGE-DD
- Changes to Data Dictionary
- Source: ISM_MAXDB71_MAXIMO
- Type: CHANGE
- Batch Size: 100
- Change Role: DEVELOPER

Below the form is a table of Migration Groups:

Migration Group	Description
DATADictionary	Data Dictionary

There is a 'New f' button at the bottom right of the table area.

Figure 10-10 Change package MYCHANGE-DD

A second Change package definition is created to track changes to applications. A new migration group is used as a duplicate copy of the APPLICATIONS migration group. This new group is called MYAPPLICATION, and all dependent groups have been removed. This package is also associated with the DEVELOPER role, as shown in Figure 10-11 on page 214.

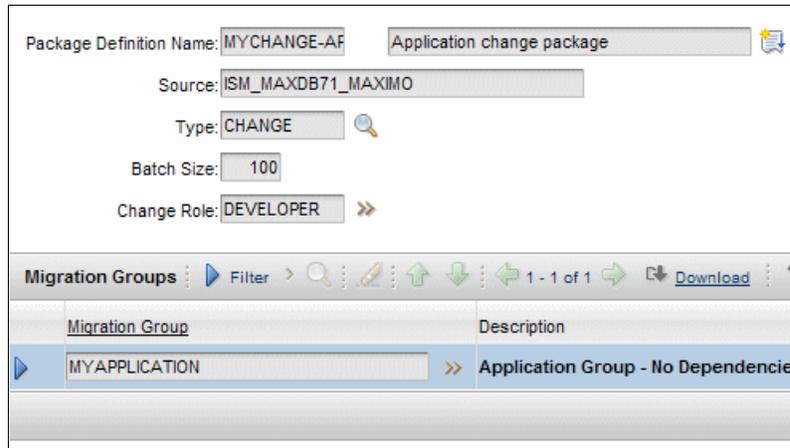


Figure 10-11 Change package MYCHANGE-APP

Tracking changes: Change tracking will be performed for every Maximo business object (MBO) for the migration group that is specified in the list. More groups can be added, as needed, as long as the package definition status is WAPPR.

10.6.5 Tracking change events

After approving and activating the packages, the developers can start the configuration efforts. These events are tracked and stored in the event tables.

When making structural changes to the database using the Database Configuration application, applying these changes will require the Admin mode to be turned on. During this time, the tracking stops. Do not perform any changes until the database configuration is completed and Admin mode is turned off in the system.

We will demonstrate how we will use the Change package for tracking. We require the following changes:

- ▶ A new domain, MYRESULTS, of type ALN was added. The values are PASSED and FAILED.
- ▶ A new attribute, which is called MYRESULTS, was created in the ASSET table. This attribute will hold the results of test values. The new domain is configured for validation.
- ▶ The new attribute for test results is added to the ASSET application and configured to validate.

After the developers have completed their configuration, we can now see the exact events from the Change package definitions, by clicking **Select Action** → **View Event Tracking Records** (Figure 10-12 and Figure 10-13).

Object	Action	Primary Keys	Event Date	Event User	Object	Object Id
MAXDOMAIN	ADD	DOMAINID=MYRESULTS	10/5/10 16:00:02	WILSON	MAXDOMAIN	768
MAXOBJECTCFG	UPDATE	OBJECTNAME=ASSET	10/5/10 16:02:04	WILSON	MAXOBJECTCFG	192

Figure 10-12 Events for package MYCHANGE-DD

Object	Action	Primary Keys	Event Date	Event User	Object	Object Id
MAXAPPS	UPDATE	APP=ASSET	10/5/10 16:18:21	LWAYNE	MAXAPPS	8

Figure 10-13 Events for package MYCHANGE-APP

This example shows the events' sample configurations that were created by the two developers. Each package definition provides the functionality to list these recorded events in the application.

Single role Change package: Depending on the strategy, you can use a single role Change package to track change events. The single Change package must include all of the migration groups that have been identified as being configured.

The Object and the Primary Keys columns are the most important. With this information, we can define the migration requirements for the Snapshot package.

For example, from the events that are shown in Figure 10-12 on page 215, we need the objects and SQL conditions that are shown in Table 10-3 for our package.

Table 10-3 Where clauses for data dictionary objects in a Snapshot package

Migration object	Object	Where clause
DMMAXDOMAIN	MAXDOMAIN	domainid in ('MYRESULTS')
DMMAXOBJECTCFG	MAXOBJECTCFG	objectname in ('ASSET')

From the events shown in Figure 10-13 on page 215, we can gather enough information to determine the objects and SQL conditions (as shown in Table 10-4) that are needed for a Snapshot package.

Table 10-4 Where clauses for application objects in a Snapshot package

Migration object	Object	Where clause
DMMAXAPPS	MAXAPPS	app in ('ASSET')

In a multiple developer environment, with several configurations taking place at the same time, often developers must learn other tools or applications and focus on completing tasks. The use of Change package functionality provides essential information, which is sometimes not documented, to prepare and deploy Snapshot packages in an ongoing basis throughout the development cycle or as part of the daily operations and maintenance tasks.

Keeping track of what is changed and by whom is a good practice and enables accountability and change process documentation. However, this tracking information is stored in the database, and the only documentation might be in the form of a tracking spreadsheet.

Next, we will show how you can extract this information for documentation and Snapshot package preparation purposes.

10.6.6 Creating ad hoc reports of change events

You can configure an ad hoc or query-based report (QBR) to gather all of the changes that are tracked for all users across all packages. You must perform several one-time tasks to configure and execute this type of an ad hoc report.

Defining reporting object structure

Create a DMPKGEVENTS object structure by clicking **Go To** → **System Configuration** → **Migration** → **Object Structures**. The following details apply to the new object structure:

1. Specify the Consumed By field for the DMPKGEVENTS object structure as REPORTING.
2. Specify the DMPACKAGEDEF MBO as the parent and DMPKGEVENTTRK MBO as the child.
3. For the DMPKGEVENTTRK child MBO, specify the Relationship as DMPKGEVENTTRK_ROOT. This specification will result in the ad hoc report only displaying the changes to root (or primary) MBOs.
4. Save the object structure.

The new object structure is shown in Figure 10-14.

The screenshot shows the configuration interface for the DMPKGEVENTS object structure. The top section contains several fields and checkboxes:

- Object Structure: DMPKGEVENTS
- Event tracking object: Event tracking object
- Query Only?:
- User Defined?:
- * Consumed By: REPORTING
- Reporting: Reporting
- Self Reference?:
- Configurable?:
- Application: DM
- Support Flat Structure?:
- Alias Conflict?:
- Outbound Definition Class: (empty field)
- Inbound Processing Class: (empty field)

Below the fields is a table titled "Source Objects for DMPKGEVENTS". The table has columns for Object, Parent Object, Object Location Path, Relationship, Object Order, and Use. The table contains two rows:

Object	Parent Object	Object Location Path	Relationship	Object Order	Use
DMPACKAGEDEF		DMPACKAGEDEF		1	<input checked="" type="checkbox"/>
DMPKGEVENTTRK	DMPACKAGEDEF	DMPACKAGEDEF/DMPKGEVENTTRK	DMPKGEVENTTRK_ROOT	1	<input checked="" type="checkbox"/>

A "New Row" button is located at the bottom right of the table.

Figure 10-14 DMPKGEVENTS object structure for reporting

This reporting object association to an application is new in Maximo Base Services Fix Pack 7.1.1.6, as illustrated in Figure 10-15 on page 218. In Fix Pack 7.1.1.5, this association is done in Application Designer.

Object Structure: DMPKGEVENTS

* Consumed By: REPORTING

Application: DM

Fund Definition Class:

d Processing Class:

Figure 10-15 Application for reporting object structure

After you complete this step, a new reporting icon is added to the Migration Manager application.

Granting security access to reporting object structure

Perform this step for users of Maximo Base Services Fix Pack 7.1.1.6:

1. Click **Go To** → **Administration** → **Reporting** → **Report Administration**. From the List tab, click **Select Action** → **Set Report Object Structure Security**, and filter the application Migration Manager.
2. Add the MAXADMIN group to the security list, and click **OK**. The security object is shown in Figure 10-16.

Report Object Structure Security

Description	Application
Event tracking object	Migration Manager

Group	Description
MAXADMIN	Maximo Administrators (Super Users)

New Row

Figure 10-16 Report Object Structure Security window

Configuring the Ad Hoc report

In the Migration Manager application, click the **Create Report** icon to display the Query-based report pop-up dialog. In Maximo Base Services Fix Pack 7.1.1.6, the dialog is multi-tabbed.

The first tab is the Style tab. Perform this configuration (Figure 10-17):

1. Specify a name for the ad hoc report (MM_ALL_EVENTS).
2. Make it public by selecting the **Public?** check box.
3. Select **Save Report** to enable the report design to be available to subsequent users and sessions.
4. Select **Summary Report** to display records in a simple row/column format.

Query Based Report

Style Select Format Submit

Help Text

Style to Use for the Report

Summary Report. Select this style to create a Summary Report. Summary reports are used to present multiple details on multiple records in row/column format. Summary reports can be grouped and sorted.

Detail Report. Select this Style to Create a Detail Report. Detail reports present details on individual records in categories by separating the individual records by page breaks. Detail reports are dynamically organized by categories.

* Report Title: Save Report?

Public? Close Window?

Submit Cancel

Figure 10-17 MM_ALL_EVENT query-based report Style tab

5. On the Select tab, configure the selected fields list, as demonstrated in Figure 10-18.

Column Order	Report Label	Field Value	Category
1	Package Definition Name	PKGDEFNAME	DMPACKAGEDEF
2	Object	OBJECT	DMPKGEVENTTRK
3	Primary Keys	PRIMARYKEYS	DMPKGEVENTTRK
4	Action	ACTION	DMPKGEVENTTRK
5	Event User	EVENTUSER	DMPKGEVENTTRK
6	Event Date	EVENTDATE	DMPKGEVENTTRK

Figure 10-18 MM_ALL_EVENTS selected fields for QBR

- Leaving the option **Apply the Current Query and Filter from the Application?** selected allows you to use the filtering of package definitions from the Migration Manager application List tab.

In our example, the filtering was done on the package Type and Active fields, as shown in Figure 10-19.

Package Definition Name	Source	Description	Type	Status	Active
			change		Y
MYCHANGE-DD	ISM_MAXDB71_MAXIMO	Changes to Data Dictionary	CHANGE	APPR	<input checked="" type="checkbox"/>
MYCHANGE-APP	ISM_MAXDB71_MAXIMO	Application change package	CHANGE	APPR	<input checked="" type="checkbox"/>

Select Records

Figure 10-19 Change Package list tracking events

Executing the ad hoc report

At this point, you can submit the report to run immediately as a PDF report. After you close the definition, you can invoke the report in the Migration Manager application by clicking **Select Action** → **Run Reports** and then clicking the saved report **MM_ALL_EVENTS**. Figure 10-20 shows the output.

Package Definition Name	Object	Primary Keys	Event User	Event Date
MYCHANGE-DD	MAXDOMAIN	DOMAINID=MYRESULTS	WILSON	10/5/10 4:00:02 PM
MYCHANGE-DD	MAXOBJECTCFG	OBJECTNAME=ASSET	WILSON	10/5/10 4:02:04 PM
MYCHANGE-APP	MAXAPPS	APP=ASSET	LWAYNE	10/5/10 4:18:21 PM
MYCHANGE-APP	MAXPRESENTATION	APP=ASSET	LWAYNE	10/5/10 4:18:20 PM

Number of Records: 4
Saved Where Clause: (dmpackagedef.active = 1 and dmpackagedef.type like '%CHANGE%')
Dynamic Where Clause:
 10/6/10 16:55:32 1 / 1

Figure 10-20 MM_ALL_EVENTS query-based report output

The report output can also be a spreadsheet (XLS), or it can be scheduled.

10.7 Admin mode

The Tivoli process automation engine requires exclusive access to the back-end database to configure structural changes to its objects. In the past, this configuration took place after shutting down the application server with the purpose of restricting user access and applying the configuration changes successfully.

With Admin mode, the application can now restrict the user access and make the corresponding database configuration changes without having to shut down the application server.

In Admin mode, cron jobs are suspended, and any event listener will stop until it is turned off.

This operating mode also affects the Migration Manager. The Migration Manager requires that Admin mode is turned on during the deployment of configuration changes made to database objects, which are also known as *structural changes*. The Migration Manager also requires that Admin mode is turned on when deploying a package that includes the MAXAPPS object, or application designer changes.

Therefore, depending on the type of package that is being deployed, the Admin mode is needed so that the changes are reflected on the user's session. However, it is always a good idea to turn this mode on during package deployment windows.

There are other considerations when working on a multiple server or cluster environment. Refer to 10.5, "Clustered environment considerations" on page 209.

10.8 Migrating hierarchical data

Historically, the migration of hierarchical data was extremely difficult. Today, that problem is solved. The following section provides a scenario for the migration of a location hierarchy. There is a unique situation that occurs with hierarchical data. That is, the object structure must be self-referencing, and it must have a database reference to itself to be self-referencing.

The following steps are necessary to create a migration for hierarchical data:

1. Identify the hierarchical data.
2. Identify the object structure to use as a template.
3. Create a self-referencing relationship.

4. Duplicate the object structure that was identified in step 2.
5. Migrate the object structure.
6. Migrate the data.

Identifying the hierarchical data for migration

In this scenario, the construction of the training facility on the 6th floor is complete. Assets now need to be placed into production, and you need to move the hierarchy from the test environment into production. Figure 10-21 shows the location hierarchy in the drill-down view.

