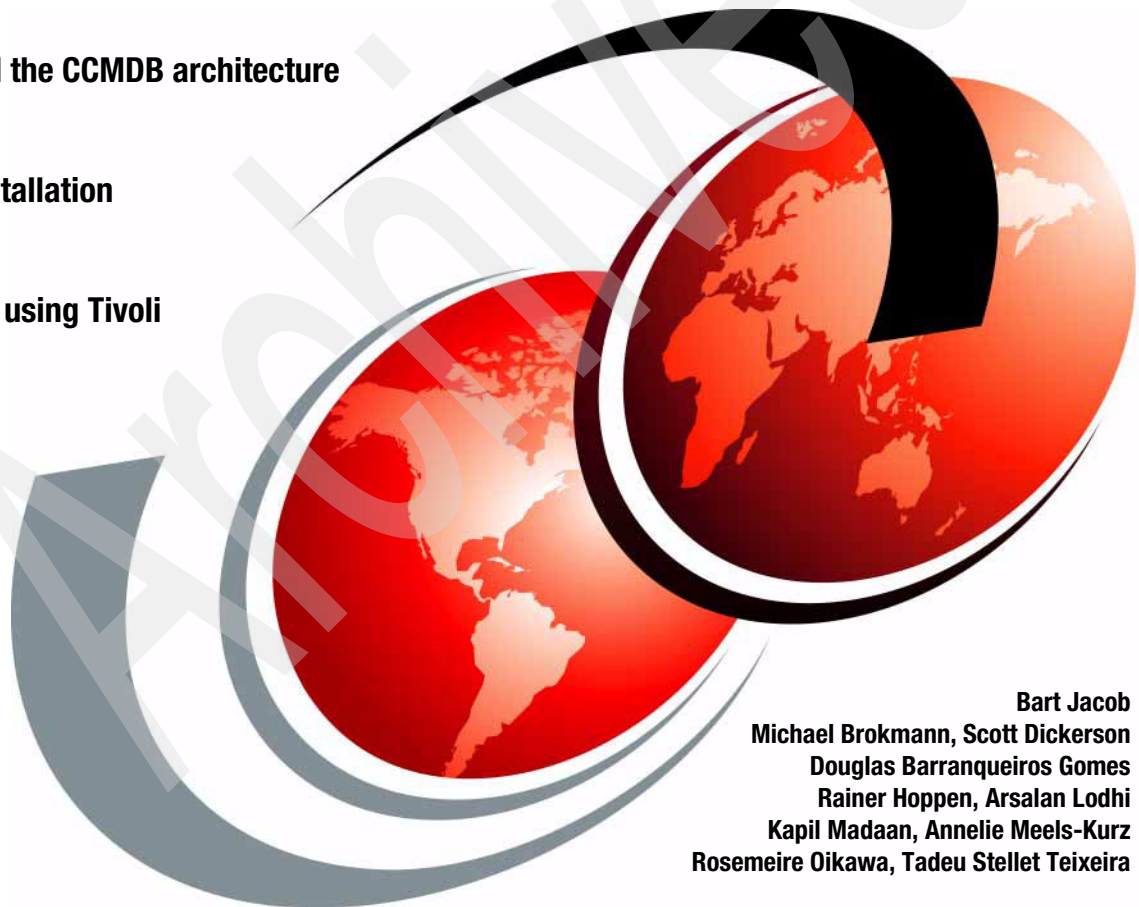


Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning

Understand the CCMDB architecture

Plan for installation

Get started using Tivoli
CCMDB



Bart Jacob
Michael Brokmann, Scott Dickerson
Douglas Barranqueiros Gomes
Rainer Hoppen, Arsalan Lodhi
Kapil Madaan, Annelie Meels-Kurz
Rosemeire Oikawa, Tadeu Stellet Teixeira



International Technical Support Organization

**Deployment Guide Series: IBM Tivoli CCMDB
Overview and Deployment Planning**

May 2008

Archived

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

First Edition (May 2008)

This edition applies to Version 7, Release 1, of IBM Tivoli Change and Configuration Management Database.

© Copyright International Business Machines Corporation 2008. 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

Figures	ix
Notices	xvii
Trademarks	xviii
Preface	xix
The team that wrote this book	xx
Become a published author	xxii
Comments welcome	xxiii
Part 1. Concepts	1
Chapter 1. CCMDB overview	3
Chapter 2. What is behind CCMDB	5
2.1 IBM Service Management	6
2.1.1 Why businesses need ISM	6
2.1.2 What IBM Service Management is	8
2.2 Information Technology Infrastructure Library	9
2.2.1 ITIL Version 3	10
2.2.2 Critical success factors to implement ITIL	12
2.2.3 IBM and ITIL	13
2.3 IBM Tivoli Unified Process	13
2.3.1 ITUP Composer	16
2.3.2 IBM Rational Method Composer overview	18
2.3.3 Method content authoring overview	21
2.3.4 Process authoring overview	23
2.3.5 How ITUP Composer works with CCMDB	24
Part 2. Components	27
Chapter 3. CCMDB components	29
3.1 Components of the IBM CCMDB	30
3.2 User interface layer	32
3.3 CCMDB discovery server and TADDM	33
3.4 CCMDB Base Services	52
3.5 CCMDB Process Manager Products	54
3.6 Integration Composer and Integration Adapter for TADDM	72
Chapter 4. CCMDB Data Layer	75

4.1	Discovered configuration items	77
4.2	Actual configuration items	78
4.3	Authorized configuration items	82
4.4	Audit: comparing Actual and Authorized CI data spaces	85
4.5	Federation of external data	86
4.6	Extensibility of the CCMDB schema	87
4.7	Where to begin to load data	87
Chapter 5. Physical components and operational model		89
5.1	Components of the physical run time environment	90
5.1.1	Physical components of the process run time	91
5.1.2	Integration Composer component	95
5.1.3	Components of the Discovery / TADDM environment	96
5.2	Operational model	97
5.3	Scalability and high availability	101
5.3.1	Scalability	101
5.3.2	High availability	103
Chapter 6. CCMDB security architecture		105
6.1	CCMDB V7.1 authentication model	108
6.1.1	Virtual Member Manager	108
6.1.2	Secure Token Service	115
6.1.3	Configuring the CCMDB for single sign-on	116
6.2	CCMDB V7.1 authorization model	124
6.3	Bringing it all together	143
Chapter 7. Integration technologies		145
7.1	Discovery Sensor	148
7.2	Discovery Library adapter and IdML files	148
7.3	TADDM application programming interface	149
7.4	Federation services	150
7.5	Maximo Enterprise Adapter Integration Framework	152
7.6	Integration Modules and Logical Management Operations	156
7.7	Launch in Context	157
7.8	Summary	158
Part 3. Planning and installation		161
Chapter 8. CCMDB installation planning		163
8.1	CCMDB components	164
8.1.1	Middleware components	164
8.1.2	Tivoli Application Dependency Discovery Manager	165
8.1.3	IBM Tivoli Change and Configuration Management Database	166
8.1.4	Console - CCMDB Web Interface	166

8.1.5 IBM Tivoli Integration Composer	166
8.1.6 Integration adapter	166
8.1.7 IBM Tivoli Unified Process Composer	166
8.2 Installation plan	167
8.2.1 Software and hardware requirements	168
8.2.2 Planning for the deployment of CCMDB	173
8.2.3 Planning for CCMDB worksheet	180
8.2.4 CCMDB topologies	183
Chapter 9. Installation	187
9.1 Topology overview	188
9.1.1 Windows multiple server deployment topology	188
9.1.2 Red Hat Linux server multiple server deployment topology	189
9.2 Installation flow	190
9.3 What you should do before you begin	191
9.4 The CCMDB launchpad	193
9.5 Middleware installation	194
9.6 Tivoli Application Discovery and Dependency Manager Installation	199
9.7 Change and Configuration Management Database installation	200
9.8 CCMDB post installation steps	206
9.8.1 Sign in with the default user ID	207
9.8.2 Granting universal access to the MAXADMIN group	208
9.8.3 Update User Security to view inactive organizations and sites	212
9.8.4 Create currency codes	214
9.8.5 Create item and company sets	214
9.8.6 Create an organization	215
9.8.7 Create a general ledger account component	217
9.8.8 Create a general ledger account	218
9.8.9 Create default insert site	220
9.8.10 Create a Work Type	221
9.8.11 CCMDB post installation steps: Solution Installer Command-Line Interface	222
9.9 IBM Tivoli Integration Composer installation	224
Part 4. Getting started	229
Chapter 10. TADDM and Process Layer integration	231
10.1 End-to-end data discovery and migration	232
10.2 Integration Composer overview	233
10.2.1 System requirements	236
10.2.2 Integration Composer components	236
10.3 IBM Tivoli Integration Adapters	238
10.4 TADDM adapter installation	239
10.4.1 Integration adapters for CI Type	240

10.4.2	Integration adapters for Actual CI (CI Instances)	240
10.4.3	Adding the TADDM Adapter to Integration Composer	241
10.5	Configuration for TADDM and Maximo integration	242
10.5.1	Setting up the TADDM integration adapters	244
10.6	Set schemas, define mapping, and run execution	251
10.6.1	Configure and execute TADDM Adapter for CI Type mapping	252
10.6.2	Configuring executing TADDM adapter for Actual CI mapping	271
10.7	Transfer of new or updated CIs after successful migration	294
10.7.1	Execute mapping through insertion	295
10.7.2	Execute mapping through an update	299
10.8	Import CI data through DLA	304
Chapter 11.	Launch in Context.	307
11.1	Launch in Context graphical user interface	309
11.2	Launch entry URL specifications and parameters	312
11.2.1	Launching the TADDM Product Console within CCMDB V7.1	316
11.3	Adding a new Launch in Context into CCMDB V7.1	321
11.3.1	Define a launch entry point	322
11.3.2	Associate the launch entry with a Signature option	322
11.3.3	Modify the Select Action menu	328
11.3.4	Allow access for everybody by defining security	330
11.3.5	Verify the new launch entry	332
Chapter 12.	Reporting.	333
12.1	BIRT architecture	334
12.2	BIRT reporting process	336
12.2.1	BIRT report development	337
12.2.2	BIRT Report Administration	338
12.2.3	BIRT Configure Reports	348
12.2.4	BIRT Run Reports	350
12.2.5	BIRT Report examples	354
12.2.6	BIRT manage reports	357
12.2.7	BIRT report queue	358
12.2.8	BIRT report localization	358
12.3	BO Crystal Reports XI Integration (BOC)	359
12.3.1	Requirements	360
12.3.2	Integration with CCMDB V7.1	361
12.3.3	Report development	362
12.3.4	Report administration	363
12.3.5	Security	363
12.3.6	Report functions not supported	368
12.4	External Report Integration (ERI)	370
12.4.1	Requirements of ERI	371

12.4.2 Enabling the ERI	371
12.4.3 Registering and running ERI Reports in CCMDB V7.1	374
Glossary	383
Related publications	391
IBM Redbooks	391
Online resources	391
How to get Redbooks	392
Help from IBM	392
Index	393

Figures

2-1 Infrastructure complexity	7
2-2 IBM Service Management	9
2-3 ITUP overview	15
2-4 ITUP versus ITUP Composer feature comparison	16
2-5 Method framework	21
2-6 Method Content Representation in RMC	22
2-7 Process authoring	24
3-1 Logical architecture overview for CCMDB V7.1	30
3-2 New sensors in Discovery Profile	37
3-3 Level 1 Discovery Profile	38
3-4 Rediscovery of configuration items	39
3-5 Extended Attribute type support	40
3-6 Model extension in TADDM	41
3-7 Extending the model using an XML based definition file	42
3-8 Manual merge	44
3-9 Prioritization rules	45
3-10 Pluggable reconciliation architecture	46
3-11 UserGroup Definition in TADDM Domain Manager Interface	48
3-12 Query based collections	49
3-13 Domain Manager topology graph	50
3-14 Launch in Context definition	51
3-15 Process Requests application	56
3-16 Process Manager Type selection	57
3-17 Process Request classification	58
3-18 Completed Process Request	59
3-19 Process Request into Configuration Management	60
3-20 Submission of the Process Request	61
3-21 Integration Module applications	62
3-22 IT Infrastructure Module Applications overview	63
3-23 CI Lifecycle application	64
3-24 Job Plans application	64
3-25 Update CI Job Plan template	66
3-26 Key Performance Indicator for Configuration Management	67
3-27 Link to Change Management application	68
3-28 Impact Analysis tab of the Change Management application	70
3-29 Predefined Change Management Job Plan templates	71
4-1 CCMDB V7.1 Data Layer	76
4-2 Interconnected graph of CI dependencies	78

4-3	Filtered CI data in Actual CI data space	80
4-4	Actual CI filter depth	81
4-5	Sample of an Authorized CI view	83
5-1	Physical component and relationship overview	90
5-2	IBM HTTP Server configuration in WebSphere Admin Console	92
5-3	WebSphere cell definitions: CCMDB default installation environment ...	94
5-4	Operational model.	98
6-1	CCMDB V7.1 security architecture	106
6-2	ISMRealm definition	110
6-3	Default definition for VMMSYNC cron task	111
6-4	Error message of security application	112
6-5	LDAP entries	113
6-6	User and group definitions synchronized into the CCMDB database ...	114
6-7	Users and groups in WebSphere Admin Console	115
6-8	Configuring the SSO domain in the WebSphere Admin Console	121
6-9	Launch in Context operation from Actual CI application	122
6-10	Login verification into TADDM using maxadmin	123
6-11	Users management dialog in TADDM Domain Manager	124
6-12	Entities relevant for authorization	125
6-13	CCMDB Security Users application	126
6-14	CCMDB Security Groups application	127
6-15	Defining Start Center of the Security Group	128
6-16	Data Restriction in Security Groups application	128
6-17	Attribute Restriction in the Security Groups application	129
6-18	Collection Restriction in Security Groups application	130
6-19	Conditional Expression definition in Security Groups application	131
6-20	Security profile of a user in the Security Users application	132
6-21	Security Group Access report	133
6-22	Predefined Security Groups of Change and Configuration Management PMPs	134
6-23	Person record	135
6-24	Actual Configuration Items application	137
6-25	Promoted Configuration Item in the Configuration Items application ..	138
6-26	Collection Definition in the Collections application	139
6-27	Permitting Access to Collection in Security Groups application	139
6-28	Publish Channels application	140
6-29	Enabling the publish channels	140
6-30	Endpoint definition.	141
6-31	External System definition.	142
6-32	TADDM User Group dialog after Access Collection synchronization ..	143
7-1	Interface technology overview.	146
7-2	MXAUTHCI Object Structure exposed as XML	153
7-3	Web Services Library application	154

7-4 Defining Endpoint Handler in Endpoint application	155
8-1 Multiple server deployment option	184
9-1 Our Windows-based lab environment	188
9-2 Our Linux-based lab environment	189
9-3 Installation flowchart	190
9-4 Launchpad Initial window	194
9-5 Middleware components	196
9-6 Windows services startup options	197
9-7 Administrative Console login	199
9-8 TADDM installation initial window	200
9-9 CCMDB installation	201
9-10 Import middleware configuration	202
9-11 Choose deployment option	203
9-12 Choose installation folder	203
9-13 Setting the TADDM password	204
9-14 Maximo installation directory	205
9-15 Installation status	206
9-16 Initial start center	208
9-17 Opening Security groups application	208
9-18 Listing default security groups	209
9-19 Selecting MAXADMIN group	209
9-20 Applications tab	210
9-21 Granting access	210
9-22 Accepting the grant	211
9-23 Save the group	211
9-24 Sign out	211
9-25 WebSphere Admin Console used to stop and start MXServer	212
9-26 User security application	212
9-27 User list	213
9-28 Viewing user detail	213
9-29 Accessing Currency Code application	214
9-30 Selecting currency	214
9-31 Entering IT items	215
9-32 Creating an organization	216
9-33 Entering a Site name	216
9-34 General Ledger application	217
9-35 Account configuration	218
9-36 Chart of Accounts application	218
9-37 General ledger component	219
9-38 Organization application	219
9-39 Create default site	220
9-40 Starting the organization application	221
9-41 Accessing Work Type definition	221

9-42	Setting Work Type information	222
9-43	ITIC installation	225
9-44	ITIC database information	226
9-45	ITIC login window	227
10-1	End-to-end data migration plan	232
10-2	Integration Composer application	233
10-3	Integration Composer and Adapter overview	234
10-4	Integration steps	235
10-5	Integration Composer components	236
10-6	Starting Integration Composer	243
10-7	Integration Composer login window	243
10-8	Integration Composer main application	244
10-9	Maximo Classification window	245
10-10	Creating TOPCICLASS in Maximo	246
10-11	Adding a database connection	247
10-12	Executing the update command in DB2 Command Editor	248
10-13	Reference classes	250
10-14	Selecting the source CI data model	252
10-15	Creating a TADDM CI Type schema in a Maximo DB	253
10-16	Define New Data Schema in Integration Composer	254
10-17	Data Source for Data Schema	255
10-18	New Data Source to Maximo target database	256
10-19	Data Schema window	257
10-20	Import CI Classification.schm	258
10-21	Import Data Schema classification	259
10-22	Error analysis	260
10-23	Errors fixed	261
10-24	Importing data schema for CI Classification	262
10-25	Define TADDM data source	264
10-26	Naming a data source	265
10-27	TADDM Data Source Connection parameters	266
10-28	Create New Mapping	268
10-29	Execute the mapping from Integration Composer interface	270
10-30	Mapping Execution Compiling	271
10-31	Mapping Execution in Progress	271
10-32	Create TADDM71 actual data schema	272
10-33	Defining a data source using TADDM7.1 Actual CI	274
10-34	Naming the data source	275
10-35	Connection parameters to TADDM DB and testing connection	276
10-36	Data schema for Actual CI	278
10-37	Data Source to target DB (Maximo) for Actual CI	279
10-38	Connection parameters to target Maximo DB	280
10-39	Import Actual CI schema in target DB	281

10-40	Import schema CCMDB V7.1 Classification	281
10-41	Import CCMDB7.1 Actual CI errors	282
10-42	Errors fixed	283
10-43	Saving the import schema	284
10-44	Execute qualifierCCMDB71ActualCI script	285
10-45	Import Actual CI mapping	286
10-46	Log into TADDM DB	287
10-47	Log into the Maximo DB	288
10-48	TADDM to Maximo Actual CI Mapping window	289
10-49	Import TADDM mapping file	290
10-50	Import TADDM dialog box	290
10-51	Saving mapping	291
10-52	Executing Actual CI mapping	292
10-53	Log into source for executing Actual CI mapping	293
10-54	Log into target for executing Actual CI mapping	293
10-55	Compiling mapping	294
10-56	Mapping success	294
10-57	Open Mapping window for Actual CI mapping	295
10-58	Check Insert box in existing mapping	296
10-59	Execute mapping with the Insert box checked	297
10-60	Mapping execution progress migration for only new CIs	297
10-61	TADDM CI loaded through DLA	298
10-62	New CI added in Maximo from TADDM with no other updates made to existing CIs in Maximo	299
10-63	Rerun mapping with updates	300
10-64	TADDM CI and its associated relationship	301
10-65	CI Relationship migrated in Maximo	302
10-66	Searching Software CI Types in Maximo	302
10-67	Activate Software CI Types	303
10-68	Activating Software CI Types	303
10-69	Software CI Types activated	304
10-70	Status of TADDM server	305
10-71	RADDM UI showing new CIs	306
11-1	Configuration for Launch in Context application	309
11-2	List of launch points	309
11-3	Specifying a URL	310
11-4	Select Action	310
11-5	New Launch Entry	310
11-6	Save Launch Entry	310
11-7	Clear Changes	311
11-8	Help for Launch in Context configuration	311
11-9	Specifying a URL	312
11-10	Prompt for downloading confignia.jnlp	316

11-11 TADDM startup window	316
11-12 Possible pop-up window for slow launch	317
11-13 Starting Launch in Context from the Actual CI application	318
11-14 TADDM Application Infrastructure view	319
11-15 CCMDB Physical Topology view.	320
11-16 Physical Infrastructure view	321
11-17 Define the launch entry point	322
11-18 Selecting Application Designer	323
11-19 Choosing Actual Configuration Items application	323
11-20 Add/Modify Signature options	324
11-21 Add/Modify Signature options window	325
11-22 Specifying the new option	326
11-23 Selecting the new option	327
11-24 Associating the launch entry	328
11-25 Modifying the Select Action menu.	330
11-26 Security groups	331
11-27 Approve the changes	332
11-28 New Select Action menu	332
12-1 BIRT architecture	334
12-2 Report Engine directory structure	336
12-3 BIRT Design Engine	337
12-4 Accessing the Report Administration application through Go To	339
12-5 Accessing the Report Administration application through the Reports option	339
12-6 Lists of reports.	340
12-7 Report details	340
12-8 Report security	341
12-9 report labels	341
12-10 View Scheduled Reports.	342
12-11 Report Application Security.	342
12-12 Application security settings	343
12-13 Individual security settings	344
12-14 View Group Security	345
12-15 View Library Files	345
12-16 Import Report	346
12-17 Import Resources	347
12-18 View Report Dependencies.	347
12-19 Report configuration	348
12-20 Report parameters	350
12-21 Selecting the report to run.	351
12-22 Run request.	352
12-23 Sample report	353
12-24 Available actions on report toolbar	354

12-25	E-mailing reports	354
12-26	Job plan list report	355
12-27	Job Detail Plan report	355
12-28	Service Level exception report	356
12-29	Releases by Classification report	357
12-30	Report Usage report	358
12-31	Setting report administrator's locale	359
12-32	Selecting a report through the application.	361
12-33	Selecting a report through the Report menu	361
12-34	Selecting security groups	364
12-35	Setting reporting application security	365
12-36	On Demand reports sub-tab	365
12-37	Entering report parameters	366
12-38	Example of BOC report	367
12-39	Report Administration application	375
12-40	Security group selection	377
12-41	Setting security by application.	378
12-42	Report availability	379
12-43	Selected Oracle BI Security Analysis Report	380

Notices

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

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

Redbooks (logo)  ®

AIX 5L™

AIX®

DB2®

Domino®

Enterprise Asset Management®

Extreme Blue™

HACMP™

IBM®

Informix®

iSeries®

Library Reader™

Lotus®

Maximo®

Rational®

Redbooks®

System z™

Tivoli®

WebSphere®

z/OS®

The following terms are trademarks of other companies:

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

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

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.

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

Active Directory, Expression, Internet Explorer, Microsoft, SQL Server, Windows Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

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

Preface

The IBM® Tivoli® Change and Configuration Management Database (CCMDB) is one of the key components of the IBM Service Management (ISM) strategy. It is the foundation for automating and supporting change and configuration management processes as described by the Information Technology Infrastructure Library (ITIL®). These process solutions provide best practice implementations of processes based not only on ITIL, but on the IBM Process Reference Model for IT and other standards as well.

This IBM Redbooks® publication provides information that can be used by clients, partners, or IBM field personnel who are looking to engage in an effort to implement change and configuration management processes in an enterprise environment utilizing the IBM Tivoli Change and Configuration Management Database (CCMDB) product. This book is divided into four parts:

- ▶ **Concepts:** Provides an overview of the CCMDB product and some of the standards that drive its development.
- ▶ **Components:** Provides the reader with a better understanding of the various components, logical and physical, that make up the product and the functions that they provide.
- ▶ **Planning and Installation:** This part provides information related to the actual installation of the CCMDB product components, including information related to hardware and software requirements.
- ▶ **Getting Started:** This part describes the initial use of the product, to allow a reader to create a demonstration or proof-of-concept around core product functions.

A companion book, *IBM Tivoli CCMDB Implementation Recommendations*, SG24-7567, provides more advanced details about the underlying components of the product and utilizing the product to support robust IT processes such as change and configuration management. The companion book focuses on the details of the data model, process engine, and the Change and Configuration management Process Management Programs (PMPs). It provides details on how these can be extended and customized to meet client requirements.

The team that wrote this book

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



Figure 1 Left to right: Michael Brokmann, Rainer Hoppen, Annelie Meels-Kurz, Rosemeire Oikawa, Tadeu Stellet Teixeira, Douglas Barranqueiros Gomes, Arsalan Lodhi, Kapil Madaan, Bart Jacob

Bart Jacob is a Senior Consulting IT Specialist at IBM Corp - International Technical Support Organization, Austin Center. He has over 25 years of experience providing technical support across a variety of IBM products and technologies, including communications, object-oriented software development, and systems management. He joined the ITSO in 1989, where he has been writing IBM Redbooks publications and creating and teaching workshops around the world on a variety of topics. He holds a Masters degree in Numerical Analysis from Syracuse University.

Michael Brokmann is a Senior IT Architect working for Software Group in Germany. He has over 10 years of experience in Systems and Service Management and a long Tivoli history. He consults for large enterprises all over Germany and gives lectures at various German universities.

Scott Dickerson was the development lead for Release Process Manager V7.1 and was involved in the Deployment Partner Program for CCMDB V7.1. He is involved with the design and implementation of future releases of CCMDB and Release Process Manager.

Douglas Barranqueiros Gomes is an IT Specialist working for IBM Global Services Strategic Outsourcing/SDC Brazil in the Automation Team. He provides deployment and support in Tivoli tools and BMC systems for outsourced customers in Global Resources. He holds a degree in Computer Science (1996) from Carioca University in City of Rio de Janeiro, Brazil.

Rainer Hoppen is an IT architect at Sparkassen Informatik in Germany. He holds a degree in Computer Science and has twenty years of experience in IT. His areas of expertise include service management, project management, and communications software.

Arsalan Lodhi is working as a Solution Architect for IBM in the US. His focus is bringing innovations through the integration of technology and business. His areas of interest include managing digital organization, firms and markets, operations, entrepreneurship, emerging technologies, and business innovation. He received his Master's degree in Business and Technology as part of a joint program of Stern Business School and the Courant Institute of Mathematics at New York University. He went to California State University, Long Beach and attended the undergraduate program in Computer Science and Computer Engineering. His first BS was from University of Karachi - FAST in Computer Science. Arsalan is a graduate of IBM Extreme Blue™, the most prestigious and challenging IBM internship program to attract business minded technical talent. He holds two patents. He has been in the IT industry for the last eight years in various roles ranging from Software Engineer to IT Architect.

Kapil Madaan is a Systems Management Consultant with Tivoli Lab Services in IBM India. He specializes in Tivoli Workload Scheduler, Tivoli Application Dependency Discovery Manager, and Change and Configuration Database Manager. He has four years of experience in IT and has a Master's degree in Computer Applications from IP University, Delhi.

Annelie Meels-Kurz is a systems management specialist at Sparkassen Informatik in Germany. Much of her eleven years of IT experience was spent in the support of mainframe banking applications and communications middleware. The last few years have been devoted to service management. Annelie holds a degree in Geography.

Rosemeire Oikawa is an IT Service Management Consultant from IBM Global Technology Services in Brazil and she is an instructor of ITIL Foundations. She holds a MBA in IT Governance from IPT-USP and is ITIL Practitioner Release and Control Certified. She has written extensively on Process Manager.

Tadeu Stellet Teixeira is an IBM Senior IT Specialist in Brazil. He has more than 15 years working in Information Technology (IT) services. He has ten years of experience in software development and project implementation, three years working as an IT Project Manager, consulting experience in industries such as

oil, steel, telecommunications, automotive, and wholesale commerce, and two years of experience in operations coordination. He has been in an IT architect position for an IBM global customer for more than one year. He is ITIL Foundations certified, ITIL Practitioner Release and Control certified, and an ITIL Foundations instructor.

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

Vijay Aggarwal
Grake Chen
Jim Collins
Carole Corley
Pam Denny
Scott Dickerson
Katherine Dunning
Bradford Fisher
Ann Marie Fred
Melanie Gurda
Jennifer R. Lee
Craig Love
Mike Mallo
Collen McCretton
Matt Posner
Bertrand Raillard
Charles Rich
John Roberts
Tom Sarasin
Jerry Saulman
Chris Schaubach
Ketan Shah
Kelvin Sumlin
Sumit Taank
Edward Whitehead
Amy Veatch

Become a published author

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

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

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- ▶ Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- ▶ Send your comments in an e-mail to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Part 1

Concepts

CCMDB overview

The IBM Tivoli Change and Configuration Management Database (CCMDB) is the foundation for the IBM Service Management (ISM) strategy. It is the foundation for core Information Technology Infrastructure Library (ITIL) process solution deliverables like Configuration and Change or Release Management. These process solutions provide best practice implementations of core ITIL processes.

The CCMDB provides a shared infrastructure as well as a set of foundation services used by different ISM process solutions (such as the previously mentioned ones) and includes the Configuration and Change Management processes that provide core management capabilities needed in an IT environment.

In addition, the CCMDB incorporates a consistent data model and data layer implementation and includes a framework for discovery of resources and its relationships.

A Configuration Management Database (CMDB), according to ITIL, is a database used to manage Configuration Records throughout their life cycle. The CMDB records the attributes of each Configuration Item (CI) and its relationships with other CIs and provides the underpinnings for IT Service Management processes.

A CI has several characteristics, a classification or type, attributes which describe the CI depending on its classification, and relationships that describe how a CI is related to other Configuration Items.

We define a CI as configuration items that are managed components of an IT Service. Configuration records within a CMDB contain information about the CI, and are maintained through their life cycles. Since CIs are managed components, they come under the control of the Change Management process.”

The IBM CCMDB solution provides an ITIL-aligned implementation of a Configuration Management Database.

Before we get into the specifics of the IBM CCMDB product and related solutions, we will provide an overview of the IBM Service Management strategy, ITIL, and the IBM Tivoli Unified Process that provides a linkage between the two.

What is behind CCMDB

In the fall of 2007, the IBM Systems Journal provided a series of papers focused on the IBM Service Management strategy and related technologies and solutions. This IBM Systems Journal is available at <http://www.research.ibm.com/journal/sj46-3.html>.

Some of the content from this chapter was extracted and paraphrased from the papers presented in the IBM Systems Journal.

2.1 IBM Service Management

IBM has developed thought leadership to improve the “state of the art” in service management for the last 25 years, and has supported others in their efforts as well. In addition to the advancement of management disciplines and technologies, IBM recognized early on that acceptance of common practices and standards is vital to achieving improved value from information technology (IT).

Advances in technologies and management disciplines provide the greatest value once they become part of and extend the body of generally accepted practices and open standards. IBM supports the advancement of practices and open standards such as the IT Infrastructure Library® (ITIL), Control Objectives for Information Technology (COBIT), ISO IEC 20000 and Carnegie Mellon University's e-Sourcing Capability Model (e-SCM). The fundamental characteristics of service management require integration and agreement on standards, not only between tools and roles within IT, but also among organizations and even industries.

IT service management is the integrated set of activities required to ensure the cost and quality of IT services valued by the customer. It is the management of customer-valued IT capabilities through effective processes, organization, information, and technology, including:

- ▶ Aligning IT with business objectives
- ▶ Managing IT services and solutions throughout their life cycles
- ▶ Service management processes like those described in ISO IEC 20000, ITIL, and the Process Reference Model for IT.

2.1.1 Why businesses need ISM

Today's enterprises face an ever-increasing problem of managing their IT processes to deliver IT services in a manner that is:

- ▶ Efficient
- ▶ Reliable
- ▶ Secure
- ▶ Consistent

At the same time, the complexity of the infrastructure needed to deliver these IT-enabled business services has been increasing rapidly. A simple example that shows the complexity of IT environments is shown in Figure 2-1 on page 7.

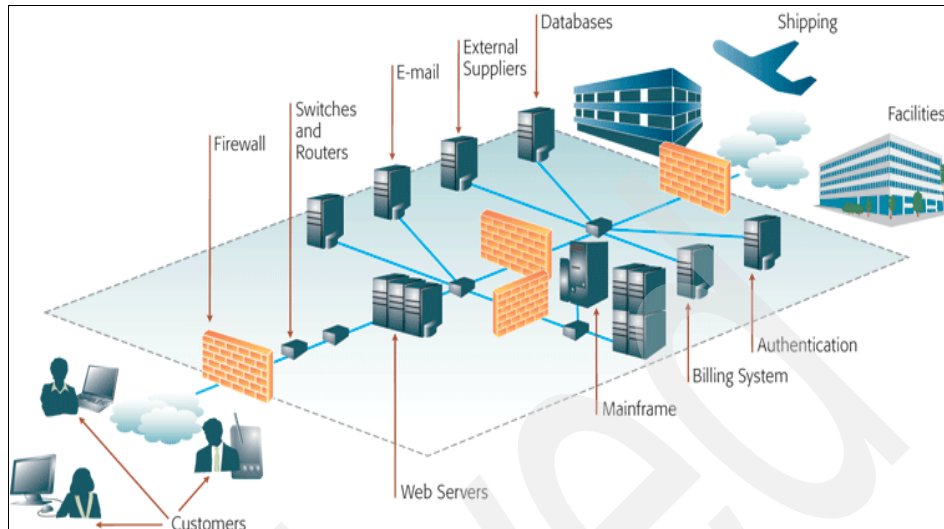


Figure 2-1 Infrastructure complexity

The following are some of the key challenges faced by businesses:

- ▶ **Complexity:** The root cause of the problems IT organizations face lies in the dramatic increase of business complexity due to heterogeneity of environments and the interconnection of applications (composite applications). Architectural and organizational issues, accelerating the proliferation of composite applications and hardware entities, and worldwide operations spanning multiple time zones, all contribute to reducing the efficiency and effectiveness of the IT organization.
- ▶ **Change:** Complexity makes for very brittle, hard-to-manage infrastructures that often break under change and whose management requires a discipline that few companies achieve without flaws. Increasing workloads, more stringent service-level assurance requirements, staff turnover, and new market opportunities all lead to pressure for change in the IT organization. Change is the leading cause of service or application disruption today, and it often results in visible business impact. In fact, our experience suggests that nearly 80 percent of all critical outages can be traced to faulty change management.
- ▶ **Cost:** Currently, operational IT labor cost constitutes almost 70 percent of the total IT budget of businesses. In the late 1990s, half of the IT labor budget was devoted to new application development and half was devoted to operations. As IT budgets have been held flat, the chief information officers of IT organizations have faced two unappealing choices: shift resources from new application development or reduce the level of support for current

applications. Both options serve to reduce the efficiency and effectiveness of IT.

- ▶ *Governance and compliance:* The introduction of government regulations, such as the Sarbanes-Oxley Act (SOX) and the Health Insurance Portability and Accountability Act (HIPAA), have put an additional burden on the IT organization to support the needs of the business to audit for compliance through the institution of better process controls and the maintenance of audit trails for IT infrastructure changes. This requires careful consideration because of the penalties of noncompliance, including criminal and civil liabilities and adverse public opinion.

2.1.2 What IBM Service Management is

For many businesses, service excellence is increasingly a competitive differentiator, as organizations need to rapidly adapt to changing conditions in the marketplace and create and deploy new services quickly and efficiently. However, service excellence can only be achieved through effective and efficient Service Management.

A fundamental goal of IT Service Management is the management of IT services and infrastructure with the same kinds of quality control that enterprises strive to use for all business processes. When this is achieved, businesses have the confidence to deploy new and updated services that are critical to their missions.

An effective IT Service Management capability reduces the time needed to deliver a company's IT services according to business policies and reduces the labor cost of the people involved in executing the processes by replacing manual IT process management with autonomic management.

IBM Service Management (ISM) is an approach designed to automate and simplify the management of business services. It concentrates on four areas of study:

- ▶ Technology integration and standards
- ▶ Improved collaboration among IT people spread across organizational silos
- ▶ Best practice based process modules to enable automated process execution
- ▶ Sharing of business-critical IT information to improve decision making

In finding workable solutions to these areas, IBM solutions cover four key areas:

- ▶ Process Managers that provide automated ITIL-aligned workflows for key IT processes
- ▶ An open, standards-based IBM IT Service Management platform

- ▶ Integration between process tasks and Operational Management Products to automate the running of those tasks from the process flow
- ▶ Best practices to help pull it all together

These four key areas are shown in Figure 2-2.

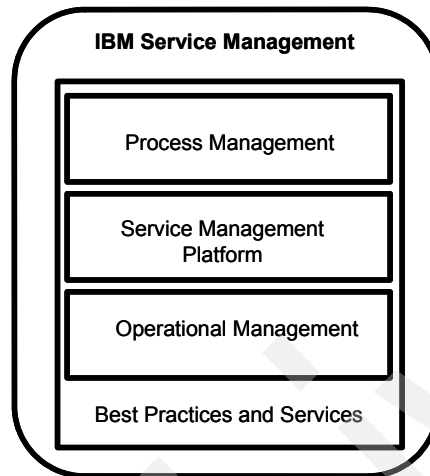


Figure 2-2 IBM Service Management

2.2 Information Technology Infrastructure Library

The Information Technology Infrastructure Library (ITIL) is an internationally recognized framework that provides comprehensive *best practice* guidelines on all aspects of end-to-end Service Management. It covers people, processes, products, and the use of partners. It began in the 1980s when the UK Government initiated an exercise to standardize its diverse IT processes.

It has evolved over the years to cover Service Support, Service Delivery, and in 2007, Version 3 was launched that includes a life cycle management approach in five core volumes: Service Strategy, Service Design, Service Transition, Service Operation, and Continual Service Improvement.

The best practices contained in ITIL are independent of tool, vendor, or industry and can be applied to an organization of any size. ITIL encourages organizations to adapt and adopt its suggestions to meet business needs and improve processes. Though there is a significant amount of detail in the books that make up the library, the books are not themselves the solution to all IT management issues. The processes require significant work to deploy at a level of detail

enabling day-to-day use, with dependencies on the three key components (process, people, and tools) of a management system.

It should be noted that although many people refer to ITIL as a standard, it is not one. Organizations cannot comply with ITIL. It is a set of guidelines that an organization can adopt and adapt to their needs.

2.2.1 ITIL Version 3

ITIL Version 3 focuses on best practices throughout the service life cycle. It focuses essentially on service and solution life cycle management, including five core volumes: Service Strategy, Service Design, Service Transition, Service Operation, and Continual Service Improvement.

These five components are briefly described here.

Service Strategy

Provides a view to align business and IT so that each brings out the best in the other. It ensures that every element of the Service life cycle is focused on customer outcomes and relates to all the companion process elements that follow. The four main activities in Service Strategy are define the market, develop the offerings, develop the strategic assets, and prepare for execution. Service Strategy encompasses the following processes:

- ▶ Strategy Generation
- ▶ Market Intelligence
- ▶ IT Financial Management
- ▶ Service Portfolio Management
- ▶ Demand Management
- ▶ Risk Management

Service Design

Provides guidance for the design of a new or changed service for introduction into the live environment, ensures there is a holistic approach to all aspects of design, and considers all aspects when changing or amending any of the individual elements of design. Service Design encompasses the following processes:

- ▶ Service Portfolio Management
- ▶ Service Catalog Management
- ▶ Service Level Management
- ▶ Capacity Management
- ▶ Availability Management
- ▶ Service Continuity Management
- ▶ Information Security Management

- ▶ Supplier and Contract Management

Service Transition

Provides guidance for the development and improvement of capabilities for transitioning new and changed services into the production environment. It focuses on the broader, long-term change management role and release practices. Service Transition encompasses the following processes:

- ▶ Change Management
- ▶ Service Asset and Configuration Management
- ▶ Knowledge Management and Service knowledge System)
- ▶ Service Release and Deployment Planning
- ▶ Performance and Risk Evaluation
- ▶ Testing
- ▶ Acquire, Build, and Test Release
- ▶ Service Release, Acceptance, and Test and Pilot
- ▶ Deployment, Decommission, and Transfer

Service Operation

Introduces, explains, and details delivery and control activities to achieve operational excellence on a day-to-day basis. Many of the familiar processes from the former service support and service delivery books of ITIL Version 2 will be found in this book. Service Operation encompasses the following processes:

- ▶ Monitoring and Event Management
- ▶ Incident Management
- ▶ Request Fulfillment
- ▶ Problem Management
- ▶ Access Management

Continual Service Improvement

Provides guidance for continual alignment of the portfolio of IT Services with the current and future business needs, growth and maturity of the enabling IT processes for each service in a continual service life cycle model, activities to support a continual process improvement plan, and how to measure, interpret, and take action. Continual Service Improvement encompasses the following processes:

- ▶ Measurement and Control
- ▶ Service Measurement
- ▶ Service Assessment and Analysis
- ▶ Process Assessment and Analysis
- ▶ Service Level Management
- ▶ Improvement Planning

2.2.2 Critical success factors to implement ITIL

As ITIL is a framework of best practices, and not a methodology; it only describes what needs to be done. ITIL does not provide guidance on how to implement the processes, so each company chooses the best way to fit ITIL to its requirements.

A key mindset when implementing ITIL is “adopt and adapt”: “Adopt” ITIL as a common language and reference point for IT Service Management and “adapt” ITIL best practices to achieve business objectives.

Generally IT organizations do not implement all ITIL processes because they do not have the budget or they decide that they do not need all the processes. Initially, not implementing all processes can be seen as a way to avoid extra costs, however, depending on the processes chosen to be implemented, choosing not to implement the other process may result in fewer benefits from the implemented processes. For example, choosing to implement Change and Release processes without implementing Configuration may result in inaccurate impact assessment when approving changes.

The service management processes selection should be done carefully, taking into consideration the relationship among all processes and not only the cost perspective and implementation complexity of individual processes.

A successful implementation of IT Service Management should:

- ▶ Be aligned with business needs, that is, business-driven not technology-driven.
- ▶ Improve staff awareness about business goals.
- ▶ Be adapted to the culture of the organization. This adaptation should be done when defining the roles, responsibilities, tools, processes, procedures, tasks, and so on. After ITSM is implemented, it should be rigorously followed.
- ▶ Have its processes clearly defined, documented, and available.
- ▶ Have its main processes integrated with each other.
- ▶ Have its inputs measurable and repeatable.
- ▶ Have IT processes tool supported and customized to fit the processes defined.
- ▶ Have processes easily changed as necessary.
- ▶ Be integrated with external suppliers.
- ▶ Properly train and communicate to all people who will use or provide IT services.
- ▶ Have clearly measurable and repeatable key performance indicators.

A successful ITSM implementation should result in improved IT customer satisfaction, better resource utilization, and improvement of customer perception of IT service quality.

2.2.3 IBM and ITIL

IBM initially contributed to ITIL with its systems management concept “yellow books” and continues to contribute as a developer, reviewer, and user of ITIL.

IBM contributed in many ways to ITIL Version 2, including authoring, quality reviews, project management, and additional support through the IT Service Management Forum. The focus of Version 2 was on Process Management practices required to enable Service Management. The ITIL service support and delivery publications contain significant contributions from IBM. The ITIL application management book, co-written by authors from IBM and other companies, is the basis for the life cycle concept in ITIL Version 3. It lays the basic groundwork for how to integrate service management practices throughout the solution life cycle.

IBM supports the development of updates and refreshes to industry-accepted best practices, including supporting the ITIL Advisory Group through quality reviews and other briefings. Thought leaders also serve on the ITIL Advisory Group and other working groups to contribute as the need arises. Our view is that ITIL is a valuable set of publications that promote best practices in service management. From a strategic outsourcing perspective, ITIL is requested by many of IBM's clients all around the globe. Companies that are implementing improvements to their service management capabilities consider ITIL a good place to start.

2.3 IBM Tivoli Unified Process

This section provides an overview description of IBM Tivoli Unified Process (ITUP) and its relationship with IT industry best practice models.

As described in 2.2, “Information Technology Infrastructure Library” on page 9, ITIL was developed as a set of Information Technology best practices and its primary goal is to define and organize IT processes. ITIL documents only what should be done; it does not show how processes should be implemented.

ITUP is a free, read only knowledge base that describes IT Service Management processes. It is an excellent reference for guidance on industry best practices and tools that can help automate processes and tasks.

ITIL describes a systematic approach to creating a service-oriented culture and practice for IT service management. The library emphasizes the central importance of meeting business requirements economically. However, IT organizations will need to look beyond ITIL to understand the IT management process disciplines that are central to delivering on the growth agenda. Leaders in IT management must handle the competing strategic priorities that force trade-offs between cost-efficiency, flexibility, and service availability.

To assist IT organizations in this critical challenge, IBM developed the Process Reference Model for IT (PRM-IT). The IBM model supplements the content of ITIL Version 2 based on IBM's extensive IT management experience, gained from managing thousands of IT environments, both large and small. The Process Reference Model for IT identifies the set of IT management processes required to move beyond a singular cost focus to principled decision making that accounts for changing business and technology conditions while managing existing systems complexity.

Neither ITIL nor the Process Reference Model for IT are directly implementable and to address the gaps between them, IBM developed the IBM Tivoli Unified Process (ITUP).

ITUP describes a comprehensive set of IT processes within an IT organization. It is aligned not only to ITIL Version 3 and the Process Reference Model for IT, but also with best practices from industry-wide specifications of IT best practices, such as ISO 20000, Enhanced Telecommunications Operations Map (eTOM), Six Sigma, and Control Objectives for Information and Related Technology (COBIT), and proposes a process framework that incorporates the best from each. Figure 2-3 on page 15 provides an overview of ITUP.

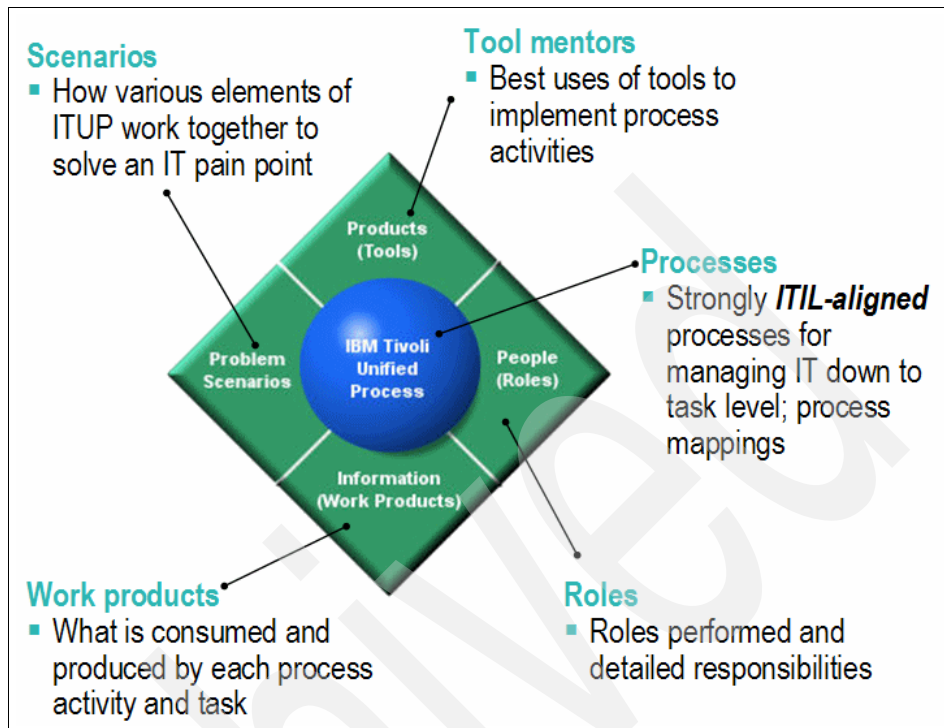


Figure 2-3 ITUP overview

Each ITUP process is defined by:

- ▶ Its purpose, goals, scope, and key performance indicators (KPIs) and relationship to other processes.
- ▶ A workflow.
- ▶ People (Roles).
- ▶ Information (Work Products by Name). This is also not described in ITIL.
- ▶ Products (Tools) that help implement aspects of the process.

In addition, problem scenarios describe how processes work together to solve important IT issues.

IBM Tivoli Change and Configuration Management Database (CCMDB) provides best practice implementations of core ITIL processes.

CCMDB provides a set of foundation services such as common UI services, installation capabilities, and the Configuration Management Database (CMDB), which provides a consistent data model and includes a framework for discovery of resources and relationships from the IT environment.

CCMDB focuses on ITIL Configuration Management and Change Management processes that provide the core management capabilities needed in an IT environment. However, additional tailored processes may be created to serve as a guideline for CCMDb operation.

2.3.1 ITUP Composer

ITUP Composer is a tool that allows for an implementation of the ITUP framework by defining and creating IT Service Management processes that will fit the business needs of an organization. ITUP Composer is shipped with CCMDb Version 7.1 as IBM Rational® Method Composer (RMC). RMC is the tool that enables the development, customization, and publication of methods and processes. ITUP Composer's key elements and concepts are described in the following sections.

ITUP Composer is the product version of ITUP. It is an ideal starting point for organizations looking to implement IT Service Management best practices and document their operational model. Only ITUP Composer has a content library that you can customize, extend, and publish using the ITUP Composer tools.

Figure 2-4 summarizes the differences between ITUP and ITUP Composer.

Feature	ITUP	ITUP Composer
Industry best practices	Y	Y
Process-level information	Y	Y
Activity-level information	Y	Y
Tool use guidance	Y	Y
Task-level information	Y	Y
Content customization		Y
Content creation		Y
Content publishing		Y

Figure 2-4 ITUP versus ITUP Composer feature comparison

The ITUP Composer contains the resources listed in the following sections.

Process library

The processes described within ITUP are closely aligned with ITIL. ITUP contains detailed process diagrams and descriptions to help you understand of processes and their relationships, making the ITIL best practice recommendations implementable. It is possible to also map ITUP to other leading process models.

ITUP is based on the Process Reference Model for IT, which was jointly developed by IBM Global Services and Tivoli. PRM-IT offers detailed process guidance for all activities that are the responsibility of an organization's Chief Information Officer (CIO), including but not limited to IT Service Management.

Tool mentors

Tool mentors describe best practices for using IBM tools in the context of specific processes. A tool mentor helps identify which IBM products and solutions to use to execute specific activities and describes in detail how to use these tools appropriately. This guidance reduces time, effort, and errors, and helps get the maximum value out of the investment.

Roles

IT staff are typically responsible for one or more roles in their job responsibility. These roles are associated with the execution of specific tasks. ITUP describes these roles and responsibilities in detail, and provides tool mentor guidance to enable staff to do their jobs more efficiently and effectively.

Work products

Work products, often referred to as artifacts, are the inputs or outputs of processes. ITUP describes the work products for each process, as well as additional information like definitions of key terms.

Scenarios

Scenarios describe common problems and best practice solutions. With these scenarios, you are able to see how to address real-world problems by improving and integrating processes, using the appropriate tool properly, and setting up the necessary roles and responsibilities.

Underlying the ITUP processes are the layers of supporting autonomic technologies. Autonomic computing provides technologies and standards to support and enable the process environments of the IBM Tivoli Unified Process and IBM IT Service Management offerings.

ITUP is intended for use by anyone who has a role in the implementation and delivery of IT Service Management. IT organizations of varying levels of maturity can use ITUP Composer as a resource to do the following:

- ▶ Deliver IT services through well-defined, repeatable processes
- ▶ Measure and improve business value
- ▶ Tailor IT services to business priorities
- ▶ Better utilize investments in system management tools and use the right tool for a given task
- ▶ Establish IT as a thought leader by harnessing key technologies to drive business value

2.3.2 IBM Rational Method Composer overview

ITUP Composer is a library of the Rational Method Composer, which is a tool platform that enables process engineers and managers to implement, deploy, and maintain processes for organizations or individual projects. Typically, two key problems need to be addressed to successfully deploy new processes.

First, development teams need to be educated on the methods applicable to the roles for which they are responsible. Software developers typically need to learn how to do analysis and design, testers need to learn how to test implementations against requirements, managers need to learn how to manage the project scope and change, and so on. Some organizations assume that developers implicitly know how to do such work without documenting their methods, but many organizations want to establish common, regulated, and repeatable practices to drive specific improvement objectives and to meet compliance standards.

Second, development teams need to understand how to apply these methods throughout a development life cycle. That is, they need to define or select a development process. For example, requirements management methods have to be applied differently in early phases of a project where the focus is on elicitation of stakeholder needs and requirements and scoping a vision, than in later phases where the focus is on managing requirements updates and changes and performing impact analysis of these requirements changes. Teams also need a clear understanding of how the different tasks of the methods relate to each other. For example, how the change management method impacts the requirements management method as well as regression testing method throughout the life cycle. Even self-organizing teams need to define a process that gives at minimum some guidance on how the development will be scoped throughout the life cycle, when milestones will be achieved and verified, and so on.

To that end, Rational Method Composer has two main purposes:

- ▶ To provide a knowledge base of intellectual capital that can be browsed, managed, and deployed. This content can include externally developed content, and, more importantly, can include customized content that can consist of white papers, guidelines, templates, principles, best practices, internal procedures and regulations, training material, and any other general descriptions of any method. This knowledge base can be used for reference and education. It also forms the basis for developing processes (the second purpose). Rational Method Composer is designed to be a content management system that provides a common management structure and look and feel for all content, rather than being a document management system in which existing documents would be stored and accessed, all in their own shapes and formats. All content managed in Rational Method Composer can be published to HTML and deployed to Web servers for distributed usage.
- ▶ To provide process engineering capabilities by supporting process engineers and project managers in selecting, tailoring, and rapidly assembling processes for their concrete development projects. Rational Method Composer provides catalogs of predefined processes for typical project situations that can be adapted to individual needs. It also provides process building blocks, called capability patterns, that represent best development practices for specific disciplines, technologies, or management styles. These building blocks form a toolkit for quick assembly of processes based on project-specific needs. Rational Method Composer also allows setting up of organization-specific capability pattern libraries. Finally, the processes created with Rational Method Composer can be published and deployed as Web sites. They can also be deployed as project plan templates for Rational Portfolio Manager.

Key terminology and concepts

To effectively develop content with the Method Composer, it is necessary to understand a few concepts that are used to organize the content. The topics include 2.3.3, “Method content authoring overview” on page 21 and 2.3.4, “Process authoring overview” on page 23.

The most fundamental principle in Rational Method Composer is the separation of reusable core method content from its application in processes. This directly relates back to the two purposes of Rational Method Composer described in 2.3.2, “IBM Rational Method Composer overview” on page 18. Almost all of Rational Method Composer's concepts are categorized along this separation. Method content describes what is to be produced, the necessary skills required, and the step-by-step explanations describing how specific development goals are achieved. These method content descriptions are independent of a development life cycle. Processes describe the development life cycle.

Processes take the method content elements and relate them into semi-ordered sequences that are customized to specific types of projects.

Note:

Method content describes what is to be produced, the necessary skills required, and the step-by-step explanations describing how specific development goals are achieved.

Process describe the development life cycle. Processes use method content elements and relate them into specific sequences to match different types of projects.

Figure 2-5 on page 21 provides a summary of the key elements used in Rational Method Composer and how they relate to method content or process. Method content is primarily expressed using work products, roles, tasks, and guidance. Guidance, such as checklists, examples, or road maps, can also be defined to provide exemplary walkthroughs of a process. On the right-hand side of the diagram, the elements used to represent processes in Rational Method Composer are presented. The main element is the activity that can be nested to define breakdown structures and how they relate to each other to define a flow of work. Activities also contain descriptors that reference method content. Activities are used to define processes. Rational Method Composer supports two main types: delivery processes and capability patterns.

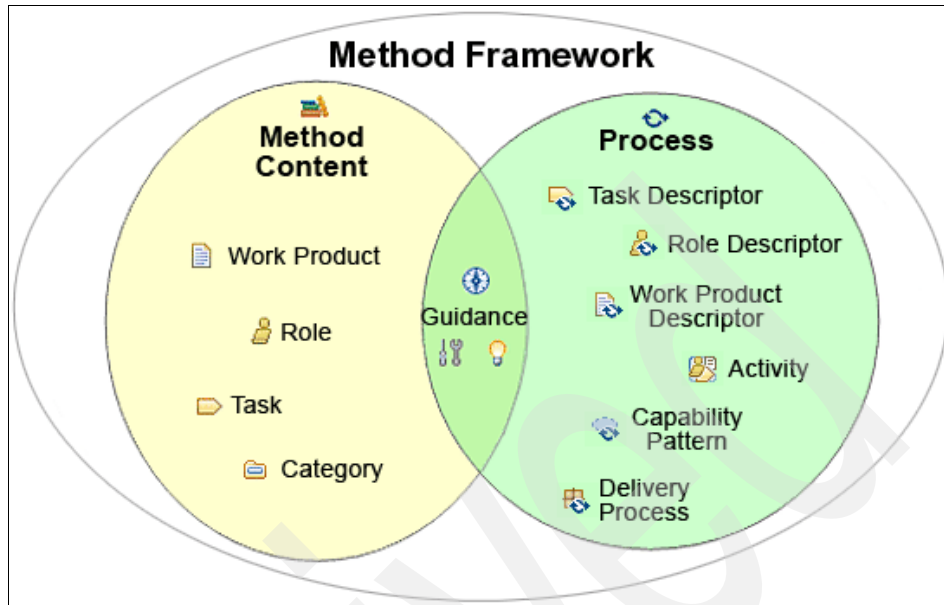


Figure 2-5 Method framework

Delivery processes represent a complete and integrated process template for performing one specific type of project. They describe a complete end-to-end project life cycle and are used as a reference for running projects with similar characteristics.

Capability patterns are processes that express and communicate process knowledge for a key area of interest, such as a discipline or a best practice. They are also used as building blocks to assemble delivery processes or larger capability patterns. This ensures optimal reuse and application of their key best practices in process authoring activities in Rational Method Composer.

2.3.3 Method content authoring overview

Method content describes roles, the tasks that they perform, the work products that are used and produced by those tasks, and supporting guidance.

Figure 2-6 illustrates how method content is represented in Rational Method Composer. Many development methods are described in publications such as books, articles, training material, standards and regulations, and other forms of documentation. These sources usually document methods by providing step-by-step explanations for a particular way of achieving a specific development goal under general circumstances. Some examples are transforming a requirements document into an analysis model, defining an architectural mechanism based on functional and nonfunctional requirements, creating a project plan for a development iteration, defining a quality assurance plan for functional requirements, redesigning a business organization based on a new strategic direction, and so on.

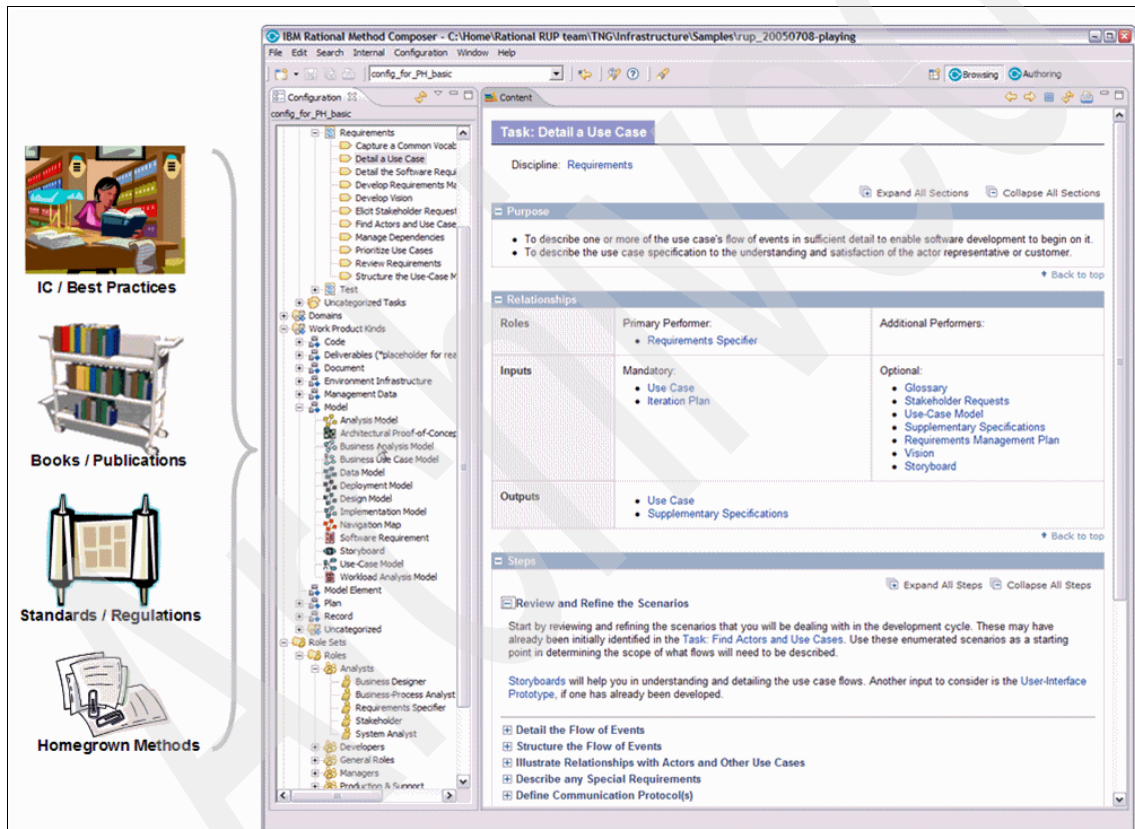


Figure 2-6 Method Content Representation in RMC

Rational Method Composer takes content such as that just described, and structures it in a specific schema of roles, work products, tasks, and guidance. This schema supports the organization of large numbers of descriptions for development methods and processes. Such method content and processes do

not have to be limited to software engineering, but can also cover other design and engineering disciplines, such as mechanical engineering, business transformation, sales cycles, and so on.

2.3.4 Process authoring overview

A process defines sequences of tasks performed by roles and work products produced over time.

Figure 2-7 on page 24 shows that processes are typically expressed as workflows or breakdown structures. Defining a strict sequence, as in a waterfall model, is as much a process as defining semi-ordered sequences in iterations of parallel work. They just represent different development approaches. Hence, for defining a process, one can take method content and combine it into structures that specify how the work shall be organized over time, to meet the needs of a particular type of development project (such as software for a online system versus software and hardware for an embedded system).

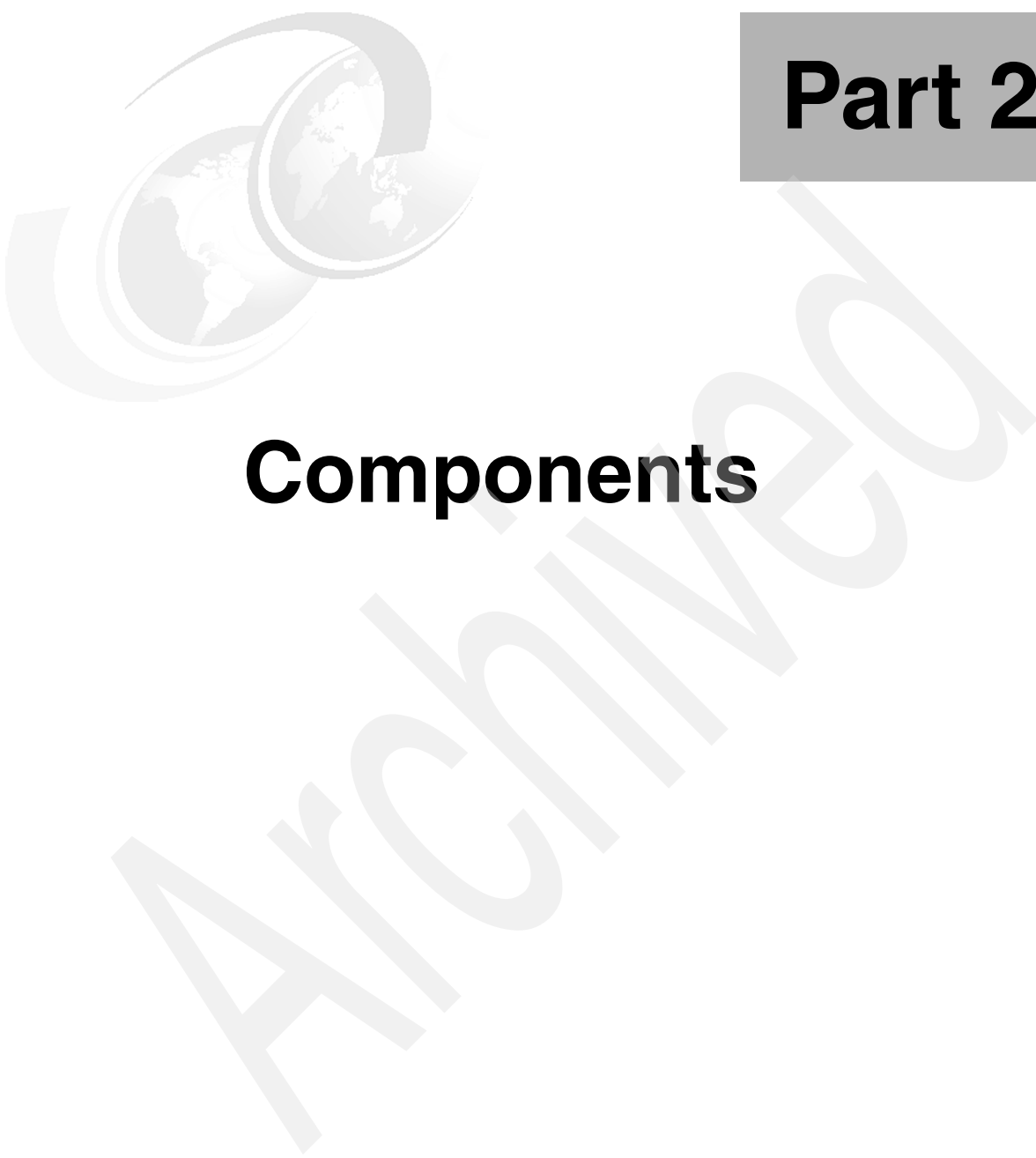
Rational Method Composer supports processes based on different development approaches across many different life cycle models, including waterfall, incremental, and iterative life cycles. Rational Method Composer also supports different presentations for process, such as work-breakdown structure or workflow presentations. It is possible to define processes in Rational Method Composer that use a minimal set of method content to define processes for agile, self-organizing teams.

ITUP Composer and CCMDB

A Tivoli CCMDB Process Manager Product (PMP) can link to the ITUP Web site to provide users with contextual guidance as they:

- ▶ Execute tasks
- ▶ Deliver a documented operational model for compliance
- ▶ Deliver greater consistency for efficiency and effectiveness
- ▶ Deliver a single source of published, standards based, and customer specific process information

More details on the ITUP Composer are provided in *IBM Tivoli CCMDB Implementation Recommendations*, SG24-7567.



Part 2

Components

CCMDB components

This chapter includes a technical overview of the IBM CCMDB components. The two major layers of the IBM CCMDB solution concept are the *Data Layer* and the *Process Layer*. Both of those layers are themselves composed of multiple logical and physical components that are leveraged in the overall solution layout. This chapter explains the components from a logical as well as from a physical and operational perspective.

3.1 Components of the IBM CCMDB

This chapter includes a technical overview of the IBM CCMDB components. The two major layers of the IBM CCMDB solution concept are the Data Layer and the Process Layer. Both of those layers are composed of multiple logical and physical components that are leveraged in the overall solution layout. This chapter explains the components from a logical as well as from physical and operational perspectives.

Figure 3-1 outlines a logical component overview of the IBM CCMDB V7.1 solution. The major building blocks are explained throughout the rest of this chapter.

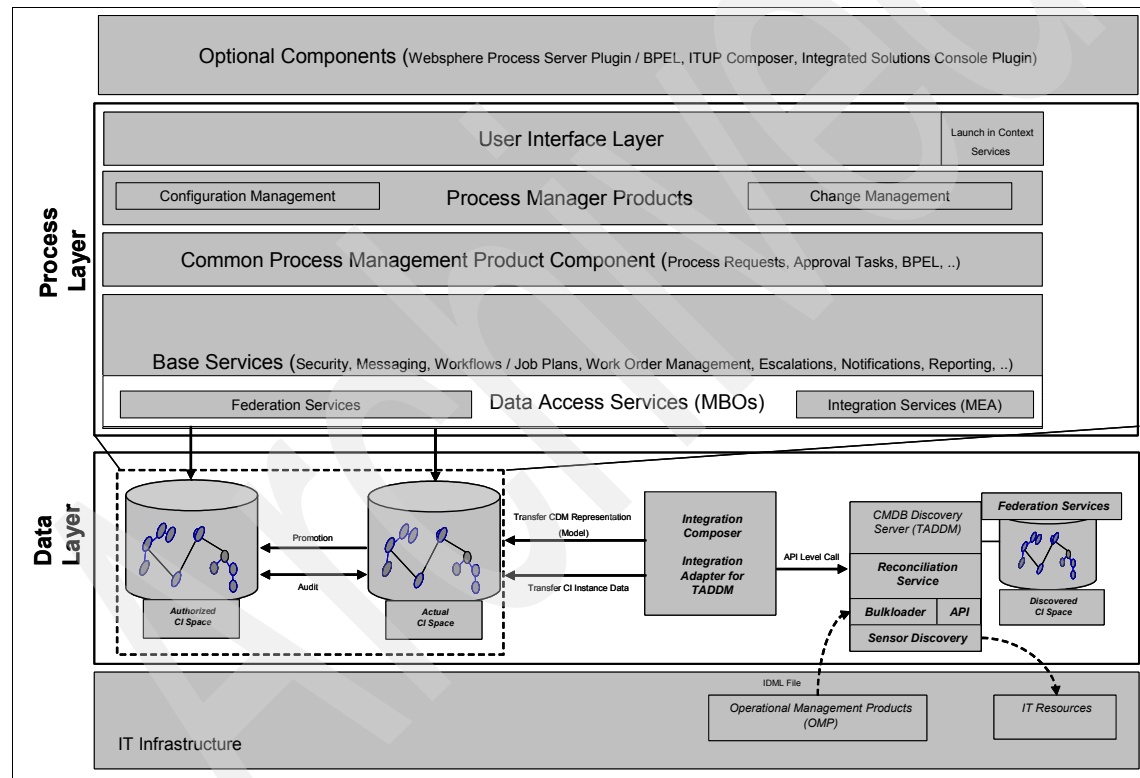


Figure 3-1 Logical architecture overview for CCMDB V7.1

Attention: Please note that Figure 3-1 on page 30 shows the logical component view of the IBM CCMDB solution. An overview of the operational model (physical layout) is documented in Chapter 5, “Physical components and operational model” on page 89.

The data layer is the base of the IBM Service Management architecture. It provides the CCMDB database structure that centralizes (not necessarily physically in one database) all of the information regarding the IT environment. The data layer is comprised of different physical databases that exchange data with respect to Configuration Items. The data layer is physically comprised of the TADDM database that holds information about discovered CIs and their relationships while the process layer database is keeping the same information or a subset of it in order to provide the appropriate data to the Process Manager implementations like Change and Configuration Management.

The Process Layer consists of a foundation that provides runtime services for Process Manager products we refer to as the *base services*. Inside this runtime environment, many common services interface to the data layer, such as Work Order Management to host Service Management process workflows, notifications, reporting, security, or integration technologies. The Process Layer also consists of the IBM Service Management Process Manager Products (PMPs) that leverage the process runtime environment for specific implementations of Service Management processes. CCMDB V7.1 provides two Process Manager products by default: Change and Configuration Management.

The following components and structures are described in more detail throughout the rest of this chapter:

1. CCMDB Discovery Server (also known as the Tivoli Application Dependency and Discovery Manager (TADDM)).
2. CCMDB Base Services.
3. The Process Manager Products (PMP) provided with the IBM CCMDB solution: the Configuration Manager PMP, the Change Management PMP, and the Common PMP as a foundation layer or service for all other Process Management products.
4. The IBM Tivoli Integration Composer (ITIC) and the Integration Adapter for TADDM.
5. Components that can be optionally implemented in the CCMDB solution layout.

6. The CCMDB data layer, with its different representations like Discovered, Actual, or Authorized. The detailed description also discusses the important topic of federation, its pros and cons, when and why to use it, and compares its usage of federation services from within the CCMDB Discovery Server or from within the Base Service Layer.
7. The operational model, including the middleware components leveraged in the CCMDB V7.1 solution. This section also touches on aspects of sizing and high availability of the solution.
8. The CCMDB security architecture.
9. The different interface technologies that CCMDB provides.

3.2 User interface layer

There are two major user interfaces provided by the overall CCMDB V7.1 solution: the CCMDB Web User interface and the interface for the TADDM discovery server.

The CCMDB Web User interface and the interface for the TADDM discovery server. TADDM actually provides two kinds of user interfaces: the Java™ Web Start-based and the Domain Manager user interface. For the purpose of our description, we refer to the TADDM user interface as a user interface.

The CCMDB Web User interface is the user interface for administering and using the process manager products. It also allows the user, given the appropriate permissions for the user, to use the tools to customize the database, workflows, and user interface behavior and style.

The CCMDB Web user interface allows launch in context to external systems. You can specify the target system of the launch operation as well as specific “land-into” views inside the Launch in Context application.

This facility is used to launch in context from the CCMDB Web User interface into different views of the TADDM user interface. Launching from the TADDM interface into the CCMDB Web User interface is not supported.

Optionally, the CCMDB Web User Interface can be used inside the IBM Solution Console or ISC, a common console used across different IBM products, including many IBM Tivoli solutions. If you want to work with CCMDB applications from the ISC console, you can do so. Please refer to the CCMDB documentation for more details.

3.3 CCMDB discovery server and TADDM

The Tivoli Application Dependency Manager (TADDM) is the discovery server in the IBM CCMDB V7.1 environment. It provides discovery aggregation and reconciliation functionality. It provides a topology view into the discovered data space, also referred to as the Discovered CI space.

There are existing IBM Redbooks publications on the TADDM capabilities and functions. We are not describing the base functions in detail in this book but rather describe the role of the TADDM server within the overall CCMDB solution context.

For more information about TADDM, refer to the IBM Redbooks publication *IBM Tivoli Application Dependency Discovery Manager Capabilities and Best Practices*, SG24-7519.

Discovery is based on the scheduled execution of a discovery profile that defines what to discover as well as the depth of discovery. Discovery execution uses discovery sensor implementations for numerous kinds of target systems, middleware, and application components. In addition, a command-line interface is provided to mass import discovered data that are provided by external operational management products (also known as Managed Software Systems (MSS)). We call this bulk loading interface the *bulkloader*. The bulkloader expects the external data that should be imported to adhere to an XML format called *IdML*. Besides the bulkloader interface, the TADDM server provides an API that also allows data to be read and imported. The API expects the data in an XML format as well, but not *IdML*.

The reconciliation process avoids generating duplicates of CIs in the system. Whether you update data using sensors, custom server templates, the bulkloader, the API, or manual data entries, the reconciliation engine in the TADDM server ensures that the data is unique if it belongs to the same entity. Reconciliation occurs while reading the data from its source and before storing it into the database. For example, if you provide the TADDM system with data for the same CI by entering data through sensor discovery and a bulkloader import, naming rules detect if data provided by the second data import belongs to a CI already present in the system.

The TADDM server also provides federation capabilities. This is to extend reporting views of the discovered CI space with data federated from external data sources. Please note that data that is federated from within the TADDM system is not transferred by the Integration Composer to the process layer of the CCMDB V7.1. This means that you cannot use data that you have federated from within the TADDM system from within a Change or Configuration Management process. If you want to federate data and use it from within the Service Management

processes, you have to set up the federation capabilities from within the process layer of the CCMDB. Please refer to *IBM Tivoli CCMDB Implementation Recommendations*, SG24-7567.

The TADDM Server is separate from the CCMDB Base Services, which provide the basis for the Service Management processes like Change and Configuration Management. TADDM can be used as a stand-alone solution if just a discovery solution is relevant without using discovered CI data in the Service Management processes. Although the TADDM server can be installed on the same system as the Base Services component, we recommend installing TADDM on a separate machine from Base Services. The TADDM server contains its own database.

Whether you implement the database on the same system as the TADDM Server or on a separate system depends on your environment and operational aspects. Both options are provided. The TADDM server also contains its own graphical user interfaces, a Web interface called the Domain Manager, and a Java Web Start based user interface referred to as the product console. You can launch into both of the TADDM consoles from the CCMDB Web interface, which is the user interface of the CCMDB used in the process environment (that is, by a Change or Configuration Manager).

You can scale the discovery environment of the overall CCMDB solution by adding additional discovery domains if your environment needs to discover a large number of systems or requires independent operations of discovery domains. Each domain server reports its data to a central instance called the enterprise discovery server. Please refer to Chapter 5, “Physical components and operational model” on page 89.

With respect to the data layer perspective of the overall CCMDB solution, TADDM as the discovery engine of the CCMDB provides the Discovered CI space. Detailed CI data, including relationships, are captured in the TADDM database. In order to expose the discovered data, the Integration Composer component of the CCMDB needs to transfer the data into the Actual CI space that is physically located in the database of the process layer. The Integration composer, a generic data import component, is specifically instructed by the TADDM Integration Adapter how to connect to TADDM, what data to transfer, and how to do the mapping between the Discovered CI space and the Actual CI space.

Note: There is a TADDM adapter available for previous versions of TADDM to transfer data into the CCMDB V7.1 database. Please note that this previous adapter transfers data into Asset representations inside the asset management database. The Integration Adapter for TADDM in the CCMDB V7.1 solution populates CI structures and instance data rather than Asset Data. The TADDM integration adapter within the CCMDB V7.1 solution uses the TADDM API, while the previous version uses a JDBC™ connection to the TADDM database.

The Integration Adapter for TADDM component actually contains two integration adapters that you run separately. The first one transfers the Common Data Model to create CI types including relationship models into the Actual CI representation of the process database of the CCMDB. If you are enhancing the data model in the TADDM environment, the model or CI type adapter must be run in order to reflect the updates in the process database. The second adapter transfers the real instance data. Please refer to Chapter 10, “TADDM and Process Layer integration” on page 231 for more information about the Integration Composer.

Important: It is very important to understand that in CCMDB V7.1 the only source for Actual CI data is to go through TADDM and transfer the data into the Actual CI space through the Integration Composer technology. You cannot manually create Actual CIs because the Actual CI representation is read only data. If you want to import your own discovered data into the Actual CI data representation, you have to import your data into the TADDM database first through the TADDM API or an IdML / DLA file and then use the Integration Composer technology. This is true for CCMDB V7.1 and might change in future releases. Please bear in mind that you can create Authorized CIs manually since they do not represent what has been discovered in the environment.

Enhancements in TADDM V7.1

Within the CCMDB V7.1 solution, the TADDM discovery server has been enhanced with new functionality. Although we do not provide a detailed description of how to use those enhanced features, we want to give an overview of the key enhancements. This list is not a full list of each and every new feature provided in Version 7.1, but highlights the most important enhancements. Some of the enhancements are not exposed to customers but are for development or IBM services people only. We do mention them though since they are important enhancements for the overall CCMDB V7.1 solution.

Note: We do expect the reader of this section to have some prior experience with previous versions of TADDM. We inserted example window highlighting some of the new functions but do not explain in detail how to use the product to get to the appropriate dialogs.