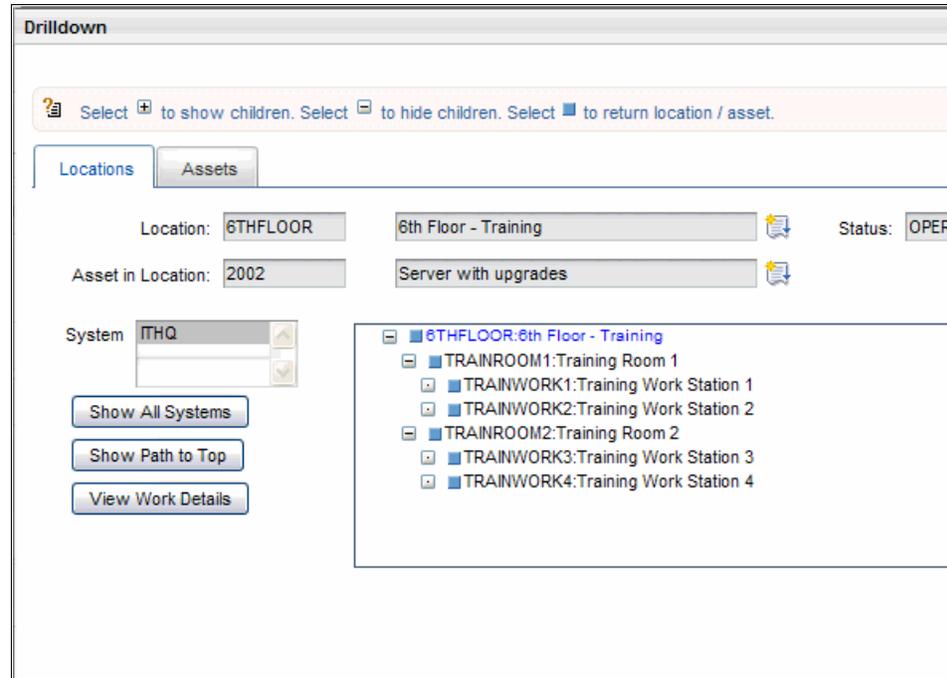


Figure 10-21 Example of location hierarchy

Identifying the object structure required as a template

After you have identified the type of hierarchical data to migrate, you will need to identify the appropriate object structure to use as a template. Because we are moving the location data, Figure 10-22 on page 223 shows that we need to use the MXOPERLOC object structure as a template for the new object structure.

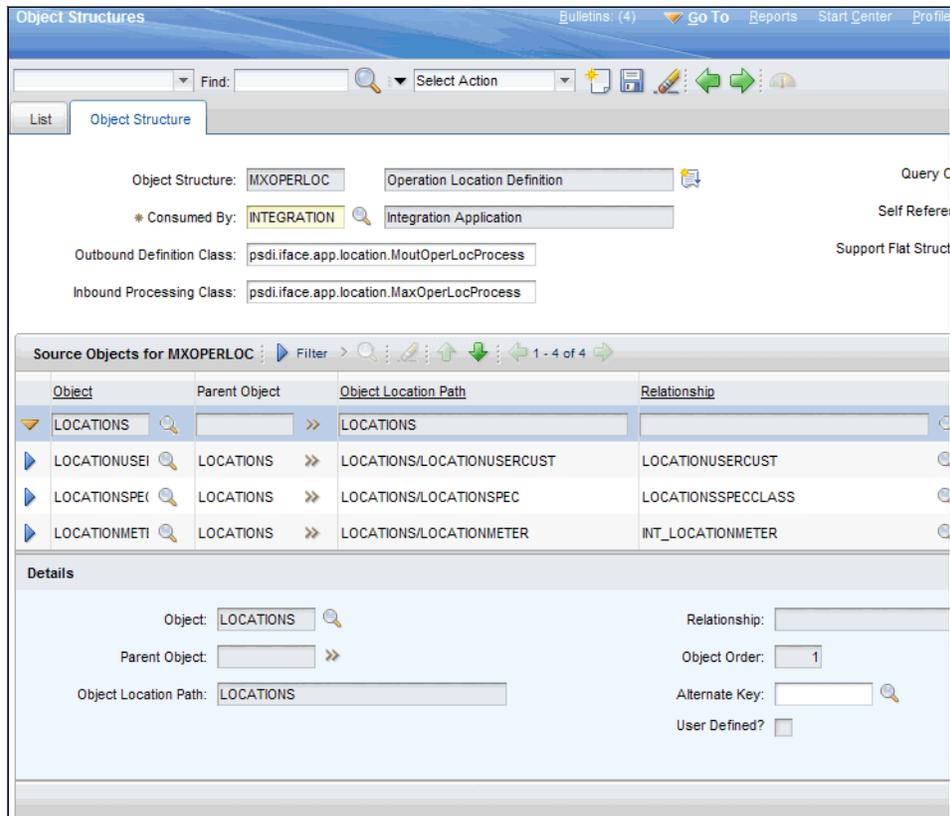


Figure 10-22 The MXOPERLOC object structure

Now that we have identified the object structure, we can determine what the relationship needs to be to create the new object structure.

Creating the new self-referencing relationship

After you have determined the nature of the relationship, as dictated by the template object structure, you will need to create the relationship using the Database Configuration application. In this case, because we use location hierarchy data, we can use the existing CHILDREN relationship to create a new relationship for self-referencing. Figure 10-23 on page 224 shows the CHILDREN relationship.

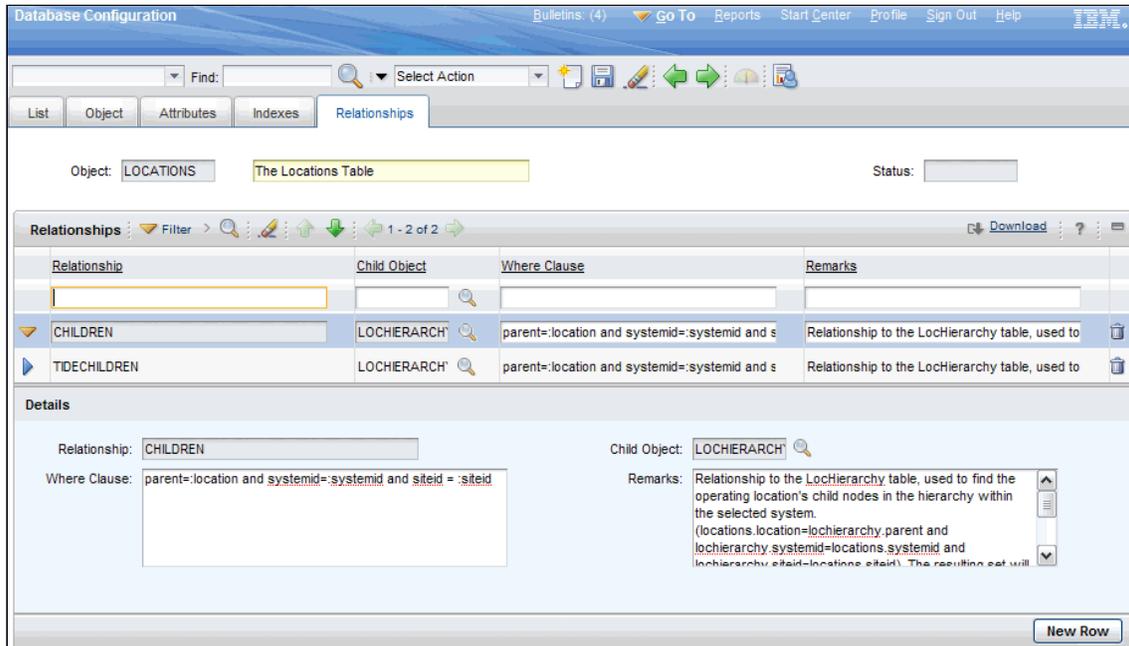


Figure 10-23 CHILDREN relationship for location

Follow these steps to set up a relationship:

1. First, click **New Row**. Enter a new relationship name. In the Child Object field, enter LOCATIONS (Figure 10-24 on page 225). Enter a new Where clause that will select all of the children for the current location, for example:

```
location in (select location from lochierarchy where parent=
:location and systemid = :systemid and siteid = :siteid
```

2. Enter remarks that define the purpose of the relationship.

Relationships: Because the child table in the relationship is the same as the parent, for every record that is selected using the Outbound processing class, every child is also selected. As a result, the processing class recursively selects all children in the tree.

Figure 10-24 on page 225 shows how we set this up.

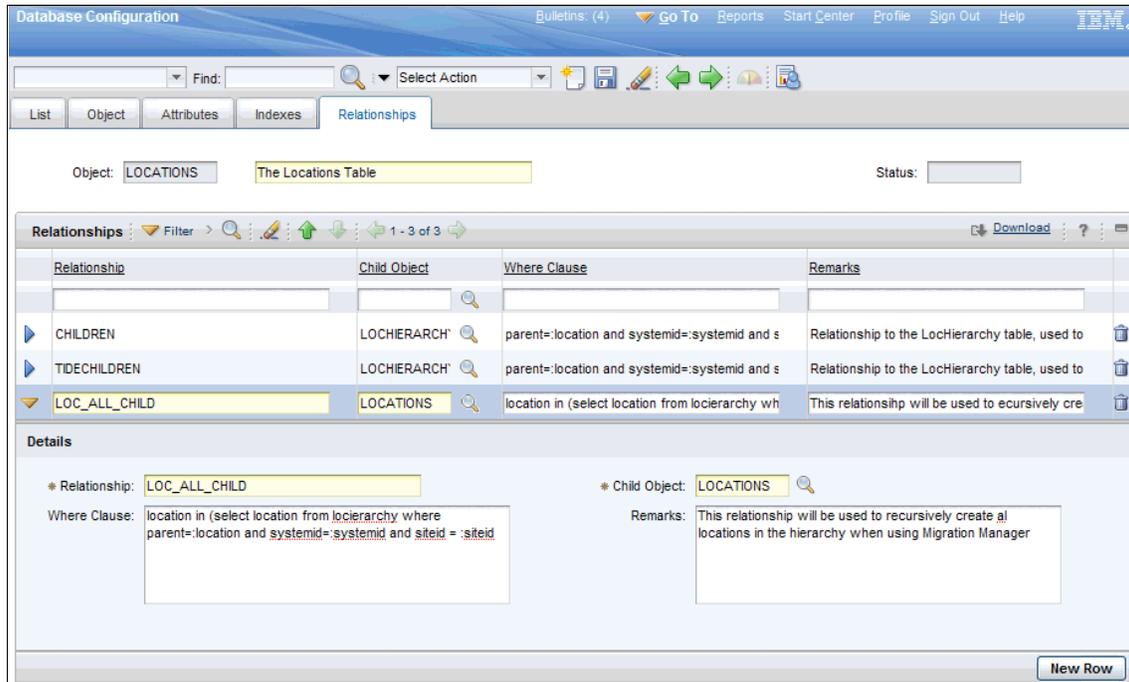


Figure 10-24 New self-referencing relationship

Save the record and return to the Object Structure application.

Creating the new object structure

Now that the new relationship is in place, you can duplicate the object structure that was identified in Figure 10-22 on page 223:

1. Click **Select Action** → **Duplicate Object Structure**. A new object structure is shown that is identical to the MXOPERLOC object structure.
2. Fill in the name of the new object structure, as indicated in Figure 10-25 on page 226.

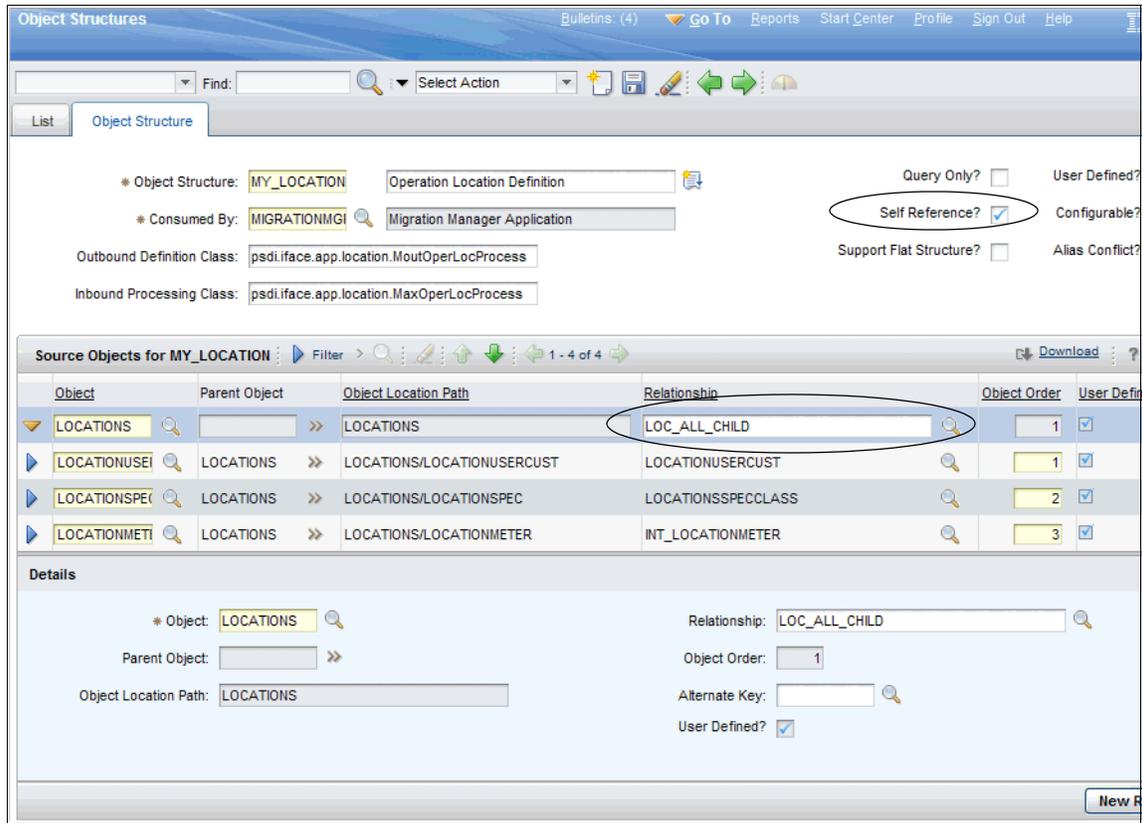


Figure 10-25 New object structure for the migration service

Note that we selected the “Self Reference?” check box and that we used the relationship that we just created.