Discovery enhancements and new sensors

There are a couple of enhancements in the area of discovery:

1. There are new sensors, such as sensors for Microsoft® Exchange or iSeries®, as well as support for SNMP V3 in the SNMP sensor that already existed. Figure 3-2 on page 37 highlights these new sensors inside a newly created Discovery Profile.

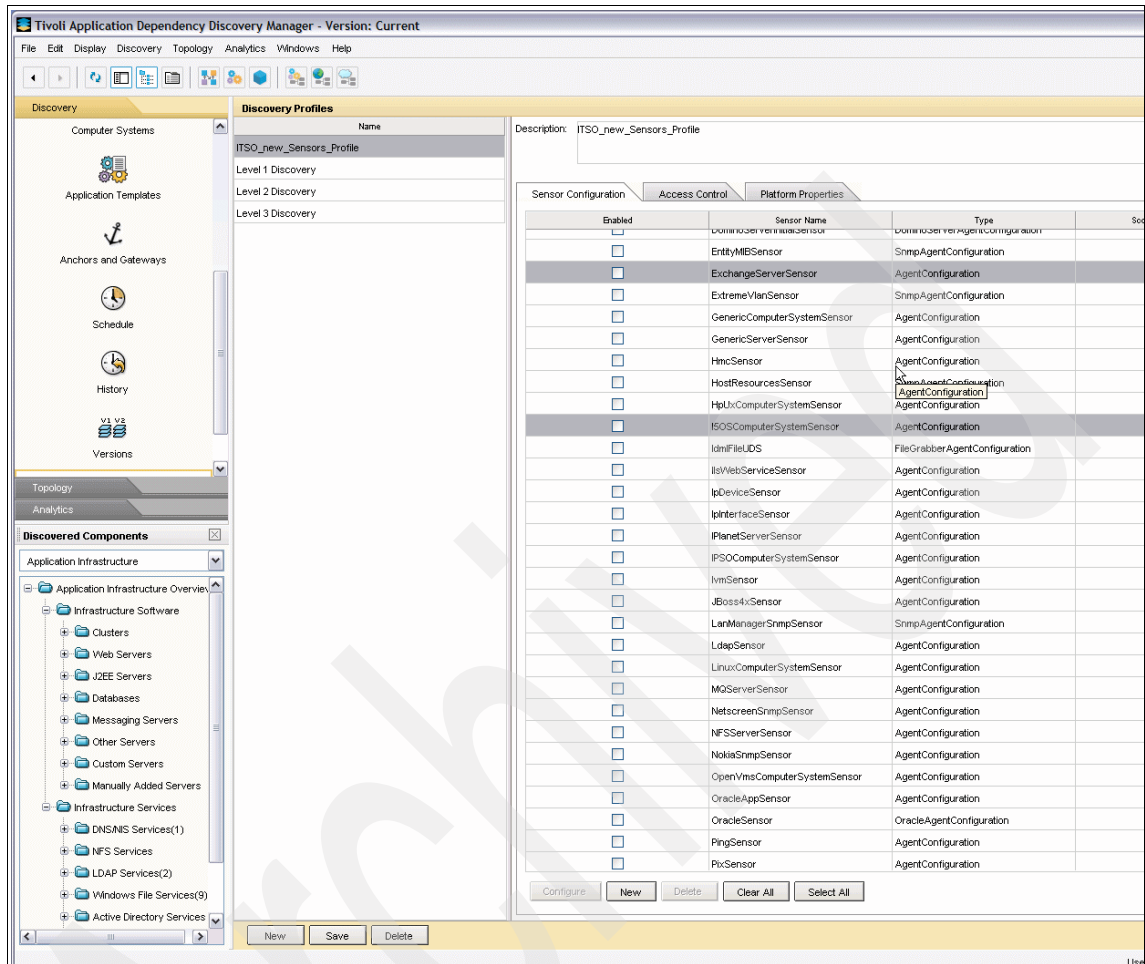


Figure 3-2 New sensors in Discovery Profile

Tip: As you can see in Figure 3-2, access controls can now be defined inside a Discovery Profile.

- The Level 1 Discovery, also known as the Stack Scan Sensor, has been enhanced to use SNMP in addition to the previous protocols to inspect the infrastructure. Figure 3-3 shows the default Level 1 Discovery Profile that has been enhanced by using an additional SNMP sensor.

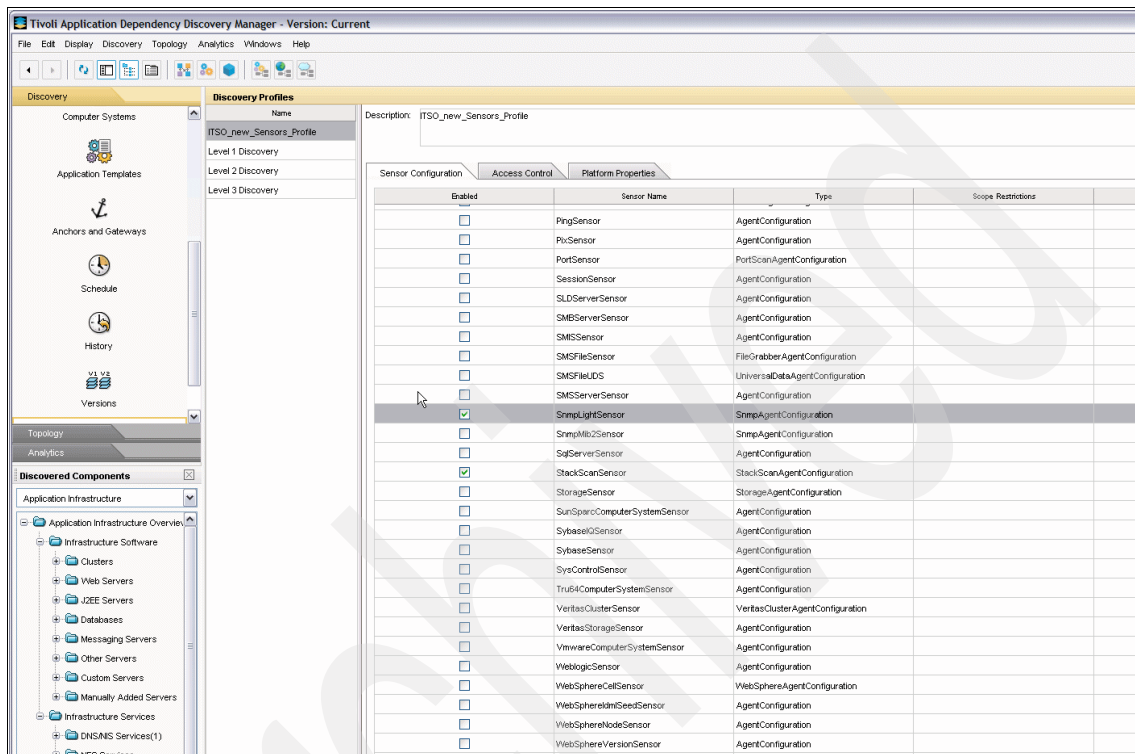


Figure 3-3 Level 1 Discovery Profile

- There are performance enhancements that make discovery runtimes faster. Enhancements have been made in order to write discovered data into the database faster so the overall elapsed time from beginning to end of a discovery is reduced. In addition, specific sensors like the WebSphere® or DB2® sensor have been specifically enhanced by using parallelism in the sensor implementations.
- Rediscovery of specific CIs. There is a new function that allows you to rediscover a single CI without having to go through the process of discovering a whole scope set or having to specify a scope just for the specific CI. You have to enable the function in the collation.properties file. The statement that you have to set to true is `com.collation.rediscoveryEnabled`.

Once you have done this, you will find a new dialog option in the Java console, as shown in Figure 3-4 on page 39.

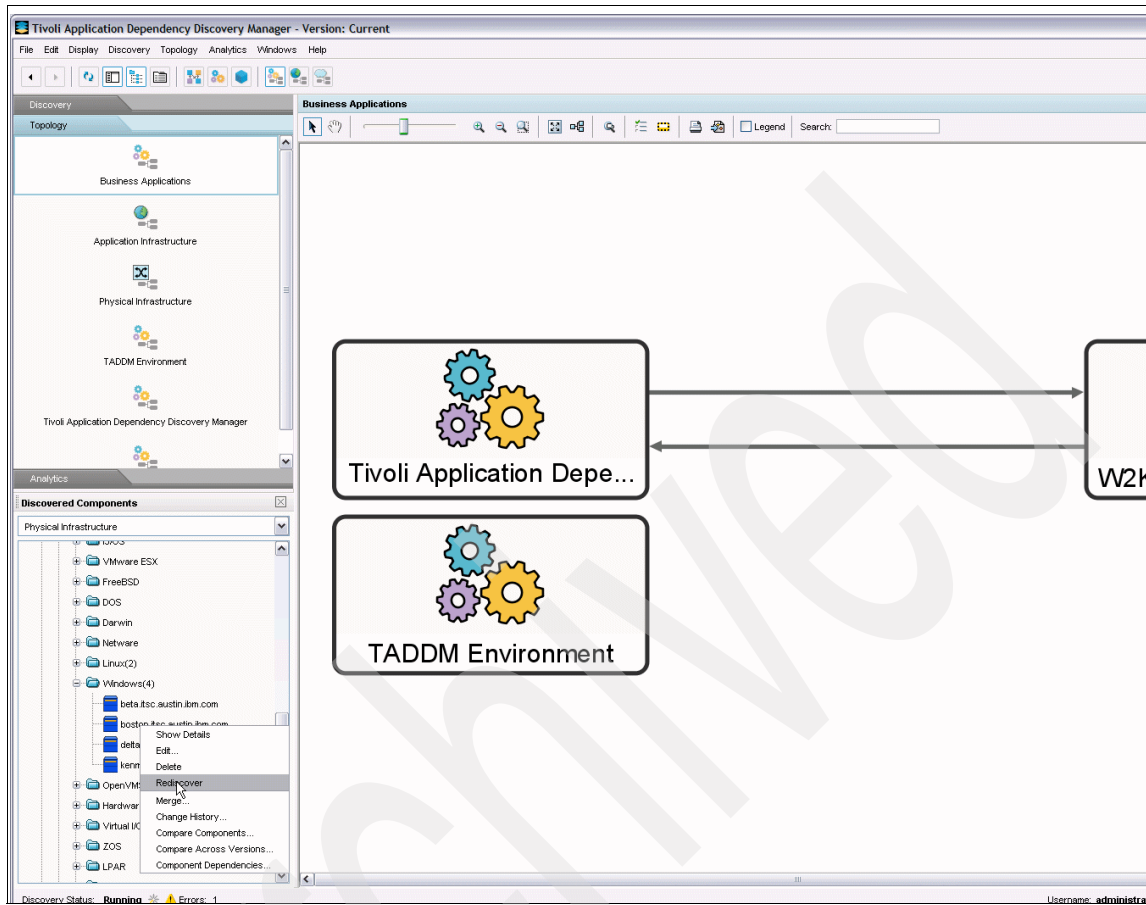


Figure 3-4 Rediscovery of configuration items

- JAR Files for the WebSphere Sensor are now included in the product. You do not have to install a complete WebSphere Application Server on the TADDM server anymore in order to discover a WebSphere environment. The provided JAR files are for WebSphere Application Server versions 5.1, 6.0, and 6.1. You can define the depth of the WebSphere discovery inside the Discovery Profile. Please refer to the product documentation for a more detailed description on how to configure this option.

Extensibility of the Common Data Model

In earlier versions of TADDM, extensions to the Common Data Model were restricted to the IBM development organization and were bound to the schedule of a product release or Fix Pack. You could only extend the model for new attributes, also referred to as Extended Attributes. There are enhancements in

Version 7.1 that enhance the Common Data Model (CDM) for new classes dynamically, including attributes. This function is not yet open to customers at this point. Its usage is restricted to the IBM development organization as well as IBM services.

1. Extended Attribute type support

You now can define extended attributes and define a data type. A data type definition was not supported in previous versions of TADDM. This function is officially supported now in the product. Please refer to Figure 3-5 for more details.

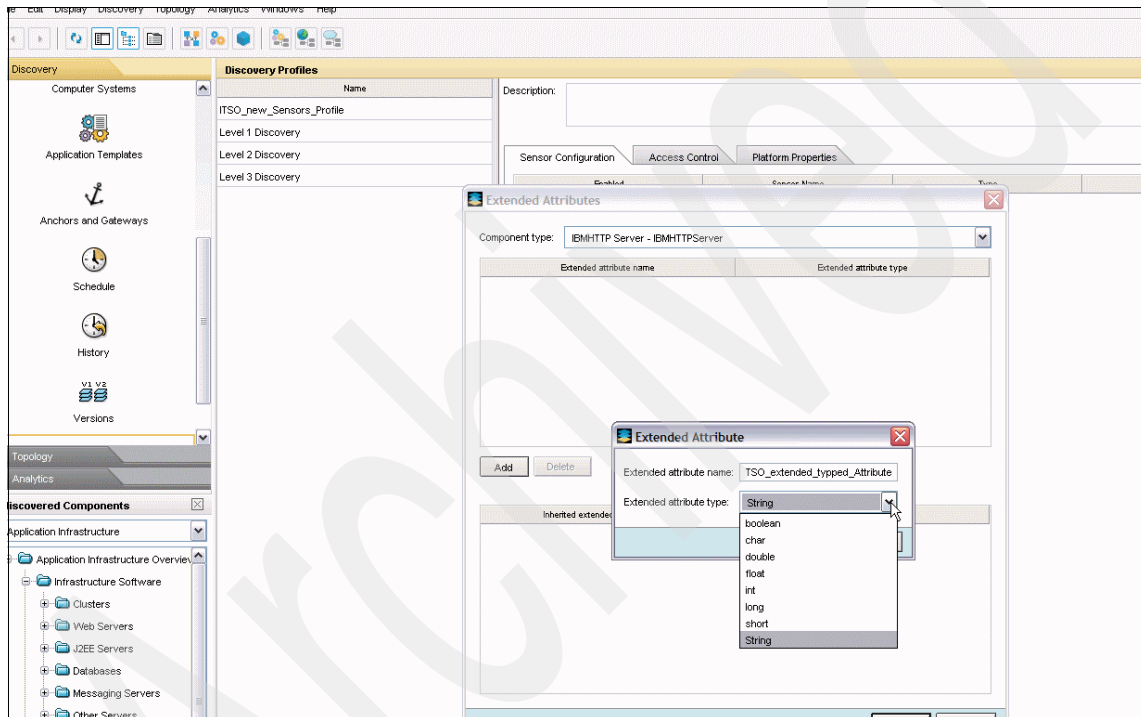


Figure 3-5 Extended Attribute type support

You can see which data types are supported in Figure 3-5.

Tip: You can now also use Extended Attributes with Custom Server Templates.

2. Dynamic Extension of the Common Data Model

The ability to dynamically define new object classes supported by the TADDM system has been introduced primarily to streamline a faster development

cycle of new pluggable sensors outside the product development cycle. As mentioned already, the extension will not be accessible by customers, but they will benefit from this functionality during their deployments by service teams.

The dynamic extensibility of the CDM allows for the introduction of new classes, the addition of subclasses to existing and newly defined classes, the definition of new attributes, and the definition of naming rules for the newly defined classes. In addition, implicit relationships between the new classes and other classes can be defined.

Although Figure 3-6 looks as though it is a part of the TADDM product console, it is a separate User Interface restricted to IBM usage. In order to define new classes, you have to use this User Interface. Alternatively, you can use XML definitions directly.

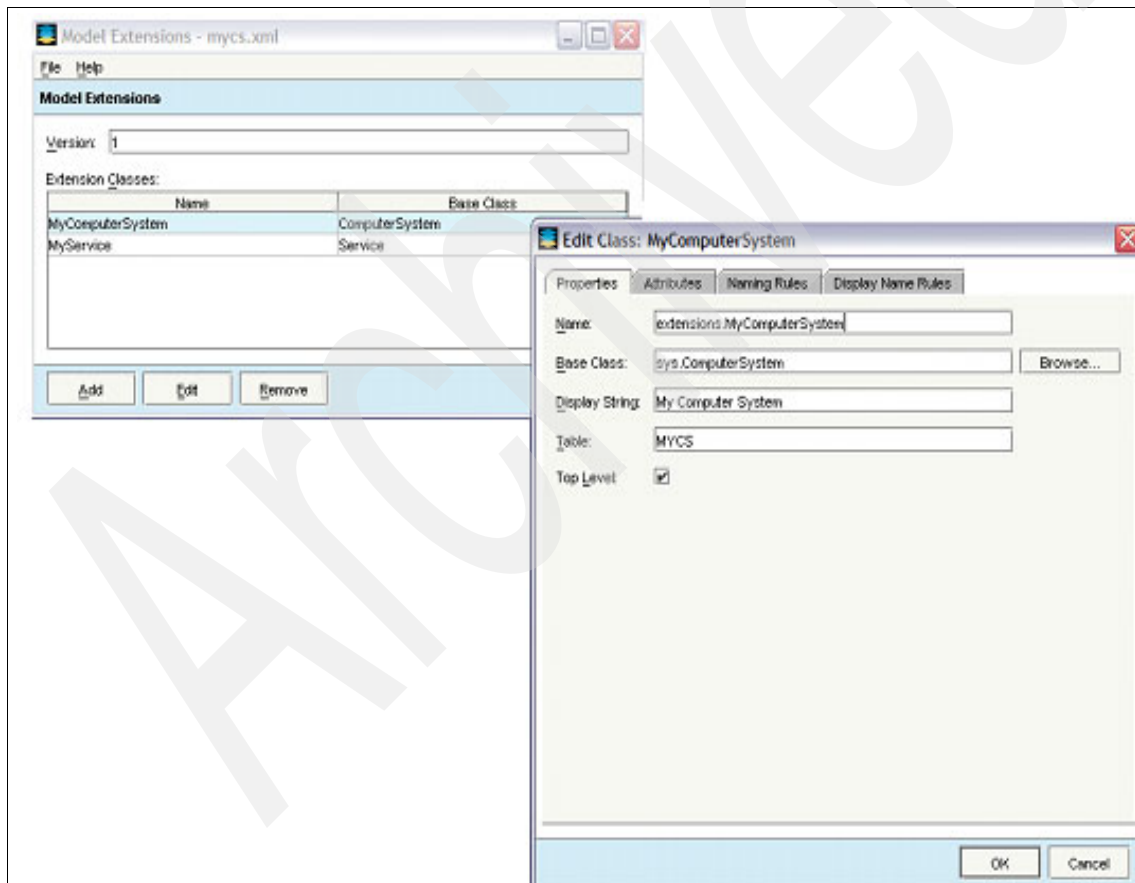


Figure 3-6 Model extension in TADDM

Figure 3-7 shows the alternative way of defining an extended class without using the UI using XML file structures directly.

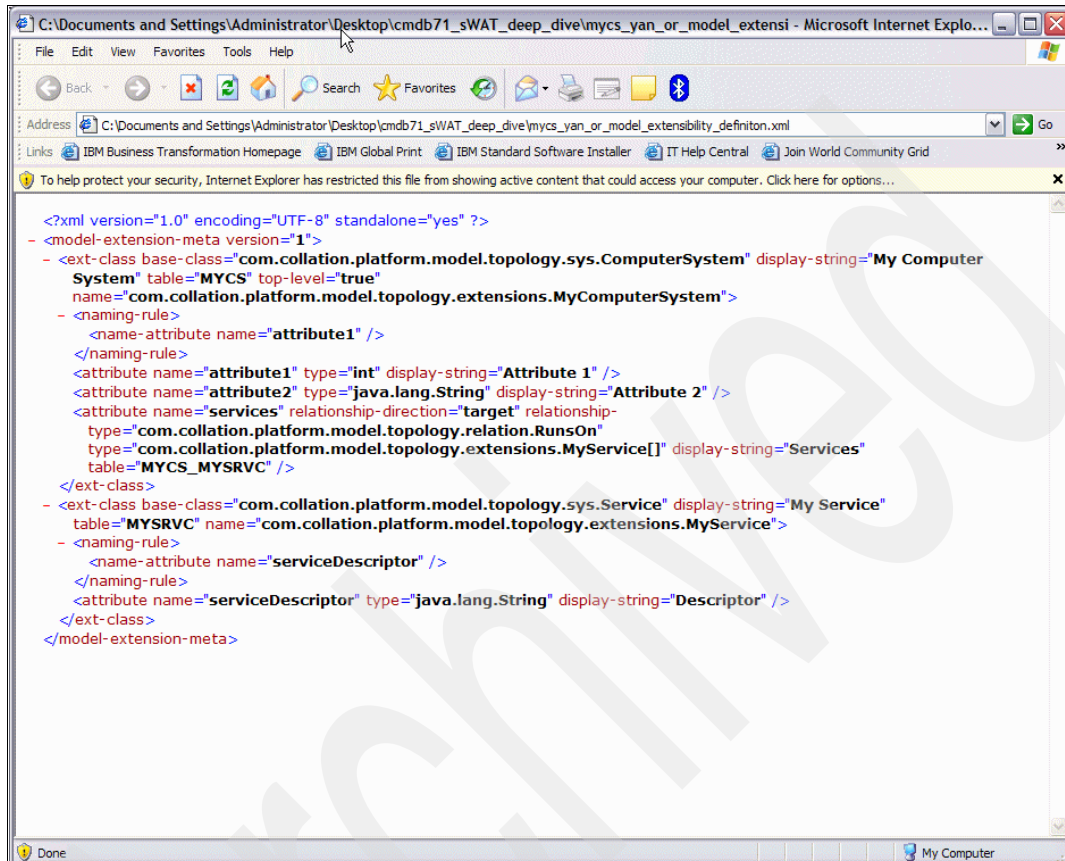


Figure 3-7 Extending the model using an XML based definition file

You can see that a new class called My Computer System is extending the base class Computer System. In addition, new attributes (attribute1, attribute2, and services) are introduced, inheriting the attributes from the base class. You also see that a naming rule for the new class is defined that uses attribute1 as the identifier for the TADDM system to generate an instance of class My Computer System. Besides the extension class for the Computer System class, you find another section in the XML file that defines a new class MyService to extend the Service Class. Before you can use the new classes in the TADDM system, some dynamic code generation, User Interface definitions, and deployment steps have to be done. We do not document these steps in this book.

Developing pluggable sensors

One of the reasons to allow dynamic extensions to the Common Data Model is to streamline the cycle time of new sensors. Prior to TADDM V, new sensors had to be part of a complete development build, either a major or minor version of the product.

Now it is possible for IBM development or IBM SWAT and service personnel to add new sensors to the TADDM system without doing a rebuild of the TADDM code base. Easier Data Model extensions support the introduction of new sensors that usually make use of their own class and attribute structures.

It is out of the scope of this book to document the process of developing sensors.

Reconciliation enhancements

There are major enhancements in the TADDM version of the CCMDB V7.1 product in the reconciliation process. The reconciliation process runs when reading data into the system, whether it comes from a sensor based discovery, a custom server template, an OMP DLA, an API call, or is entered into the system manually. Based on defined naming rules, it identifies data belonging to the same CI and tries to avoid duplicates. The enhancements belonging to this process are *attribute prioritization*, a manual way to reconcile data (also referred to as *manual merge*), and a way to plug in your own reconciliation logic to the system. We briefly describe these enhancements:

1. Manual reconciliation (manual merge)

The manual reconciliation enables the user to manually combine CIs using the TADDM User Interface without using a naming rule. For example, if you detect that you have two CIs in the TADDM database and you know that they should be represented as one, you can use the Manual Merge function to bring them together in one object. Within this process you also converge the naming rules of the objects you are bringing together if they are different.

Figure 3-8 shows an example of how to make use of the merge functionality. From the tree view in the left frame of the console or from a topology view, you have to select at least two objects that you want to merge together. Please bear in mind that they have to be of the same type. For example, you cannot merge a Windows® and a Linux® Computer System object.

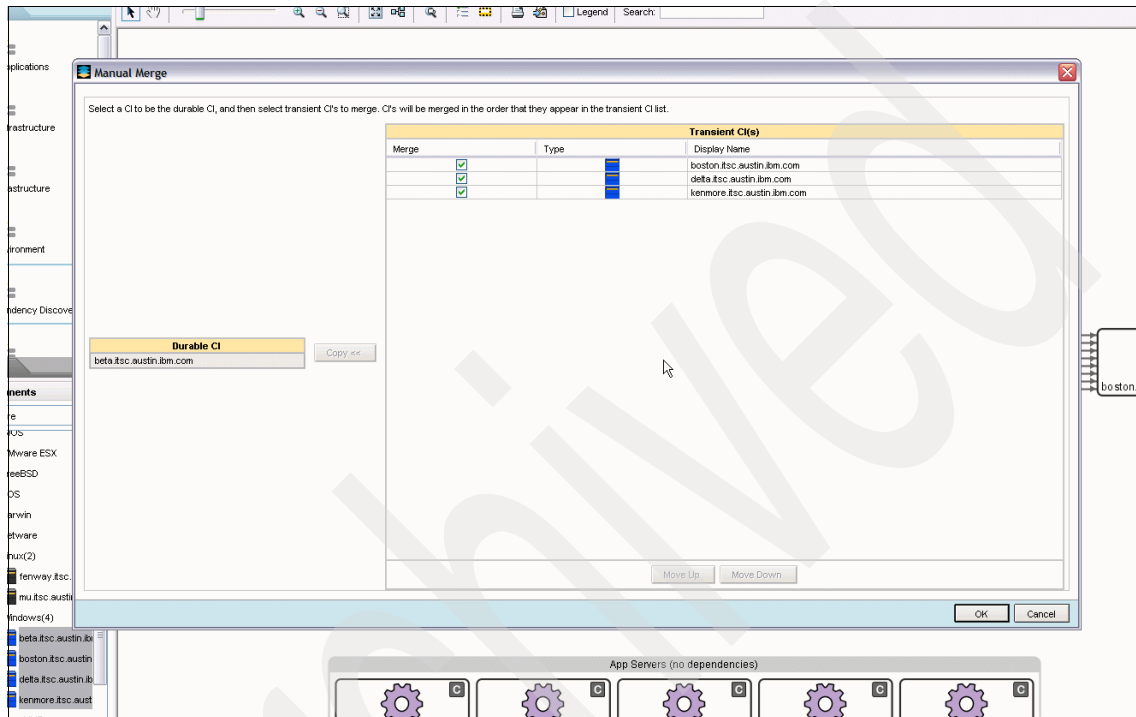


Figure 3-8 Manual merge

As depicted in Figure 3-8, you have to designate one of the CIs that you have selected as the *durable CI*, which is the object that will stay in the database after you merge the CIs together. The transient CI(s) (you can actually select more than one given they are of the same type) are those objects that will be merged into the durable CI when merging the objects together and will be deleted from the database after the merge operation.

The GUID alias table in the TADDM database is updated during a merge operation so that after the merge operation, when you start a discovery or load data from a DLA book, the operation will update the object specified as the durable object during the merge operation.

2. Attribute Prioritization

In TADDM versions prior to Version 7.1, when reading data into TADDM for the same CI attribute (for example, capturing an attribute specifying the disk size of a computer), from different sources (like a sensor discovery and a DLA import), TADDM would use the data from the last source that brought data into the system. You can call that approach a “last win” approach. In TADDM V7.1, you can specify the priority of data sources. That means you can define which data source you rely on most to provide data for specific attributes. A data source can be a sensor, a DLA, the user interface, or any other source writing data to the system. The dialog to define prioritization rules can be launched from the TADDM Java GUI Edit Menu (Figure 3-9).

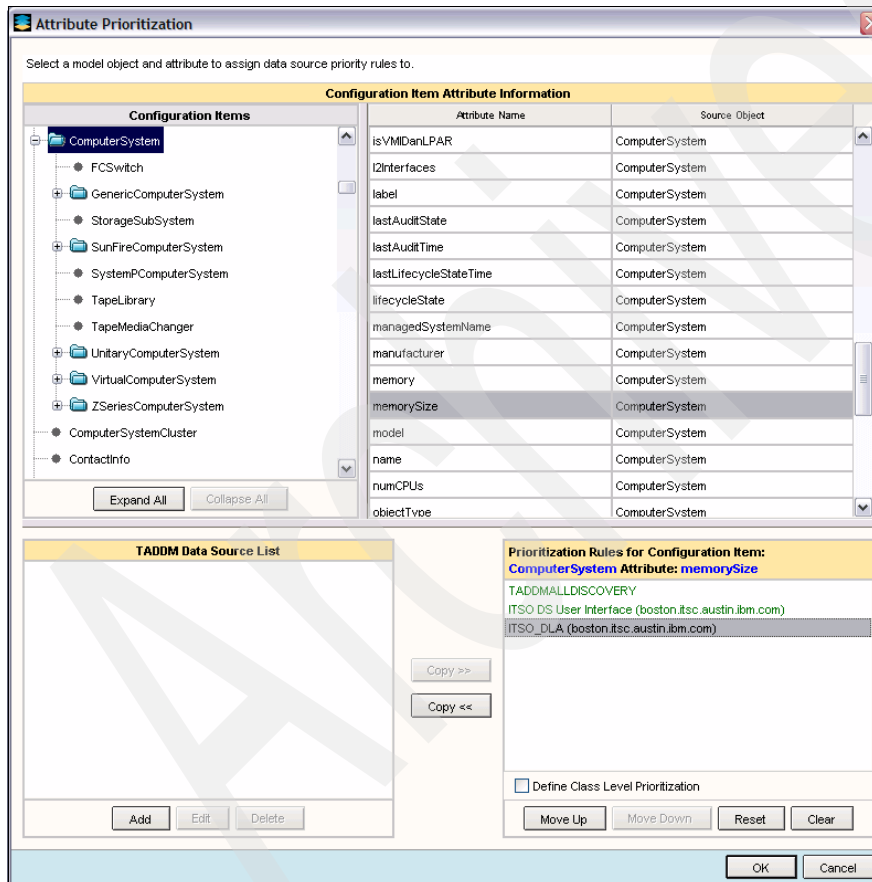


Figure 3-9 Prioritization rules

In order to define rules, you have to define Data Sources first. In Figure 3-9 on page 45, you see three Data Sources defined. All of them have been moved to the frame on the lower right side (using the **Copy** button) in order to use them to define a rule for a specific attribute.

The data sources that have been defined in the example are TADDMALLDISCOVERY, which is a default Data Source specifying all sensors, ITSO DS User Interface, a defined Data Source specifying a manual way to enter data into the system, and ITSO_DLA, which specifies a DLA that imports data into the system through the bulkloader facility.

In the lower right frame of the dialog you can specify the priority order that you would like to be applied for data entering the system for a specific attribute or a whole class structure (if you select the **Define Class Level Prioritization** check box).

In the upper right frame of the dialog, you see that the memorySize attribute is define for the priority rule, which is an attribute of the ComputerSystem Class. The data source that has the primary role of being a data provider is the sensor discovery (TADDMALLDISCOVERY).

Please refer to the product documentation for more details on how to set up attribute prioritization and best practice recommendations.

3. Reconciliation plug-In

If you want to plug in your own logic before data is stored into the database, TADDM V7.1 provides a facility for providing your own reconciliation logic. An example would be if you want to look up specific data already in the TADDM database and manipulate the incoming data appropriately to your needs before you finally store the incoming stream in the database. You would have to code the logic (Java) and plug it into the reconciliation framework. In TADDM V7.1 this facility is available to IBM Development, SWAT, and Service personnel only.

Figure 3-10 depicts the behavior of the reconciliation framework and plug-in.

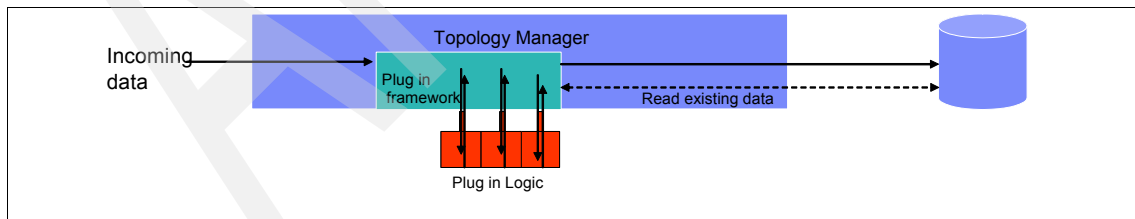


Figure 3-10 Pluggable reconciliation architecture

The reconciliation is the actor between the incoming data stream and the point in time before data gets persisted into the database.

Enhancements for an Enterprise Discovery Server Environment

There are multiple enhancements to the implementation of a discovery environment consisting of multiple discovery domain servers and a central instance called the enterprise discovery server (also referred to as the Enhancements for an Enterprise Discovery Server Environment (eCMDB) instance). The most prevalent ones are:

1. You now have topology views (all supported views except the Application Infrastructure View) through the Domain Manager user interface for an eCMDB environment. These topology views can span data being synchronized from different discovery domains.
2. The default synchronization level between the Domain servers and the enterprise server is switched to *full synchronization* in TADDM V7.1.

You now can have cross-domain references or relationships visible at the eCMDB level. Prior to Version 7.1, you had to use overlapping scope sets in the discovery domains in order to reflect cross-domain business application relationships. In V7.1, a topology builder doing post synchronization steps has been introduced in order to allow cross domain relationships without setting up overlapping discovery scope sets.

Note: The eCMDB or enterprise discovery server instance, if it is being used, is the interface for the Integration Composer, which synchronizes data from the Discovered CI space to the Actual CI space.

Security enhancements

The most prevalent enhancement is the security integration between the process layer run time and the discovery environment (TADDM) of the CCMDB V7.1 solution. This allows the process layer and the discovery component to share a common user and group repository based on a Directory Server (either IBM Directory Server or Active Directory®). This allows a single sign-on between the CCMDB V7.1 Web User Interface used in the process layer of the CCMDB and TADDM. The Launch in Context facility leverages this shared security infrastructure and passes a security token from the process layer to the TADDM server in order to launch without a reauthentication into a specific view of the TADDM system. Please refer to Chapter 6, “CCMDB security architecture” on page 105 for a more detailed description of the overall CCMDB V7.1 security architecture.

One of the other security enhancements that has been introduced in the discovery environment of the CCMDB V7.1 is the ability to define groups. You use the Administration function in the Domain Manager User Interface to define groups of users to whom you then assign roles in order to assign specific authorizations to the members of the groups.

Figure 3-11 shows the Domain Manager User Interface function that defines a new group in order to assign permissions to that group.

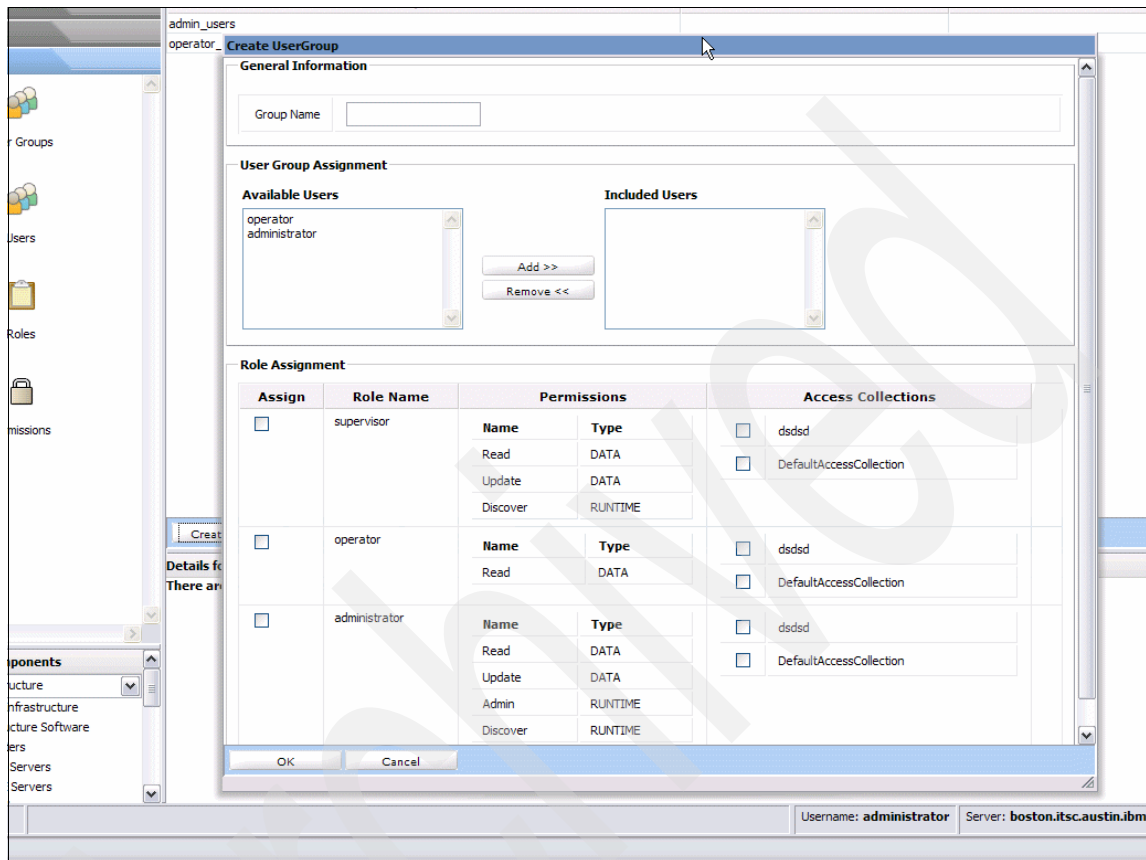


Figure 3-11 UserGroup Definition in TADDM Domain Manager Interface

Miscellaneous enhancements

In this section, we provide an arbitrary list of further enhancements introduced in the TADDM V7.1 component of the CCMDB V7.1 solution.

1. Query based collections.

You now have the ability to define collections, and through this ability, access collections as well, by creating a query. For example, you can define a collection (access collection) displaying or restricting access to a dynamically changing group of objects, define a query, and create a collection out of it. Figure 3-12 on page 49 highlights this functionality. Please refer to the product documentation for more details on how to set this up.