Migrating the object structure

Now that you have created the object structure, you need to create a migration package that will migrate the object structure and the relationship.

You can either create a new set of migration groups, or you can use the predefined migrating groups. If you use the predefined migration groups, you will need to set a number of SQL Where clause values to ‘1=2’ to limit the records that are migrated (Figure 10-26 on page 227).

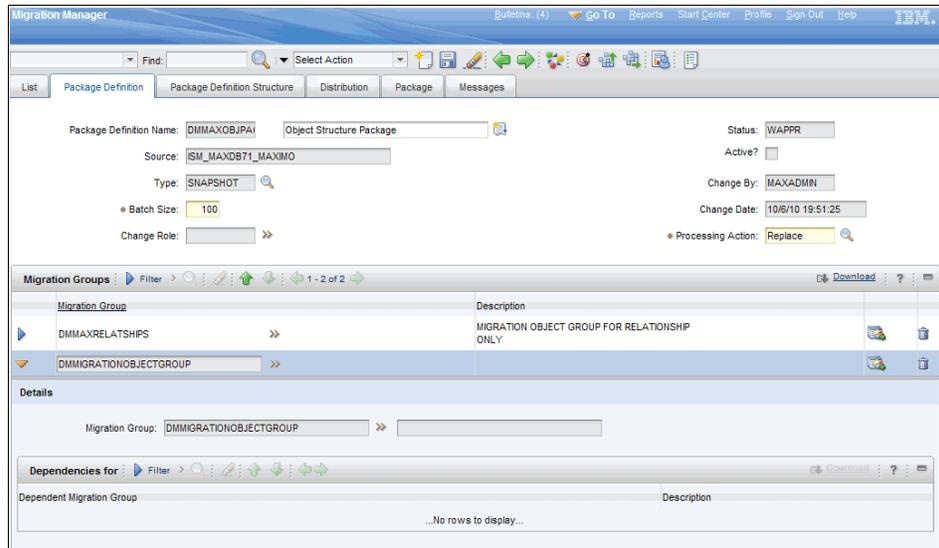


Figure 10-26 New migration package for moving the new object structure

In this example, you can see that the relationship and the object structure are the only migration objects in the group. Similarly, the groups are limited by the Where clause, as demonstrated in Figure 10-27 on page 228.

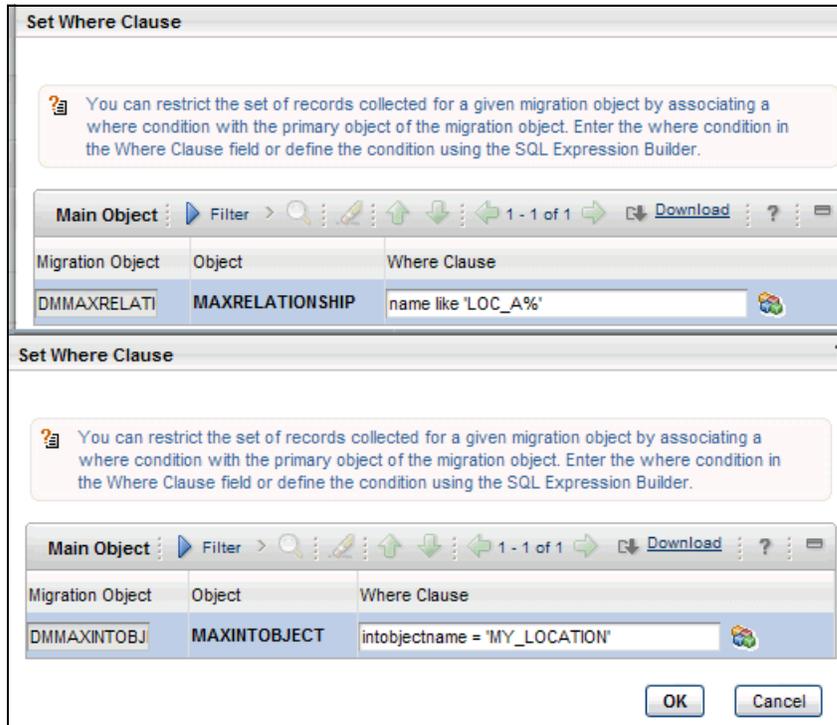


Figure 10-27 Setting the Where clause for several groups in a limited migration package

We find it easier to create new migration groups with limited object structures and simply set the SQL Where clause for the limited migration groups. This approach speeds up the package processing time.

Migrating the object structure

Finally, you can now migrate the location hierarchy. Simply create the package. The migration group only has a single object structure, as shown in Figure 10-28 on page 229.

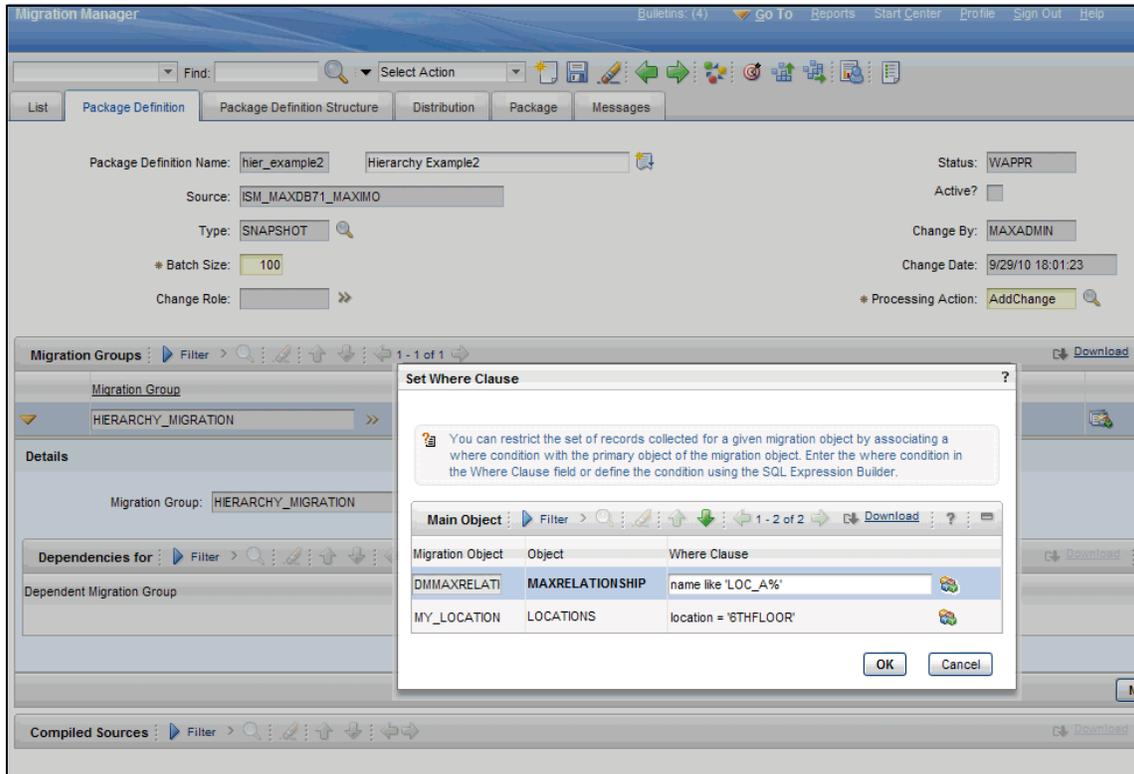


Figure 10-28 Location hierarchy migration example

At this point, simply process the migration package.

10.9 Setting up logging for the Migration Manager

The *Migration Manager Guide*

(http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tamit.doc_7.1/pdf/mam71_migration_mgr_guide.pdf) and the *IBM Maximo Asset Management, IBM Tivoli Asset Management for IT, and IBM Tivoli Service Request Manager System Administrator Guide*

(http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tamit.doc_7.1/pdf/mam71_sys_admin_guide.pdf) describe the general use of the logging application. In this section, we discuss specific ways to set up the system to provide the necessary output that is required to troubleshoot any failed migration packages.

To set up logging to provide optimal results, perform the following steps:

- ▶ Set up the root logger.
- ▶ Set up the root logging folder.
- ▶ Set up an appender for logging the Migration Manager output.
- ▶ Set the logging level for the Migration Manager service.

Manage the root logger

Perform the following steps to manage the root logger:

1. Navigate to the Logging application. Click **Select Action**, as illustrated in Figure 10-29. Click **Manage Maximo Root Logger**.

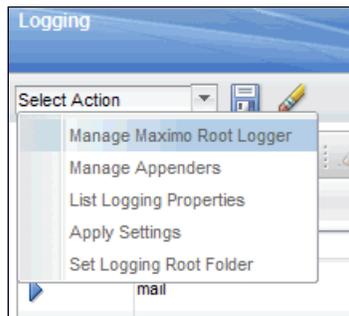


Figure 10-29 Manage the root logger

2. A new dialog box appears, as shown in Figure 10-30.

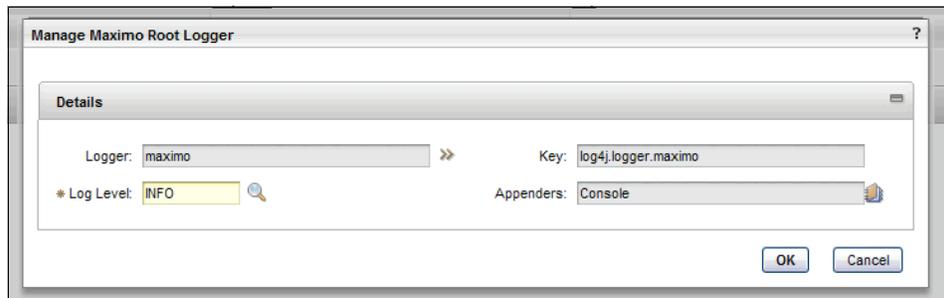


Figure 10-30 The root logger dialog box

3. You can set the root logging level and change the main appender. Be aware that if nothing specific is set for any one service, all of the logging for the system is affected.

Setting the logging folder

Next, you must set up the root logging folder. If you decide not to set up the root logging folder, be aware that the location of logs for WebSphere® by default without setting the root folder is normally in the following folder structure:

```
\\ibm\WebSphere\AppServer\profiles\ctgAppSrv01\maximo\logs
```

Figure 10-26 on page 227 shows how to set the Root logging folder if you want to set the root folder to another location. This dialog box is accessed by clicking **Select Action** → **Set Logging Root Folder**.

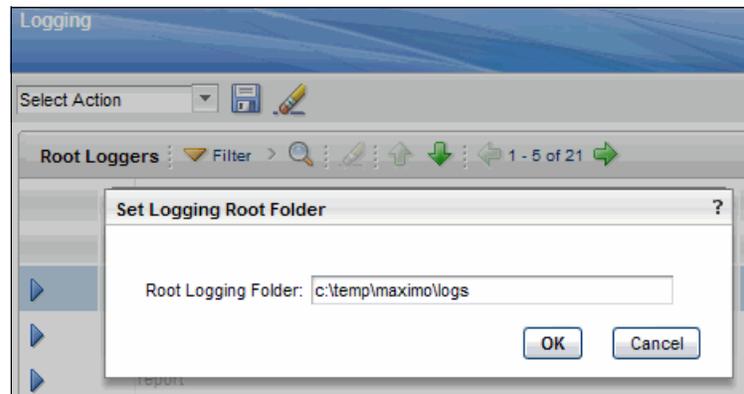


Figure 10-31 A custom root logging folder definition

Creating a new appender

The next step is to create a unique appender for you to isolate the Migration Manager activity from all other logging activity. This task will enable you to more easily read the error log when you need to isolate an error incident:

1. Navigate to the Manage Appenders dialog box, as shown in Figure 10-32 on page 232, by clicking **Select Action** → **Manage Appenders**.

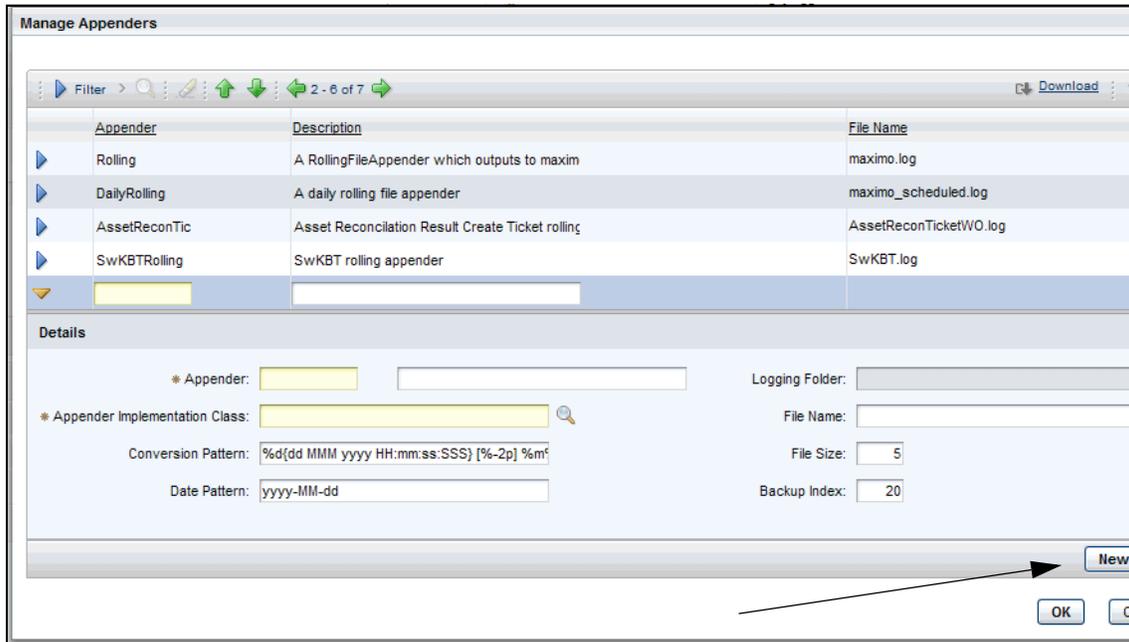


Figure 10-32 Manage Appenders dialog box

2. Click **New Row** and enter the name of the appender that you want to create. Add the description, and then, select the Appender Implementation Class.
3. Add the File Name that you want the appender to use, and then, click **OK**. An example is shown in Figure 10-28 on page 229 for your reference.

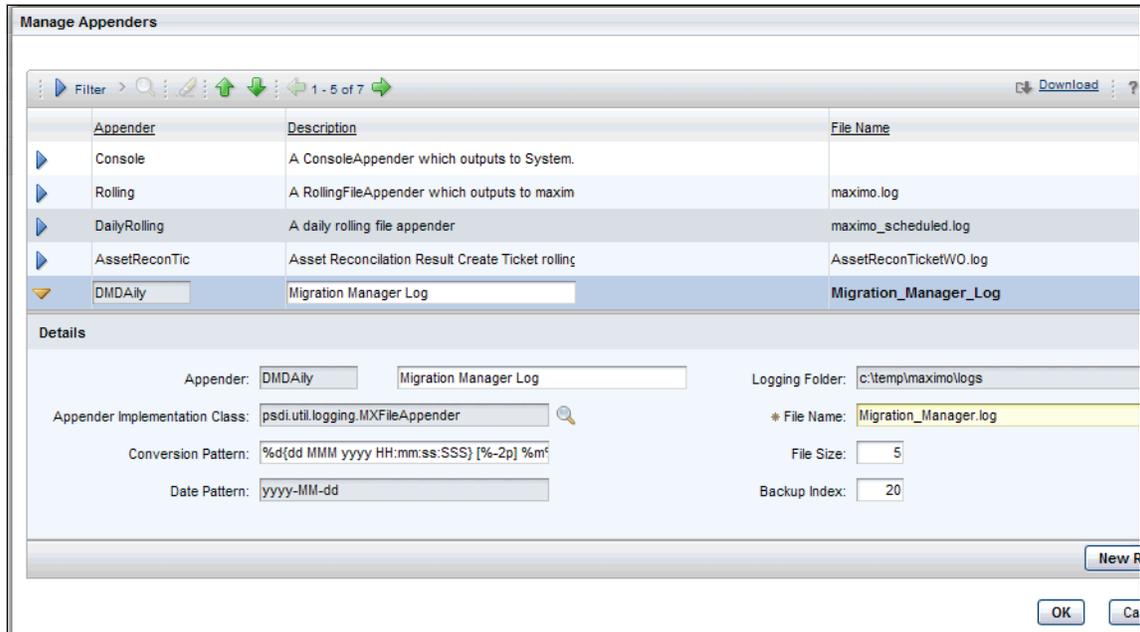


Figure 10-33 Example of a new appender

You can see that you can set the file size and the backup index, as well. We recommend using the defaults until your experience demonstrates a need for change.

Setting the logging level for the Migration Manager service

Access the specific root logger for the Migration Manager in the list search for dm:

1. Click the arrow to expand the dm entry, as shown in Figure 10-34. Set the Log Level to DEBUG.

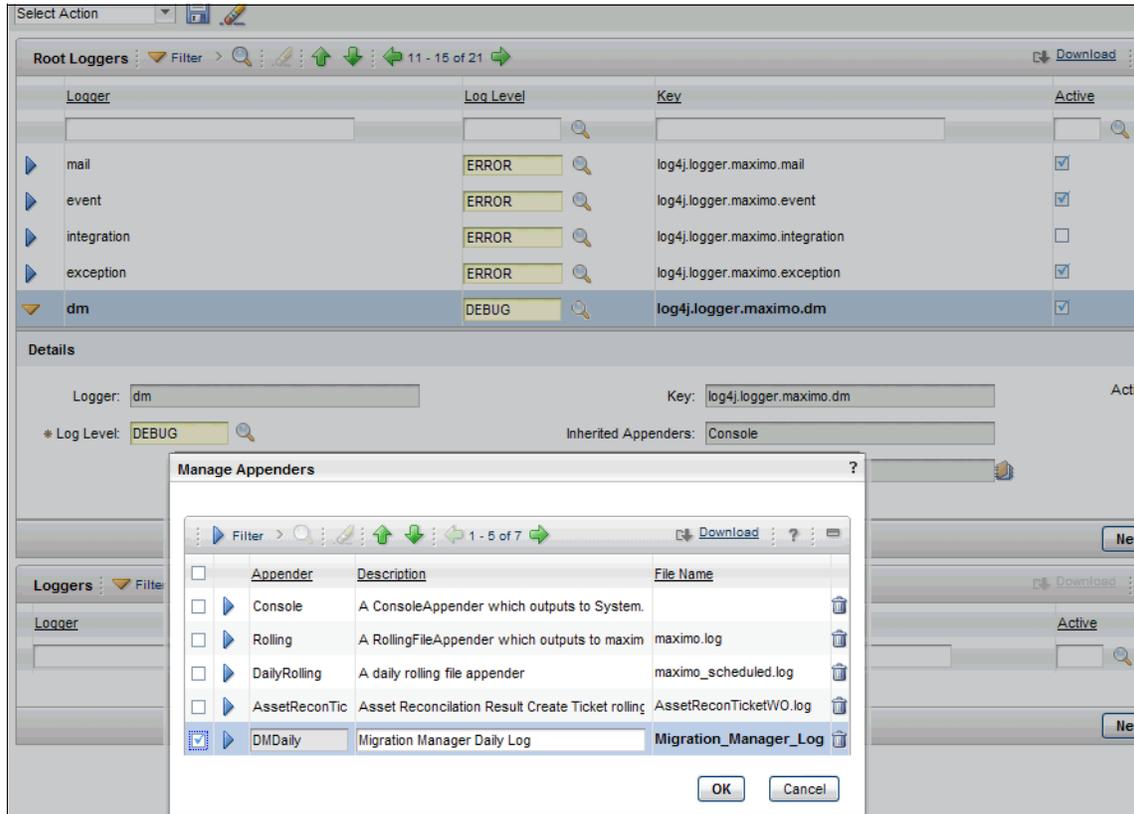


Figure 10-34 Setting the appender

2. Click the icon next to the Appender field, and select the appender that you created. Click the small square before the appender DWDaily, and then, click **OK**.
3. As instructed in the *IBM Maximo Asset Management, IBM Tivoli Asset Management for IT, and IBM Tivoli Service Request Manager System Administrator Guide*, click the **Save** icon, and then, click **Select Action** → **Apply Settings**.
4. Accept the dialog box responses, and then start to use the system.

Figure 10-35 shows the result of the previous process after processing a package with an error. Refer back to this graphic in Chapter 11, “Troubleshooting” on page 239 (shown in Figure 11-5 on page 249).

```

30 Sep 2010 17:12:03:850 [DEBUG] Thu Sep 30 17:12:03 EDT 2010*****[DmAppBean.deployPackage(). Deploy begins *****
30 Sep 2010 17:12:04:546 [INFO] Migration Manager deployment started for packageHierarchy_Example_ISM_MAXIDB71_MAXIMO_20100928160956, type CFGDATA and configuration order 1.
30 Sep 2010 17:12:04:546 [DEBUG] object name MAXRELATIONSHIP
30 Sep 2010 17:12:04:546 [DEBUG] Operation Sync
30 Sep 2010 17:12:04:546 [DEBUG] Processing record LOCATIONS-LOC_ALL_CHILD
30 Sep 2010 17:12:04:546 [DEBUG] Action is Replace
30 Sep 2010 17:12:04:562 [DEBUG] *****
30 Sep 2010 17:12:04:562 [DEBUG] object name MAXRELATIONSHIP
30 Sep 2010 17:12:04:562 [DEBUG] Operation Sync
30 Sep 2010 17:12:04:562 [DEBUG] Processing record SPRELATEDASSET-LOCPARENT
30 Sep 2010 17:12:04:562 [DEBUG] Action is Replace
30 Sep 2010 17:12:04:562 [DEBUG] *****
30 Sep 2010 17:12:04:562 [INFO] Migration Manager deployment started for packageHierarchy_Example_ISM_MAXIDB71_MAXIMO_20100928160956, type CFGDATA and configuration order 2.
30 Sep 2010 17:12:04:562 [ERROR] could not deploy configuration data for package Hierarchy_Example_ISM_MAXIDB71_MAXIMO_20100928160956. Please check log entries for more info
psdi.ut11.MXApplicationException: BMXAA1281E - object structure MY_LOCATION does not exist.
    at psdi.iface.mos.MBDMXLU11.getMosName(MBDMXLU11.java:453)
    at psdi.iface.mic.Micservice.loadData(Micservice.java:1506)
    at psdi.dm.pkg.DMPackage.deployStagingData(DMPackage.java:1414)
    at psdi.dm.pkg.DMPackage.deployPackage(DMPackage.java:906)
    at psdi.webClient.beans.dm.DMAppBean.deployPackage(DMAppBean.java:321)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:79)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:618)
    at psdi.webClient.system.session.AsyncProcess.run(AsyncProcess.java:95)
    at java.lang.Thread.run(Thread.java:810)
30 Sep 2010 17:12:04:593 [ERROR] could not import configuration records for package Hierarchy_Example_ISM_MAXIDB71_MAXIMO_20100928160956.
Please review log file for more details!
psdi.ut11.MXApplicationException: BMXAA1281E - object structure MY_LOCATION does not exist.
    at psdi.iface.mos.MBDMXLU11.getMosName(MBDMXLU11.java:453)
    at psdi.iface.mic.Micservice.loadData(Micservice.java:1506)
    at psdi.dm.pkg.DMPackage.deployStagingData(DMPackage.java:1414)
    at psdi.dm.pkg.DMPackage.deployPackage(DMPackage.java:906)
    at psdi.webClient.beans.dm.DMAppBean.deployPackage(DMAppBean.java:321)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:79)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:618)
    at psdi.webClient.system.session.AsyncProcess.run(AsyncProcess.java:95)
    at java.lang.Thread.run(Thread.java:810)
30 Sep 2010 17:12:04:593 [ERROR] Import fails for package Hierarchy_Example_ISM_MAXIDB71_MAXIMO_20100928160956 with type CFGDATA and configuration data order 2.
30 Sep 2010 17:12:04:640 [ERROR] could not deploy package Hierarchy_Example_ISM_MAXIDB71_MAXIMO_20100928160956. Please check log entries for more information.
null
30 Sep 2010 17:12:04:640 [ERROR] could not deploy package Hierarchy_Example_ISM_MAXIDB71_MAXIMO_20100928160956. Please check log entries for more information.
java.lang.Exception: psdi.ut11.MXApplicationException: BMXAA1281E - object structure MY_LOCATION does not exist.
    at psdi.dm.pkg.DMPackage.deployStagingData(DMPackage.java:1414)
    at psdi.dm.pkg.DMPackage.deployPackage(DMPackage.java:906)
    at psdi.webClient.beans.dm.DMAppBean.deployPackage(DMAppBean.java:321)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

```

Figure 10-35 Sample error log

10.10 Start Center visibility into configurations

It is beneficial to managers, developers, and implementers to be able to view the key configurations that have been set up in a development environment, not only from within a particular application, such as Workflow Designer, but also from a Start Center. This at-a-glance Start Center-based view into key configurations enables stakeholders to determine the number and extent of configurations and, consequently, prepare Snapshot migration packages.

A Start Center can be designed relatively rapidly to deliver this type of a view, provided that a consistent naming convention has been used in creating the various configurations. Queries can be constructed to retrieve the desired configurations, based on the naming convention used. Start Center result set portlets can be associated with those queries. Each portlet can be configured to display the most relevant information from the development and migration perspectives.

For example, to retrieve and display workflow processes that might be required to be migrated, this simple query can work:

```
processname like 'RB%' and active = '1'
```

This query can now be associated with a Start Center result set portlet that displays key workflow process information, such as:

- ▶ Process
- ▶ Description
- ▶ Object
- ▶ Process revision
- ▶ Enabled
- ▶ Active
- ▶ Changed by
- ▶ Changed date

When the Start Center is saved and associated with the appropriate security groups, the key information is immediately visible to users when they log in to the production environment. Figure 10-36 on page 237 shows an example Start Center (for visibility, only part of the screen is shown) on which portlets provide visibility into business objects, workflow processes, and security groups.