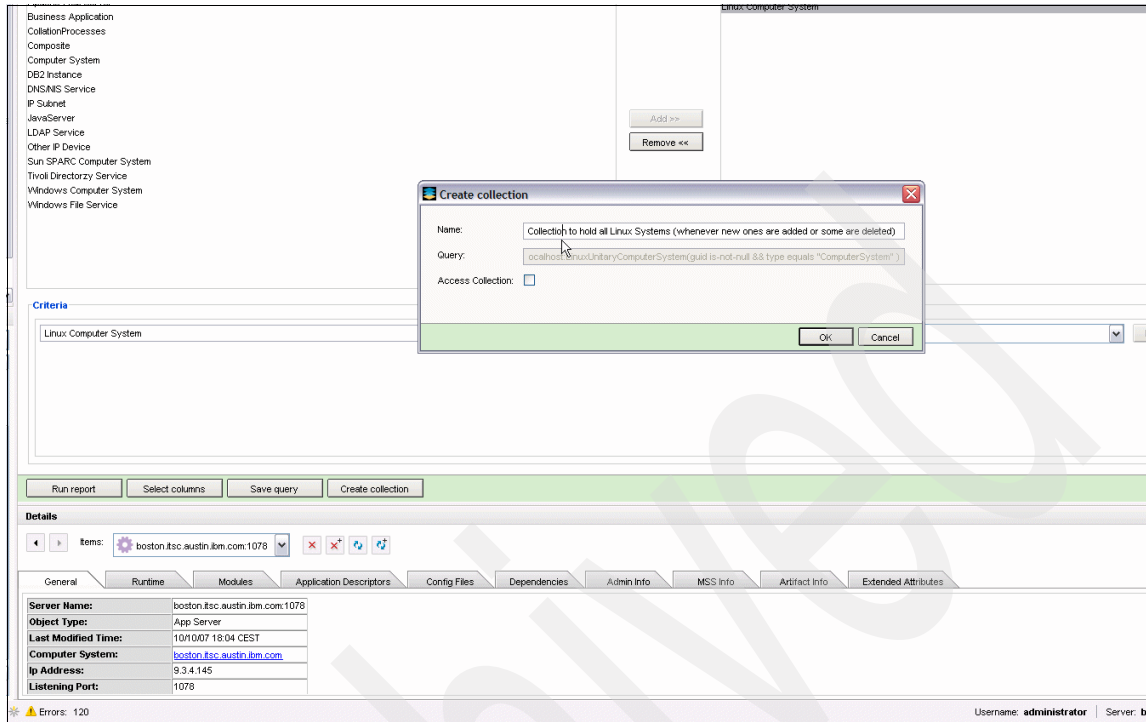


Figure 3-12 Query based collections

2. There is now support the Cygwin SSH implementation for communication between the TADDM server and the Windows Gateway and Anchor.
3. Support for extended attributes in Custom Server Templates.
4. Several Performance enhancements, primarily in the bulkloader and API.

5. Enhancements within the Domain Manager User Interface.

The usability of the Domain Manager user interface has been improved, primarily by adding the ability to view topology graphs inside the Domain Manager Web interface, as shown in Figure 3-13.

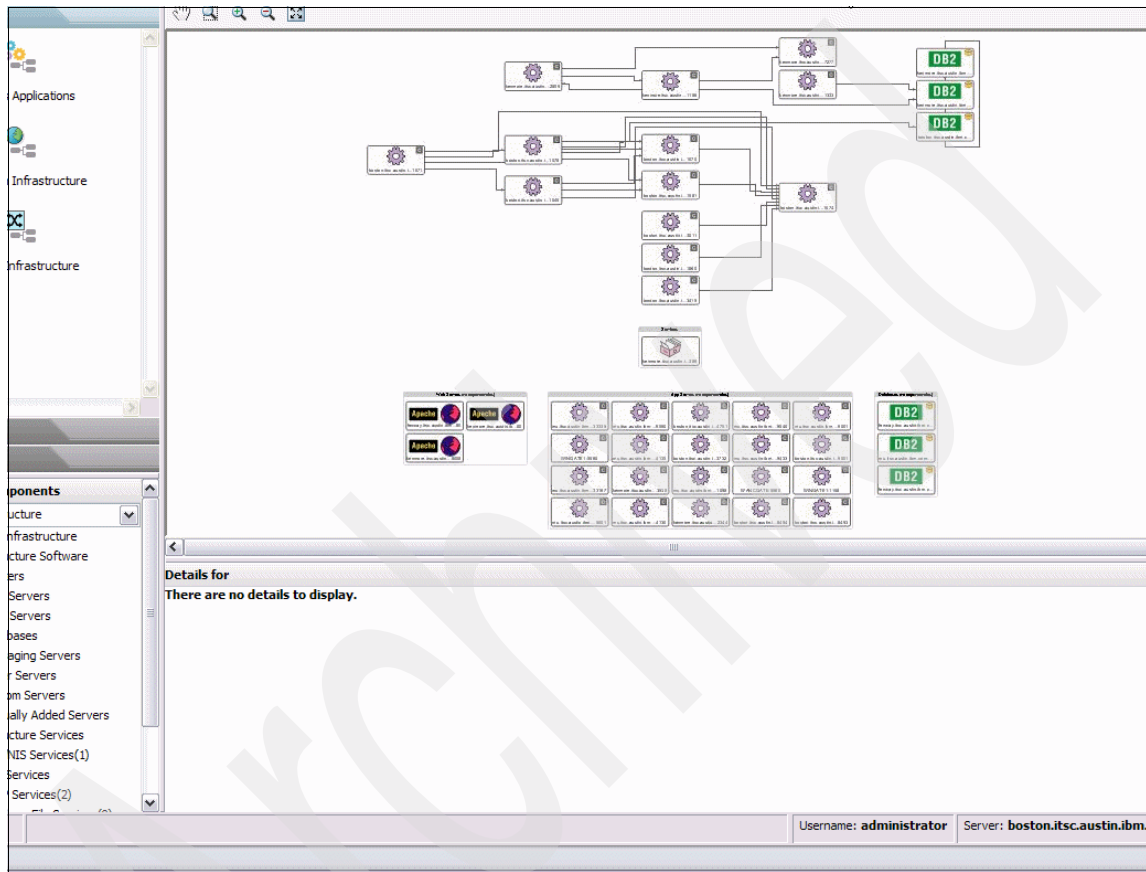


Figure 3-13 Domain Manager topology graph

Most of the functionality exposed through the TADDM Java UI is exposed through the TADDM Domain Manager interface in V7.1. The most obvious missing function in the Domain Manager UI is the ability to start and configure discoveries.

The capability to launch in context into different views (Business Topology, Application Infrastructure, Details view, Change History, and so on) of the TADDM user interfaces (Domain Manager as well as the Java UI) has been enhanced. This is most often used when launching in context from the CCMDB V7.1 Web user interface that is the interface of the process layer of

the CCMDB. Inside the process environment, there is a Launch in Context application that allows you to define launch points into TADDM and other related operational management products. Figure 3-14 gives an example of a predefined launch point defined in the CCMDB V7.1 Web user interface.

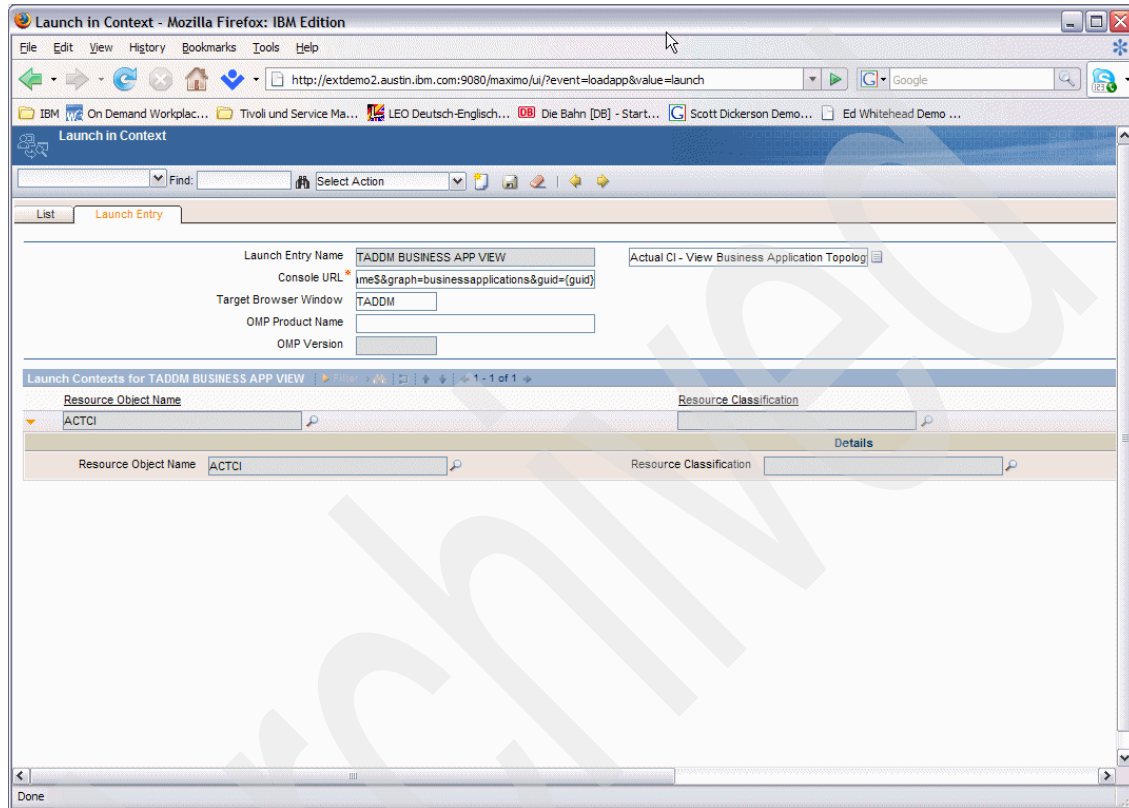


Figure 3-14 Launch in Context definition

It specifically defines a launch point from the Actual CI configuration application inside the process layer of the CCMDB V7.1 into the TADDM user interface, in this case the business application topology. The qualifier for launching in context into the TADDM system is the General Unique Identifier or GUID. The GUID is being used as a variable in the Console URL definition.

3.4 CCMDB Base Services

The *Base Services* component of the CCMDB V7.1 solution is the component that provides the platform upon which process manager products like Change Management or Release Management run and provide their specific functions. It is the foundation layer of the IBM Service Management process layer with several technical functions, including the following:

- ▶ A common security model to support users, user groups, sites, organizations, security profiles, persons, or person groups.
- ▶ A facility for all applications to share a common approach to display information to users of the system. It provides a common user interface look and feel across all applications that leverage the common process run time environment. So you will experience the same user experience regardless if you are a user of Configuration Management, Change Management, or any other process management product.
- ▶ A work management platform that allows you to configure and run end-to-end process definitions (predefined or your own) for any kind of Service Management process. It provides technologies like Job Plans, workflows, activities, and activity groups in order to automate processes and guide different personas through a process.
- ▶ A way to send a notification to a person or group. This can either be an e-mail or a message in the start center of specific user roles.
- ▶ A reporting engine based on BIRT technology. The reporting engine is by default installed on every application server in the WebSphere Cell of application servers if you have a multi-server environment, but please bear in mind that the reporting run time is running in its own Java Virtual Machine. Reports are administered from the Web user interface. Report definitions are stored in the process layer database. Report security is based on the security model provided by the base services layer.
- ▶ Integrated tooling to customize data, application layouts (user interfaces), or process definitions. The base services include integrated tooling to add, update, or modify new objects to the process layer database. From a CCMDB V7.1 perspective, this, provides the ability to manually add Authorized CI object classes or attributes without having synchronized data from the Discovered CI space. You can also use the *Database Configuration Application* in the Base Services to create new database objects that are federated from different data sources. In addition, there is an Application Designer application that allows you to set users of the system to have different views into the system than provided by the default installation. Finally, there are capabilities provided to create or modify definitions of Job Plans or Workflows in order to model your Service Management processes.

- ▶ Escalation services that are scheduled against the process layer database in order to verify defined conditions and behave accordingly in the case of deviances or exceptional behavior of the system or processes. For example, a Process Manager Product can set up an escalation to verify whether an activity has occurred by a certain time, and notify someone if it has not.
- ▶ Integration services to external systems and data sources through the Maximo® Enterprise Adapter (MEA) technology. Making use of the integration services allows you to batch-load or synchronize data with an external system in real time. For example, this provides a way to load data into the authorized CI space from an external system that you regard as a trusted source of information. Alternatively, you can use this technology to provide data from the CCMDB to an external system to work with data you have captured in the Actual or Authorized CI space. In Figure 3-1 on page 30, the integration services are part of the data access services, because what you exchange with external systems is data, either in or out of the CCMDB system. The first step in defining what to share with external systems is to define which data (Integration Objects, which refer to Maximo Objects, which refer to data in the database) you want to share.
- ▶ A data access layer that provides the services for the applications to access the process layer database. Data is stored in database tables. To access data from an application like Change or Configuration Management, these database tables are encapsulated with business logic inside Java objects, which we refer to as a Maximo Business Object (MBO). The data access layer in the CCMDB V7.1 solution provides access to the Authorized and Actual CI representations. It does not directly connect to the Discovered CI space that the TADDM server is maintaining.

From an implementation point of view, the base services component (with the exception of the reporting engine) is a J2EE™ application that runs in the WebSphere environment. It is deployed as part of the core CCMDB installation step. Refer to Chapter 8, “CCMDB installation planning” on page 163 and Chapter 9, “Installation” on page 187.

In summary, the base services component is the core run time process environment that provides all the facilities that process manager products like Change and Configuration Management leverage.

3.5 CCMDB Process Manager Products

A Process Manager Product (PMP) is a system for managing the execution of a process.

You can think of a process request as a ticket with a written note on it that is forwarded to various people (or entities) to perform various actions that, in the end, result in the objective of the process.

Change and Configuration Manager PMPs are delivered as part of the IBM CCMDB V7.1. Besides these two PMPs delivered with the CCMDB V7.1, another PMP, referred to as the Common PMP, is part of the CCMDB V7.1. The Common PMP does not provide specific functionality in terms of reflecting a Service Management process, but rather provides some common functionality that is used by all other implementations of PMP products. Please refer to “Common PMP” on page 54 for more a more detailed description of the Common PMP implementation.

IBM provides Process Manager Products aligned to the ITIL model for the various Service Management disciplines. IBM has implemented the theoretical ITIL process descriptions in its software in order to allow users of the system to be automatically guided through a process.

Note: Although the Configuration and Change Management Process Manager Products are the ones that we are focus on in this book because they are part of the CCMDB V7.1 package, there are additional PMPs available, such as Release Management. More PMPs are planned and are in the development phase.

Common PMP

The Common Process Manager Product or Common PMP is a specialized process manager implementation that gets installed as part of the CCMDB install. You do not have to install it separately through the Process Solution Installer; it is automatically installed for you when you install the CCMDB package.

The purpose of the Common PMP is to provide enablement for other process management products that implement a real Service Management process flow. Both the Configuration and Change Management PMPs build on top of the applications and functions that it provides.

If compared to the base services component, the Common PMP provides application functionality rather than technology services.

The most prevalent application that all other PMPs leverage from the Common PMP is the process request application. A process request forms the general interaction model into and between process manager products. This is because all PMPs are supposed to answer specific questions that are focused on organizational productivity.

An example of a process request is a *Request for Change (RFC)*, which is a request in the Change Management PMP. The intention is to ask the Change Management application to create a change object. The way you do this is to create a request; in the case of Change Management, you create a process request that is classified as a request for change with even more specific classification details on the nature of the change that you are planning.

Another example of a process request is a request to update data inside the CCMDB data layer, the Authorized CI representation specifically. In this case, we classify the request as a *Configuration Update Request*. This allows a controlled process to be initiated to update the CCMDB database and, through this procedure, make sure the Authorized CI space really contains a trustworthy information base.

Process requests can be entered into the system by a user of the system, but they can also be used between PMPs. An example of PMP to PMP integration using a process request is a Configuration Update Request, which is a step in a Change Management process flow towards the Configuration Management PMP in order to update the CCMDB with data that has changed through the Change Implementation itself. Each Process Manager Product provides a set of Web service interfaces. These interfaces provide the ability to create, update, query, or cancel a request from a different process manager. Another example of a PMP to PMP process request, although outside the scope of the core CCMDB, is a request from Change Management to Release Management to add a Change object into a Release object.

In order to relate these concepts to how it looks in the product interface, we have included some figures depicting the process request application and some examples of specific process requests. We do not explain in this chapter how to navigate the user interface in detail in order to get to the Process Request application.

Figure 3-15 shows an empty process request panel that can be launched from the *Change Management* module or the *IT Infrastructure* management module from within the GoTo link in the CCMDB Web user interface.

Process Requests - Microsoft Internet Explorer

Address: http://kenmore.itsc.austin.ibm.com/maximo/ui/?event=loadapp&value=spmcomsr

Links: IBM Business Transformation Homepage, IBM Global Print, IBM Standard Software Installer, IT Help Central, Join World Community Grid, IBM Internal Help Homepage

Process Requests

Find: [] Select Action: []

List Process Request Related Records Log

Process Request: 1003 Owner: [] Owner Group: [] Process State: DRAFT

User Information

Requestor: MAXADMIN

Name: []

Phone: []

E-mail: []

Process Request Details

Description: []

Priority: []

Process Manager Type: []

Site: []

Requested Completion: []

Request Classification Details

Classification: []

Class Description: []

Classification Attributes

Attribute	Description
No rows to display...	

New Row

Target CIs

Configuration Item	Target Description
No rows to display...	

Select: [] Clear All: [] New Row: []

Figure 3-15 Process Requests application

There are some fields or attributes already pre-populated, such as the Process Request number, and the Requester and Name fields. The Requester field is the name of the user who is entering a process request while the process request number is auto-generated by the system.

We only want to highlight the most important fields at this point in order to make the concept of a process request easier to understand. One of the fields that need to be specified is the *Process Manager Type* field. When you click the lens icon beside the field, it brings up a selection panel that lists all the process managers that have an implementation of a process request implementation (Figure 3-16 on page 57).

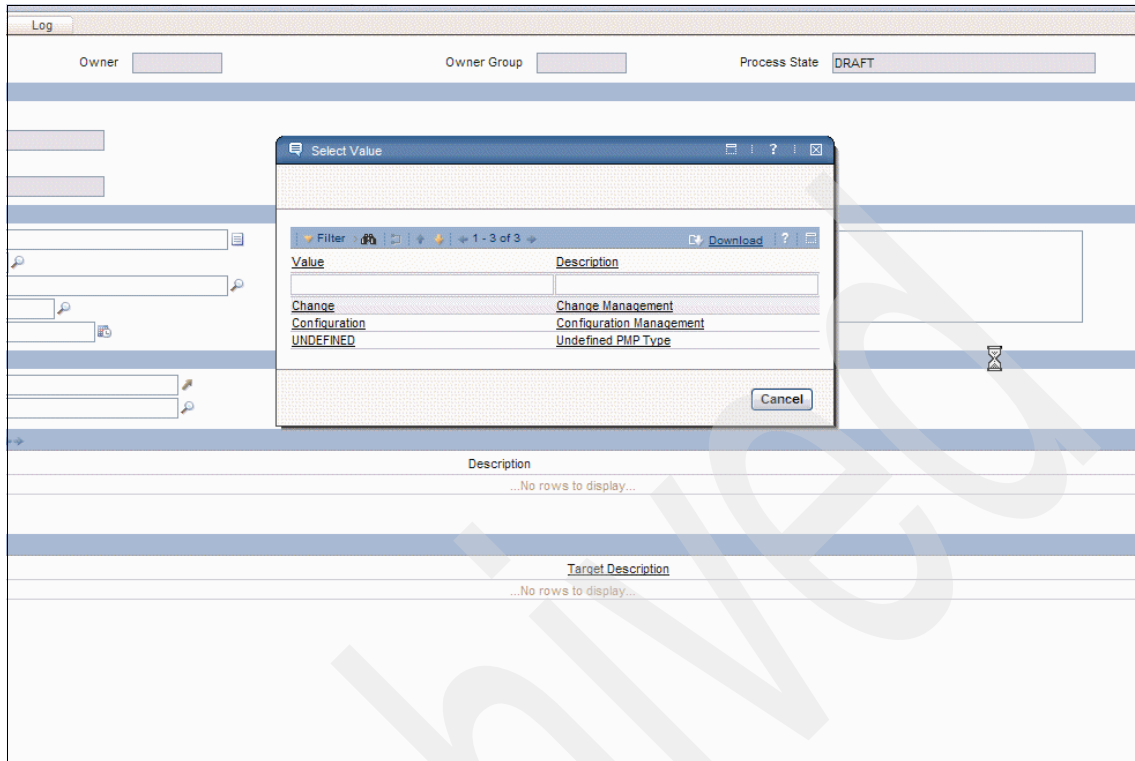


Figure 3-16 Process Manager Type selection

You can see that at this time we have a system with just Change and Configuration Management PMPs installed. We can select which PMP we want to send a process request to. We selected Change Management in this case.

Once you have defined which PMP you want to send the request to, the next thing you have to do is to classify the request with more detailed information. Classifications are another general concept inside the system that is used in many places to add more detailed information. When using the small arrow to the right of the *Classification* field, you bring up a window that allows you to do the classification.

Note: You can also move into the Classification application, do your selection, and move your selection back into the originating window.

Figure 3-17 shows the list of possible options to select from when classifying the request.

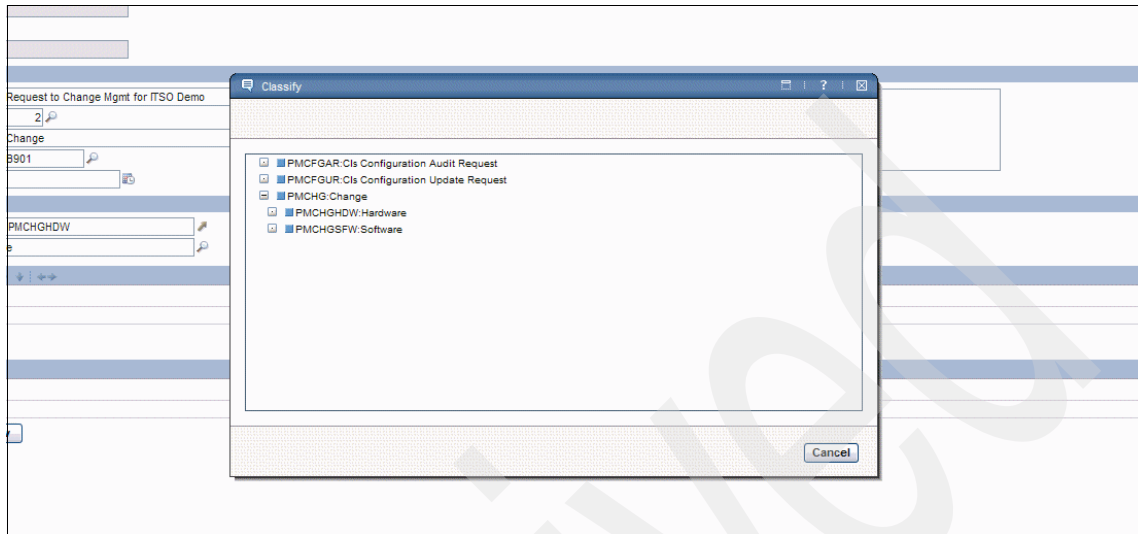


Figure 3-17 Process Request classification

You can see at this point that we have just the option to classify the request to Change Management either as a Hardware or Software Change. You can add your own additional classifications in the classifications application to align to your organizational needs.

You can also see that two further classifications to a process request in the Configuration Manager, Configuration Audit Request and Configuration Update Request, are available for selection from the classification window.

We selected **Software** as a further classification for the change that populates the overall process request window, as shown in Figure 3-18 on page 59.

The screenshot shows a web browser window with the address bar displaying a URL from the IBM Maximo system. The page title is "Process Requests". Below the title bar, there are tabs for "List", "Process Request", "Related Records", and "Log". The "Process Request" tab is active, showing details for request 1004. The form includes sections for "User Information" (Requestor: MAXADMIN, Name: MAXADMIN, Phone, E-mail), "Process Request Details" (Description: Request to Change Mgmt for TSO Demo, Priority: 2, Process Manager Type: Change, Site: B901, Requested Completion), "Request Classification Details" (Classification: PMCHG \ PMCHGSFW, Class Description: Software), "Classification Attributes" (Attribute, Description), "Target CIs" (Configuration Item, Target Description), and "Select" buttons.

Figure 3-18 Completed Process Request

If a classification has defined its own attributes, those would automatically show up in the *Classification Attributes* section. You can see that you can also attach CI information in the *Target CIs* section of the window. In the case of an RFC, you would do this in case you know already the CI or group of CIs that are targets of the change that you request through the process request.

Figure 3-19 shows a process request to the Configuration Management PMP, classified as an Update CI Request.

The screenshot displays the IBM Tivoli Configuration Management PMP web interface. The browser address bar shows the URL: `http://kenmore.itsc.austin.ibm.com/maximo/ui/?event=loadapp&value=pmcomsr`. The page title is "Process Requests". The interface includes a search bar, a "Select Action" dropdown menu, and a toolbar with various icons. The main form is titled "Process Request" and contains the following fields:

- Process Request:** 1005
- Owner:** (empty)
- Owner Group:** (empty)
- Process State:** DRAFT

The "Requestor" section includes:

- Requestor:** MAXADMIN
- Name:** MAXADMIN
- Phone:** (empty)
- E-mail:** (empty)

The "Request Details" section includes:

- Description:** Request to Configuration Mgmt for ITSO Demo
- Priority:** (empty)
- Process Manager Type:** Configuration
- Site:** B901
- Requested Completion:** (empty)

The "Classification Details" section includes:

- Classification:** PMCFGUR
- Class Description:** CIs Configuration Update Request

The "Attributes" section includes a table with the following columns: Attribute and Description.

Attribute	Description
HGRECD	All Changes Recorded
OWNER	Notification Person ID
ASON	Reason to Update
CONOTE	Request Note
ICID	RFC / Change WO ID

The "Action Item" section includes a table with the following columns: Action Item and Target Description.

Action Item	Target Description
...No rows to display...	

At the bottom of the page, there are buttons for "Clear All" and "New Row".

Figure 3-19 Process Request into Configuration Management

After you have specified and classified your process request, you have to submit the request. In order to do this, you have to select an action from the Action drop-down menu, as shown in Figure 3-20 on page 61.

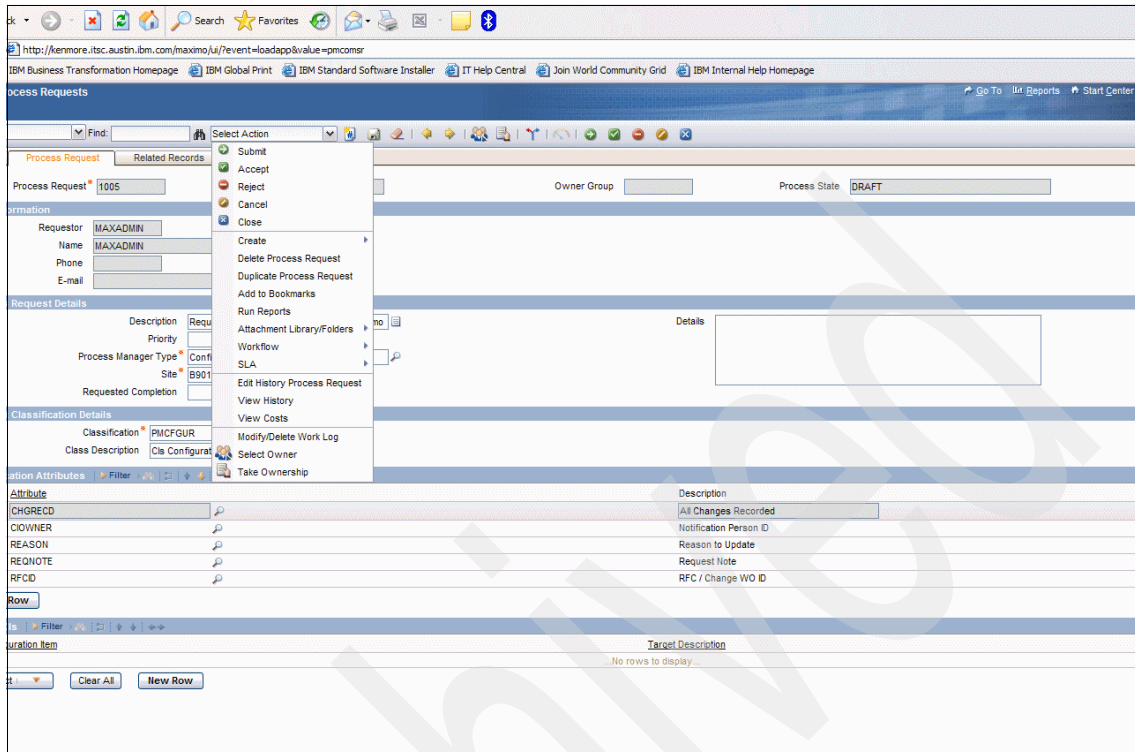


Figure 3-20 Submission of the Process Request

The first action you take is submit the request by using the Submit option of the drop-down menu. This will initiate a request in the system that gets assigned to someone with a specific role to approve the request before a Change Management object gets created.

We have explained the Common PMP in some detail because it is important to understand the basics for other PMPs like Change and Configuration Management.

There are further applications that are provided within the Common PMP that we do not explain in detail, such as the foundation to interconnect a PMP to an Operational Management Product (OMP) like Tivoli Provisioning Manager or Tivoli Configuration Manager in order to initiate some action from within a step in a process into the OMP. In Figure 3-20, a PMP can initiate a software distribution action inside the Tivoli Provisioning Manager as the OMP. The Common PMP provides the technology for the integration between the PMP and OMP layer. It does not provide concrete implementations of integrations; instead, it provides the foundation to do this task.

For example, the Integration Modules, the Logical Management Operation, or the Launch in Context applications inside the Integration Module Menu are provided by the Common PMP (Figure 3-21).

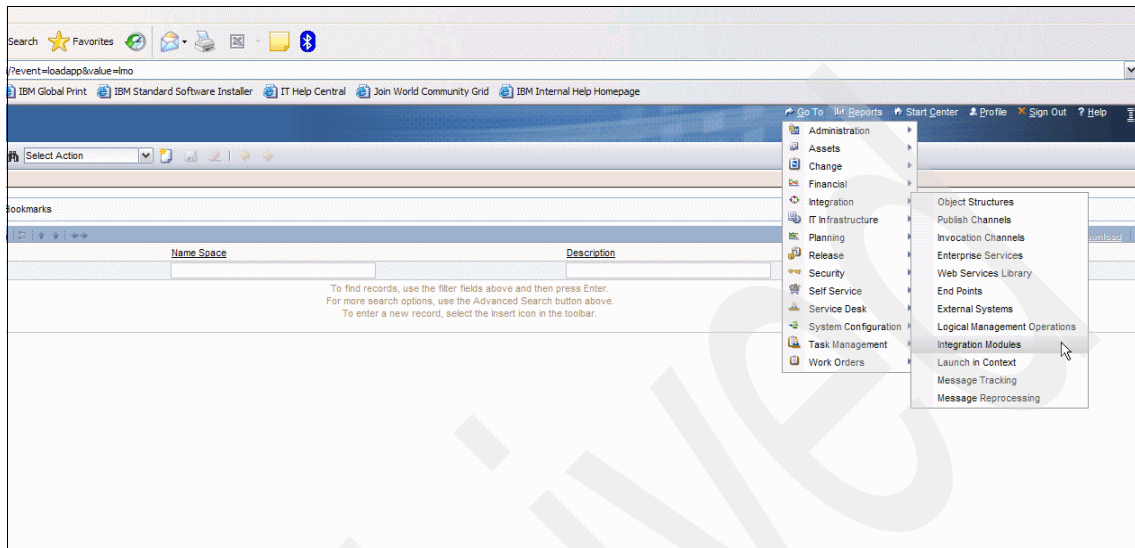


Figure 3-21 Integration Module applications

There are many organizations that have implemented IBM WebSphere Process Server to host implementations for business oriented workflows. They may have a desire to combine their business and IT Service Management processes. To support such scenarios and use cases, all PMPs ship support for BPEL flows running on the IBM WebSphere Process Server. In this case, the business process flow hosted on the IBM WebSphere Process Server is using specific interfaces provided by the PMPs. The interfaces are provided at the Activity level of a process flow. BPEL support is optional.

Configuration Management PMP

The Configuration Process Manager Product is the part of the IBM CCMDB V7.1 solution that enables IT organizations to identify, control, account for, and audit all Configuration Items in the IT Infrastructure through the Change and Configuration Management database. This brings process control to the data inside the data layer of the CCMDB. Otherwise, it would just be another configuration database for persisting configuration data. It stores a logical model of the IT infrastructure containing configuration items, their attributes, and relationships. The data is represented in three spaces: the discovered, the authorized, and the actual CI representations. The Configuration Management PMP primarily operates on the Authorized and the Actual CI spaces while also

allowing you to launch from the Actual to the Discovered CI representation that is maintained in the TADDM database.

The Configuration Management PMP provides several applications to work with and maintain CIs. At install time, they are installed as part of the CCMDB package.

Most of the applications that the Configuration Management PMP provides are grouped in the *IT Infrastructure* module that you can launch from the GoTo Link in the CCMDB Web User Interface, as shown in Figure 3-22.

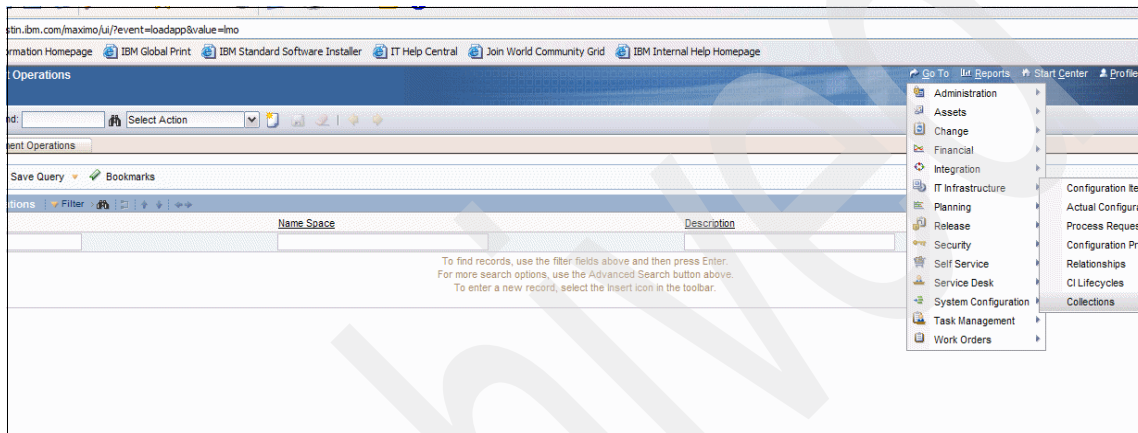


Figure 3-22 IT Infrastructure Module Applications overview

There are applications for maintaining Configuration Items (Authorized CIs), maintaining the Actual Configuration Items, generating process requests into the Configuration Management (Audit and Update CI requests; please refer to “Common PMP” on page 54 for more details on process requests), and to maintain or create relationships or CI Lifecycles.

Defining life cycles defines states that CIs of a specific type can be in, as well as which states are regarded as protected states, in which case a change to a CI would require a Change Request (RFC).

The Configuration Management PMP provides default life cycles according to ITIL, but you can provide your own life cycle definitions within the CI Lifecycle application, as depicted in Figure 3-23.

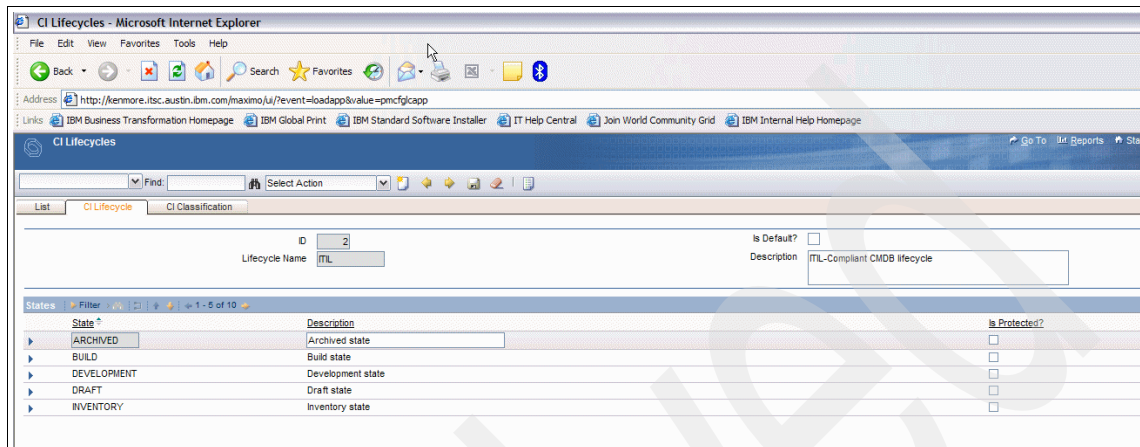


Figure 3-23 CI Lifecycle application

The Configuration Management PMP provides two process implementations in V7.1, *Verify* and *Audit CI* as well as *Control CI* (also known as *Update CI*). Both of them require a process request to be submitted and approved before they are actually initiated.

You can find the definitions of those processes as predefined templates in the Job Plan application, as shown in Figure 3-24.

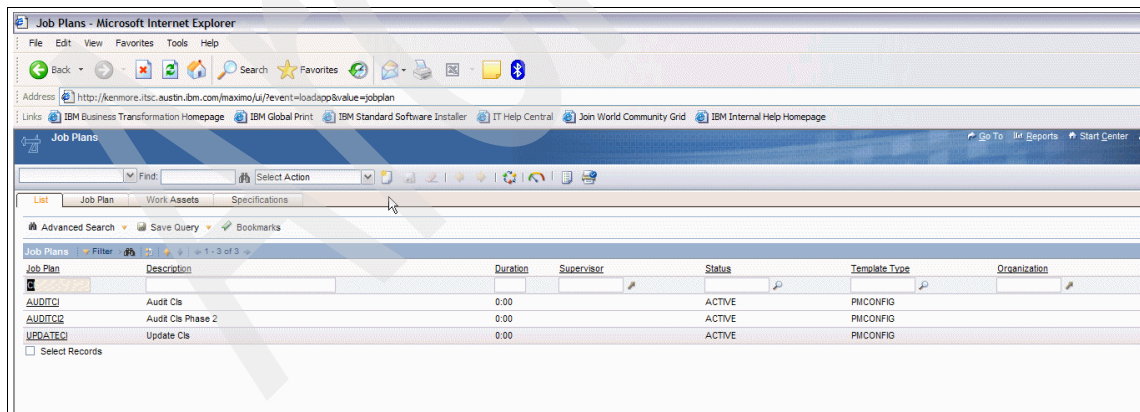


Figure 3-24 Job Plans application

The definitions above define steps in order to move users with the appropriate role through formal process steps for the Audit and Update requests.

CI Auditing (and remediation) provides a formal request and process for CI audits. An audit is defined as a comparison between the Actual and the Authorized CI representations in order to identify any variances between what the environment should look like (Authorized) versus what it actually looks like (Actual). Remediation as a possible step inside the Audit process (Audit Phase 2) supports the creation of an RFC to eliminate the variance through a controlled process or directly update the Authorized CI space with what is defined in the Actual CI space.

Update CI or CI control provides a formal request and process for making changes to the Authorized CI space. This provides a tight control over CI changes. Change Management, for example, can generate a process request to Configuration Management to update the CCMDB database with the changes made within the change implementation.

As you can see, the Audit process template is divided up into two phases, the first phase being more related to planning and defining the audit, while the second phase is more related to the analysis of the comparison results of the audit.

If you link to the definitions of the UPDATECI job plan template, you will see that this job plan has different phases, steps, or activities, as shown in Figure 3-25.

The screenshot shows the IBM Tivoli CCMDB Job Plans interface. The browser address bar indicates the URL: <http://kenmore.itsc.austin.ibm.com/maximo/ui/?event=loadapp&value=jobplan>. The interface has a navigation bar with tabs: List, Job Plan, Work Assets, and Specifications. The Job Plan tab is selected, showing details for the UPDATECI job plan. The details section includes fields for Status (ACTIVE), Template Type (Configuration PMP), Duration (0.00), Classification, Class Description, and Launch Entry Name. There are also fields for Default WO Class (PMCFGWG), WO Priority (0), Interruptible? (unchecked), Flow Controlled? (checked), Suspend Flow Control? (unchecked), Flow Action, and Flow Action Assist? (unchecked). A Responsibility section shows Supervisor, Crew, and Lead fields. Below the details is a table of Job Plan Tasks:

Sequence	Task	Description	Nested Job Plan	Duration
1	10	Review Update CI request and work plan		0.00
2	20	Determine requested CI changes		0.00
3	30	Make CI attribute changes if requested		0.00
4	40	Make CI relationship changes if requested		0.00

At the bottom, there is a section for Planned Labor with a table that has columns for Task, Craft, Skill Level, Labor, Quantity, Hours, and Rate. The table currently shows "No rows to display..."

Figure 3-25 Update CI Job Plan template

These tasks define the steps of the formal process to update a CI in the CCMDB database. You can modify them to fit your needs. There are additional items provided by the Configuration Manager PMP, for example, predefined roles implemented as security groups in the system (Configuration Manager, Configuration Auditor, and so on), predefined reports, and KPIs or Start Centers for the different types of Configuration Management users.

One of the predefined Key Performance Indicator definitions that you can use in the Start Center of a Configuration Manager or any other user of the system is the *Percentage of Update CI Processes completed* (Figure 3-26 on page 67).

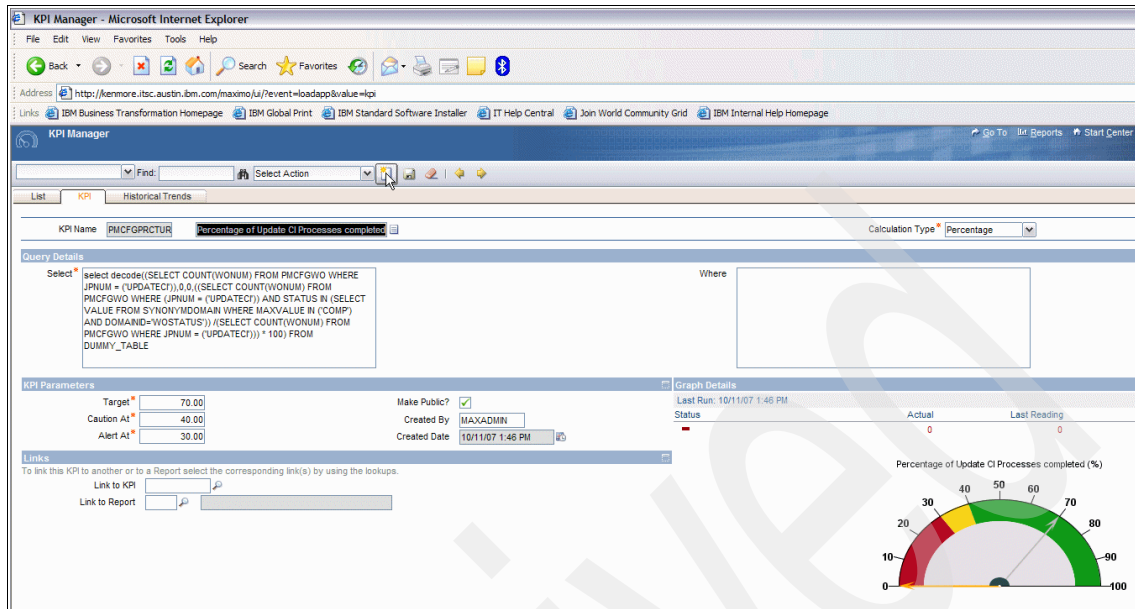


Figure 3-26 Key Performance Indicator for Configuration Management

For more details on the Configuration Management PMP, please refer to *IBM Tivoli CCMDB Implementation Recommendations*, SG24-7567.

Change Management PMP

The Change Management PMP provides the ability to control and implement changes according to your organization's change process. The Change Management process begins with an RFC (process request classified as a change request) that needs to be accepted and approved by someone having the appropriate role. Therefore, a Request for Change has to be created in order to start the Change process.

The CCMDB change management process enables you to manage changes from the initial request (RFC), through the assessment, the implementation, and up to the final review, often referred to as the Post Implementation Review (PIR). The Change Management PMP provides predefined process templates aligned with ITIL best practices that you can modify according to your organizational needs.

Change Management is integrated with Configuration Management in different ways. You can choose CIs as targets of a change and use the relationships to understand the potential impact of the change on other CIs in the infrastructure. Change Management does interact with Configuration Management to demand

that an RFC be created before certain CIs can be updated in the CCMDB database. This happens when Change Management generates a process request to Configuration Management for updating a CI (an Update CI process request). For more details on process requests, please refer to “Common PMP” on page 54.

Change Management also interacts with Release Management to add Changes to a specific or any release. Please bear in mind that the Release Management PMP is a separate package of the CCMDB.

The Change Management PMP provides different applications in order to administer and use the Change process. You can find these applications in the Change Module of the CCMDB Web User Interface, as shown in Figure 3-27.

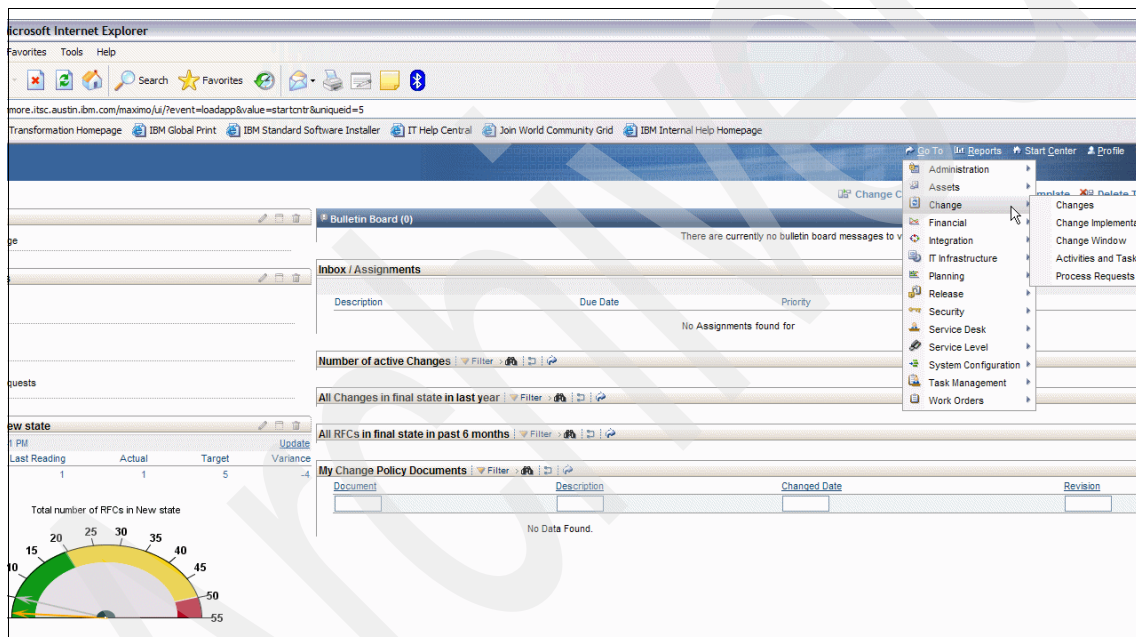


Figure 3-27 Link to Change Management application

As you can see in Figure 3-27, there are different applications belonging to the Change Management PMP: the *Change* Application itself to handle changes that have been requested by an RFC, and the *Change Implementation Schedule* application, which provides a view in change management that shows the start and end dates for changes and included implementation tasks for selected configuration items. Different views are provided for this Change Calendar-like application.

The next application is the *Change Window* application that allows you to define maintenance periods of when CIs are allowed to have maintenance applied or be allowed to be changed. Defining change windows for CIs or groups of CIs defines the baseline for detecting change window conflicts. A conflict could occur, for example, when you define a change where you may identify implementation tasks for these changes whose schedules do not conform to the defined or allowed change windows for these CIs.

The Process Request application allows you to generate a process request to Change Management, that is, when correctly classified, an RFC.

One of the most important functions inside the Change Management PMP is the *impact analysis* function. The reason for doing impact analysis as part of the Change Management process is to identify and document all the consequences of implementing a change request. Once all the consequences are documented, the subsequent steps of the process will use this information. For example, approvals and implementation scheduling will depend on accurate impact analysis. The goals of the impact assessment is to record the results of the impact analysis done by some subject matter experts, facilitate the creation of implementation tasks to do work identified during the analysis, and assist the Change Analyst to identify CIs which will be impacted by this change.

Important: Impact analysis is not simply a matter of identifying the CIs related to the targets of a change. We can find those relationships by analyzing the data in the CCMDB database. But being related to the CI defined as a target in the change does not mean a CI is impacted by default. For example, we discover that the implementation of a Change will require the installation of a software update to a server. The CCMDB database shows relationships between this server and other CIs that depend on it. Are these related CIs impacted by this change? If the update can be installed on the server without any degradation of service, then they are not. Whether the installation of the update requires that the server being updated be offline depends on the update design. This information will need to be investigated during impact analysis before the Change Analyst can identify the impact of the change.

As part of the change application, the Change Management PMP provides a specific tab that will be used by one or more Change Analysts to both document the results of their assessments and identify CIs impacted by this change, as shown in Figure 3-28.

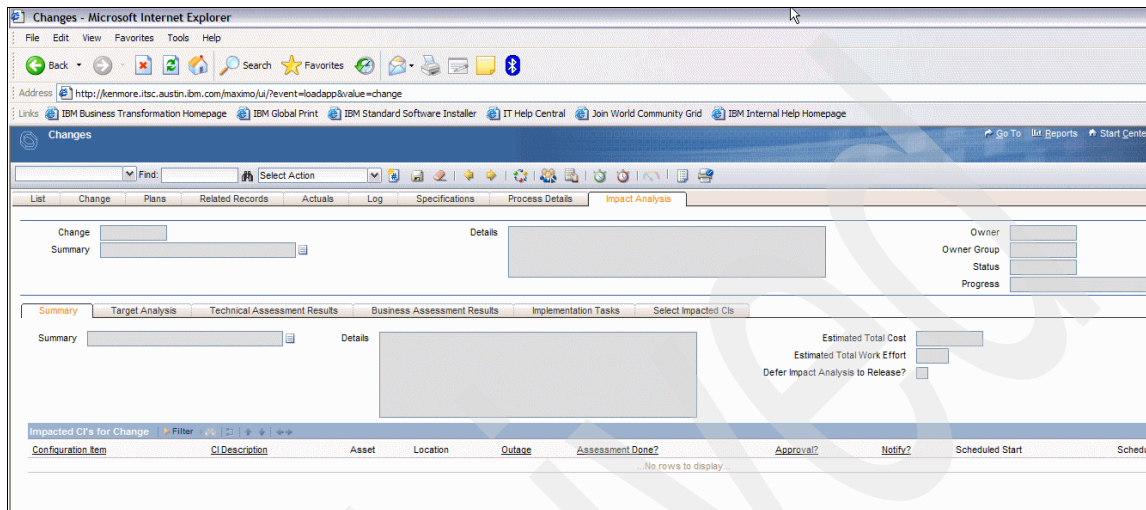


Figure 3-28 Impact Analysis tab of the Change Management application

In Figure 3-28, you can see several sub-tabs to structure the impact assessment work.

The Change Management PMP provides additional assets as part of the package, for example, predefined *Job Plans* for exemplary ITIL-aligned process flow definitions, *Roles* like Change Manager, Change Administrator, Change Owner, Change Analyst, Change Approver, Change Implementer, or Change Requester. In addition, predefined Start Centers for the different roles are provided; these Start Centers leverage predefined Key Performance Indicator views. And, finally, many out-of-the box reports are provided for Change Management.

One of the elementary facilities of the Change Management PMP is to provide predefined end-to-end process flows. These are modelled as Job Plans in the system. The Change Management PMP provides three predefined end-to-end process flows, as shown in Figure 3-29 on page 71.

Job Plan	Description	Duration	Supervisor	Status	Template Type	Organization
CHG-F1	Assessment	0.00		ACTIVE	ACTIVITY	
CHG-F2	Approval	0.00		ACTIVE	ACTIVITY	
CHG-F3	Schedule	0.00		ACTIVE	ACTIVITY	
CHG-F4	Implement the Change	0.00		ACTIVE	ACTIVITY	
CHG-F5	Post Implementation Review	0.00		ACTIVE	ACTIVITY	
CHG-F6	Business Assessment	0.00		ACTIVE	ACTIVITY	
CHG-F7	Technical Assessment	0.00		ACTIVE	ACTIVITY	
CHG-F8	J2EE Implementation	0.00		ACTIVE	ACTIVITY	
CHG-P-F1	Standard Change JobPlan	0.00		ACTIVE	PROCESS	
CHG-P-F2	Pre-Approved JobPlan	0.00		ACTIVE	PROCESS	
CHG-P-F3	J2EE Application JobPlan	0.00		ACTIVE	PROCESS	

Figure 3-29 Predefined Change Management Job Plan templates

In Figure 3-29, you see more than three predefined Job Plans for Change Management. This is because Job Plans can be nested. You can recognize the top level Job Plans that model the end-to-end process flows at the bottom of the window, as they have a template type of PROCESS rather than ACTIVITY. The Job Plans with a template type of ACTIVITY are used in the top level job plans. The three examples provided by the Change Management PMP for end-to-end process flows are:

1. *Standard Change Job Plan* to represent a standard change
2. *Pre-Approved Job Plan* to represent a pre-approved change
3. *J2EE Application Job Plan* to represent a specific example of steps necessary to change a J2EE application

Within each of the Job Plan definitions, the steps and tasks to be taken by different roles of the organization (who are involved in the Change implementation) are defined. Looking into the template for the Standard Job Plan, you recognize four phases that have been defined. The steps are Assessment, Approval, Schedule, and Implement the change. These phases are aligned with ITIL. You can see that those phases are modelled as (nested) Job Plans themselves. Inside these nested Job Plans, you would find specific tasks related to those phases.

For more details on the Change Management PMP, please refer to *IBM Tivoli CCMDB Implementation Recommendations*, SG24-7567.

3.6 Integration Composer and Integration Adapter for TADDM

IBM Tivoli Integration Composer (ITIC), formerly known as Fusion, is a component or tool to primarily import data from an external system into the actual data representations of the process layer database. The actual data representations are separated between representations of asset data structures and CI data structures. In the case of the CCMDB V7.1 environment, we focus on loading CI data rather than asset data into the system. Please bear in mind that CI data can be linked to asset data through appropriate relationships in the process layer database.

Inside the CCMDB V7.1 environment, the main use case for the Integration Composer is to transfer data from the TADDM database that holds the discovered data to the Actual CI space inside the process layer database. The transfer of data from the TADDM discovery environment to the CCMDB process layer environment is actually a two-step process.

First, the model or CI type data (based on the Common Data Model) are transferred to the process environment in order to reflect the appropriate CI class structures inside the process layer database. These class structures and attribute definitions are used throughout the system, foremost in the Authorized and Actual CI applications. Think of the model as the backbone for the real or instance data. If you do not have a model that defines how to represent a Linux Computer system or Business Service, the instance data cannot be transferred. You need a schema in which you input the real data. Every time a change to the model is being done on the TADDM side of the CCMDB environment (like adding an additional attribute or adding new classes), you need to synchronize these changes into the process layer database.

Second, the instance data is brought over. These is the real data that you would see on the TADDM console. This is your Linux server *ABC* or the Business service *ITSO Application*.

The IBM Tivoli Integration Composer is actually a tool for data migration with a separate installation program. It is not part of the middleware or CCMDB installation. There is a separate install program for the Integration Composer that runs its own Java Virtual Machine; it does not run in the WebSphere environment of the CCMDB. You can even install the ITIC on a different system from TADDM and the process environment of the CCMDB. It just needs appropriately defined connections to the TADDM environment and the system hosting the process layer database of the CCMDB.

The ITIC data migration component needs instructions for the source and target databases that it needs to connect to, it requires information about the schema of the source and target data sources, and it usually requires some mapping instructions on how it needs to map data from the source into the target representation of data inside the process layer database. A mapping is a set of expressions that tells ITIC how to transform data on its way from the source to the target.

These instructions are given through specialized adapters that comprise schema information for the source and target data sources as well as appropriate mapping information.

CCMDB V7.1 provides an adapter called the *Integration Adapter for Tivoli Application and Dependency Manager*. This adapter provides the instructions for ITIC to connect to the TADDM environment on one side and to the process environment on the other side in order to transfer data from the discovered CI space to the Actual CI space. Different adapter files are provided for exchanging the data model or CI type information as well as the instance data.

For a more detailed description of how to implement and configure the ITIC and the Adapter for TADDM, please refer to Chapter 10, “TADDM and Process Layer integration” on page 231.

CCMDB Data Layer

This chapter provides an overview of the data layer of the CCMDB architecture. Additional details are provided in *IBM Tivoli CCMDB Implementation Recommendations*, SG24-7567.

The CCMDB Data Layer contains three data spaces that hold CIs and process artifacts and relationships between these objects to provide everything from a dependency mapping of the discovered environment to a specification of authorized CIs that define the specific aspects and characteristics of CIs you wish to tightly control and manage, and relationships with process artifacts, such as Request for Changes (RFCs).

The CCMDB supports the Tivoli Common Data Model (CDM) across all three data spaces. The CDM is a logical information model that is used to support the sharing of consistent data definitions and the exchange of data between Tivoli management products concerning managed resources and components of a customer's business environment.

This chapter gives a conceptual overview. Later chapters will provide implementation details of the data layer, including several use cases like Federation, Promotion, or Launch in Context.

Figure 4-1 reflects the three CI data spaces of the CCMDB V7.1 solution, its interoperability, and its relationship to other data structures, such as Process Artifacts.

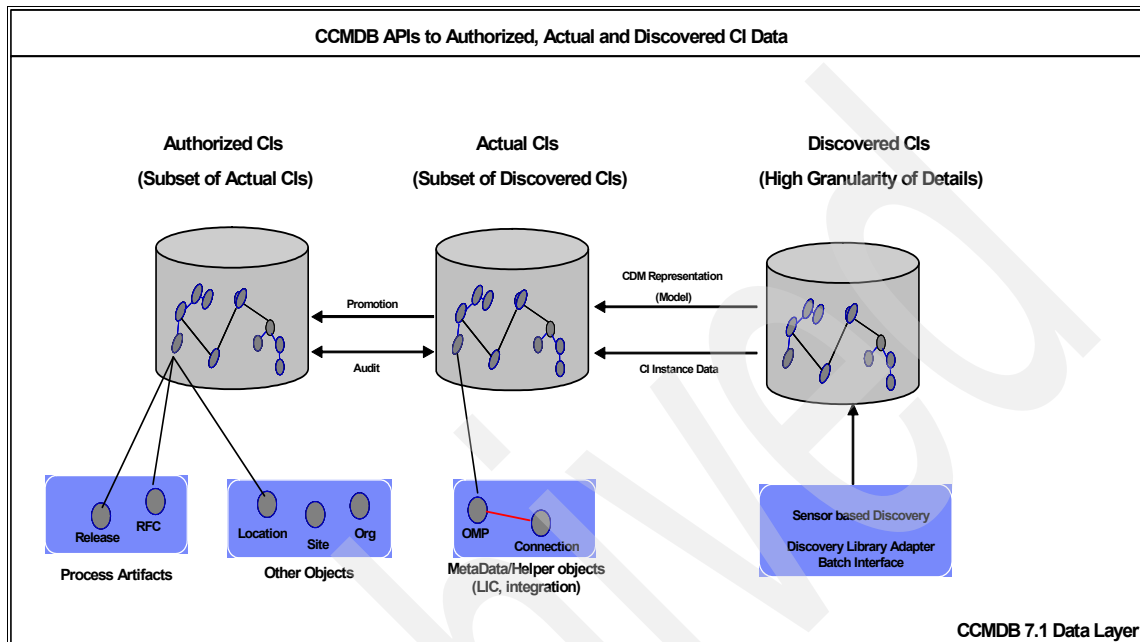


Figure 4-1 CCMDB V7.1 Data Layer

In Figure 4-1, the high granularity of details in reference to the discovered CIs implies that the data discovered may be more detailed than needed for Actual or Authorized CIs. When the data is imported as an Actual CI, the unnecessary details may be filtered out.

4.1 Discovered configuration items

The Discovered CIs data space of the CCMDB contains information discovered in the heterogeneous IT environment. This includes CIs and relationships discovered using sensor discovery and those loaded through Discovery Library Adapters (DLAs). Operational Management Products (OMPs) that are managing CIs in the environment and persisting data in their own database can synchronize selected data through the DLA batch interface, also referred to as the bulkloader, into the Discovered CI data space.

The TADDM component of the CCMDB provides the discovery engine for the solution as well as a set of discovery services, such as naming and reconciliation, unique resource identification and attribute prioritization, and change history on discovered CIs and relationships. These discovery capabilities provide an accurate dependency mapping between CIs, which includes many different types of relationships, such as logical, physical, and application topologies.

For CIs and relationships that are not discoverable through the discovery sensor capabilities and are not available through the IBM-provided DLAs, the discovery toolkit can be used to build custom DLAs to bring data from other sources into your CCMDB discovered CI data space.

A discovered CI and its relationships can be quite deep and complex, as indicated in Figure 4-2 on page 78. A specific computer system called EmailServer1 is shown, where the computer system CI itself can contain a significant number of objects and attributes that make up the computer system (for example, physical components and software components on the computer system), while these components can have relationships to other CIs. The computer system, for example, has relationships with logical CIs such as business services that it supports, and each of these would contain relationships with other CIs.

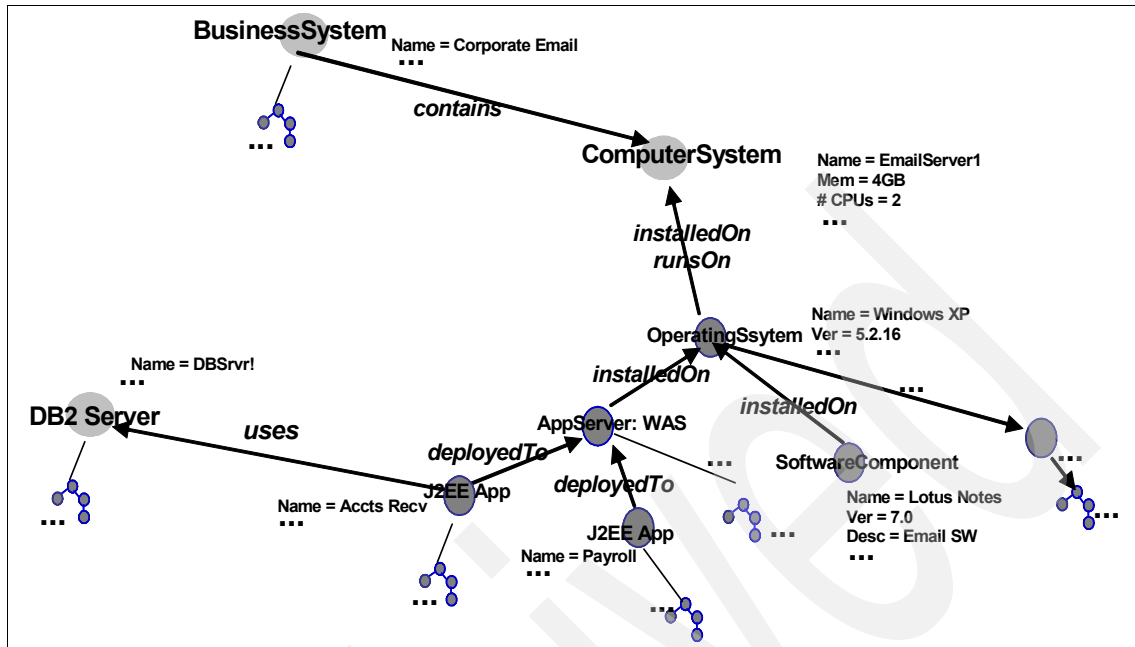


Figure 4-2 Interconnected graph of CI dependencies

The interconnected graph of CIs and relationships can get deep very quickly in the discovered CI space of the CCMDB, so that the CCMDB can easily contain a very large amount of data. While this data can be very useful for understanding significant detail about the IT environment, it is typically considerably more information than is needed or desired when you are interested in managing a specific set of CIs through applications and processes, such as Configuration Management, Change Management, or Release Management. Thus, there is a need to utilize a subset of the data in the Discovered CI space while still allowing the ability to refer back to the entire set of interconnected dependencies and attributes.

4.2 Actual configuration items

A filter can be applied that results in a subset of the CIs and relationships of the Discovered CI data space being copied over into the Actual CI data space. This allows the system to deal with a reasonable subsets of the entire set of discovered data, but still contains everything needed to perform all of the Process Management and Service Management capabilities that are desired, such as CI auditing as part of Configuration Management or Change Management.

For example, the filter could be set such that the Actual CI space only contains computer systems that are production servers, if it is determined that initially only these systems would come under Process Management. The server systems may be a very low percentage of the data that is in the Discovered CI space, so keeping the amount of data in the Actual CI space to just what is needed will indeed improve performance; however, the full set of discovered CI data and relationships is still accessible by launching the Discovered CI data space.

Actual CI filter

A configuration option allows the administrator to specify which CI types to bring into the Actual CI space from the Discovered CI space. It is best to set the filter for just what is necessary to meet the needs of the various teams that are performing process management and service management on the CI data. The filter can be expanded or changed over time to bring in more CI data as needed to support new requirements or processes. Refer to Chapter 10, “TADDM and Process Layer integration” on page 231 for more details.

Note that when you specify bringing in a CI into the Actual CI space that is a top-level CI, it will bring over the entire child tree beneath it. For example, a ComputerSystem is a top-level CI, so when each computer system CI is copied over into the Actual CI space, it will bring over all other CI objects that are related to it in a "downward" notion.

This Actual CI filter is provided by the IBM Tivoli Integration Composer (ITIC) adapter to determine which CI instances are brought over into the Actual CI data space. For each CI Type specified, all instances of that CI Type in the Discovered CI space are copied into the Actual CI space.

If the ComputerSystem CI Type was specified in the filter to bring computer systems into the Actual CI space, Figure 4-3 shows an example of what CIs and relationships would be copied over from the Discovered CI space.

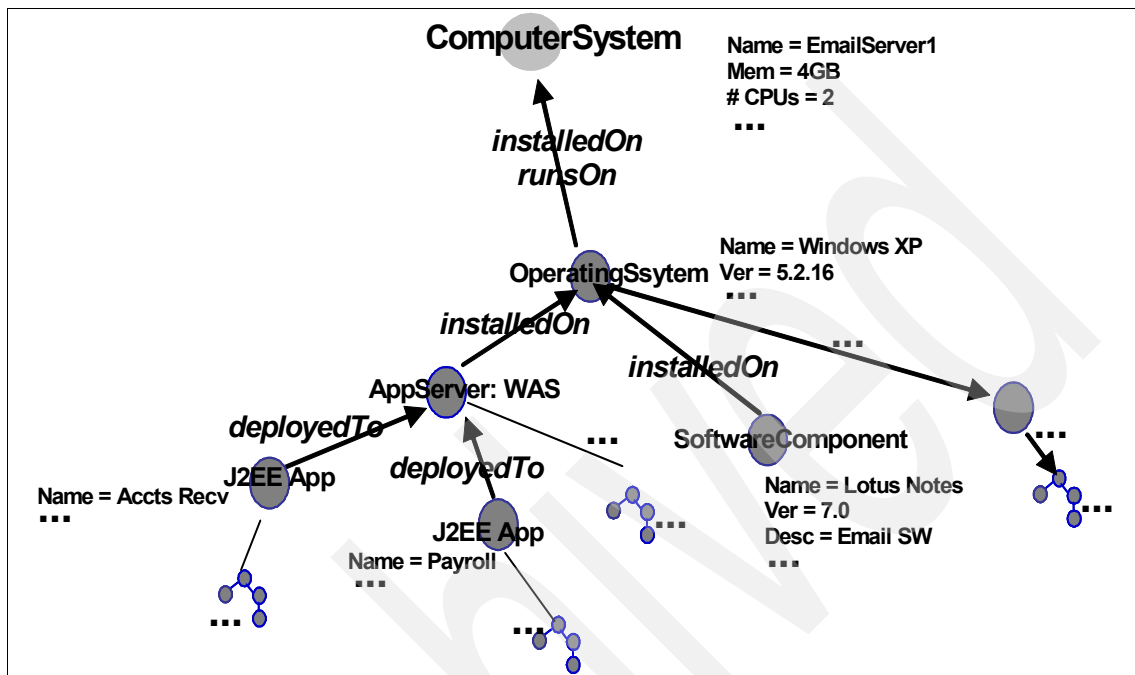


Figure 4-3 Filtered CI data in Actual CI data space

Note that the business service called Corporate Email and the database server called DBSrvr1 (and their relationships to EmailServer1) shown in Figure 4-2 on page 78 are not copied over in this example, because those CI Types were not specified as part of the Actual CI filter. If you want to bring over the relationships of the Computer System CI to the Business Service CI, you have to specify that all instances of the CI Type Business Service have to be copied as well.

Because the discovered CIs (particularly top-level CIs) can be quite deep and some can contain a significant number of objects and attributes, it is recommended that you set the Actual CI filter depth to an appropriate level for how deep you will be using the data within the ISM processes and applications.

While the filter specifies the breadth regarding which CI Types you want to bring over, the depth specification allows you to control the depth of the relationship hierarchy that you want to bring over from the discovered into the Actual CI data space.

database. If not, the ITIC adapter data request will need to go back to the discovery domains to satisfy the request, which will result in an increased number of data requests and network traffic during the time which the ITIC adapter is running, as well as a longer total elapsed time for the overall operation to complete.

Actual configuration items are administered through the Actual Configuration Items application.

4.3 Authorized configuration items

Once you have specified which CIs should be copied over into the Actual CI space, the next step is to create Authorized CIs. The process of creating Authorized CI instances from Actual CI instances is called promotion.

Authorized CIs are subject to control and modification by the Change Management and Configuration Management processes in the CCMDDB and are the target object for many operations within the overall IBM Service Management solution.

An Authorized CI is typically a simple definition with a small number of relationships and attributes. Earlier it was mentioned that a computer system CI can be quite complex with a large number of attributes and a fair number of relationships with other objects; however, an Authorized CI for a computer system would contain only the attributes or values and relationships to objects that you care to bring under change control.

For example, to meet corporate standards for a production e-mail server, there are certain characteristics that you want to ensure are met at all times (for example, it has 4 GB of memory on it, it is running Windows XP (any version), and it has Lotus® Domino® Version 7.0 installed on it). You define these characteristics in the Authorized CI space. For the production e-mail computer system called EmailServer1, you build the setup shown in Figure 4-5 on page 83 as a set of Authorized CIs, attributes, and relationships in the Authorized CI space. This reflects your authorized state.

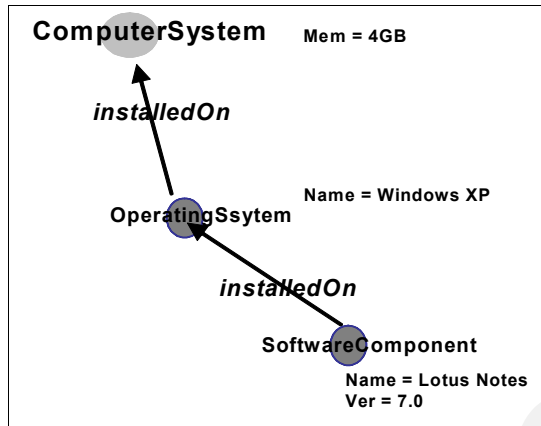


Figure 4-5 Sample of an Authorized CI view

Note that there are only three Authorized CI objects that were defined (a **ComputerSystem**, an **OperatingSystem**, and a **SoftwareComponent**), a single relationship between each of them, and a small number of attributes.

These are the only aspects of the **EmailServer1** CI that you wish to manage and tightly control. You can revise the Authorized CI definition at any time, such as adding new relationships to other Authorized CIs or adding new attributes. This Authorized CI can now be put under change control and used within the ISM processes, such as **Configuration** or **Change Management**.

Creating Authorized configuration items

Authorized CIs can be created either manually through the CCMDB Web user interface or through the CI object structure that provides an API layer for managing Authorized CIs. In other words, you can use the MEA integration technology of the CCMDB solution to synchronize CI data from external data sources. A typical use case would be to synchronize CI data that is not discoverable.

The primary method of creating an authorized CI is through a process called promotion. The promotion creates either a duplicate or a subset of an actual CI in the authorized CI data space.

It is important to note that there is at most one Authorized CI for an Actual CI, meaning an Authorized CI can be associated with an Actual CI or it can exist on its own without a link to any Actual CI.

If a user wants to create an Authorized CI manually, the user specifies the definition of an Authorized CI in the Configuration Items application (what are the attributes and values, relationships to other Authorized CIs, and the attributes

and values of those). They can specify the Actual CI to which the Authorized CI is linked, or as previously stated, Authorized CIs can be created within the CI application completely independent of an Actual CI (in which case there is no link specified to an Actual CI). For example, you may want to create Authorized CIs for computer systems that you have received through procurement, but which are not in the network yet, and therefore cannot be seen through discovery. You could create an Authorized CI for each computer system and drive them through the change process to get them deployed into your IT environment. Once these computer systems are found during discovery, you can go back and create the link between the Authorized CI and the Actual CI. There are also cases where Authorized CIs may be created that will never have an Actual CI, for example, in the case of logical CIs that will never come in through discovery, you only desire to have them in the Authorized CI space.

The other method of creating Authorized CIs through the user interface is through promotion. There are two ways to promote an actual CI to an authorized CI. The first way is to copy every detail (attribute, relationship, and so on) over to the authorized CI data space.

In order to filter out details of the CI data that you do not require within your Service Management processes, a second way of promotion is supported.

A set of Actual CIs are promoted through a CI Hierarchy (also sometimes called an Authorized CI Template). A CI Hierarchy definition is similar to an Authorized CI definition except that it is not associated with a given instance of a CI; it is a template definition that stands on its own. Just like with an Authorized CI, the CI Hierarchy contains a set of attributes and values and relationships between objects. In the Actual CI application, the user can then select a number of Actual CIs and indicate that they wish to promote them to a given CI Hierarchy, which will create an Authorized CI instance for each of the Actual CIs selected. The Authorized CIs created will contain whatever objects, attributes, and relationships are specified in the CI Hierarchy selected during promotion. Note that when promoting Actual CIs to a CI Hierarchy, if you do not specify copying the attributes during the promotion, then it will use the attribute value that is in the CI Hierarchy and not what is specified in the Actual CI (useful when you want all of the Authorized CIs you are creating through promotion to look the same). A CI Hierarchy can be used as a filter, but can provide a way to set default values for specific attributes as well. For more details on how to use the promotion process, please refer to *IBM Tivoli CCMDB Implementation Recommendations*, SG24-7567.

Regardless of which way you create an Authorized CI in the authorized CI data space, you can change its definition as required to support Service Management processes. You can add or delete attributes, although the related Actual CI keeps

a different structure. Usually you would extend the Authorized CI scheme with attribute definitions to persist or federate data that is not discoverable.

Other data artifacts in the CCMDB Data Layer

The CCMDB contains more than just CIs; it also contains process artifacts that are used in a Service Management environment within the processes and applications that manage the CIs. Examples of process artifacts are RFCs, Change Records, and Release Records. Process artifacts have relationships with Authorized CIs, as shown in Figure 4-1 on page 76, and are an integral part of the data.

In addition to Actual and Authorized CI data structures, the database can hold Asset-based data representations. Assets are represented as analogous to CIs in an actual and authorized data space. CIs and Assets are linked through appropriate relationships in the database. Authorized CIs are linked to authorized Assets while Actual CIs are linked to actual Assets. Asset data structures are usually supporting commercially oriented processes like IT- or Enterprise Asset Management® while CI representations are foremost oriented to supporting more technically oriented processes like Configuration or Change Management.

4.4 Audit: comparing Actual and Authorized CI data spaces

Comparing Actual to Authorized CIs is referred to as an Audit. A CI Audit is part of Configuration Management.

The process of auditing a set of CIs requires defining a set of rules that specify the criteria of how to link the Authorized to the Actual CIs and what exactly to compare during the Audit.

The Audit process can be executed in a scheduled or on demand basis. The result of the Audit is showing either a consensus or deviation between the Actual and Authorized CI data that is compared.

You can, for example, see which authorized CI does not have a corresponding Actual CI or which Actual CI does not have an Authorized CI. You can also see deviations within the attribute values of the two CI data spaces.

A key use case for the Audit process is to verify if a change that has been planned based on the Authorized CI data space is reflected in the real IT environment. A periodic discovery of the IT environment is capturing the current details of the CIs and feeds the (filtered) details into the Actual CI data space.

4.5 Federation of external data

The CCMDB allows for data sourced in an external repository to be made available to the Service Management processes and applications by configuring the federation capabilities of the underlying database or federation support tools and using the CCMDB tooling available (Database Configuration and Application Designer applications) to make the federated data accessible for display in applications.

In fact, federation can be enabled on any type of object in the process layer database portion of the CCMDB, whether it be additional attributes on CIs (for example, the owner of a computer system) or additional attributes on a process artifact (for example, the contact point for a Request for Change that was synchronized from a third-party service desk). Federated data is accessed in a read-only manner, the source of the data is the owner, and changes should only be made at the source.

When there is additional data you would like to make available in the CCMDB, the question arises as to whether model extensions should be made to physically import the additional data into the CCMDB or whether the data should be federated in (where the data is not physically copied into the CCMDB but is made available in a read-only manner through the federation capabilities).

If there is a new attribute or a new CI Type that you would like to include in a Configuration Management audit, then you would have to physically populate the attribute into the CCMDB. Federated data cannot be used in a Configuration Management audit.

If the attribute is something that exists in an external management system and you wish to simply make it available for displaying additional detail about an object in an application (for example, whenever you look at a computer system you want to also see the color, so you federate in the color attribute), then it would be best to make the data available through configuring it as a federated attribute.

Please refer to *IBM Tivoli CCMDB Implementation Recommendations*, SG24-7567 for more details and an implementation example of how to customize the Actual Configuration Items application to dynamically display federated data at run time. Please note that you can also use the federation capabilities within the CCMDB discovery environment. Given that you have simply the requirement to produce reports out of your detailed discovered CI data space and want to enrich the discovered data with federated data from external data sources, you can do so. Please bear in mind though that you cannot transfer data that you federate in the TADDM discovery environment to the Actual CI data space. Using

the federation capabilities inside the TADDM environment is for reporting purposes only.

4.6 Extensibility of the CCMDB schema

The CCMDB schema, that is, the classifications of class types, relationships, and attributes, is extensible in a number of ways. Extensions can be made within the TADDM discovery component or they can be made within the process run time database.

Extensions made to the CDM implementation in TADDM will ensure that both TADDM and the Base Services have the same schema definition of the CDM. This is based on the fact that the schema definitions are propagated through the ITIC TADDM CI Type adapter.

TADDM supports the creation of extended attributes through the user interface. It also allows for the addition of new CI Types, new relationship types, new discovery sensors, or customizing naming rules, all of which require the use of a toolkit. Extensions may also be performed within the CCMDB process layer database, so you may find cases where you wish to perform model extensions there. For example, you may choose to add a non-discoverable attribute directly to an Authorized CI and expose the attribute in an application user interface and have the attribute value manually provided by the user. The manually entered attribute may not be something that you would like to have added to the CDM and may not provide any value by making the extension in TADDM; it is only ever used on the Authorized CI. There are various tools for making extensions within the CCMDB solution, including the Database Configuration and Application Designer applications.