Accelerator Objects Filter			
Object	Extends Object	Entity	Class
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
TKOWNERHISTORY		TKOWNERHISTORY	com.ibm.tsd.app.ticket.PmOwnerHistoryS
SOLUTION		SOLUTION	com.ibm.tsd.pmsol.app.PmSolutionSet
TICKET		TICKET	psdi.app.ticket.TicketSet
PROBLEM	TICKET	PROBLEM	com.ibm.tsd.pmpo.app.PmProblemSet
SR	TICKET	SR	com.ibm.tsd.pmsrv.app.PmSrvReqSet
TKSTATUS		TKSTATUS	com.ibm.tsd.app.ticket.PmTKSLAHoldSta
RELATEDRECORD		RELATEDRECORD	psdi.app.ticket.RelatedRecordSet
WORKLOG		WORKLOG	psdi.app.ticket.WorkLogSet
CLASSTRUCTURE		CLASSTRUCTURE	com.ibm.ism.cci.app.classcatalog.CCIC
INBOUNDCOMM		INBOUNDCOMM	psdi.common.emailstnr.InboundCommSe
Set Graph Options			

Accelerator Workflow Filter			
Process	Description	Object	Process Revision
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
A61REPLY	User Replied	SR	1
A61	Request Fulfillment	SR	5
A612	Receive and Approve Request	SR	6
A612.1	Receive Service Request and Apply Response Plan	SR	3
A612.2	Validate User Profile	SR	1
A612.3	Confirm Configuration Information	SR	1
A612.4	Verify Support Status	SR	2
A612.6	Decline Further Action	SR	2
A612.7	Categorize Service Request	SR	4
A613	Fulfill or Route Service Request	SR	10
Graphical View			

Accelerator Security Groups Filter	
Group	Description
<input type="text"/>	<input type="text"/>
A61.REQUESTOR	RF Requestor Group
A61.ANALYST	RF Analyst Group
A61.ADMIN	RF Administrator Group
A61.MANAGER	RF Manager Group
A65.ANALYST	Incident Analyst Group
A65.ADMIN	Incident Administrator Group
A65.MANAGER	Incident Manager Group
A65.REQUESTOR	Incident Requestor Group
A65.OWNER	Incident Owner Group

Figure 10-36 Sample Start Center showing business objects, workflow processes, and security groups



Troubleshooting

This chapter discusses the approach for the practitioner to apply when the migration packages do not migrate as expected. We present the most common experiences we have experienced. We provide detailed techniques to help you determine the cause of the migration failure. We show the method for you to use to correct the package to reprocess it for successful migration.

This chapter provides the following sections:

- ▶ 11.1, “Common migration package failures” on page 240
- ▶ 11.2, “Methods to solve migration package failures” on page 248
- ▶ 11.3, “Techniques to prevent migration package failures” on page 254

11.1 Common migration package failures

This section details the most common occurrences for failures that we have encountered. The *Migration Manager Guide* (http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tamit.doc_7.1/pdf/mam71_migration_mgr_guide.pdf) briefly discusses the three major types of errors that are found when using the application. Package creation, distribution, and deployment are the three categories that classify these errors. This section deals with the common ways that deployment fails.

11.1.1 Package deployment fails due to installation differences

The installation and product documentation clearly states that the Migration Manager only works when the software installations are identical between the Source system and the Target system. We have experienced several occasions in which this situation is not the case. When this situation is not true, the following approach will assist you in correcting the situation.

Identifying the installation differences

Review the system information in both the Target system and the Source system. Click **Help** → **System Information** in the upper-right corner of the application (Figure 11-1).

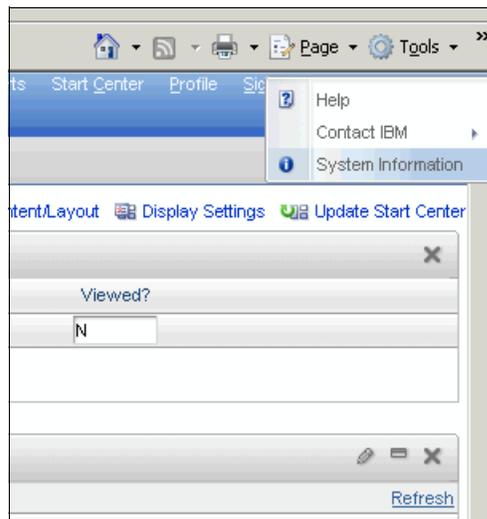


Figure 11-1 Accessing the system information

Next, review the components that are installed on the Source system and compare them to the Target system. Carefully review the information that is presented in Figure 11-2.

System Information	
App Server	IBM WebSphere Application Server 6.1
User Name	MAXADMIN
Version	Service Desk Integration MEA 7.2.0.00 Build 20091111D DB Build V7200-03 IBM Content Catalog 7.1.null.00 Build 1 DB Build V710-04 SRM Service Request Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 IBM Tivoli Configuration Management 7.2.0.01 Build 20100218D2 DB Build V7201-01 IBM Tivoli Change Management Content 7.2.0.00 Build 20091111D DB Build V7200-14 IBM Tivoli Integration Composer 7.2.1.0 Build 201003231145 DB Build V7210-02 IMBot SRM Service Desk 7.2.0.00 Build \${env.BUILD_NAME} DB Build V7200-05 SRM Service Desk Content - Best Practices 7.2.0.1 Build 20100316D2 DB Build V7201-01 SRM Service Desk Content - Classification 7.2.0.0 Build 20091111D DB Build V7200-05 IBM Tivoli Common Process Components 7.2.0.01 Build 20100109D DB Build V7201-02 SRM Service Desk Content - Best Practice Users 7.2.0.0 Build 20091111D DB Build V7200-05 Naming Reconciliation Services 1.0.0.0 Build 200907211418 DB Build V7200-04 SRM Service Desk 7.2.0.1 Build 20100316D2 DB Build V7201-04 IBM Tivoli Asset Management for IT 7.2.1.0 Build 201003231509 DB Build V7210-96 SRM Screen Capturer 7.2.0.0 Build 20091111D DB Build V7200-02 IBM Tivoli Common Process Components for Service Providers 7.2.0.0 Build 20091105D DB Build V7200-04 SRM Problem Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 SRM Solution 7.2.0.1 Build 20100316D2 DB Build V7201-01 SRM Incident Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 SRM SLA Hold 7.2.0.1 Build 20100316D2 DB Build V7201-03 SRM Service Catalog Base 7.2.0.1 Build 20100316D DB Build V7201-05 IBM Tivoli Asset Management for IT for Service Providers 7.2.1.0 Build 201003231509 DB Build V7210-11 SRM Service Desk for Service Providers 7.2.0.0 Build 20091111D DB Build V7120-03 CI PMP Content 7.2.0.00 Build 200911081214 DB Build V712-01 SRM Search 7.2.0.1 Build 20100316D2 DB Build V7201-02 SRM Service Desk Demo Data 7.2.0.0 Build 20091111D DB Build V7200-04 SRM Survey Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 IBM Tivoli ITIL V3 Support 7.2.0.0 Build 201004112300 DB Build V7200-02 IBM Tivoli UI Widgets 7.2.0.01 Build 201002251643 DB Build V7201-02 IBM Tivoli Change Management 7.2.0.01 Build 20100215D2 DB Build V7201-02 ServiceProvider 7.1.1.1 Build BUILD DB Build V7121-03 HFDB Build HF7121-01 IMI for SRM Service Desk 7.2.0.00 Build 201002200030 DB Build V7200-02 CCI PMP 7.2.0.01 Build 201002151749 DB Build V7201-05 SRM Service Catalog Content 7.2.0.1 Build 20100316D DB Build V7201-06 Base Services 7.1.1.6-LA20100506-1424 Build 20091208-1415 DB Build V7116-173 HFDB Build HF7116-0
Server OS	Windows Server 2003 5.2 build 3790 Service Pack 2
License #	BMXAA4039E - MRO Software Internal Use Only

(c) Copyright IBM Corp. 2007

OK

Figure 11-2 System Information

This example shows that many products are in the Source system. In certain cases, you might install several content packages and, after having done so, decide not to use several of the content packages. Subsequently, you decide not to install the content on the Target system. See Figure 11-3.

System Information	
App Server	IBM WebSphere Application Server 6.1
User Name	MAXADMIN
Version	Service Desk Integration MEA 7.2.0.00 Build 20091111D DB Build V7200-03 IBM Content Catalog 7.1.null.00 Build 1 DB Build V710-04 SRM Service Request Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 IBM Tivoli Configuration Management 7.2.0.01 Build 20100218D2 DB Build V7201-01 IBM Tivoli Change Management Content 7.2.0.00 Build 20091111D DB Build V7200-14 IBM Tivoli Integration Composer 7.2.1.0 Build 201003231145 DB Build V7210-02 IMBot SRM Service Desk 7.2.0.00 Build \${env.BUILD_NAME} DB Build V7200-05 IBM Tivoli Common Process Components 7.2.0.01 Build 20100109D DB Build V7201-02 SRM Service Desk Content - Best Practice Users 7.2.0.0 Build 20091111D DB Build V7200-05 Naming Reconciliation Services 1.0.0.0 Build 200907211418 DB Build V7200-04 SRM Service Desk 7.2.0.1 Build 20100316D2 DB Build V7201-04 IBM Tivoli Asset Management for IT 7.2.1.0 Build 201003231509 DB Build V7210-96 SRM Screen Capturer 7.2.0.0 Build 20091111D DB Build V7200-02 IBM Tivoli Common Process Components for Service Providers 7.2.0.0 Build 20091105D DB Build V7200-04 SRM Problem Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 SRM Solution 7.2.0.1 Build 20100316D2 DB Build V7201-01 SRM Incident Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 SRM SLA Hold 7.2.0.1 Build 20100316D2 DB Build V7201-03 SRM Service Catalog Base 7.2.0.1 Build 20100316D DB Build V7201-05 IBM Tivoli Asset Management for IT for Service Providers 7.2.1.0 Build 201003231509 DB Build V7210-11 SRM Service Desk for Service Providers 7.2.0.0 Build 20091111D DB Build V7120-03 CI PMP Content 7.2.0.00 Build 200911081214 DB Build V712-01 SRM Search 7.2.0.1 Build 20100316D2 DB Build V7201-02 SRM Service Desk Demo Data 7.2.0.0 Build 20091111D DB Build V7200-04 SRM Survey Management 7.2.0.1 Build 20100316D2 DB Build V7201-01 IBM Tivoli ITIL V3 Support 7.2.0.0 Build 201004112300 DB Build V7200-02 IBM Tivoli UI Widgets 7.2.0.01 Build 201002251643 DB Build V7201-02 IBM Tivoli Change Management 7.2.0.01 Build 20100215D2 DB Build V7201-02 ServiceProvider 7.1.1.1 Build BUILD DB Build V7121-03 HFDB Build HF7121-01 IMI for SRM Service Desk 7.2.0.00 Build 201002200030 DB Build V7200-02 CCI PMP 7.2.0.01 Build 201002151749 DB Build V7201-05 SRM Service Catalog Content 7.2.0.1 Build 20100316D DB Build V7201-06 Base Services 7.1.1.6-LA20100506-1424 Build 20091208-1415 DB Build V7116-173 HFDB Build HF7116-0
Server OS	Windows Server 2003 5.2 build 3790 Service Pack 2
License #	BMXAA4039E - MRO Software Internal Use Only
(c) Copyright IBM Corp. 2007	
OK	

Figure 11-3 System Information

Note through careful examination between the Source and Target systems that the Target system does not have the same installation packages as the Source system. When this situation occurs, you have an inherent problem with migration, because the first thing that the Migration Manager does is to check the package

manifest against the database (MAXVARS table) to make sure that the product installations are identical.

11.1.2 Package deployment fails due to incorrect Source designation

Deployments will fail if you choose a package incorrectly. This situation occurs when you use a central file server to store packages for several environments. This situation is not that uncommon and occurs most often when you develop a project and have many packages that are all located in the same folder.

Review the package manifest

The package manifest provides you with the information that is needed to analyze the contents of the package. Often, the package manifest reveals that the Source system is the consuming environment. Use of the manifest enables you to deploy the package correctly. In Figure 11-4, you see the Source system information.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <PACKAGEMANIFEST>
- <PACKAGEHEADER>
  <PKGDEFNAME>Hierarchy_Example</PKGDEFNAME>
  <SOURCE>ISM_MAXDB71_MAXIMO</SOURCE>
  <PACKAGE>Hierarchy_Example_ISM_MAXDB71_MAXIMO_20100928170057</PACKAGE>
  <BASELANGUAGE>EN</BASELANGUAGE>
  <CREATEBY>MAXADMIN</CREATEBY>
  <CREATEDATE>2010-09-28T17:00:57-04:00</CREATEDATE>
- <COMPONENTVERSION>
  <RDBMS>DB2/NT : SQL09053</RDBMS>
```

Figure 11-4 Package manifest XML

The XML tag Source contains the information that you need to validate against your Target system.

11.1.3 Package deployment fails due to inconsistent manifest data with package file name

When you download the migration package file from your Source server to your computer, the file name might change unexpectedly. This situation will create an inconsistency that causes the deployment to fail.

The name of the file changes when the browser settings rename the downloaded file if the browser sees the file already in its downloaded location. This situation happens most often with metadata.

To determine if this is the case, navigate to the download location and make a note of the filename of the package. Now, open the package and extract the manifest. If the name of the package in the manifest does not match the name of the compressed file, simply rename the package compressed file to match the information in the manifest.

11.1.4 Package deployment fails due to improperly combined object structures

The following error occurs when you associate one migration group to another migration group on which there is a dependency and you do not use the dependency. This error occurs because dependencies are processed *and committed* into the database *before* the next group that depends on that data.

Herein lies a challenge for you. If your dependent data group is large, but it depends on a small part of a large data migration group, setting up the dependency in the migration group or package will affect performance. This situation is true, because there is no way of filtering the dependent data group. Therefore, you have two choices to alleviate this problem.

First, you can create a migration package with only the particular object structure in question. Then, filter the package for the specific small record set. Migrate this package, and then migrate the original package without the dependency. Because you just processed the data on which the original set is dependent, you will have no problems.

Second, you can go ahead and create the dependency and process the data. This method will take longer and depending upon the amount of data in the dependency, it might take a long time to process the migration.

11.1.5 Security configuration migration errors

In this section, we cover the migration errors that are related to the security configuration.

Security group and Start Center association

Every security group is associated with a corresponding Start Center. If a user belongs to multiple security groups, that user might be assigned to multiple Start Centers as a result of the accumulation of Start Centers that are associated with each security group.

When migrating a security group, be careful to ensure that the associated Start Center has been previously migrated in a separate package or will precede the

security group configuration in the current package. If this action is not done, the migration of the security group will fail with the following error:

```
[ERROR] java.lang.NullPointerException at  
psdi.dm.procclass.DMMaxGroupProcess.setAdditionalData  
(DMMaxGroupProcess.java:114)
```

During the deployment of the migration package containing the security group, the Migration Manager will attempt to establish the link between the security group and its associated Start Center. Because the Start Center does not exist in the Target production environment, the Migration Manager reports the deployment error.

An alternative approach to the security group migration is to exclude the reference to the Start Center before performing the migration. Access the Object Structure application and display the DMMAXGROUP object structure. From the Select Action menu, you execute the Exclude/Include Fields action. In the pop-up dialog box that appears, you exclude the SCTEMPLATEID attribute from the MAXGROUP business object. Click OK to save the change, and close the dialog. This action will ensure that the Start Center template reference is not carried into the migration package in the Source environment. After the migration completes, you must manually establish the link between the security groups and the corresponding Start Centers. Consider this important trade-off part of the migration planning activities.

11.1.6 Security group and group reassign errors

When migrating users, the following deployment error is reported:

```
BMXAA6695E - The MBO could not be batch validated for object  
GROUPUSER. The error is GROUPNAME  
psdi.util.MXApplicationException: BMXAA0028E - Your security privileges  
do not allow access to the selected option
```

Users are migrated with the DMMAXUSER object structure. Before users are migrated, security groups are migrated with the DMMAXGROUP object structure. In the DMMAXUSER object structure, the GROUPUSER business object associates users with the security groups of which they are members. Specific security authorizations are required to assign a user to a group. A given user is granted the authorization to add other users to a chosen group.

When migrating users with the Migration Manager, the user or administrator performing the deployment of the migration package must have the authorization to assign users to a security group. If this authorization has not been granted, the validation rules against GROUPUSER Maximo Business Object (MBO) fail with the deployment error stated previously. Ensure that the user performing the

migration is an administrative user that has been granted requisite authorizations. Typically, to perform migrations, clients use the MAXADMIN user account or an account duplicated from MAXADMIN.

11.1.7 Impact of invalid or deprecated MAXVARS

Clients encountered deployment errors when the migration package included the DMMAXVARS object structure and the data for the MAXVARS table. The error reported is:

```
BMXAA4116E - Maxvar type is not valid
```

The MAXVARS table holds a number of system-wide, organization-level or site-level flags. Several of these flags can be set through various Select Action menus of the Organization application. Every MAXVARS table entry must be associated with a corresponding MAXVARTYPE table entry. The entry in the MAXVARTYPE table determines if the particular MAXVAR is system-wide, and organization level or site level. The MAXVAR business object enforces validations that ensure that the entries in the MAXVARS and MAXVARTYPE tables match.

MAXVARS and MAXVARTYPE entries are created solely through installation or upgrade programs. If the installation or upgrade of the same product has completed in a production environment, the MAXVARs must exactly match between the two production environments. The MAXVAR business object does not offer the capability to add a new MAXVAR entry into the MAXVARS table.

Consequently, the Migration Manager only supports the update of existing MAXVAR entries that are subject to the validations that are enforced by the MAXVAR business object.

Over the course of several product releases, a subset of MAXVAR has been deprecated (no longer in use with the product).

IBM has recognized a defect in the upgrade scripts supplied with Tivoli's process automation engine that causes orphan or deprecated MAXVAR entries to be left behind in the production environment after upgrading to release 7.1.x. The migration of MAXVAR that includes these entries causes the migration to fail. The defect has been remediated with the release of Maximo Base Services (MBS) Fix Pack 7.1.1.6. For more details, refer to this link on the IBM support website:

<http://www-01.ibm.com/support/docview.wss?uid=swg1IZ61692>

A preventive measure that clients can take is to execute the following SQL statement in both environments to determine that the number of MAXVARS entries matches:

```
select count(*) from maxvars
```

The following SQL statement ensures that there is at least one corresponding entry in the MAXVARTYPE table for an entry in the MAXVARS table:

```
select count(*) from maxvars where exists (select * from maxvartype  
where maxvars.varname=maxvartype.varname)
```

The number of records returned by the two SQL statements must match.

Another consideration to successfully migrate MAXVARS is to track the changes that have been made in the development environment and only migrate those MAXVARs that were modified.

11.1.8 Attached documents and document types

When migrating a Snapshot package with the Replace processing action, a client encountered the following deployment error:

```
Error occurred while processing DMDOCTYPES (Object Structure number 1.  
Primary Object is: DOCTYPES. Key is: Attachments). Error is: BMXAA0876E  
- Cannot delete this record
```

The DMDOCTYPES object structure supports the migration of attached document types. The DOCINFO business object is a member of this object structure and the child of the DOCTYPES business object. When migrating a Snapshot package with the Replace processing action, the Migration Manager will add entries into the DOCINFO table, if entries do not exist in the underlying Target environment database. The Migration Manager will update existing entries into the DOCINFO table and, finally, will attempt to delete those entries in the DOCINFO table that are not present in the XML document contained within the migration package.

If the Target production environment contains living attached documents, so that the entry in DOCINFO is linked to an actual application record, such as a service request or a purchase order, the DOCINFO business object validation rule prevents any attempt to delete that DOCINFO record. This situation results in the Snapshot migration package failing deployment with the error shown previously.

You can use the following preventive measures:

- ▶ Perform the initial migration of DOCTYPES before the production goes live. This step ensures that there are no living attached documents tied to an application record.
- ▶ Perform all Snapshot migrations with the AddModify processing action before and after production goes live. This option ensures that the Migration Manager will not attempt to delete DOCINFO records that are present in the Target database but not present in the XML document that is being processed during deployment.

11.2 Methods to solve migration package failures

You can use the following methods to remediate the package installation errors that are described in the preceding section. The order is not important, and you can use any combination of methods as the case requires. When you encounter an error, make note of the error that is given in the application on the Messages tab.

11.2.1 Use of the Logging application

The following sections explain the Logging application.

The Migration Manager Guide

The Migration Manager Guide

(http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.mam.doc_7.1/pdf/mam71_migration_mgr_guide.pdf) briefly describes how to access and set up the Logging application to create log files for you to use in system analysis when encountering an error. Refer to “Migration Manager application logger” in the fourth section in Chapter 4, “Troubleshooting migration”, of this manual to see how to set up the logging application for the Migration Manager specifically. This section describes how to analyze the log files and instructs you to take various actions based on what information the log file contains.

Reading the log file

When you encounter an error in the migration, immediately access the log file that you set up in the logging application. When you open the log file, navigate to the end and work backwards to find the error. It will look similar to Figure 11-5 on page 249. Normally, an error will create several stack traces that you can identify out of the standard log file entries. When you encounter the error, look carefully for text similar to the following example.

Example 11-1 Log file

```
[9/28/10 17:06:55:359 EDT] 00000092 SystemOut    0 28 Sep 2010
17:06:55:359 [ERROR] Import fails for package
Hierarchy_Example_ISM_MAXDB71_MAXIMO_20100928170057 with type CFGDATA
and configuration data order 4.
[9/28/10 17:06:55:390 EDT] 00000092 SystemOut    0 28 Sep 2010
17:06:55:390 [ERROR] Could not deploy package
Hierarchy_Example_ISM_MAXDB71_MAXIMO_20100928170057. Please check log
entries for more information.
```

Note the following information:

- ▶ First, the phrase “Import fails for package” will identify the section of the log that will alert you to the location of the error in the package.
- ▶ Second, the package file identify that will direct you to the offending file is in this statement (Figure 11-5).

```
[9/28/10 17:06:55:265 EDT] 00000092 SystemOut    0 28 Sep 2010 17:06:55:265 [DEBUG] *****
[9/28/10 17:06:55:265 EDT] 00000092 SystemOut    0 28 Sep 2010 17:06:55:265 [INFO] Migration Manager deployment started for packageHierarchy_Example_ISM_MAXDB71_MAXIMO_20100928170057, type
CFGDATA and configuration order 4.
[9/28/10 17:06:55:281 EDT] 00000092 SystemOut    0 28 Sep 2010 17:06:55:281 [ERROR] Could not deploy configuration data for package Hierarchy_Example_ISM_MAXDB71_MAXIMO_20100928170057.
Please check log entries for more information.
pdi.util.MXApplicationException: BMXAA1281E - Object Structure MY_LOCATION does not exist.
  at pdi.iface.msc.MboXMLUtil.getName(MboXMLUtil.java:453)
  at pdi.iface.msc.MboXMLUtil.loadData(MscService.java:1506)
  at pdi.dm.pkg.DMPackage.deployStagingData(DMPackage.java:1414)
  at pdi.dm.pkg.DMPackage.deployPackage(DMPackage.java:906)
  at pdi.webclient.beans.dm.DMAppBean.deployPackage(DMAppBean.java:321)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:79)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:618)
  at pdi.webclient.system.session.AsyncProcess.run(AsyncProcess.java:95)
  at java.lang.Thread.run(Thread.java:810)
[9/28/10 17:06:55:359 EDT] 00000092 SystemOut    0 28 Sep 2010 17:06:55:359 [ERROR] Could not import configuration records for package Hierarchy_Example_ISM_MAXDB71_MAXIMO_20100928170057.
Please review log file for more detail.
pdi.util.MXApplicationException: BMXAA1281E - Object Structure MY_LOCATION does not exist.
  at pdi.iface.msc.MboXMLUtil.getName(MboXMLUtil.java:453)
  at pdi.iface.msc.MboXMLUtil.loadData(MscService.java:1506)
  at pdi.dm.pkg.DMPackage.deployStagingData(DMPackage.java:1414)
  at pdi.dm.pkg.DMPackage.deployPackage(DMPackage.java:906)
  at pdi.webclient.beans.dm.DMAppBean.deployPackage(DMAppBean.java:321)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:79)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:618)
  at pdi.webclient.system.session.AsyncProcess.run(AsyncProcess.java:95)
  at java.lang.Thread.run(Thread.java:810)
[9/28/10 17:06:55:390 EDT] 00000092 SystemOut    0 28 Sep 2010 17:06:55:390 [ERROR] Import fails for package Hierarchy_Example_ISM_MAXDB71_MAXIMO_20100928170057 with type CFGDATA and
configuration data order 4.
[9/28/10 17:06:55:390 EDT] 00000092 SystemOut    0 28 Sep 2010 17:06:55:390 [ERROR] Could not deploy package Hierarchy_Example_ISM_MAXDB71_MAXIMO_20100928170057. Please check log entries
for more information.
null
[9/28/10 17:06:55:406 EDT] 00000092 SystemOut    0 28 Sep 2010 17:06:55:406 [ERROR] Could not deploy package Hierarchy_Example_ISM_MAXDB71_MAXIMO_20100928170057. Please check log entries
for more information.
java.lang.Exception: pdi.util.MXApplicationException: BMXAA1281E - Object Structure MY_LOCATION does not exist.
  at pdi.dm.pkg.DMPackage.deployStagingData(DMPackage.java:1542)
  at pdi.dm.pkg.DMPackage.deployPackage(DMPackage.java:906)
  at pdi.webclient.beans.dm.DMAppBean.deployPackage(DMAppBean.java:321)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:79)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:618)
  at pdi.webclient.system.session.AsyncProcess.run(AsyncProcess.java:95)
  at java.lang.Thread.run(Thread.java:810)
Caused by:
pdi.util.MXApplicationException: BMXAA1281E - Object Structure MY_LOCATION does not exist.
  at pdi.iface.msc.MboXMLUtil.getName(MboXMLUtil.java:453)
  at pdi.iface.msc.MboXMLUtil.loadData(MscService.java:1506)
  at pdi.dm.pkg.DMPackage.deployStagingData(DMPackage.java:1414)
  ... 8 more
[9/28/10 17:06:55:546 EDT] 00000092 SystemOut    0 28 Sep 2010 17:06:55:546 [DEBUG] Tue Sep 28 17:06:55 EDT 2010*****finally block *****
[9/28/10 17:07:12:859 EDT] 00000056 SystemOut    0 28 Sep 2010 17:07:12:859 [INFO] CLASSIFICATION: mbosets (3), mboos (5)
[9/28/10 17:07:12:859 EDT] 00000056 SystemOut    0 28 Sep 2010 17:07:12:859 [INFO] CLASSSTRUCTURE: mbosets (2), mboos (4)
[9/28/10 17:07:12:859 EDT] 00000056 SystemOut    0 28 Sep 2010 17:07:12:859 [INFO] CRONTASKDEF: mbosets (38), mboos (76)
[9/28/10 17:07:12:859 EDT] 00000056 SystemOut    0 28 Sep 2010 17:07:12:859 [INFO] CRONTASKINSTANCE: mbosets (50), mboos (105)
[9/28/10 17:07:12:859 EDT] 00000056 SystemOut    0 28 Sep 2010 17:07:12:859 [INFO] CRONTASKPARAM: mbosets (46), mboos (78)
[9/28/10 17:07:12:859 EDT] 00000056 SystemOut    0 28 Sep 2010 17:07:12:859 [INFO] DMCFGGROUP: mbosets (6), mboos (7)
[9/28/10 17:07:12:859 EDT] 00000056 SystemOut    0 28 Sep 2010 17:07:12:859 [INFO] DMPDEPENDENCY: mbosets (1), mboos (1)
[9/28/10 17:07:12:859 EDT] 00000056 SystemOut    0 28 Sep 2010 17:07:12:859 [INFO] DMPDEPENDENCY: mbosets (1), mboos (1)
```

Figure 11-5 Reading the log file and finding the error

Interpreting the error

Now that you know the location for the file, open the Migration Package with a compressed file editor, and then, open the offending XML file (Figure 11-6).

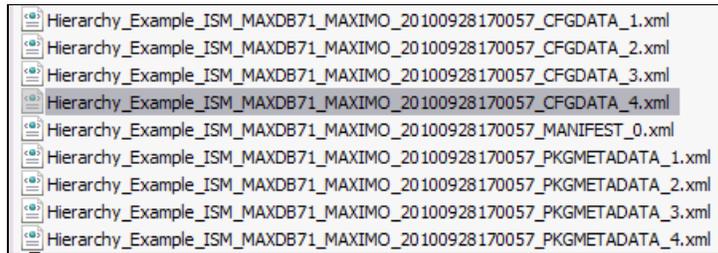


Figure 11-6 Configuration file with error identified

So, as instructed in the log file, CFGDATA number 4 is where the error lies.

Analyzing the solution

Open the XML file in your editor or viewer of choice. In Figure 11-7, you can see how the XML looks with Internet Explorer.

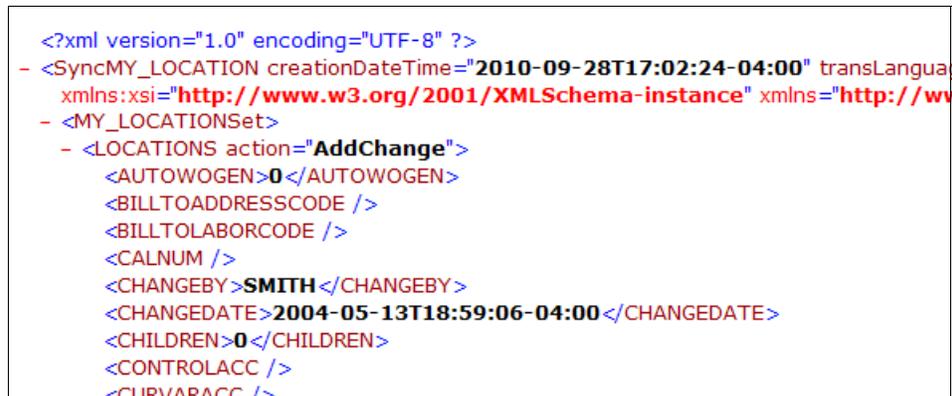


Figure 11-7 Examining the XML file

At this point, refer again to the original error, as shown in Example 11-2.

Example 11-2 Original error

BMXAA1281E - Object Structure MY_LOCATION does not exist.

Note at the top of the XML file that the action is to the SYNC the Object Structure called MY_LOCATION.

When a synchronization action takes place, the system requires that the data is in the system or, at least, has a validation reference point.

In this case, the validation failed because the Migration Manager did not yet process the Object Structure itself. Either the data was not part of the package, or the processing order was incorrect. In this case, the Object Structure definition was not included (Figure 11-8).

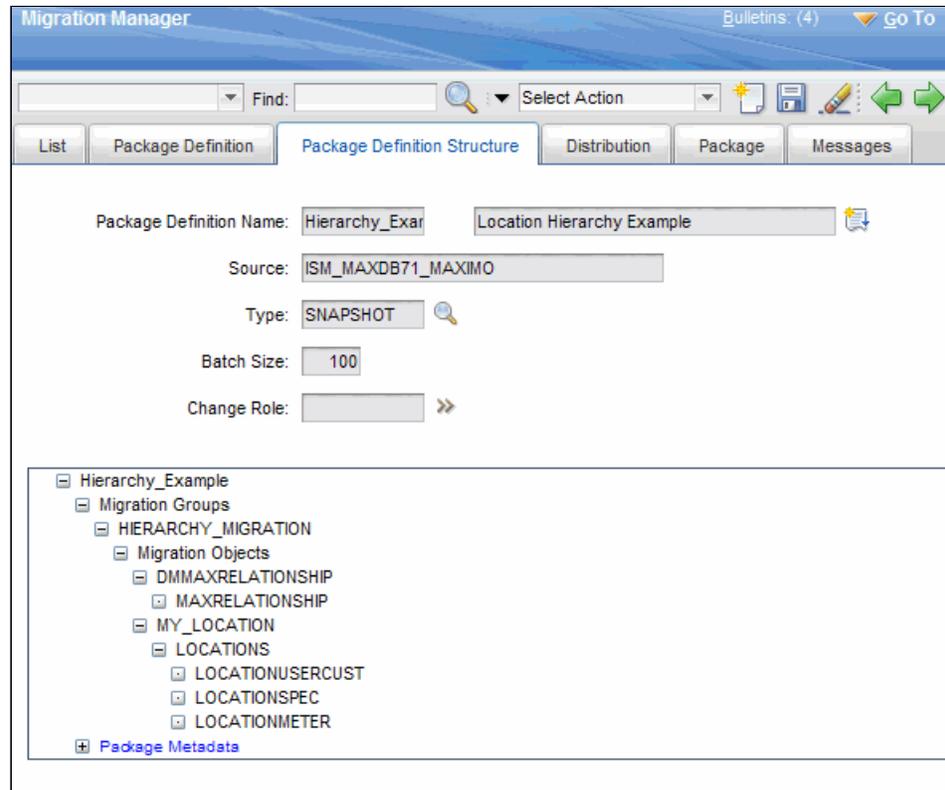


Figure 11-8 Package definition structure missing migration object definition

This situation occurs when new object structures are created and not migrated as part of the Data Dictionary migration. You can remedy this situation in one of two ways.

Applying the solution

First, you can create a dependency to the migration group for the data dictionary. In this isolated scenario, this approach is time consuming and not recommended.

Second, you can simply add the object structures that are required to your migration package definition. This technique allows you to migrate discrete data

artifacts without requiring you to migrate the entire database. Doing so, however, will cause this same error, because the way in which the migration groups are processed does not include a re-fetch to the processed data, so the package will again fail.

The solution is to create a separate package for only the DMMAXINTOBJ object structure. Create a migration group with this structure alone. Then, create a package with only that new group. After these steps are complete, you can then migrate the package separately *prior* to the migration of the solution package.

Figure 11-9 shows how to create this package.

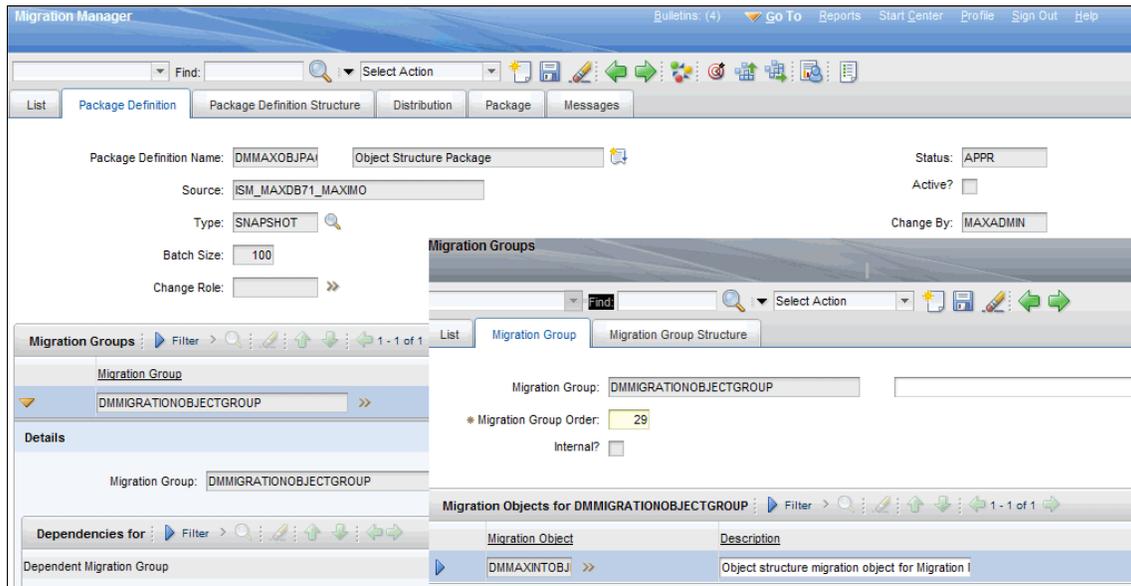


Figure 11-9 Object structure migration group

This approach demonstrates the flexibility and power of the Migration Manager and its use when troubleshooting various migration scenarios.

11.2.2 Modifying the migration package XML

In 11.1, “Common migration package failures” on page 240, we discussed the problem of installations that differ. The Migration Manager was never intended to support migration across differing installation packages; however, we recognize that there are times when this situation cannot be avoided.

To work around this situation, perform these steps:

1. Modify the Data Dictionary SQL Where clause on the DMMAXVAR object structure and filter out the entry for the missing installation package. Repeat this process for all other object structures.
2. Go to the entry in the Manifest XML file, and remove the differing package entry.
3. Modify the various XML files that have limited information about content that does not filter out of the package through the SQL Where clause modification.
4. Repeat the process until all of the differing XML tag entries are removed.

XML file identification

Refer to the previous section and Figure 11-5 on page 249 to determine the XML file that contains the error.

Modifying package manifest XML

First, you must open the XML file. Then, remove the offending entry by using an XML editor that allows you to easily manipulate the file.

Direct XML modification

After you have corrected the manifest, you need to correct the XML files. Reprocess the package, and note the subsequent error. Open the log file and identify the offending entry. Search for the key in the log file that is identified in the error message. Figure 11-10 shows an entry with a highlighted key.