4.7 Where to begin to load data

As explained, there are different manifestations of CI data in the CCMDB. Discovered CIs are populated within TADDM and provide a view of the discovered IT environment. TADDM discovery sensors perform discovery while the DLAs provide a mechanism for loading additional data that conforms to the CDM. On the other end of the spectrum are the Authorized CIs, which are the entities that you explicitly build, which are subject to tightly managed control. You can create Authorized CIs without having a Discovered or Actual CI. So where should you load CIs that you wish to put into the CCMDB?

If it is IT data, or discoverable in any way, then it can be loaded into the CCMDB by using the TADDM discovery capabilities. Having data come in through TADDM discovery allows for the use of a key set of discovery services, such as naming and reconciliation, and the ability to specify a unique identification of a resource. As subsequent discoveries occur, changes are captured so that a change history for a resource exists, which can be valuable for problem analytics. And, of course, with periodic discoveries, we can copy Discovered CIs into the Actual CI space that forms the basis for the CI auditing between Authorized CIs and Actual CIs. If there is a need for any one of these discovery-related capabilities or any of the others not mentioned here that are available with TADDM, then the CI data should be brought into the CCMDB through the TADDM discovery capabilities.

It is possible that you have a situation where you have CIs for which there is only a single source (you do not have a need to reconcile these objects with some other source) and you do not have a need to perform an audit between an authorized version of the CI and what would currently be in a “discovered” environment. In this case, it is feasible that you could create these CIs directly in the Authorized CI space in the CCMDB. An example of this type of data would be a logical hierarchy of business services and business applications that you have defined in an external system that you wish to be able to bring under change control, so you decide to use the CI MEA (Maximo Enterprise Adapter) to create Authorized CIs directly in the Authorized CI space without bringing them in through the discovery services.

It is reasonable that you will actually use both of these methods as you continue to build out your CCMDB, using discovery as the workhorse to bring in the discovered environment as well as directly creating Authorized CIs in the cases where you have single-sourced data that you do not wish to bring in through TADDM. It is important to recognize that if you feel that you will at some point need to reconcile these objects you are directly creating in the Authorized CI space with objects from another data source, or you will need to audit these CIs for changes, then you should give consideration to whether this data should instead initially be brought in through the TADDM discovery services of the CCMDB solution.

Physical components and operational model

In Chapter 3, “CCMDB components” on page 29, we describe the components of the CCMDB V7.1 solution from a functional perspective. Our focus is on the logical components, such as the Base Services, the Process Manager Products, Data Access Services, or the Data Layer in general rather than the physical run time environment.

In this chapter, we now describe the run time environment that is used to host the logical components that we described:

- ▶ We describe the major building blocks of the physical run time environment, their role in the overall solution, and their relationships to each other.
- ▶ We describe the environment from the perspective of an operational model. We explain the protocols used for component interaction and different options for deploying the solution.
- ▶ We give you a rough understanding of what needs to be taken into account in order to size the environment and refer to aspects of high availability.

5.1 Components of the physical run time environment

Figure 5-1 highlights the components of the CCMDB V7.1 run time environment.

Please note that although the components are drawn in separate boxes, they do not necessarily have to be deployed on different physical systems.

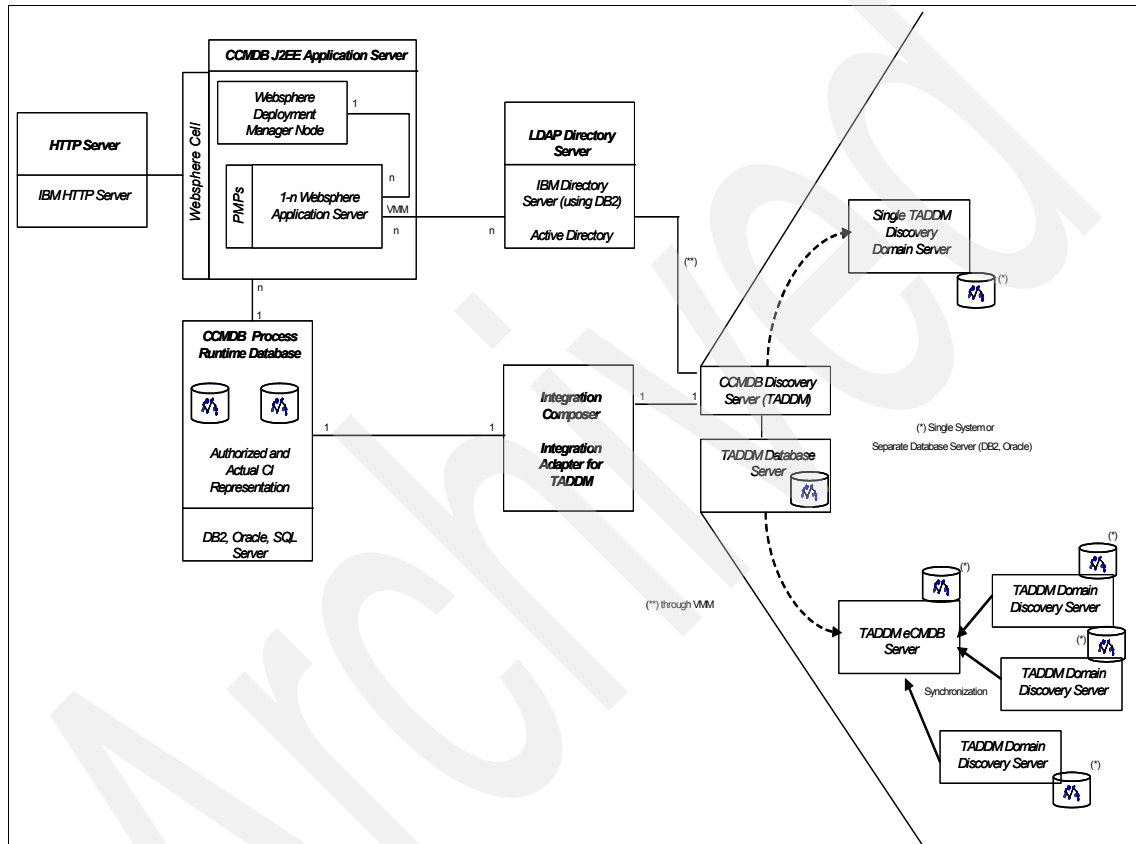


Figure 5-1 Physical component and relationship overview

The HTTP Server, the WebSphere Application Server, the CCMDB Process Runtime Database, and the LDAP Directory Server comprise the environment that hosts the Process Manager Products. In other words, this is the environment that is needed to run Service Management process implementations.

The discovery environment is reflected on the right hand side of Figure 5-1 and is labeled as a TADDM component.

Note: Please note that Figure 5-1 on page 90 presents two different options for implementing the TADDM environment. Either you use a single Domain Discovery Server or you implement TADDM in multi-domain environment front-ended by an instance of an enterprise discovery server (eCMDB).

The Integration Composer is the interface between the discovery and the process run time environment.

5.1.1 Physical components of the process run time

The HTTP Server is serving the user requests of those users participating in a Service Management process, such as Change or Configuration Management. It forwards these requests to the application server that handles the business logic of the applications. CCMDB V7.1 by default uses the IBM HTTP Server V6.1 Fix Pack 9. The HTTP server is by default installed by the middleware installer. In addition, it installs and configures the WebSphere Application Server Plug-in for the IBM HTTP Server (IHS) in order to set up the communication path between the HTTP server and the Application server.

You can configure the settings of the IHS from the WebSphere Admin Console. In our environment, we use the following URL in order to launch the console:

`http://fenway.itsc.austin.ibm.com:9060/ibm/console`

Figure 5-2 shows the WebSphere Admin Console.

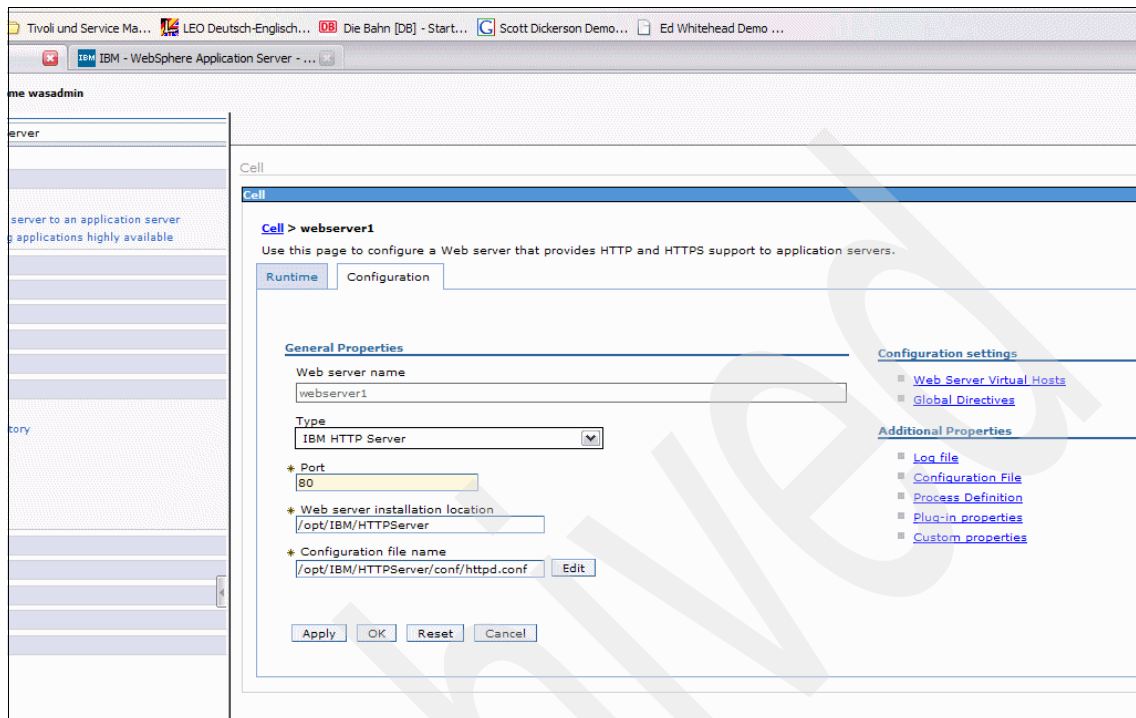


Figure 5-2 IBM HTTP Server configuration in WebSphere Admin Console

Another key component is the J2EE application server environment, which provides run time services to the applications and hosts applications such as Change and Configuration Management itself. In CCMDDB V7.1, IBM WebSphere Application Server is the only supported J2EE implementation. The version that is used is IBM WebSphere Application Server Network Deployment V6.1.0.9 (equivalent to Fix Pack 9).

The WebSphere middleware itself is installed by the middleware installer while the applications (including the PMPs) are installed by the CCMDDB installer.

The installation by default creates a WebSphere cell, that is, in WebSphere terminology, an administrative domain, named *ctgCell01*. Inside this cell is a node that is considered the Deployment Manager Node. It provides central administration services for all other nodes in the cell by communicating to the node agents on each of the systems in case you have a multisystem environment. By default, the Deployment Manager Node is named *ctgCellManager01*.

Each node, which is an entity primarily for administrative purposes, hosts one to many application servers.

Finally, the application server is the server that hosts and runs the application itself, in our case, the CCMDB applications. There is a special application server on the ctgCellManager01 node named gander whose purpose is to run the administration application for the cell.

A second node is created at install time to host the application server on which the CCMDB application server is running. The node by default is named ctgNode01. The application server is the MXSERVER application server.

The main application that is used for Change and Configuration Management is an application called Maximo provided in the Maximo.ear file. This EAR file contains the code for all base services, PMPs, and all other administrative applications.

Note: By default, in a single system environment, both nodes are defined on the same physical system, although logically separated. In a large environment, we expect the Deployment Manager Node to be on a separate physical system.

All these definitions are shown in the WebSphere Admin Console. We use the following URL in our environment in order to launch the console:

`http://fenway.itsc.austin.ibm.com:9060/ibm/console`

Looking at the cell structure of our default deployment, you can see the defaults that have been created for the cell itself, the nodes, including the Deployment Manager node, the application server, and the applications.

Note: We used the default installation and installed the middleware environment all on the same box. If you change the default values for a reason, such as deploying into an existing WebSphere cell, then the structure shown in Figure 5-3 on page 94 will look like slightly different.

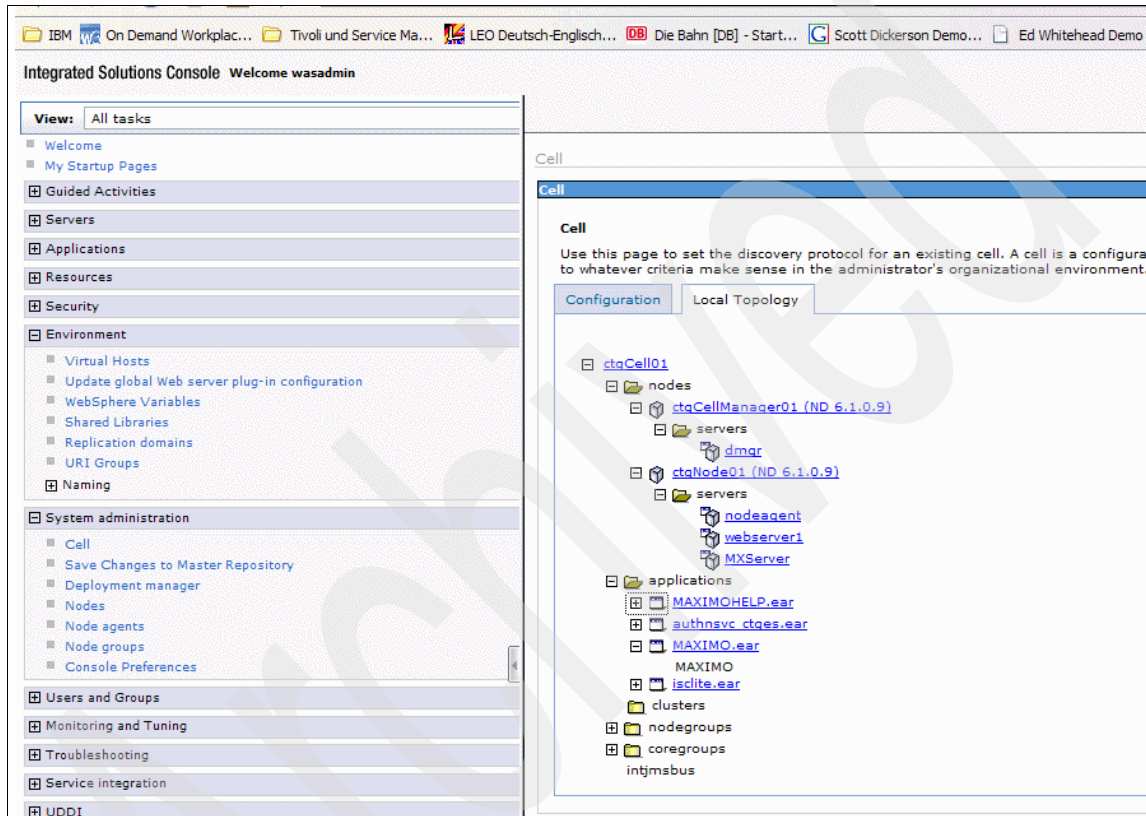


Figure 5-3 WebSphere cell definitions: CCMDB default installation environment

Each cell has exactly one Deployment Manager node. It can have multiple node agents, each hosting one to many application server instances, for example, to scale the environment in order to serve a high number of user requests. This explains the 1:n relationship between the Deployment Manager node and the application server node in Figure 5-3.

For the CCMDB solution, the WebSphere Application Server environment is using the Virtual Member Manager technology (VMM) inside its security implementation. We explain this in detail in Chapter 6, “CCMDB security architecture” on page 105. VMM abstracts the usage of different Directory Server

implementations. CCMDB V7.1 makes use of an LDAP Directory Server for persisting user, group, and user to group membership definitions as well as passwords.

CCMDB V7.1 can either use the IBM Directory Server V6.1, which is the default or Microsoft Active Directory, as a directory server implementation. IBM provides the IBM Directory Server within the CCMDB package. The installation is performed by the Middleware Installer while Active Directory is not provided by IBM or installed by the installation program. You have to install Active Directory on your own if you prefer to use it.

We show an n:n relationship between the WebSphere Application Server environment and the Directory Server. This is because VMM can hide different Directory Server implementations and instances to synchronize user and group related data into the process run time database using a VMM cron task.

The final major component of the process run time environment is the database system. We also refer to it as the CCMDB Process Runtime Database. There are numerous tables inside the database, for example, table structures to keep data for Actual and Authorized configuration items but also structures for administrative purposes, such as to keep permission data to specify which user is allowed to access which kind of objects and data in the system.

By default, the instance name that is created at installation time is called *ctginst1*, while by default the name of the database itself is *MAXDB71*. If there are multiple application servers deployed inside the WebSphere cell, either on one physical system (and on one node) or spread over multiple physical systems, they all access the same database instance. This is why we show an n:1 relationship between the WebSphere Application Server environment and the database.

The database server is supported on DB2, Oracle®, or Microsoft SQL Server®. Please refer to the product documentation for details on the specific versions. CCMDB provides DB2 in its package, while Oracle or SQL Server have to be acquired separately. In addition, the Middleware Installer installs DB2 by default, while you have to install Oracle or SQL Server manually.

5.1.2 Integration Composer component

The Integration Composer is a data migration component. Its primary purpose in the CCMDB environment is to transfer discovered data into the Actual CI space of the environment. Please refer to Chapter 10, “TADDM and Process Layer integration” on page 231 for more details on the data migration responsibilities of the Integration Composer.

The Integration Composer is a Java application that can either run on a system of its own, it can run on the same box as the CCMDB J2EE Application Server, or it can run on the TADDM server. It requires two appropriate connection definitions to the source and target systems of the data migration. These definitions are referred to as data source definitions.

You can install the Integration Composer from a command line or from the CCMDB launchpad. We used the launchpad for the installation in our lab environment and installed the Integration Composer on the same system as the CCMDB J2EE Application Server. The integration composer actually uses its own tables inside the MAXDB71 (CCMDB process run time database) to maintain its own metadata.

While the Integration Composer is a generic data migration component, it needs to be configured with specific information about the source and target of the migration process, schema, and mapping instructions. This is what the Integration Adapter for TADDM actually uses for migrating data within the CCMDB V7.1 environment.

In CCMDB Version 7.1, there can only be one Integration Composer environment, because CCMDB V7.1 supports data migration from only one TADDM environment, either from a single domain server or from an Enterprise Domain Server (referred to as the eCMDB server).

5.1.3 Components of the Discovery / TADDM environment

Figure 5-1 on page 90 actually highlights two different options for using TADDM as the discovery solution of the CCMDB. Depending on the size or organizational structure of your environment, you can either use a single discovery domain server or you can use several discovery domain servers that synchronize their data with the central Enterprise Discovery Server, often referred to as the eCMDB server.

Each TADDM server in the environment requires its own database, regardless if it is a domain or enterprise server instance. The database system supported for CCMDB V7.1 is either DB2 or Oracle. IBM provides the DB2 software inside the CCMDB V7.1 package.

You have the option to use the TADDM database on a separate or on the same system as the TADDM application server. This again depends on your environmental and organizational requirements. The general recommendation is to have the database implemented on a separate system as the TADDM application server.

The Integration Composer (Integration Adapter for TADDM) is either pointed to transfer data from the single domain server or the eCMDB server.

Note: If you have a distributed discovery environment using multiple separated domain discovery servers or using an eCMDB environment, you can point the Integration Composer to only one TADDM instance. In CCMDB Version 7.1, you cannot transfer data from more than one discovery environment into the process run time database.

There are slight differences in the platforms that CCMDB V7.1 supports regarding the different physical components mentioned previously. Although the installation manual provides detailed information about the supported platforms and operating system levels, we summarize the platform support in Table 5-1.

Table 5-1 Operating system support for CCMDB components

HTTP Server (IBM HTTP Server)	J2EE Server (WebSphere)	Database Server (DB2, Oracle, SQL Server)	Directory Server (IBM Directory Server, Active Directory)	Integration Composer	TADDM Discovery Server (Database DB2, Oracle)
Windows	Windows	Windows	Windows	Windows	Windows
Red Hat Linux (Intel®)	Red Hat Linux (Intel)	Red Hat Linux (Intel)	Red Hat Linux (Intel)	Red Hat Linux (Intel)	Red Hat Linux (Intel and z)
AIX®	AIX	AIX	AIX	AIX	SUSE Linux (Intel and z)
				Solaris™	Solaris
				HP-UX	

5.2 Operational model

After introducing and explaining the roles of the various physical components, in this section we explain their interaction from a run time or operational perspective. Although it is not an extensive description, it should give you insight into how the various components communicate and could operate in case your sizing requires you to have the system operate in a multi-system deployment.

Figure 5-4 shows aspects of an operational model for the overall CCMDB V7.1 solution.

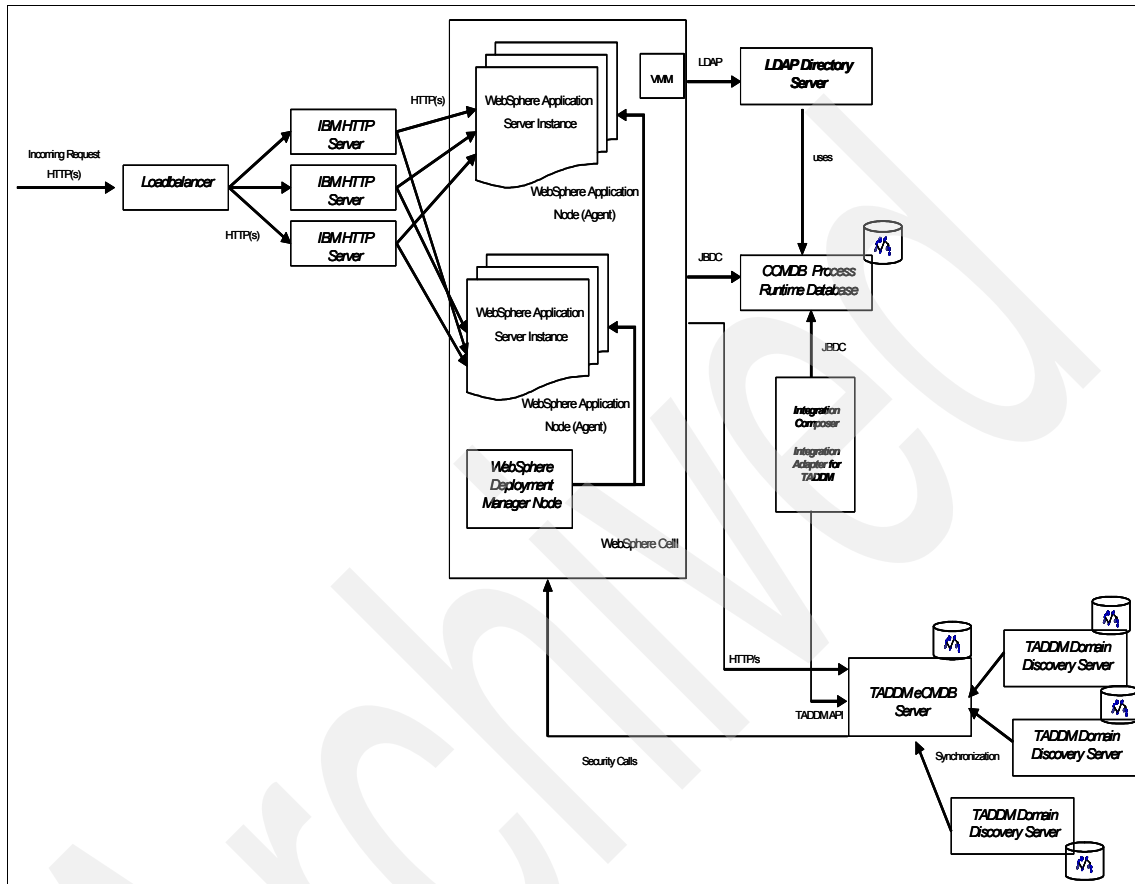


Figure 5-4 Operational model

We do not introduce network components like firewalls in Figure 5-4. Since CCMDB is following the paradigm of a distributed application, in real customer environments, multiple firewalls could segregate different entities of the solution.

For example, there usually is a firewall between a load balancing system and the HTTP Server layer, while in some cases firewalls are implemented between the HTTP layer and the application server, as well as between the application server and the database layer. This very much depends on the security policies of your organization.

Some organizations also use a reverse proxy in front of the HTTP server layer in order to capture incoming requests and have them authenticated through a centralized security facility. Tivoli Access Manager for eBusiness is an example of such a centralized security facility.

We do not embed firewalls or other security components in further discussions in this chapter. Please be aware that the load balancer component shown in Figure 5-4 on page 98 is not part of the CCMDB V7.1 solution itself.

The incoming user requests flow into the system either through the HTTP or HTTPS protocol. The request is usually captured by a load balancing system and spread into the HTTP server layer. The HTTP server layer can consist of multiple physical systems in order to satisfy high availability as well as performance requirements.

Each HTTP server forwards the request to the application server layer, which is a WebSphere Application Server environment in the case of the CCMDB solution. As described in 5.1.1, “Physical components of the process run time” on page 91, the CCMDB J2EE applications are deployed into a WebSphere cell, which can either be an existing one in your environment or one that is created at installation time.

In order to spread the load, you can deploy the J2EE application into a multisystem, multiapplication server environment. Since the WebSphere environment scales horizontally as well as vertically, you can either deploy multiple application servers on one physical box in case you have enough system resources or you can spread multiple application servers onto different physical systems. Each cell consists of one WebSphere Deployment Manager Node and one or more nodes running one or more application server instances. By default, as in our lab environment, both nodes including the MxServer application server instance are hosted on the same physical system. If you use different physical systems for hosting additional application servers, there is always one node agent implemented on each box in order to receive management instructions from the Deployment Manager Node inside the cell.

CCMDB makes use of VMM, a component of WebSphere being used for security purposes. VMM communicates through the LDAP protocol to one or multiple LDAP directory servers. The directory server infrastructure can consist of multiple physical systems and use replication techniques in order to synchronize their data. If you choose to use the IBM Directory Server instead of using Microsoft Active Directory, the directory server implementation will use an underlying DB2 database to maintain its physical data in appropriate table structures. This database can, but does not have to, be the same system as being used for the CCMDB Process Runtime Database.

The J2EE application running in the WebSphere environment uses a JDBC connection to the CCMDB Process Runtime Database. In order to simplify the setup shown in Figure 5-1 on page 90, we show just one connection from the WebSphere cell into the Database system. Technically, each CCMDB (MXSERVER) Application Server is actually using its own JDBC connection to the database.

We have so far explained the communication paths between the components of the process environment. Since this run time environment is very much user oriented (the users in various roles using the Service Management processes), we expect that the operational model of the J2EE environment will be aligned to the operational model that you probably have already in place for existing J2EE business applications.

The Integration Composer is using a JDBC connection to connect to the CCMDB Process Runtime Database on one side to keep its own metadata as well as to load CI type and instance data into the database. On the other side, it uses a TADDM API call by default on port 9530 to the TADDM Discovery Server, which is either a single domain or an enterprise domain discovery server to pull CI type and instance data from the TADDM system.

The TADDM environment is usually not considered an environment to be driven by many users directly. From an operational perspective, it is a back-end facility to discover data and expose this data into the process layer of the CCMDB solution. Nevertheless, if you plan your operational setup, you have to consider that you probably want to launch in context into one of the different TADDM topology views, either the detail or the change view from within the CI or Actual CI applications in the process environment. If you want to do this, you have to consider that there is an HTTP(S) connection from the WebSphere Application Server instance that is satisfying the user request flowing into the primary TADDM server instance.

In addition, the TADDM server calls out to the WebSphere environment for security purposes. This is to authenticate a user or validate an incoming LTPA token in case of a launch in context operation. You should plan for this path of communication from the eCMDB server instance rather from the domain servers in case you plan a multi-domain discovery environment.

In case you have a distributed TADDM environment with multiple domains and an eCMDB instance, the Launch in Context is directing the HTTP(s) request into the eCMDB system. There is a specific servlet on the TADDM server that is satisfying these requests. The launch in context operation can use the Domain Manager or Java Console of the TADDM environment. So you have to make sure that when you have a firewall between the WebSphere Application Server and the TADDM server that the request is getting through the firewall.

The TADDM environment itself is targeted to discover primarily your datacenter environment. This usually requires the discovery requests to cross firewalls in order to get to the target system. TADDM provides an entity called an Anchor Server that helps discover those systems inside a firewall protected zone so that you just have to open one port from the TADDM Domain Server to the Anchor Server (SSH).

Note: Please note that the discovery is not initiated from the eCMDB server but from the Domain Discovery server.

For more details on anchor host or firewall ports, please refer to the IBM Redbooks publication *IBM Tivoli Application Dependency Discovery Manager Capabilities and Best Practices*, SG24-7519.

5.3 Scalability and high availability

The CCMDB V7.1 solution is designed for being used in large enterprise environments. In this section we give you a rough understanding of what the determining factors for scalability and high availability are. We do not provide a detailed sizing guide within this chapter.

5.3.1 Scalability

The CCMDB design allows you to scale for discovering and maintaining a high number of systems as well as satisfy a high number of user requests to work with the data that has been discovered within Service Management process implementations like Change or Configuration Management.

The TADDM discovery component of the CCMDB V7.1 solution primarily scales up by adding additional discovery domains to the environment and synchronizing the data into the eCMDB instance. The factors that you have to take into account when sizing your discovery environment, for example, when to add additional domains, how many systems you can cover in one domain, how to best layout the system resources, or how to size your database system, are covered in *IBM Tivoli Application Dependency Discovery Manager Capabilities and Best Practices*, SG24-7519.

For the process run time environment, the primary determining factors that have to be taken into account for scalability are:

- ▶ The number of users working with the system.
 - How many users are working at the same time actively with the system from within the different processes that are implemented? The average number of users active at a time is referred to as *concurrent users*. Usually one third of the overall registered user population is considered to be a concurrent user.
 - Of what type are these users? Are they power users and heavily using the system or are they just logging into the system, doing some steps and leaving the system, or do they remain logged in while in between activities? We refer to this type of user as a *walk up users*.
- ▶ What is the expected response time for general user operations? We refer to response time in this case as the time being used for generating a UI request and receiving a rendered window in the CCMDB Web Browser. We consider a request a synchronous request when there is no user “think time”.
- ▶ What processes are running on the system? How effectively are they used? Are there a lot of background activities like escalations or notifications being used?

The CCMDB process run time environment, since it is based on a J2EE (WebSphere) application server infrastructure, can be scaled to a large number of users and high performance either by using a horizontal or vertical scalability approach.

Vertical scaling means adding hardware resources (processors or RAM) to a single node. Adding more resources to the physical system, you can then add additional logical application server instances to the system. Each logical application server is running in its own Java Virtual Machine, so you have to take into account things like Heap Space or Thread definitions. There is a limit of concurrent users that a single logical application server can serve, so it is not that you can just add physical resources to the system and dedicate more heap space to the logical application server. You can take approximately 50 concurrent users per application server instance as a rough guideline.

Horizontal scaling means that you are adding more physical systems into the WebSphere cell and adding logical application servers into those nodes in order to spread the workload among them.

You can combine horizontal and vertical scaling techniques. This means you are implementing multiple physical servers, each hosting multiple logical application servers running the application.

Please note that spreading the load over multiple physical systems hosting multiple logical application servers is also targeting the aspect of high availability. This means in case one application server instance is down, the others can still serve the user requests. If you need full fail-over capability, you have to deploy multiple application server instances on at least two physical nodes.

In this book, we do not provide absolute numbers of required system configurations and resources, since they can differ a lot depending on the platform being used and environmental circumstances. Therefore, we provide two examples that are aligned to real life customer implementations of the process run time environment. These are to give you a rough understanding of what is needed to scale the CCMDB environment:

- ▶ Intel based environment, 200 concurrent users, separation of application and database server:
 - 1 x Application Server: Four CPUs (Intel Quad), 8 GB RAM
 - 2-4 JVM™ Instances (two to four logical application server instances)
 - Database Server: Two CPUs, 4 GB RAM
- ▶ UNIX® based environment, 400 concurrent users, two physical application server nodes, separation of application and database server, designed for high availability:
 - 2 x Application Server: Eight CPUs, 16 GB RAM (each)
 - 16 JVM Instances (16 logical application server instances) spread over the 2 physical nodes
 - Database Server: Eight CPUs, 16 GB RAM

As a general rule of thumb, adding 50 concurrent users to the system requires you to add approximately one CPU and 1 GB of RAM of physical resources to the system. You also should consider a logical application server instance to the system.

5.3.2 High availability

The goal of a high availability approach is to prevent single point of failures in the system. The requirement is to always provide the ability to respond to user requests or perform back-end operations like discovery in the environment.

The TADDM discovery environment can be designed for high availability by leveraging an external high availability solution such as Tivoli System Automation for Multiplatform. Using Tivoli System Automation for Multiplatform in order to fail over components of the TADDM environment is documented in *IBM/IBM Tivoli Application Dependency Discovery Manager Capabilities and Best Practices*, SG24-7519.

The CCMB process run time environment is leveraging its underlying middleware technology in order to satisfy the requirements of high availability:

1. Use a cluster of HTTP Servers behind a load balancing system in order to receive incoming user requests. In case one of the Web Servers is down, the load is spread by the load balancer to the remaining systems.
2. Use a cluster of logical application server instances to host the Maximo / CCMDB J2EE application. You can spread the load to multiple logical application servers on one physical system or application server instances distributed over multiple physical systems. In case one of the application servers is down or in maintenance, the remaining application servers can take over the load. With respect to high availability, you should at least have two physical systems in your application server cluster in case one system is down or in maintenance.
3. LDAP directory server implementations make use of techniques like replication and referral. Replication is the technique to copy data from master to several subordinate servers. IBM Directory server even makes use of a concept referred to as *peer-to-peer replication*, which allows you to define multiple masters. You can replicate the data between those instances. These techniques not only allow you to scale the environment, but they also prevent single point of failures.
4. For the database system use an external high availability solution like Tivoli Systems Automation for Multiplatform, HACMP™, VERITAS Cluster, or a solution recommended by the vendor of the Database you are using in your implementation. You should also consider SAN technologies for data replication purposes.

The only component of the overall CCMDB V7.1 solution that can be regarded as a single-point-of-failure is the Integration Composer. Since the Integration Composer is operating in a batch-oriented mode and does not have to satisfy user requests on demand, we regard this as acceptable.

CCMDB security architecture

In this chapter, we explain the security architecture of the overall CCMDBV 7.1 solution.

The CCMDB solution is considered a concept that is comprised of multiple physical and logical components. The main components are the process run time and the discovery environment. In order to support a sophisticated user experience, a common security model between those major components has been implemented.

Note: The TADDM Discovery environment, if used in a stand-alone fashion, can still rely on its own security model. If you do not rely on single sign-on, a common user and group Repository, or a Launch in Context capability, TADDM can rely on its own security implementation. Even if you just want to transfer data from the Discovered CI space to the Actual CI space through the Integration Composer, TADDM can rely on its own security model.

In this chapter, we primarily focus on components of the solution that are involved in the common security model for authentication, authorization, and single sign-on (SSO). Our focus is not on other aspects of security, such as transport layer security or encryption standards being used in the CCMDB solution.

We provide insight into key areas of the system that are either installed and customized by default or need to be adjusted to your organizational needs. We provide example windows in order to make your customization as easy as possible.

Figure 6-1 represents the common CCMDB V7.1 security architecture.

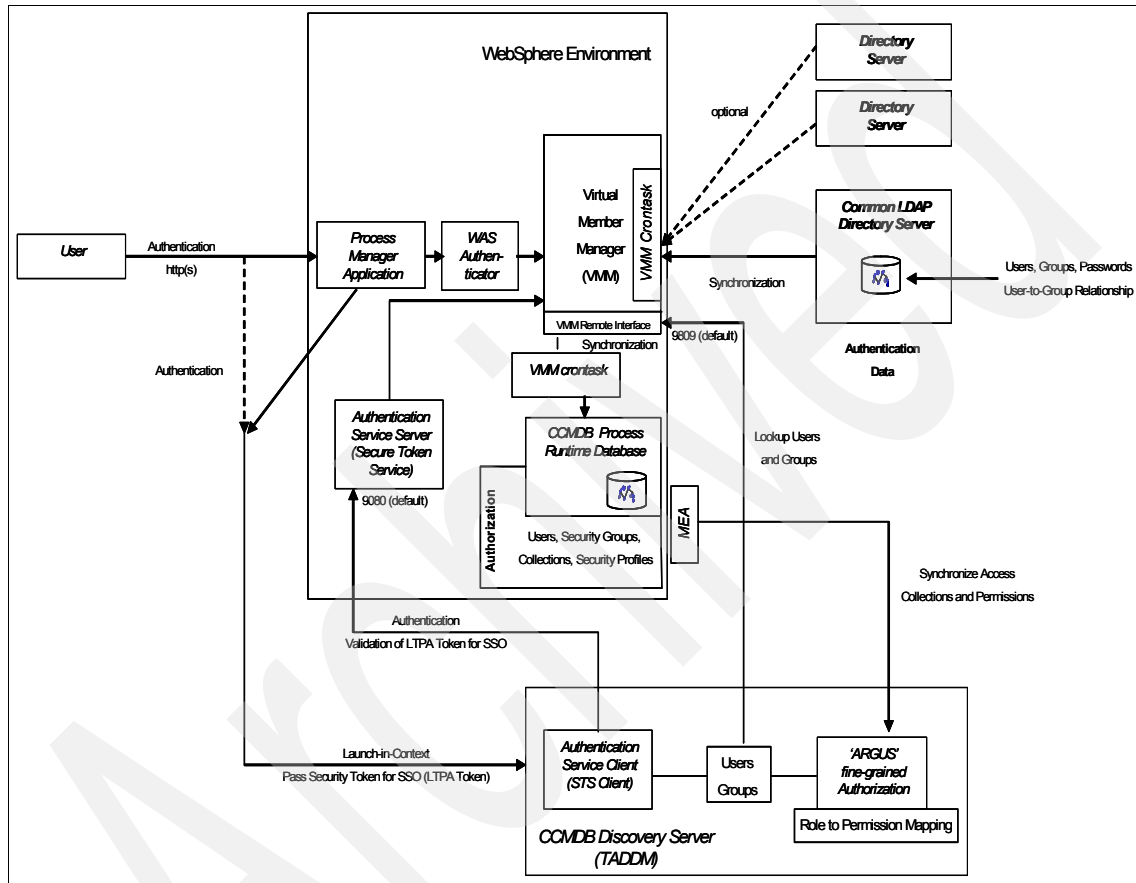


Figure 6-1 CCMDB V7.1 security architecture

The two major CCMDB V7.1 components with respect to security are the process environment hosted inside the J2EE WebSphere Application Server run time environment and the CCMDB Discovery Server, referred to as TADDM. We consider the process environment the leading component of the solution since most of the users of the CCMDB will be users of the Service Management processes rather than the discovery environment. We consider the discovery environment a data center component that collects data and provides the data to different Service Management solutions. TADDM is not considered a component that many users in the organization work with. Therefore, we start our explanation of the security model inside the WebSphere area of the CCMDB.

6.1 CCMDB V7.1 authentication model

The Process Manager Products are hosted inside a J2EE WebSphere environment and rely on the facilities that the WebSphere Application Server provides.