```
{19:46:342 [DEBUG] *****
{19:46:348 [DEBUG] Object name HAXMESSAGES
{19:46:348 [DEBUG] Operation Sync
{19:46:348 [DEBUG] Keys StatusChangeFailure~solution
{19:46:355 [DEBUG] Action is Replace
{19:46:355 [DEBUG] *****
{19:46:362 [DEBUG] Object name HAXMESSAGES
{19:46:362 [DEBUG] Operation Sync
{19:46:362 [DEBUG] Keys srvdefShouldBeActive~pmsscrrvdef
{19:46:365 [DEBUG] Action is Add
{19:46:365 [DEBUG] *****
```

Figure 11-10 Key on which the failure occurred

The log file will next identify the XML file (as noted in Figure 11-5 on page 249). Open the XML file, and search for the key. Either fix the error or remove the entry, as required. Save the file, and replace it in the compressed file package.

Repeat the process until package migrates successfully

Reprocess the package again.

Important: This process is time consuming and tedious. Be patient.

Not all package metadata is readily identifiable. The data validation process is maintained by the Maximo Business Objects (MBOs). Therefore, you must be patient and take care to process the packages sequentially, until all of the packages process. Then, you will have a “clean” environment that you can move forward into production.

11.3 Techniques to prevent migration package failures

In this section, we discuss the ways in which you can use the knowledge presented so far to prevent migration package errors. The knowledge presented in this publication is cumulative to this point, and we hope that you will have successful migration experiences with this information.

11.3.1 Start with identical environments

To whatever extent possible, always have identical environments. We strongly recommend that you refrain from trying to migrate between differing environments from a product standpoint. Migrating between differing environments most often applies to secondary “content” installations.

11.3.2 Ensure that you understand migration group and object structure relationships

The object structures presented to you with the product are thoroughly tested and work as designed. However, as the scenarios presented in this book demonstrate, often you will migrate portions of data and not all of the packages all at one time. Simply because IBM presents you with a complete package with an entire object structure that consists of all dependencies does not mean that you must process all packages with all dependencies.

When you create your packages, do so with the understanding that dependent data can be migrated first in a separate package. When you do this (and the previously migrated configurations have not changed), there is no need to re-migrate that data, which is what occurs when you include that migration group as a dependency.

Taking this view will not only help to prevent failures but will also greatly increase the speed of your migrations.

11.3.3 Making separate packages work in a dependent fashion

We have discussed how to understand the structural relationship between migration groups. With this understanding, you can create separate packages for each of the various groups that require sequential processing to quickly and accurately migrate small portions of configuration data.

11.3.4 Making use of the SQL Where clause for package groups

When you have assembled all of the various packages that you require to migrate your configuration changes, the last step is to modify the SQL Where clause to limit the package content to only those configuration changes that you want to migrate.

The proper use of the SQL Where clause will ensure that you migrate only the data that is required and will ensure that dependent data groups have the data that is required for proper validation (Figure 11-11).

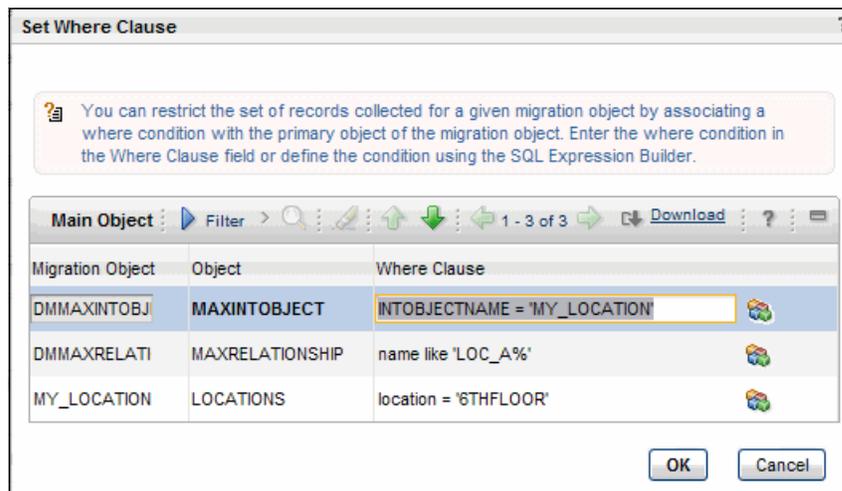


Figure 11-11 SQL Where clause example



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 Web server. Point your Web browser at:

<ftp://www.redbooks.ibm.com/redbooks/SG247906>

Alternatively, you can go to the IBM Redbooks website at:

ibm.com/redbooks

Select **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG247906.

Using the Web material

The additional Web material that accompanies this book includes the following files:

<i>File name</i>	<i>Description</i>
------------------	--------------------

SG247906.zip

Zipped migration samples

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space: 10 MB minimum

Operating System: Windows/Linux/UNIX

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

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 websites are also relevant as further information sources:

- ▶ *IBM Maximo Asset Management 7.1, IBM Tivoli Asset Management for IT 7.1, IBM Tivoli Change and Configuration Management Database 7.1.1, IBM Tivoli Service Request Manager 7.1 Application Developer Guide:*
http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tamit.doc_7.1/pdf/mam71_app_dev_guide.pdf
- ▶ *Migration Manager Guide:*
http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.mam.doc_7.1/pdf/mam71_migration_mgr_guide.pdf
- ▶ *IBM Maximo Asset Management, IBM Tivoli Asset Management for IT, IBM Tivoli Service Request Manager System Administrator Guide:*
http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tamit.doc_7.1/pdf/mam71_sys_admin_guide.pdf
- ▶ Best practices publication for data loading tools and capabilities:
<http://www.ibm.com/developerworks/wikis/download/attachments/130515354/TpaeEcosystemDataIntegrationBestPractices.pdf?version=2>
- ▶ Object structure support for attached documents:
<http://www-01.ibm.com/support/docview.wss?uid=swg24027858>
- ▶ Migration Manager V7.1.1.5 features:
http://www-01.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&dc=DB560&dc=DB520&uid=swg21393047&loc=en_US&cs=utf-8&lang=en
- ▶ Migration Manager V7.1.1.5 ticket templates:
http://www-01.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&dc=DB560&dc=DB520&uid=swg21393063&loc=en_US&cs=utf-8&lang=en

- ▶ Migration Manager V7.1.1.5 classifications:
http://www-01.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&dc=DB560&dc=DB520&uid=swg21393048&loc=en_US&cs=utf-8&lang=en
- ▶ Log setup for Migration Manager:
<http://www-01.ibm.com/support/docview.wss?uid=swg21397377>
- ▶ Start Center migration:
<http://www-01.ibm.com/support/docview.wss?uid=swg21427580>
- ▶ Change size limit when uploading large Migration Manager packages:
<http://www-01.ibm.com/support/docview.wss?uid=swg21408312>
- ▶ Migration Manager preview V7.1.1.6:
<http://www-01.ibm.com/support/docview.wss?uid=swg21414562>
- ▶ Run Integrity Checker before Migration Manager:
<http://www-01.ibm.com/support/docview.wss?uid=swg21299691>
- ▶ Oracle length semantics affects Migration Manager:
<http://www-01.ibm.com/support/docview.wss?uid=swg21411696>
- ▶ No support for IBM Maximo Enterprise Adapter (MEA) interface table migration:
<http://www-01.ibm.com/support/docview.wss?uid=swg21299697>
- ▶ Migrating conditions:
<http://www-01.ibm.com/support/docview.wss?uid=swg21389955>

How to get IBM Redbooks publications

You can search for, view, or download IBM Redbooks publications, Redpapers, web docs or Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this website:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

A xii, 1, 29, 53, 84, 109, 140, 177, 199, 245
Actions 148, 170
Admin Mode On 137
Application Designer 63, 110, 112
APPLICATION migration groups 113
APPLICATIONAUTH object structure 78
APPLICATIONS group 135
APPLICATIONS migration group 213
applications, creating 63
APPLYSLA key value 111
APPSECURITY 56, 113, 203
 duplicating 56
 migration group 65
APPSECURITY group 126
APPSECURITY migration group 113
Automation Scripts 170

B

BPM migration group 89

C

CATALOGNAME service catalog 178
Catalogs 181
CatalogTemplate 188
Change management 198
Change migration packages 205
Change package 210
change package 198
Change Package Event Tracking 41
Circular Relationships 195
classification data, migrating 158
Classifications 155, 170
communication template 96
Communication Templates 142, 148
communication templates, accessing 85
Condition Expression Manager 53
CONDITION table 124
Conditional Expression Manager 62
Conditional Expression Manager. 69
conditional expressions 53, 124
conditions 28, 30, 52, 115, 150, 176, 216, 260

 managing 69
configuration items (CIs) 154, 200
Configure Conditional Properties 64
Control Properties 63
Create New/Modify Template 132
Create Report 218
cron task 86, 99

D

Data Dictionary 23, 30, 130, 140, 194, 251
 SQL 253
Data Dictionary content 134
Data Restrictions 54
DATADictionary 55, 127, 203
 group 134
 migration group 123, 213
dependency groups 135
deprecated MAXVARS 246
DMACTION object structure 87, 94, 148, 186
DMACTIONGROUP object structure 87, 94, 148, 186
DMAPPLICATIONAUTH migration object 80
DMAPPLICATIONAUTH object structure 78
DMCLASSIFICATION 195
DMCLASSIFICATION object structure 155, 157, 185
DMCOMMTEMPLATE object structure 88, 98, 105, 149
DMCONDITION migration object 128
DMCONDITION object structure 55, 72
DMCRONTASKDEF object structure 100
DMCTRLGROUP 203
 object structure 65, 67
DMCTRLGROUP migration object 129
DMCURRENCY object structure 32
DMDOCTYPES object structure 247
DMESCALATION object structure 89, 94, 98–99
DMLANGUAGE object structure 203
DMMAXGROUP migration object 129
DMMAXGROUP object structure 56, 71, 120, 181, 185
DMMAXINTOBJECT migration object 72
DMMAXINTOBJECT object structure 78, 121, 133

DMMAXLAUNCHENTRY migration object 128
DMMAXLAUNCHENTRY object structure 182
DMMAXMENU migration object 128
DMMAXMODULES migration object 128
DMMAXSERVSECURITY migration object 129
DMMAXUSER migration object 129
DMMAXUSER object structure 56, 58, 245
DMPERSON object structure 55, 88
DMPKGEVENTS object structure 217
DMROLE object structure 88, 98, 148
DMSCTEMPLATE migration object 128
DMSECURITYRESTRICT object structure 72
DMSECURITYRESTRICT object structure 71, 74
DMSIGOPTFLAG migration object 129
DMSIGOPTFLAG object structure 183
DMSIGOPTFLAG migration object 129
DMSIGOPTFLAG object structure 60, 65, 67, 182
DMTKTEMPLATE object structure 184
DMWFPROCESS object structure 149, 184
DOCINFO records 248
DOCINFO table 105
Duplicate Object Structure 225

E

E xiii
EAR file, building 106
E-mail Account Creation service offering 177
embedded URLs 208
ESCALATION cron task 100, 102
escalationpoint 104
escalations, accessing| 85
Event Tracking Records, resetting 207
Event Tracking Records, viewing 215
Expression Builder 175

F

Failure Reporting 61

G

GL Components 54
Global Data Restriction 70
global data restrictions 68
global security restrictions 76
 migrating 121
groups xvi, 7, 35, 51, 84, 113, 140, 156, 165, 178,
199, 244
 adding new groups 55

migration 56

H

hierarchical data, migrating 221
HIERARCHYPATH 161
HIERARCHYPATH non-persistent field 161

I

Integration Framework (MEA) 199–200
invalid MAXVARS 246
ISMACCEPT process 147

J

Java deployment Enterprise Archive (EAR) 106
Java Virtual Machines (JVM) 25
Job Plan Plans 194
Job Plans 169

L

L_ Multi-language table 202
LANGUAGE table 203
Lightweight Directory Access Protocol (LDAP) 76
log files, reading 248
Logging application 248
 reading the log file 248
LONGDESCRIPTION table 203

M

Manage Appenders 231
Maximo Asset Management 200
Maximo Business Objects (MBOs) 245, 254
Maximo EAR 200
MAXOBJECTCFG 46, 127, 216
MAXPRESENTATION table 122
MAXVARS table 243
MAXVARTYPE table 246
Meta Data 194
migration xi, 3, 29, 51, 84, 109, 140, 155, 165, 177,
197, 239, 248, 258
 groups 56
Migration Groups 66, 80
Migration Manager 66, 73, 165, 173, 201, 206
migration object structure 133
Migration Object Structures application 135
migration of DOCTYPES 248
migration order values 193
Migration Package 178

migration package failures 248
migration plan document 24
MIGRATIONMGR 133
multiple XML files 199
MXOPERLOC object structure 222
MYESCALATION migration group 91
MYMAXAPPS migration object 128
MYMAXPRESENTATION object structure 122
MYQUERY object structure 133

N

Nested Job Plans 194
new Snapshot package 136
Notifications 142

O

object structures 6, 31, 55, 87, 112, 140, 163, 178, 199, 244
 accessing 132
 APPLICATIONAUTH 78
 creating 71, 77, 112, 121
 DMACTION 87, 186
 DMACTIONGROUP 87, 186
 DMAPPLICATIONAUTH 78
 DMCOMMTEMPLATE 88
 DMCONDITION 55
 DMCRONTASKDEF 100
 DMCTRLGROUP 65, 67
 DMESCALATION 89
 DMMAXGROUP 56, 71, 120
 DMMAXINTOBJECT 78, 121
 DMMAXUSER 56, 58
 DMPERSON 55, 88
 DMROLE 88
 DMSIGOPTION 60, 65, 67
 DMWFPROCESS 184
 duplicating 121, 157, 225
 migrating 226
 migration 122
 Migration Manager 202
 PMSC_ASSETATTRIBUTE 186
 PMSC_AUTOSCRIP 186
 PMSC_COMMODITIES 183
 PMSC_DOCINFO 171
 PMSC_JOBPLAN 170, 183
 PMSC_JPACTION 183
 PMSC_Offering 171
 PMSCATALOG 182

RBSECURITY01 57
SECURITYRESTRICT 71
 standard 120, 132, 144
 system lookups 122
Offerings 170
out-of-the-box migration package template 179

P

package deployment failing 244
packages 9, 30, 58, 105, 119, 163, 178, 201, 242
 RBSECPKG01 57
Parent Classifications 195
person groups, accessing 85
person records, accessing 85
PMSC_ASSETATTRIBUTE object structure 186
PMSC_AUTOSCRIP object structure 186
PMSC_COMMODITIES object structure 183
PMSC_DOCINFO 170
PMSC_DOCINFO object structure 171, 185
PMSC_JOBPLAN 170
PMSC_JOBPLAN object structure 170, 183
PMSC_JPACTION object structure 183
PMSC_OFFERING 170
PMSC_OFFERING migration group 172
PMSC_OFFERING object structure 185, 187
PMSC_Offering object structure 171
PMSC72_CATALOG migration group 178, 180, 187
PMSC72_CatalogOffering 194
PMSC72_CatalogTemplate 180, 195
PMSC72_CatalogTemplate Package Definition 178
PMSC72_OFFERING migration group 178, 180, 187
PMSCATALOG object structure 182

Q

queries, creating 132

R

R 200
RBAUTHORIZATIONS01 migration group 78
RBGLOBALSE01 migration group 73
RBSECPKG01
 package 57
RBSECURITY01 object structure 57
Redbooks Web site 260

Contact us xv
Report Date 111
Reset Event Tracking Records 207
RESOURCES migration group 89
Response Plans 194
roles 143, 148
roles, accessing 85
root logger, accessing 234
root logger, managing 230
Run Reports 220

S

Save Query 132
Save Report 219
SCTEMPLATEID attribute 245
SEARCHWHERE key value 111
security configuration migration errors 244
Security Groups 62, 181
 creating 54
SECURITYRESTRICT object structure 71
SECURITYRESTRICT table 71, 120–121
service catalog 177
Service Groups 170
service invocation 198
Set Logging Root Folder 231
Set Report Object Structure Security 218
Set Where Clause 136–137, 146
Signature Option 64
 adding or modifying 63
sigoptflag 190
Sigoptions 62
Snapshot migration package 204
Snapshot package 210
Source Objects 78
SQL Data Dictionary 253
SQL filter criteria 145
SQL Where clause 255
SQL Where clause commands 211
standard object structures 132, 144
Start Center 131
STATUS key value 111
system information screen 242
system lookups 122

T

Test Results 131
Ticket Templates 170
Tivoli Service Request Manager, object structures

170

U

URLs, embedded 208
users, creating and managing 55

V

View Documents 170

W

web services 199
Work Order 64
Work Order Tracking 110
Workflow Designer 141, 148, 170
workflow modifications 140
WORKORDER object 54



Migration Use Cases with the Migration Manager

(0.5" spine)
0.475" <-> 0.875"
250 <-> 459 pages



Migration Use Cases with the Migration Manager

Covers Tivoli Service Request Manager, CCMDB, and Tivoli's process automation engine migrations



Learn about Migration Manager implementation strategy

Experiment with a variety of real-life scenarios

Review migration troubleshooting tips

The Migration Manager enables you to migrate configuration content from one production environment to another. The typical use is to migrate configuration content from a development environment to a test environment and then on to production for the Tivoli process automation engine and its applications, such as IBM Tivoli Change and Configuration Management Database (CCMDB) and IBM Tivoli Service Request Manager. The goal of migration is to ensure that your production environment fully meets the needs of your users.

This IBM Redbooks publication covers the most common migration use cases with the Migration Manager. Of course, these use cases are only a small subset of the possible migration scenarios that can be performed by the Migration Manager, but they were chosen to be representative of the capabilities of the Migration Manager.

In addition to these use cases, the book presents a migration strategy and a comprehensive chapter about troubleshooting possible migration problems when using the Migration Manager.

We strongly suggest that you read Chapter 1, "Migration strategy" on page 1 first before reading the other chapters. This chapter will give you a good foundation for all of the migration scenarios covered in the book.

This book will be a reference for IT Specialists and IT Architects working on migrating configuration content from one production environment to another using the Migration Manager.

**INTERNATIONAL
TECHNICAL
SUPPORT
ORGANIZATION**

**BUILDING TECHNICAL
INFORMATION BASED ON
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7906-00

ISBN 0738435090