There are two main components that are relevant for the CCMDB security. The Virtual Member Manager (VMM) and the Secure Token Service (STS). These WebSphere based security components are involved in providing authentication and authorization services to WebSphere based applications.

6.1.1 Virtual Member Manager

IBM WebSphere Application Server V6.1 offers a federated user repository feature that makes it easy to access and maintain user data in multiple repositories. This feature is called the Virtual Member Manager (VMM). It provides the ability to map entries from individual user repositories into a single virtual repository. The federated repository consists of a single realm, which is a set of independent user repositories. Each repository may be an entire external repository or, in the case of LDAP, a sub-tree within that repository. The root of each repository is mapped to something called a base entry within the federated repository, which is basically a starting point within the hierarchical namespace of the virtual realm. VMM provides a repository-independent programming interface that shields the application from the details of the repository implementation.

CCMDB V7.1 uses the Virtual Member Manager technology to shield the process run time environment from the LDAP provider being used. All interactions between the applications and the directory server flow through the virtual member manager. Its common interface masks the differences of the LDAP provider implementation, be it IBM Directory Server or Active Directory Server or different implementations in the future.

CCMDB relies on LDAP directory server implementations, IBM Directory Server and Microsoft Active Directory being supported in Version 7.1, to maintain user and group entries as well as the user to group relationships. The relationship defines which user is a member of which group. In addition, passwords for users are maintained in the LDAP implementation.

Note: You have to enter users, groups, passwords, and user to group relationships in LDAP. You are not supposed to add these entries in the CCMDB applications directly. You have to do that either using the command line to import LDIF files or the appropriate user interface for the LDAP implementation that you use in your environment. Please note that the administration application for the IBM Directory Server is not deployed by default when you install the middleware through the CCMDB Middleware Installer. You have to deploy it manually.

As Figure 6-1 on page 106 shows, there can be multiple directory server implementations at run time that get virtualized by the VMM technology.

In our environment, we use one IBM Directory Server instance to hold user and group data (Figure 6-2 on page 110). We used the default settings at installation time. In this case, you can see in the WebSphere Application Server Admin console's security definition that a realm called ISMRealm has been defined. In addition, you can see that the definition of the base entry has been defined as ou=SWG,o=IBM,c=US. This is the root entry that is the starting point in the repository for our user and group data.

Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure > Federated repositories

By federating repositories, identities stored in multiple repositories can be managed in a single, virtual realm. The realm can consist of identities in the file-based repository that is built into the system, in one or more external repositories, or in both the built-in repository and one or more external repositories.

Configuration

General Properties

✦ Realm name
ISMRealm

✦ Primary administrative user name
wasadmin

Server user identity

☒ Automatically generated server identity

☐ Server identity that is stored in the repository
Server user ID or administrative user on a Version 6.0.x node
Password

☒ Ignore case for authorization

Repositories in the realm:

Add Base entry to Realm... Use built-in repository Remove

Select	Base entry	Repository identifier	Repository type
☐	ou=SWG,o=IBM,c=US	ISMITDS	LDAP:IDSG

Additional Properties

- Property extension repository
- Entry mapping repository
- Supported entity types

Related Items

- Manage repositories

Apply OK Reset Cancel

Figure 6-2 ISMRealm definition

The identifier is set to ISMITDS by default.

Once user and group data has been entered into the Directory Server, it is synchronized into the CCMDB process run time database. The synchronization between LDAP and the process environment is one-way. It is controlled by a cron task that is defined at CCMDB installation time. The name of the cron task is VMMSYNC.

The definition of the cron tasks are in the Cron Task Setup application inside the Platform Configuration module, which you can find inside the System Configuration module.

Figure 6-3 on page 111 shows the default definition for the VMMSYNC cron task definition.

Back | Search | Favorites | IBM Business Transformation Homepage | IBM Global Print | IBM Standard Software Installer | IT Help Central | Join World Community Grid | IBM Internal Help Homepage

Cron Task Setup | Go To | Reports

Find: | Select Action

Cron Task

Cron Task: VMMSYNC Invokes WebSphere VMM APIs to populate data

Class: psdi.security.vmm.VMMSyncCronTask

Access Level: FULL

Task Instances | Filter | 1-1 of 1

Cron Task Instance Name	Schedule	Run as User	Active?	Keep History?	Max Number
VMMSYNC01	5m*****	MAXADMIN	☒	☐	0

Parameters | History

Task Parameters | Filter | 1-5 of 8

Parameter	Value	Description
Credential	smartway	VMM admin credentials.
GroupMapping	<?xml version="1.0" encoding="UTF-8" ?><DO	The USER XML used by the VMM task.
GroupSearchAttribute	cn	VMM search attribute to query group records.
Principal	cn=wasadmin,ou=users,ou=SWG,o=IBM,c=US	VMM admin principal.
SynchAdapter	psdi.security.vmm.DefaultVMMSynchAdapter	VMM synchronization adapter.

Details

Parameter: Principal VMM admin principal.

Value: cn=wasadmin,ou=users,ou=SWG,o=IBM,c=US

Figure 6-3 Default definition for VMMSYNC cron task

In the Schedule field, the default is to synchronize data from LDAP into the process layer database every five minutes. You can customize the interval to your needs. You can see additional parameters that get configured by default, such as the principal attribute used by the cron task to connect to the local VMM service. By default, this is the wasadmin user. Another parameter is the GroupSearchAttribute. This is the LDAP group attribute used to search for groups under the configured directory sub-tree.

Note: If you are planning to use an existing Directory Server with predefined object structures, you have to customize the VMMSYNC cron task appropriately. If you are planning to use multiple LDAP repositories at run time, you would have to set up multiple VMM cron task definitions.

VMM is the only supported option in CCMDB V7.1 for setting up a connection to the Directory Server. Although technically possible, connecting directly to LDAP without going through the VMM interface is not supported in Version 7.1.

If you try to create a user or group in the process runtime security application, you will receive the pop-up message shown in Figure 6-4.

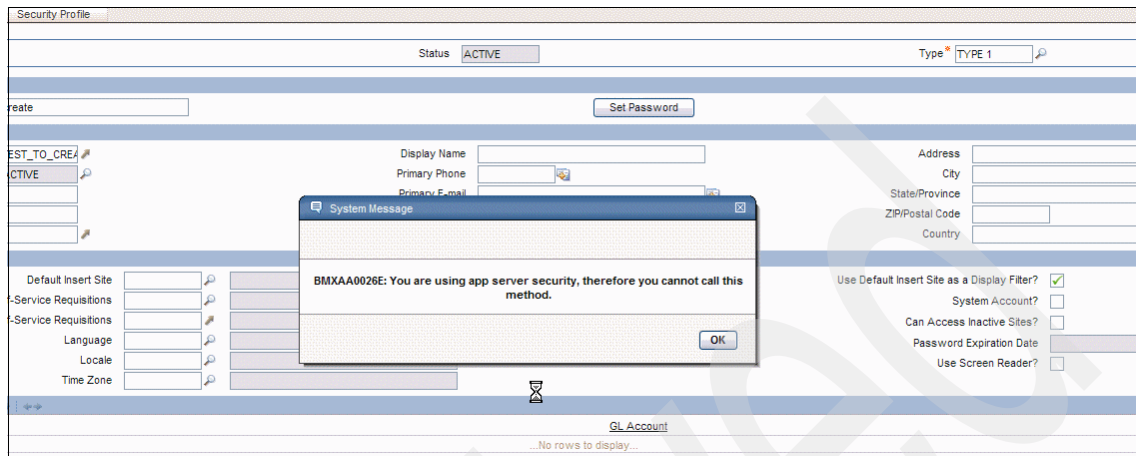


Figure 6-4 Error message of security application

The pop-up message tells you that the system is configured for Application Server security, and that you are not allowed to add users or groups by using the Security Application. You have to use the LDAP interface directly and wait for the synchronization to happen.

For the purposes of this book, we use a simple utility called the *LDAP Browser* in order to search for and enter data into the IBM Directory Server. As you can see in Figure 6-5 on page 113, we created a group called MBGROUP that has two members, MAXADMIN and MICHAELB. We created the user michaelb while MAXADMIN is a default user in the system.

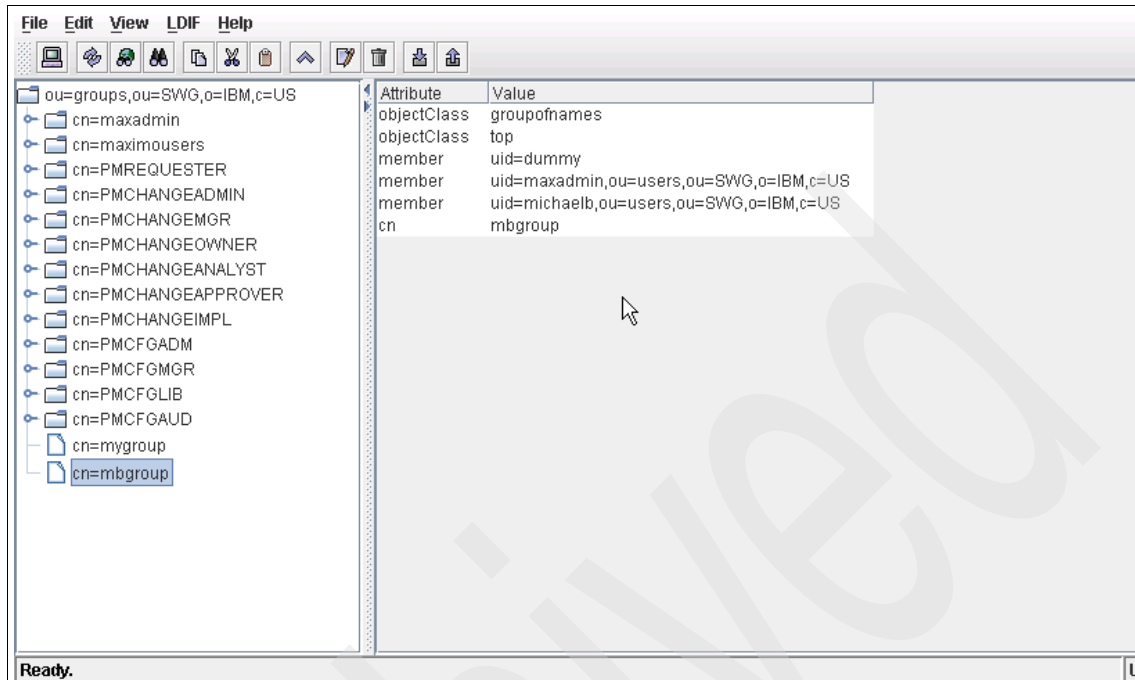


Figure 6-5 LDAP entries

After some time (five minutes by default) the new user, the new group, and the membership definition are synchronized into the process layer database of the CCMDB. You can find them in the Security Module applications for users and groups. Users are persisted in the MAXUSER table, groups are persisted in the MAXGROUP table, while membership definitions are stored in the USERGROUP table.

Figure 6-6 shows the synchronized group definition and user membership definitions in the process environment.

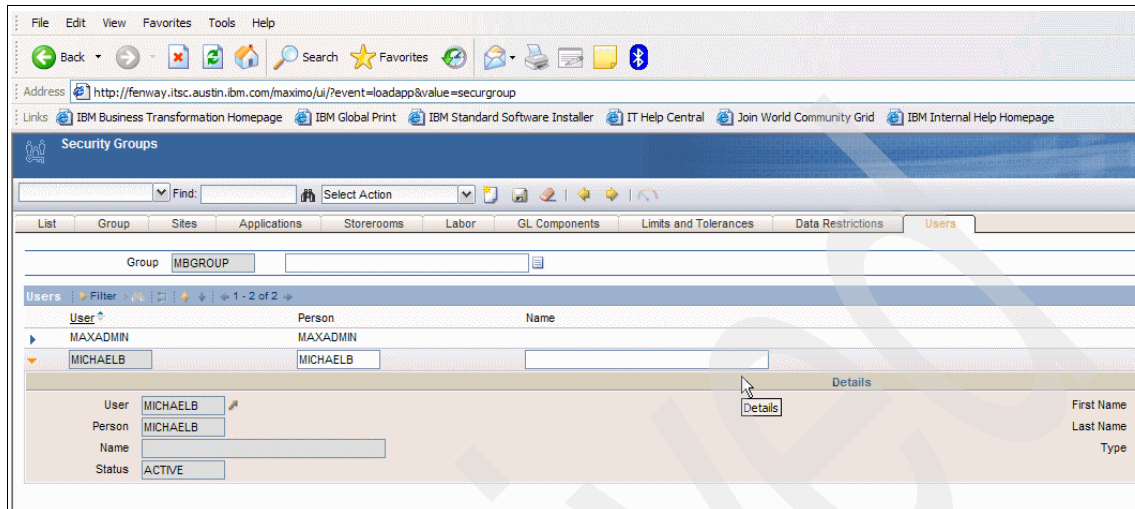


Figure 6-6 User and group definitions synchronized into the CCMDB database

You see the two users that we have defined in LDAP as members of the MBGROUP.

You also can see user and group entries in the WebSphere Admin console. They are listed under the Users and Groups section, as shown in Figure 6-7 on page 115.

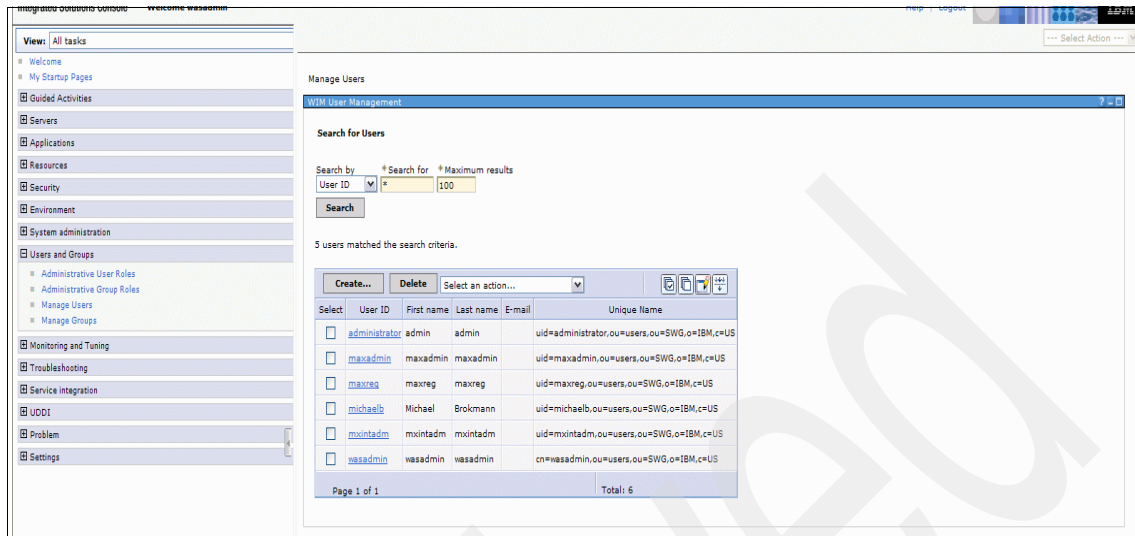


Figure 6-7 Users and groups in WebSphere Admin Console

To summarize, the primary purpose of the VMM component is to federate directory server implementations. This forms the baseline for authentication. You need user and group data in order to allow users get into the system.

6.1.2 Secure Token Service

In addition to what the VMM component provides, there is the need for an authentication service that makes use of the user and group data stored in the directory server implementation. Authentication is the process of determining if a user is the one who he claims to be. The WebSphere Application Server security allows you to define different authentication mechanisms like Basic Authentication, Digest Authentication, Certificate-based Authentication, or Forms-based Authentication.

CCMDB V7.1 uses the Lightweight Third Party Authentication (LTPA) mechanism of WebSphere Application Server security. LTPA usually is used in an interoperability context when multiple applications need to share a common security token that identifies a user so that the user does not have to log in to multiple applications, but rather is provided a single-sign on experience.

The instance of the WebSphere Application Server security model that generates and validates LTPA tokens is called the Secure Token Service, sometimes also referred to as the *Embedded Security Service (ESS)*.

The LTPA token provides features like authenticating the user with a secure mechanism, since it is encrypted (confidentiality) and signed (integrity), and also provides a validity period feature. The LTPA token is very useful when propagating an identity, which means that you can pass along the LTPA token in the different tiers of the architecture, and still keep the identity of the caller.

The Secure Token Service is the second major security facility that CCMDB V7.1 leverages from the WebSphere Application Server security model in order to pass a security token from the CCMDB process environment into the TADDM environment when a user tries to launch in context from a Process Manager Product into the Discovered CI space of the TADDM environment. After TADDM receives the token through the URL of a launch in context operation, it validates the token by calling the Secure Token Service.

In order for TADDM to call the STS instance in the WebSphere Application environment, it has an implementation of an Authentication Service client (STS client).

In case a user logs into TADDM directly, TADDM first looks up the user and group by calling the remote VMM interface and then authenticates the user by calling the STS instance in the WebSphere environment using the local STS client implementation.

6.1.3 Configuring the CCMDB for single sign-on

Now that we have explained the general concepts of WebSphere security being used in the CCMDB V7.1 security model, we show how to configure the solution. We focus primarily on the TADDM side since most of the configurations in the process environment are provided by default.

You have to customize different configuration files on the TADDM server. We recommend that you refer to Figure 6-1 on page 106, which describes the interaction model of the various components. It also highlights the usage of key port definitions. This makes it easier to understand what you are configuring.

Configuring the collation.properties file

Most of the required configurations are done by configuring key-value pairs in the collation.properties file. It is located in the \$COLLATION_HOME/dist/etc directory on your TADDM server.

As mentioned already, TADDM can still rely on its own independent security model for both authentication and authorization.

If you want to set up a common security model for the overall CCMDB solution, then you have to configure TADDM as follows:

1. In the collation.properties file, search for the Security properties section:

```
#=====
# Security properties
#=====
com.collation.security.internal=6297357493
com.collation.security.privatetruststore=true
com.collation.security.enablesslforconsole=true
# This property turns on data level security. When set to 'true', it
# means that users will only be allowed access to CI's
# in access collections to which they have been granted access. When
# set to 'false', users can still be assigned access to
# access collections but access wont actually be checked.
com.collation.security.enabledatalevelsecurity=false
com.collation.security.enforceSSL=false
# The user management module used by this CMDB server.
# Possible values are:
# "file" for a TADDM file-based user registry
# "ldap" for an LDAP user registry
# "vmm" for a WebSphere Federated Repositories-configured user
registry
com.collation.security.usermanagementmodule=vmm
com.collation.security.auth.sessionTimeout=240
com.collation.security.auth.searchResultLimit=100
```

You need to set the `com.collation.security.usermanagementmodule=vmm` in order to define that TADDM leverages the Virtual Membership Manager to get access to the users and groups defined in LDAP. By doing this, you have to define users and groups only once in LDAP. They then get used within the process run time and the TADDM environment.

2. Once you have configured TADDM to use the VMM based security, search for a section in collation.properties called Federated Repositories:

```
#=====
# Federated Repositories/ESS
# Authentication & SSO
#=====
# FQDN of the machine hosting WebSphere,
# Federated Repositories and ESS
com.collation.security.auth.WebSphereHost=Specify the FQDN (fully
qualified domain name) of the WebSphere Application Server that is hosting
your CCMDB process environment
# WebSphere system port (default = 2809)
```

`com.collation.security.auth.WebSpherePort=`*You need to specify port 9809 is you want the TADDM server to lookup users and groups in LDAP through the VMM Remote Interface on the WebSphere Application server. The default port 2809 mentioned above is misleading.*

`com.collation.security.auth.VMMAdminUsername=`*Specify the WebSphere Admin name here, for example wasadmin*

`com.collation.security.auth.VMMAdminPassword=`*Specify the password for the WebSphere Admin here*

`com.collation.security.auth.VMMUserSearchBase=`*You do not need to specify this*

`com.collation.security.auth.VMMGroupSearchBase=`*You do not need to specify this*

`com.collation.security.auth.ESSClientTrustStore=`*This is optional and needs to be specified if you want to use SSL between the TADDM Server authentication client and the STS instance on the WebSphere Application server*

`com.collation.security.auth.ESSClientTrustPwd=`*This is optional and needs to be specified if you want to use SSL between the TADDM Server authentication client and the STS instance on the WebSphere Application server*

As you can see in the snippet of the `collation.properties` file above, we added explanations for those parameter values that you have to specify when configuring the TADDM server for the federated repository approach.

Note: You specify the password in cleartext. When the TADDM server restarts, it reads the password and writes it back to the file in an encrypted way. You have to restart the TADDM server for the changes to take effect.

This concludes our configuration to direct TADDM to use VMM to look up users and groups in the LDAP directory through the remote VMM interface.

Configuring the `ibmessclientauthncfg.properties` file

The next thing you have to configure is the communication between the authentication service client on the TADDM server to the authentication service implementation (STS) on the WebSphere Application server. TADDM uses the authentication client service in order to authenticate the users looked up through VMM and validate the token received from the process environment to support single sign-on. The STS service is installed and configured by the CCMDB installation in the WebSphere environment while the STS authentication client is installed through the TADDM installation. You have to take some configuration steps on the TADDM side in order to define the communication channel between the STS client and server services.

On the TADDM Server, search for the `ibmessclientauthncfg.properties` file. It is in the `$COLLATION_HOME/dist/etc` directory:

```
# This is the URL for the ESS Authentication Service
authnServiceURL=http://localhost:9080/TokenService/services/Trust
#authnServiceURL=https://localhost:9443/TokenService/services/Trust
```

This is the URL that the authentication service client on the TADDM server uses to call back to the Security Token Service on the WebSphere Application Server in order to authenticate the TADDM user or validate the LTPA token that TADDM receives through the Launch in Context call from the process environment. You just have to replace the `localhost` string with the FQDN of your WebSphere Server in case you did not change any security settings on the WebSphere Server. In case you use an SSL connection between the STS client and server implementation, use port 9443 instead of port 9080.

Copying orb.properties and iwsorbutil.jar files

Next, you need to copy two files, `orb.properties` and `iwsorbutil.jar`, on the TADDM server from the `$COLLATION_HOME/dist/lib/WebSphere/6.1` directory to the `$COLLATION_HOME/dist/external/jdk-1.5.0-Windows-i386/jre/lib` directory (for `orb.properties`) and the `$COLLATION_HOME/dist/external/jdk-1.5.0-Windows-i386/jre/lib/ext` directory (for `iwsorbutil.jar`).

The directory specifying the JDK™ (`jdk-1.5.0-Windows-i386`) depends on the platform that you have used to install your TADDM server.

This is done so that the WebSphere libraries (which are the same used for WebSphere discovery purposes on the TADDM server as well) that ship with TADDM will recognize `sas.client.props`, which is required to make an authenticated client connection to WebSphere.

Note: In our environment, the files have already been placed in the appropriate directory structure after installation. Nevertheless, we recommend to checking to see if they are available in the target directory structure.

Configuring the sas.client.props file

You need to configure some parameters in the sas.client.props file, located in \$COLLATION_HOME/dist/etc. Search for a section called Client Security Enablement. You need to set the following parameters:

com.ibm.CORBA.securityServerHost= *Set this to the FQDN of your WebSphere Application Server*
com.ibm.CORBA.securityServerPort= *Set this to port 9809 if you did not change anything for the STS service on WebSphere. The product documentation refers to port 2809, this is misleading.*
com.ibm.CORBA.loginUserId= *Set this to the WebSphere admin user on you WebSphere Application server, e.g. wasadmin*
com.ibm.CORBA.loginPassword= *Set this to the password of the WebSphere admin user on the WebSphere Application server*

These configuration steps are necessary in order to allow the authentication service client on the TADDM server to call out to the authentication service (STS) on the WebSphere server in order to authenticate a user and verify the LTPA token that has been passed into the TADDM environment by a launch in context call.

You will find further configuration steps in the product documentation:

- ▶ To encrypt the login password that you specified in sas.client.props
- ▶ To secure the communication channel between the authentication service client on the TADDM server and the authentication service server (STS) on the WebSphere Application server either by using SSL or by using client authentication

Both steps are optional. You can skip them if you do not want to set up a secure communication or do not care to encrypt the password in the property file (for example, being in a test environment). Otherwise, please refer to the product documentation for these additional steps.

Configuring the single sign-on domain

There is one additional step in the series of configuration steps. You have to define the single sign-on domain. This is the domain for which the LTPA token is valid. You have to configure the domain in the WebSphere Admin console (Figure 6-8 on page 121). Select **Secure administration, applications, and infrastructure**, click the **Web Security** link, and use the single-sign-on section. Set the domain name to your domain that is used for the systems of the CCMDB solution. In our environment, we specified .austin.ibm.com. Please note the period sign in front as the leading character.

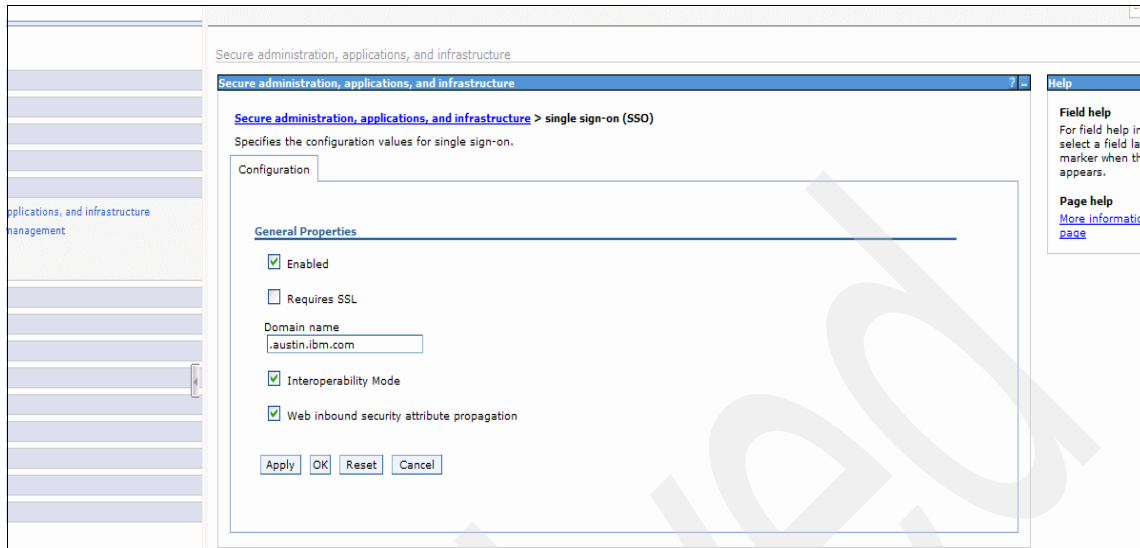


Figure 6-8 Configuring the SSO domain in the WebSphere Admin Console

Validating the configuration

If you took all of the configuration steps above, you should be able to launch in context from the process environment into a specific view of the TADDM environment. This view can be a topology view, a detailed view of a CI, or a view specifying the change history of the CI in context.

The Launch in Context application in the process environment defines specific URL strings that specify the target view inside the TADDM environment for your launch in context operation. For example, you can launch in context from the Actual CI application into one of the TADDM topology views by using the **Action** drop-down menu, as shown in Figure 6-9.

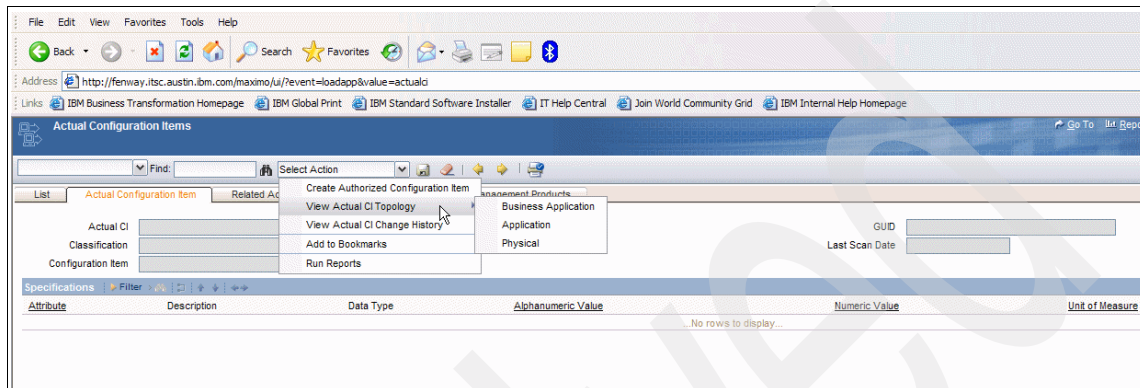


Figure 6-9 Launch in Context operation from Actual CI application

Selecting one of the action menu entries spawns a connection to the URL defined for the specific view in the Launch in Context application. For example, to connect to the Business Application View from the Actual CI application, the following URL is used:

```
http://boston.itsc.austin.ibm.com:9430/cdm/servlet/LICServlet?default.port=9433&default.server=boston.itsc.austin.ibm.com&graph=businessapplications&guid={guid}
```

Take special note of the graph type string and the GUID string. The graph type string specifies the target view to launch into, while the GUID is the identifier of the CI in the Actual CI application that gets used in the URL string.

The LTPA token is passed as part of this URL invocation to the TADDM environment.

For more details and examples of launch in context operations, please refer to Chapter 11, “Launch in Context” on page 307.

Another proof-point for your configuration setup is to log in from the TADDM product or domain manager interface. You should be able to log in into TADDM with predefined users like maxadmin or even wasadmin, as shown in Figure 6-10 on page 123.

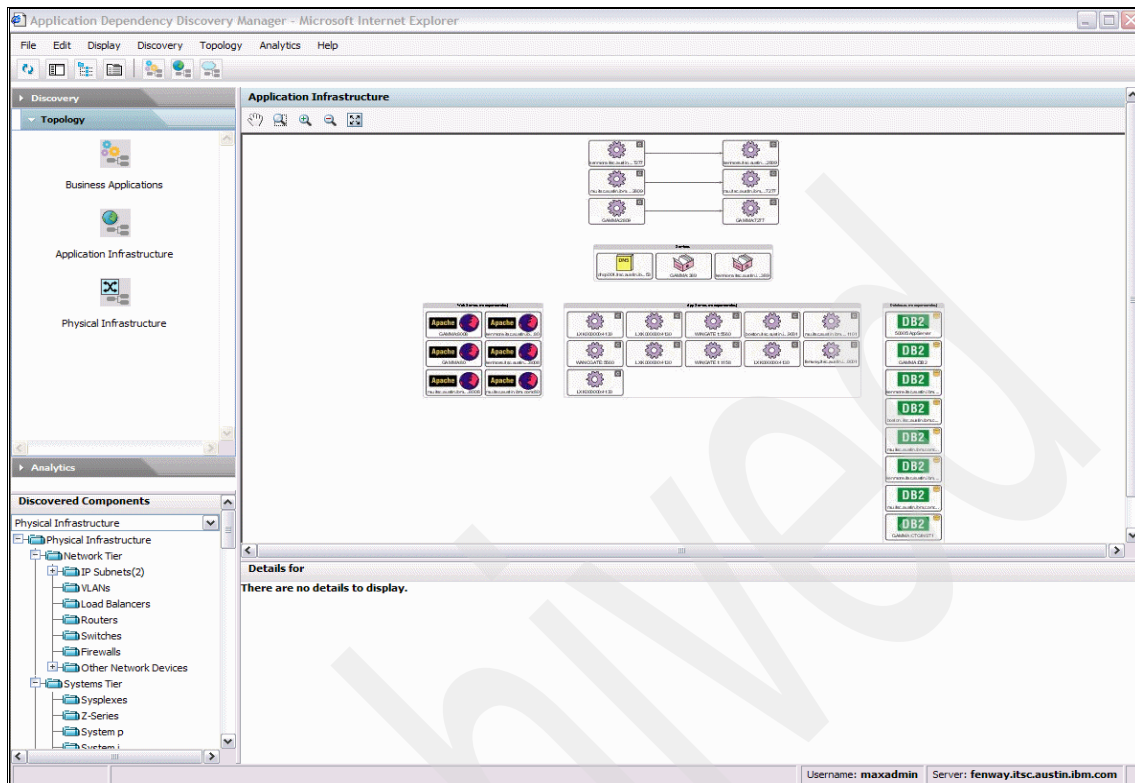


Figure 6-10 Login verification into TADDM using maxadmin

We logged into the TADDM domain manager user interface using maxadmin. The logged in user is shown in the lower right corner.

Important: You cannot log in using the administrator account as you normally do when using the file based security configuration of TADDM. You have to create the administrator UID in LDAP first before you can log into TADDM with this account.

If you are using the domain manager user interface of TADDM and go to the Users dialog in the administration section, you will recognize that the buttons for creating, editing, and deleting users are greyed out (Figure 6-11). This is because the user management should be done through LDAP. The same is true for the Groups management dialog.

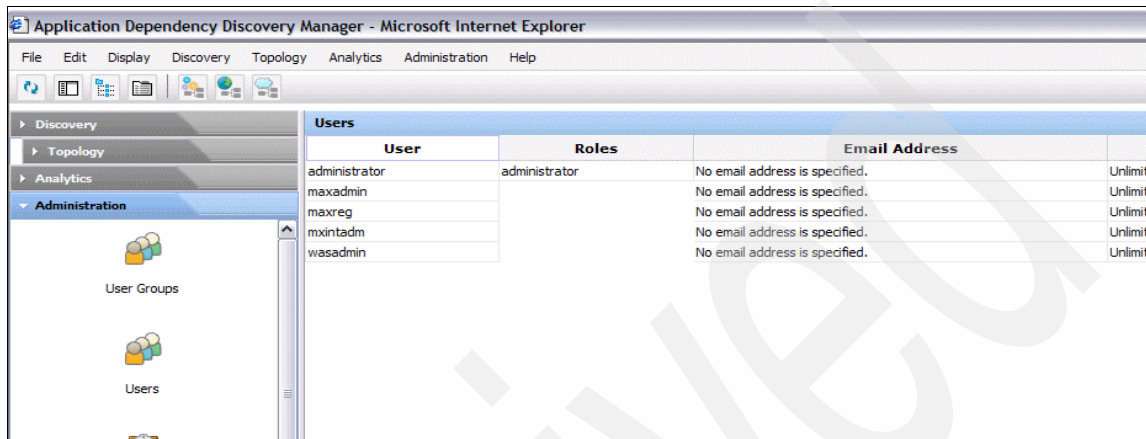


Figure 6-11 Users management dialog in TADDM Domain Manager

Important: You need to create the administrator login before you can see the Administration section inside the Domain Manager interface. This is because users like maxadmin do not have the authority to see this section. The Administrator account is predefined in the TADDM authorization policy file but needs to be created as a user first in order to log in. For further details, please refer to 6.2, “CCMDB V7.1 authorization model” on page 124.

Referring to the overall security architecture diagram in Figure 6-1 on page 106, we did not cover the topics of authorization and synchronization of access collections so far. We explain this in the following section.

6.2 CCMDB V7.1 authorization model

In the previous section, we explained the authentication model of the CCMDB solution, which heavily relies on WebSphere security components and LDAP directory server implementations. The directory server manages the users and groups, as well as their relationships.

Authorizations to CCMDB applications (for example, Configuration and Change Management applications or any other kind of application that is used to administer the process environment) are managed in the Security Groups application of the process environment. Security Groups are the place where you define what a user, who is a member of a group, can do in the system.

Note: As we have mentioned, TADDM can still rely on its own independent security model. We do not explain how to configure access collections and their assignments to users and groups in TADDM in this chapter. We focus on the common security model of the overall CCMDB solution, where you define collections and permissions to security groups in the process environment and then synchronize them to the TADDM server.

Figure 6-12 highlights the object types involved in the authorization process inside the process environment.

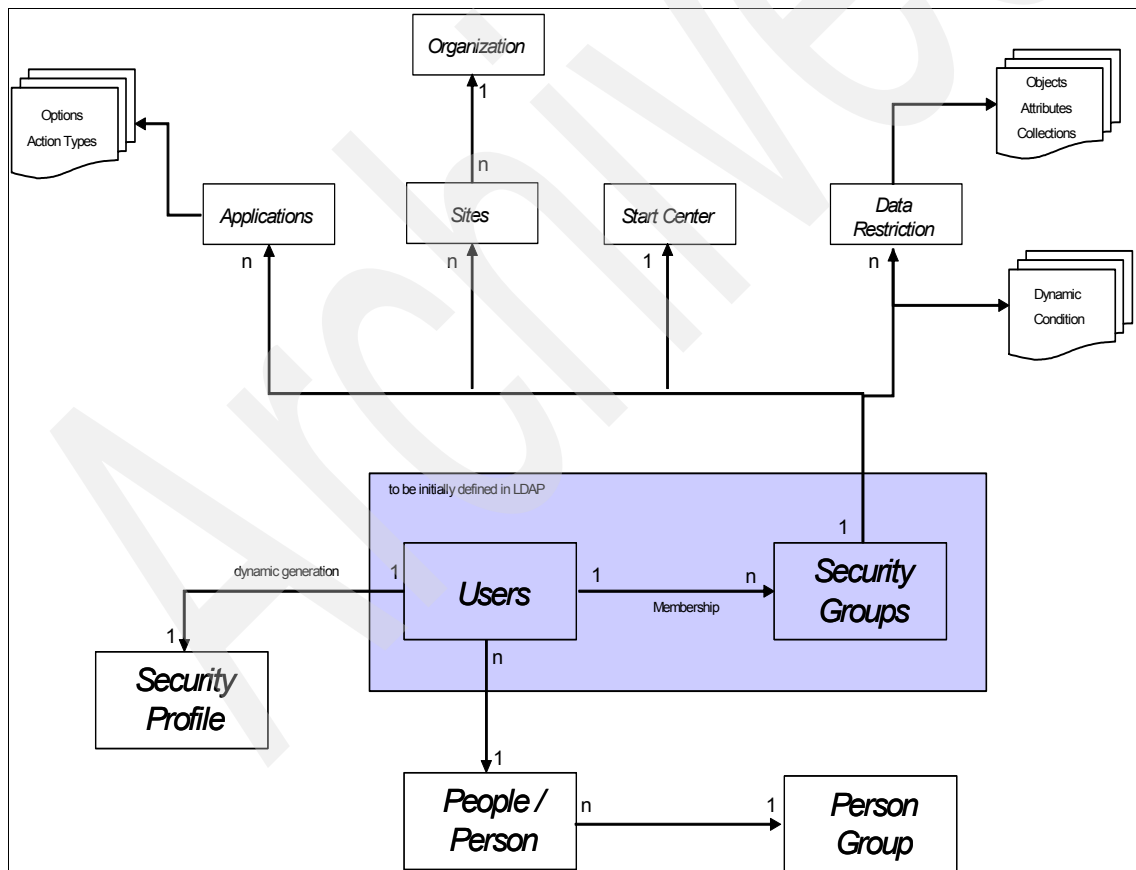


Figure 6-12 Entities relevant for authorization

Users and groups are the key entities involved in determining the authorization or security profile of a user. As mentioned, users and groups, including their relationships, are created and maintained in an LDAP Directory Server. They then are synchronized into the CCMDB process run time database with the help of the VMM cron task.

The User application is used to maintain CCMDB specific attributes of user records that are not maintained in LDAP. That means you can enrich user specification attributes in the database with additional data that has not been synchronized through the VMM cron task from the LDAP Directory server (Figure 6-13).

Figure 6-13 CCMDB Security Users application

For example, you can add all the user settings in the User Settings section. Please bear in mind that you *cannot* create a new user in the User Application. You are also not allowed to generate a new password or change the password of a user in the User application. You have to do this in the directory server.

A user is a member of one or more Security Groups. When a user tries to access an application, security checks are performed in order to see what the maximum access is based on the combination of the group memberships the user has.

The Security Groups Application is where you define permissions. Permissions are never defined for a user directly; they are always defined to a group to which a user needs to belong in order to get access to some specific objects and data.

Figure 6-14 shows the definitions for the group we created called MBGROUP.

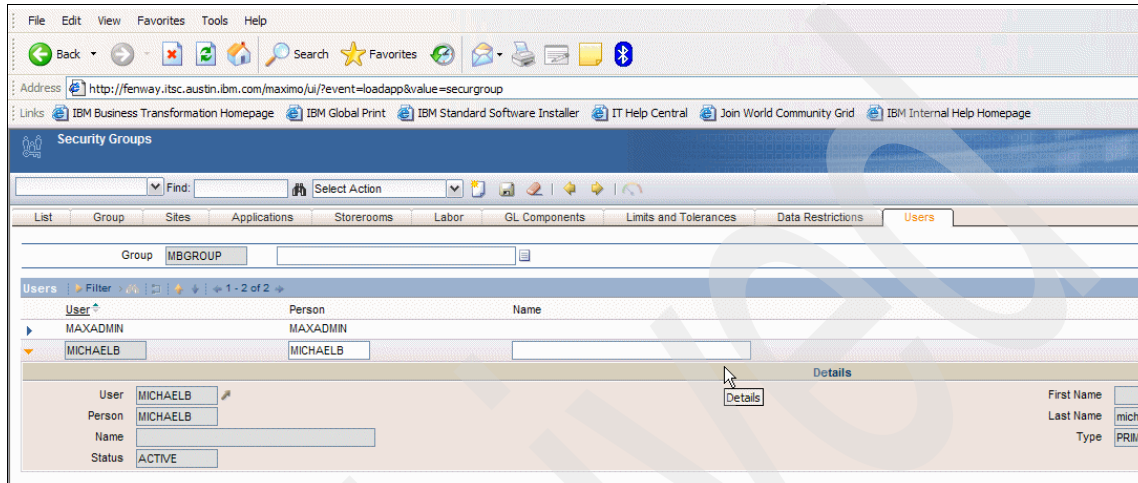


Figure 6-14 CCMDB Security Groups application

You can see several tabs that allow you to define different types of permissions. We do not explain all possibilities in detail, but just highlight the most important ones.

The Applications tab that is shown in Figure 6-14 does allow you to specify which applications in the process environment a user should be able to have access to and in which way he can access the application. You see different applications listed in Figure 6-14, such as the Actual Configuration Items application, the CI Lifecycles application, or the CI Types application. All of the applications mentioned above are actually part of the Configuration Management PMP.

In addition to the specification of which application a user can get access to, you specify which type of access a user is allowed for the specific application.

Furthermore, you can select options that are usually accessible in the menu bar or Action Menu of the appropriate application.

The Sites tab allows you to specify if a user has access to data from all Sites or a specified Site. A Site is an entity in the database that partitions the data for organizational purposes. A Site belongs to an organization. An organization could be, for example, the engineering department of a company or the Duesseldorf location in Germany.

You can specify which Start Center a user will see when he logs into the system. In our example, we defined Start Center number 6, which is a template for a Configuration Administrator role (Figure 6-15).

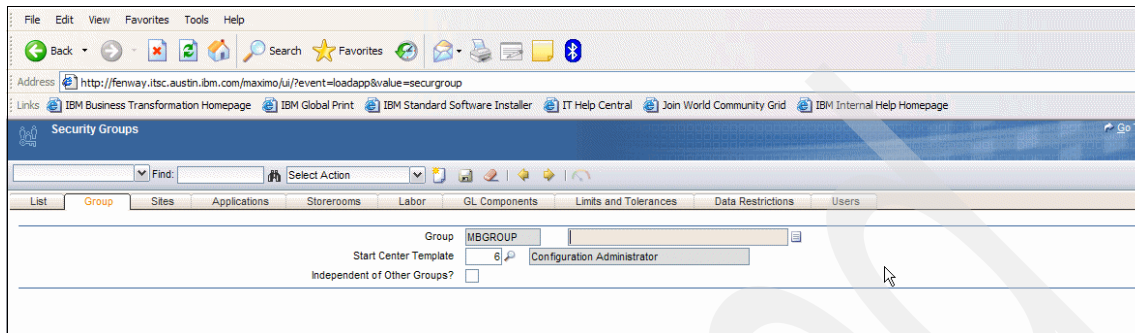


Figure 6-15 Defining Start Center of the Security Group

After you defined which applications a user has access to, you can further restrict the data that a user can access through these applications. For example, you specified that a user has access to different applications that are needed for fulfilling a role as a configuration manager. In order to further restrict the user to only use specific CIs, you can use the Data Restrictions tab in the Security Groups application (Figure 6-16).

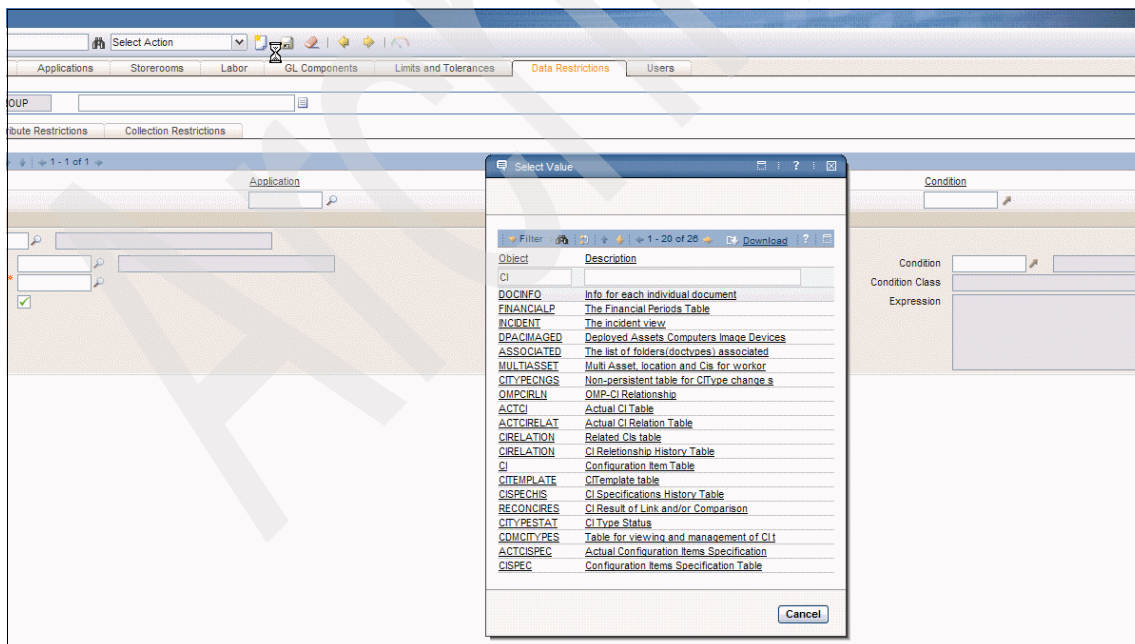


Figure 6-16 Data Restriction in Security Groups application

The Object Restrictions sub-tab is used to define which database objects (Maximo Business Objects) the user has access to. In Figure 6-16 on page 128, you can restrict the user to the CI object, which represents the (Authorized) CI table while not giving the user access to the Actual CI Table.

If you further restrict the user to specific fields of the object or database table, you can do so in the Attribute Restrictions sub-tab of the Security group application.

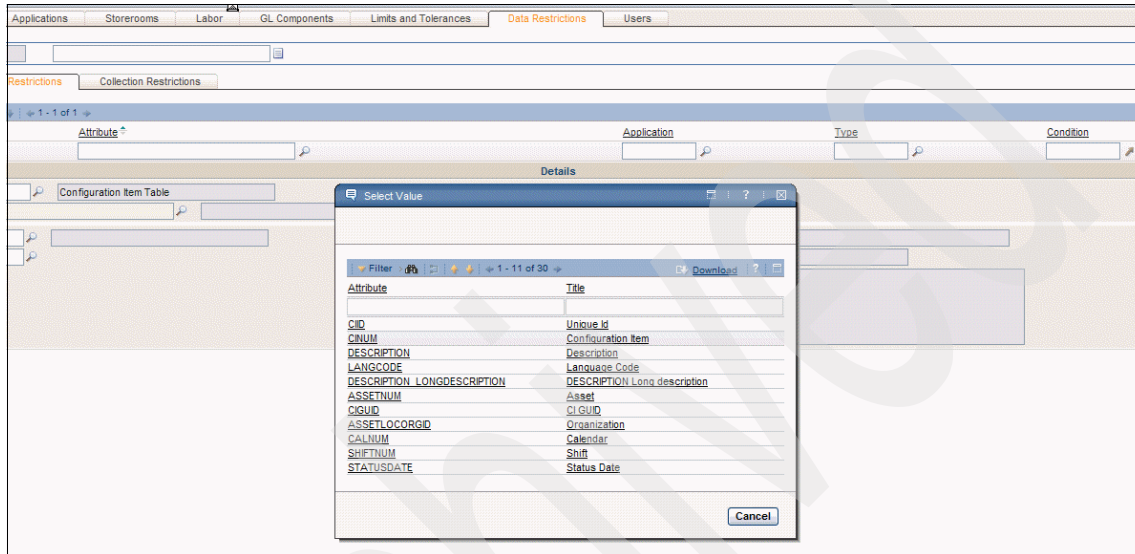


Figure 6-17 Attribute Restriction in the Security Groups application

If, for example, you want to restrict the access of the users of the group to specific collections of CIs (for example, all Linux Servers, all Windows Servers, the Storage Subsystems, or the Business Application Internet Banking), you can do so by using the Collection Restrictions sub-tab in the Security Group application (Figure 6-18). You have to define a collection before you can restrict the access to the collection in the Collections Application. If you do not specify any collection restrictions, access to all collections is granted by default.

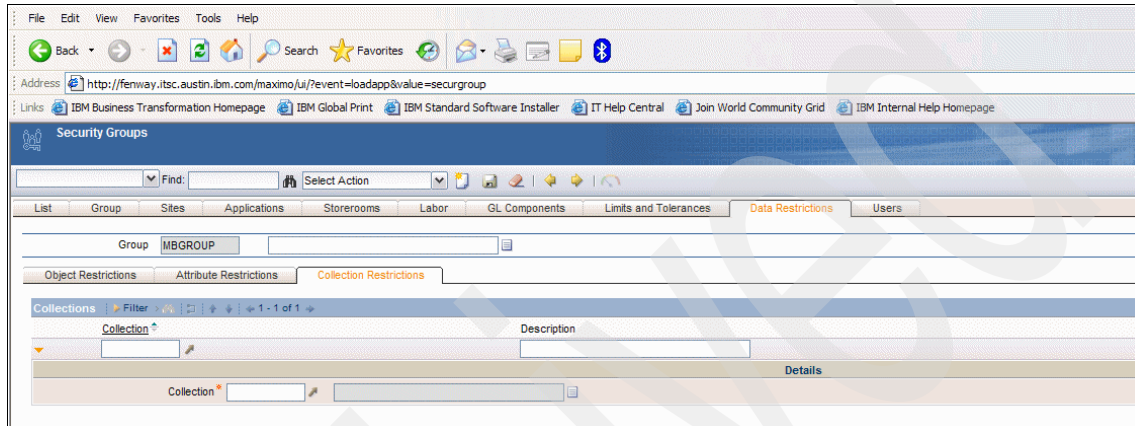


Figure 6-18 Collection Restriction in Security Groups application

As you can tell, the security model is very powerful in order to define access controls aligned with your policies and organizational structures.

Besides defining access restrictions to objects, attributes, or collections, you can even define conditional expressions that specify circumstances under which you allow access to objects, attributes, or CI collections. Conditional expressions are defined in the Conditional Expression® Manager application and are used for multiple purposes in the system. For example, they can be used for dynamically controlling the user interface behavior (to present or not present some fields to the user under specific conditions) or for permitting access to data in the system.

Conditional access can be defined at all levels of the data restrictions inside the Security Groups application. You can define them for Objects, Attributes, and CI Collections using the appropriate sub-tabs of the Data Restrictions tab.

We used the Object Restrictions sub-tab to highlight our example of conditional access definitions (Figure 6-19 on page 131).

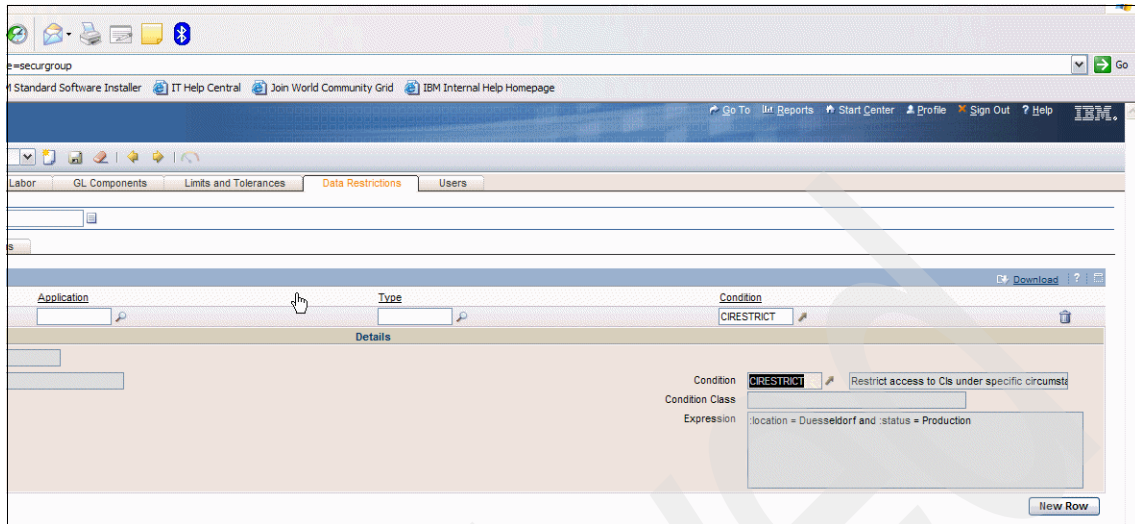


Figure 6-19 Conditional Expression definition in Security Groups application

After we have defined a condition called CIRESTRICK in the Conditional Expression Manager application, we can use this condition to define under what circumstances the user gets access to the CI object. In this example, we use a simple condition where only the user gets access to the CI object table when the location attribute of the CIs is equal to Duesseldorf and the status attribute of the CIs equal Production.

You can either use expressions, as used in Figure 6-19, or Java classes to define conditional behavior in the system.

The access a user finally experiences is defined by the combination of permissions of the groups the user belongs to. In order to see at run time which access the user has, the Security Users application provides the Security Profile tab that generates a dynamic view of access permissions, as shown in Figure 6-20.

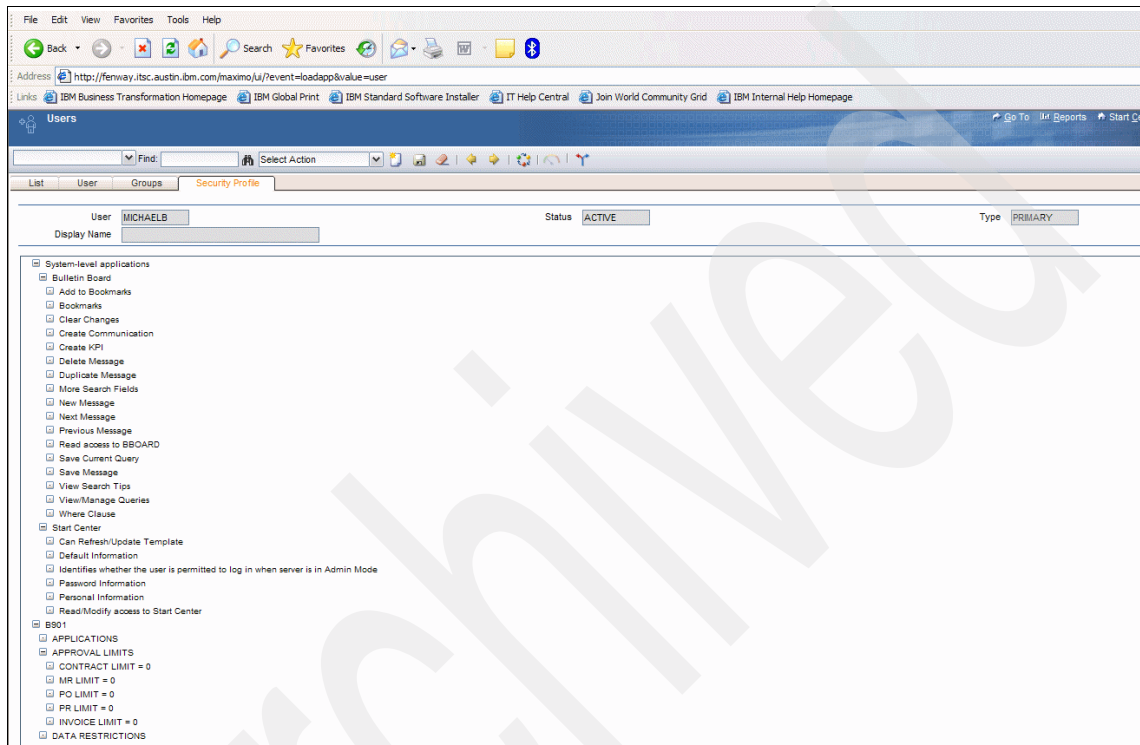


Figure 6-20 Security profile of a user in the Security Users application

You can also use the CCMDB reporting facility in order to generate a report for showing all permissions of a user. There is a predefined report called Security Group Access that lists all the access definitions for a group. Here is an example of such a report restricted to the group MBGROUP (Figure 6-21 on page 133).

Security Group: MBGROUP		Description:	
Independent?	N	Start Center:	5
Password Lasts this Number of Days:		Days Before Password Expires to Warn User:	
Sites			
Site	Description	Organization	Active?
ALL SITES	All Sites		
Applications			
Application	Option	Restriction	
Bulletin Board	Add to Bookmarks		
Bulletin Board	Clear Changes		
Bulletin Board	Create Communication		
Bulletin Board	Create KPI		
Bulletin Board	Delete Message		
Bulletin Board	Duplicate Message		
Bulletin Board	New Message		
Bulletin Board	Next Message		
Bulletin Board	Previous Message		
Bulletin Board	Read access to BBOARD		
Bulletin Board	Save Message		
Bulletin Board	Bookmarks		
Bulletin Board	More Search Fields		
Bulletin Board	Save Current Query		
Bulletin Board	View Search Tips		
Bulletin Board	View Manage Queries		
Bulletin Board	Where Clause		
Start Center	Identifies whether the user is permitted to log in when server is in Admin Mode		
Start Center	Default Information		
Start Center	Password Information		

Figure 6-21 Security Group Access report

We have explained in some detail what you can do in order to restrict access to objects and data in this section. Very often, the question of how to restrict access to specific CIs is a common question in actual deployments.

We also want to highlight that the PMPs of the CCMDB, Change and Configuration Management, provide predefined security groups. Figure 6-22 shows the predefined security groups after an initial CCMDB installation.

We limited the search filter by using the string PM to search for all groups with PM embedded.

Group	Description	Independent of Other Groups?	Authorize Group for All Sites?	Authorize Group for All Storerooms?
PM		☐	☐	☐
PMCFGADM		☒	☒	☐
PMCFGAUD		☒	☒	☐
PMCFGCLR		☒	☒	☐
PMCFGMGR		☒	☒	☐
PMCHANGEAD		☐	☒	☐
PMCHANGEAN		☐	☒	☐
PMCHANGEAP		☐	☒	☐
PMCHANGEEM		☐	☒	☐
PMCHANGEEMG		☐	☒	☐
PMCHANGEOW		☐	☒	☐
PMREQUEST		☐	☒	☐

Figure 6-22 Predefined Security Groups of Change and Configuration Management PMPs

These groups are defined in the LDAP directory server at installation time.

In addition to all the entities we described so far in this chapter, Figure 6-1 on page 106 includes further entities like Person / People and Person Groups.

You can define and modify these entities in the People and Person Groups applications. Both of them are located inside the Resources module, which is located inside the Administration module.

Although these entities are not directly involved in security aspects, they have a relationship to those entities that are involved in security aspects.

A person record (created and modified in the People application) captures personal information about a human entity. It is the primary record holding information about an individual. The record is independent of other records. This means that a person record can exist without a user record. An example of this would be a person who has reported an incident and is shown in an “affected by” field, but there is not a user record for this individual because he does not necessarily have to log into the system.

A user record always requires a person record. If you create a user in the LDAP directory server, it gets synchronized into the process database and shows up in the Security Users application. A person record is created automatically for you as well. Multiple users can be assigned to the same person record. You have to change the Person Record entry of a user in the Security Users application.

So what is the purpose of a Person record in addition to a User record? Figure 6-23 shows the person record MICHAELB that automatically has been created for the user michaelb after we created the user in the directory server. You can see that the person record allows you to specify the times of availability of a person, for example, to accommodate shift times.

The screenshot shows the 'Modify Person Availability' dialog box in the Security Users application. The dialog box is titled 'Modify Person Availability' and contains the following information:

- Person:** MICHAELB
- Calendar:** [Empty field]
- Display From:** 10/15/07
- Work Dates:** A table showing the availability of the person for the week of 10/15/07 to 10/21/07.

Work Date	Day	Start Time	End Time	Work Hours Reason Code	Description
10/15/07	Monday				
10/16/07	Tuesday				
10/17/07	Wednesday				
10/18/07	Thursday				
10/19/07	Friday				
10/20/07	Saturday				
10/21/07	Sunday				

The background shows the main form for the person record with fields for address, status, and calendar organization. The status is set to 'ACTIVE'.

Figure 6-23 Person record

A Person Group is a group of people (or Person records) who fulfill the same job role. A Person Group can be used in assigning work from within a Job Plan Template to a group rather than to an individual person. You can define alternate people for each primary member in the group for notifications and task assignments. Each primary member can have one or more alternate persons to contact in case they are not available.

Synchronization of permissions to TADDM access collections

The final thing that we explain in this section is the synchronization of CI collections and permissions that you have defined for these collections in the process run time environment into the TADDM environment so that they can be reused in the TADDM environment. We refer to this process as *synchronizing access collections*.

This allows you to define collections of configurations items in the Collection application, and then use them inside the Security Groups application to define permissions for this collection and finally synchronize them to TADDM. TADDM creates them as access collections inside its own security facility and defines the permissions you have synchronized in order to locally verify access to those access collections. They are created using the operator role with READ permissions.

So when a user tries to display a topology view in TADDM, the TADDM security verifies if the user is allowed to access the CIs based on the permissions defined. The TADDM security facility is sometimes also referred to as ARGUS.

Note: Collections of CIs are defined in the process environment for authorized configuration items. In order to synchronize the collections into the TADDM environment, the authorized CIs have to have a link (relationship) to the appropriate actual configuration items. Synchronizing collections of actual CIs would not be of much purpose since actual CIs are transferred from the TADDM environment through the Integration Composer.

The following sequence of figures explains the steps that are prerequisites in order to synchronize CI collections into the TADDM environment from an operational perspective.

Make sure you have brought over CIs from TADDM into the Actual CI space using the ITIC adapter. The CIs show up in the Actual Configuration Items application (Figure 6-24 on page 137).

Actual Configuration Items

Find: Select Action

List **Actual Configuration Item** Related Actual Configuration Items Operational Management Products

Actual CI: GUID:

Classification: Last Scan Date:

Configuration Item:

Specifications 1 - 9 of 9

Attribute	Description	Data Type	Alphanumeric Value	Numeric Value
COMPUTERSYSTEM_FQDN	COMPUTERSYSTEM_FQDN	ALN	GRUENE	0.00
COMPUTERSYSTEM_TYPE	COMPUTERSYSTEM_TYPE	ALN	IpDevice	0.00
MODELOBJECT_LASTMODIFIEDBY	MODELOBJECT_LASTMODIFIEDBY	ALN	system	0.00
MODELOBJECT_LASTMODIFIEDTIME	MODELOBJECT_LASTMODIFIEDTIME	NUMERIC		1,193,101
COMPUTERSYSTEM_SIGNATURE	COMPUTERSYSTEM_SIGNATURE	ALN	9.3.4.193	0.00
MODELOBJECT_CONTEXTIP	MODELOBJECT_CONTEXTIP	ALN	9.3.4.193	0.00
MODELOBJECT_BIDFLAG	MODELOBJECT_BIDFLAG	NUMERIC		3.00
MODELOBJECT_CREATEDBY	MODELOBJECT_CREATEDBY	ALN	system	0.00
MODELOBJECT_DISPLAYNAME	MODELOBJECT_DISPLAYNAME	ALN	GRUENE	0.00

Figure 6-24 Actual Configuration Items application

It is important that the GUID attribute, which you can find in the upper right corner of the dialog, is transferred from TADDM. This is the unique identifier of the CI in the overall CCMDB solution.

In order for a CI collection to be synchronized into the TADDM environment, at least one CI that has been promoted from the Actual to the Authorized CI space has to be in the collection (Figure 6-25). This is because TADDM needs to be aware of the GUID of the CI when the access collection gets synchronized.

The screenshot displays the 'Configuration Items' application interface. At the top, there's a navigation bar with links like 'Go To', 'Reports', 'Start Center', and 'Profile'. Below this is a search bar with 'Find:' and 'Select Action' buttons. The main content area is divided into two sections: 'Configuration Item' and 'Related CIs'. The 'Configuration Item' section shows details for 'GRUENE-1492', including its classification 'TOPICCLASS \ SYS.COMPUTERSYSTEM', class description 'SYS.COMPUTERSYSTEM', and various attributes like Asset, Location, Item, Service, and Service Group. The 'Related CIs' section shows a list of related items. On the right side, there's a 'Attachments' section with fields for Status (NOT READY), CI Location, Site, Organization, Item Set, Calendar, Shift, CI Owner, Change Window, and RFC Number. Below these sections is a 'Specifications' table with columns for Attribute, Description, Data Type, Alphanumeric Value, Numeric Value, and Unit of Measure. The table lists various attributes related to the computer system, such as ARCHITECTURE, ASSETID, ASSETTAG, AUTOSTART, CPDEVICEID, CATEGORY, CONFIGLASTUPDATE, CPUCORESENABLED, CPUCORESINSTALLED, and CPUCORESENABLED.

Attribute	Description	Data Type	Alphanumeric Value	Numeric Value	Unit of Measure
COMPUTERSYSTEM_ARCHITECTURE	COMPUTERSYSTEM_ARCHITECTURE	ALN			
COMPUTERSYSTEM_ASSETID	COMPUTERSYSTEM_ASSETID	ALN			
COMPUTERSYSTEM_ASSETTAG	COMPUTERSYSTEM_ASSETTAG	ALN			
COMPUTERSYSTEM_AUTOSTART	COMPUTERSYSTEM_AUTOSTART	NUMERIC			
COMPUTERSYSTEM_CPDEVICEID	COMPUTERSYSTEM_CPDEVICEID	ALN			
COMPUTERSYSTEM_CATEGORY	COMPUTERSYSTEM_CATEGORY	ALN			
COMPUTERSYSTEM_CONFIGLASTUPDATE	COMPUTERSYSTEM_CONFIGLASTUPDATE	ALN			
COMPUTERSYSTEM_CPUCORESENABLED	COMPUTERSYSTEM_CPUCORESENABLED	NUMERIC			
COMPUTERSYSTEM_CPUCORESINSTALLED	COMPUTERSYSTEM_CPUCORESINSTALLED	NUMERIC			
COMPUTERSYSTEM_CPUCORESENABLED	COMPUTERSYSTEM_CPUCORESENABLED	NUMERIC			

Figure 6-25 Promoted Configuration Item in the Configuration Items application

You then have to define a Collection in the Collections application and make the CIs that you want to refer to members of the collection (Figure 6-26 on page 139). Please note that Assets can also be members of a collection; they are not synchronized though.

Go To Reports Start Center Profile

Find: Select Action

List Collection

Collection: COLLTOSY Attachments

Active? ☒

Collection Details Filter 1-3 of 3

ID	Asset	Location	C/Asset/Location Description	Site
GRUENE-1492				
ITSONT00.ITSC.AUSTN.IBM.COM-1161				
KLKZZ7C-1362				

Figure 6-26 Collection Definition in the Collections application

The final prerequisite step allows a security group access to the collection. You configure this in the Security Groups application, as shown in Figure 6-27.

Go To Reports Start Center Profile Sign Out

Find: Select Action

List Group Sites Applications Storerooms Labor GL Components Limits and Tolerances Data Restrictions Users

Group: PMCFGMR

Object Restrictions Attribute Restrictions Collection Restrictions

Collections Filter 1-1 of 1

Collection	Description
COLLTOSY	itsocollection

Figure 6-27 Permitting Access to Collection in Security Groups application

The synchronization leverages the Maximo Enterprise Adapter (MEA) integration technology. Please refer to Chapter 7, “Integration technologies” on page 145 for more details on MEA. Once collections are defined and added to a security group, they get synchronized into the TADDM environment. This is a synchronous process.

The synchronization is not enabled by default. To enable the synchronization, you have to perform some configuration steps in the process environment of the CCMDB.

Inside the Integration module, you can find the Publish Channels application. Inside this application, we use the string coll for searching for the channels with respect to synchronizing collections and their authorizations (Figure 6-28).

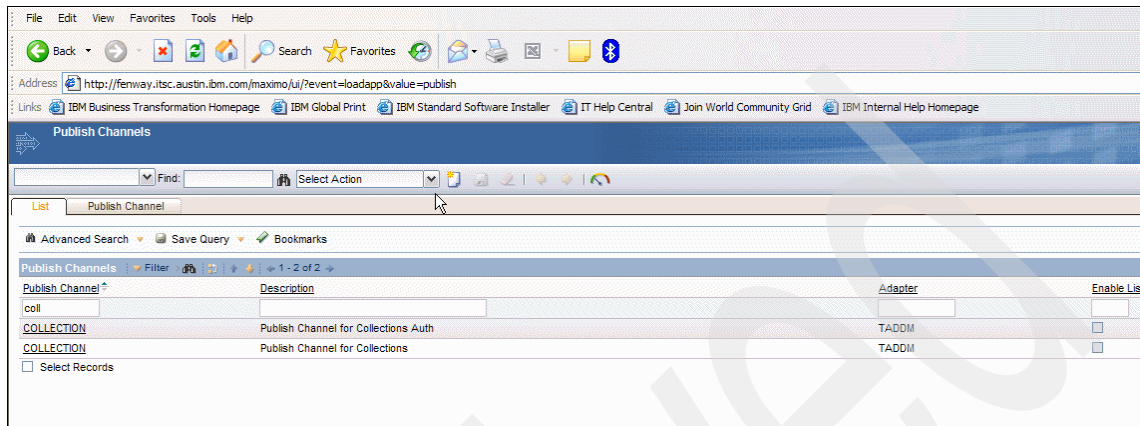


Figure 6-28 Publish Channels application

You have to enable the event listener for each of the two records (channels for collection and collection authorization synchronization). You can do so either by selecting the appropriate action from the Action drop-down menu or you can check the Enable Listener field (Figure 6-29).

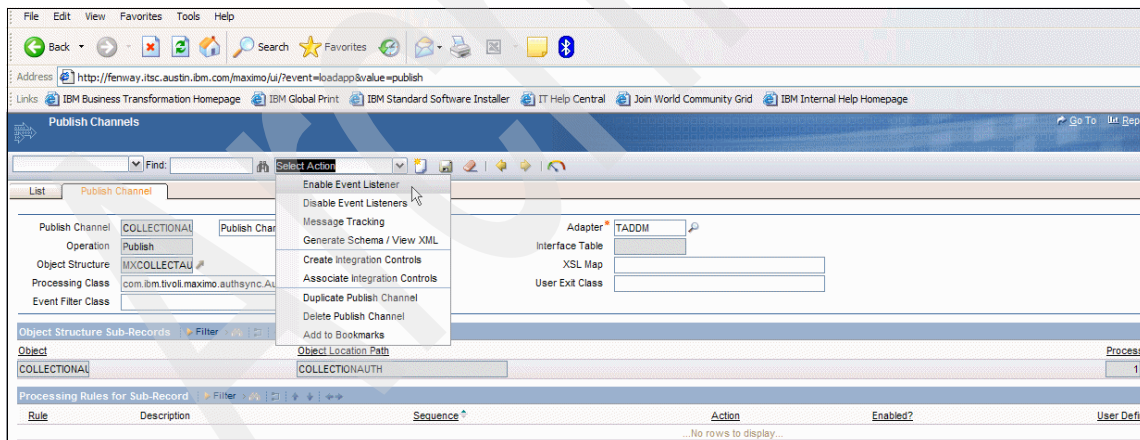


Figure 6-29 Enabling the publish channels

Do not forget to save the record.

The next thing you have to do is to define the properties for the synchronization target, also called the endpoint. You configure this in the Endpoints application

inside the Integration module. In the Endpoints application, go to the TADDMEP definition, as shown in Figure 6-30.

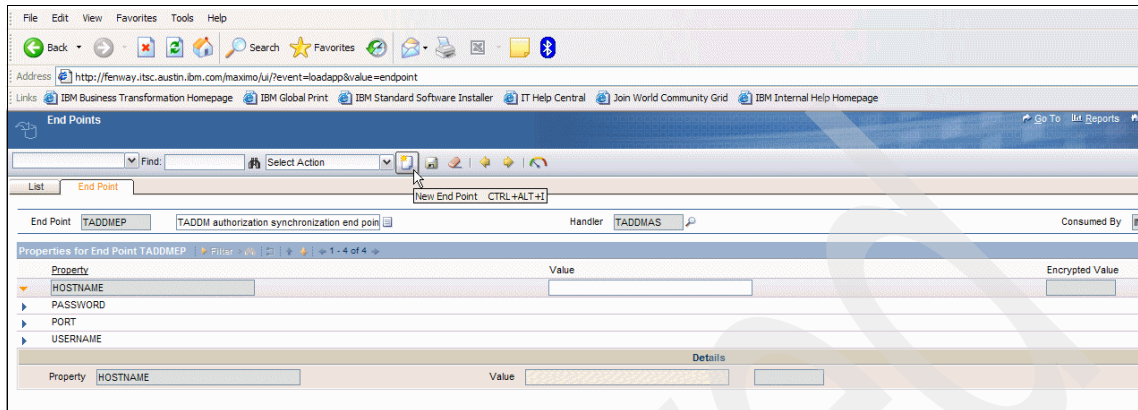


Figure 6-30 Endpoint definition

In the window above, you have to specify the parameters for your TADDM server host name (FQDN), the user name (wasadmin) and password, and the port that the integration code is connecting to in the TADDM environment. Use port 9530 (or 9531 if you are using SSL) if you did not make any changes in the TADDM environment. Do not forget to save the endpoint definition record.

The final configuration step you have to take is use the External Systems application inside the Integration module and activate the record labeled TADDMES. In this record, the parameters (host name, user, and so on) are set to the values that you previously have defined in the Endpoints application.

On the Publish Channels sub-tab of the External Systems application, you have to enable both entries, that is, the entry for the collections (COLLECTIONPC) and the entry for the collections authorization (COLLECTIONAUTHPC), as shown in Figure 6-31.

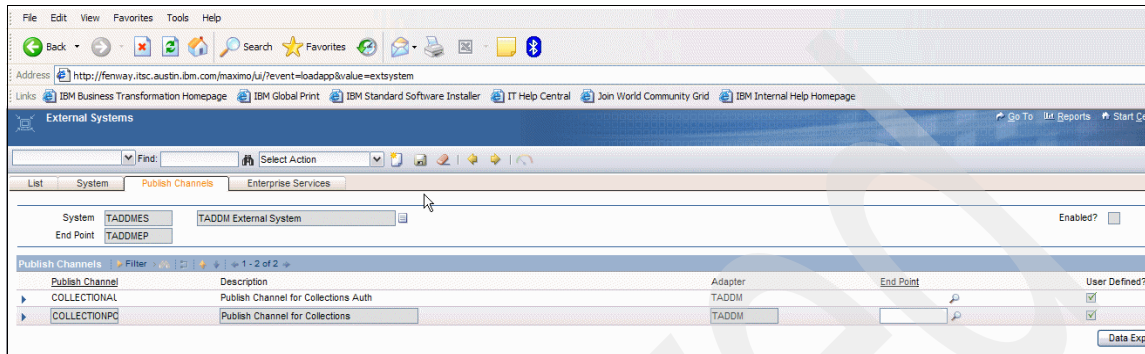


Figure 6-31 External System definition

Do not forget to save the record.

If you configured all the steps correctly, the collection you have defined in the Collections application and added to a Security group in the process environment should show up in the TADDM domain manager administration section. Figure 6-32 on page 143 shows that the collection we defined, COLLTOSY, for the PMCFGMR security, now shows up in the TADDM User Group dialog.

If you followed and completed the configurations steps to set up the common security model by configuring VMM, STS, TADDM, and the Publish Channels, the major steps that you have (some are optional) to take while operating the solution in a production environment after the initial configuration setup are the following:

- ▶ Add users to the LDAP Directory Server using the appropriate administration utility (or use predefined users).
- ▶ Add groups to the LDAP Directory Server using the appropriate utility (or use predefined groups).
- ▶ Define user to group membership in the LDAP Directory Server using the appropriate utility (or use predefined memberships).
- ▶ Wait for the VMM cron task to synchronize the definitions into the process run time database. The default schedule is to wait for five minutes.
- ▶ Define collections for groups of configuration items in the collections application (optional step).
- ▶ Define permissions to groups using the Security Groups application. Optionally, you can restrict the access to the collection of configuration items you have defined in the previous step.
- ▶ Wait for synchronization of collections and permissions to happen in the background into the TADDM environment.
- ▶ Work inside the process environment and launch in context into TADDM to verify the single sign-on.

Integration technologies

As has already been discussed, there are different manifestations of CI data in the CCMDB.

Discovered CIs are populated within TADDM and provide a view of the discovered IT environment. The majority of the data is discovered by leveraging the built-in sensor technology of the TADDM discovery component. CI data from the discovered CI data space is bridged in a filtered way into the Actual CI data space and promoted with additional filter settings into the Authorized CI data space. The Authorized CI data space is what the various process management products work with.

Discovering CIs and forwarding the required data into the Actual and Authorized CI spaces is the backbone of how CI data is collected, maintained, and used. Nevertheless, there is no green field in any environment. There are existing data sources keeping discovered data, there are systems management solutions that keep data to consider, there are data sources that need to be federated, external systems that need to consume or provide data from or to the CCMDB in a synchronous, asynchronous, or batch oriented way, or there is a need to launch in context from a CCMDB application to an external system. These are just some examples that require the CCMDB appropriate interface technologies. The CCMDB, once it gets introduced into an IT environment, is more than just another database to hold some data. It is an anchor point of the IT environment.

The IBM CCMDB V7.1 solution provides various interface technologies that can be used to exchange data in different ways for different purposes. This section provides an overview of the various interface technologies and their operational areas within the solution scope.

Figure 7-1 provides an overview of the various interface technologies related to the data layer of the CCMDB solution.

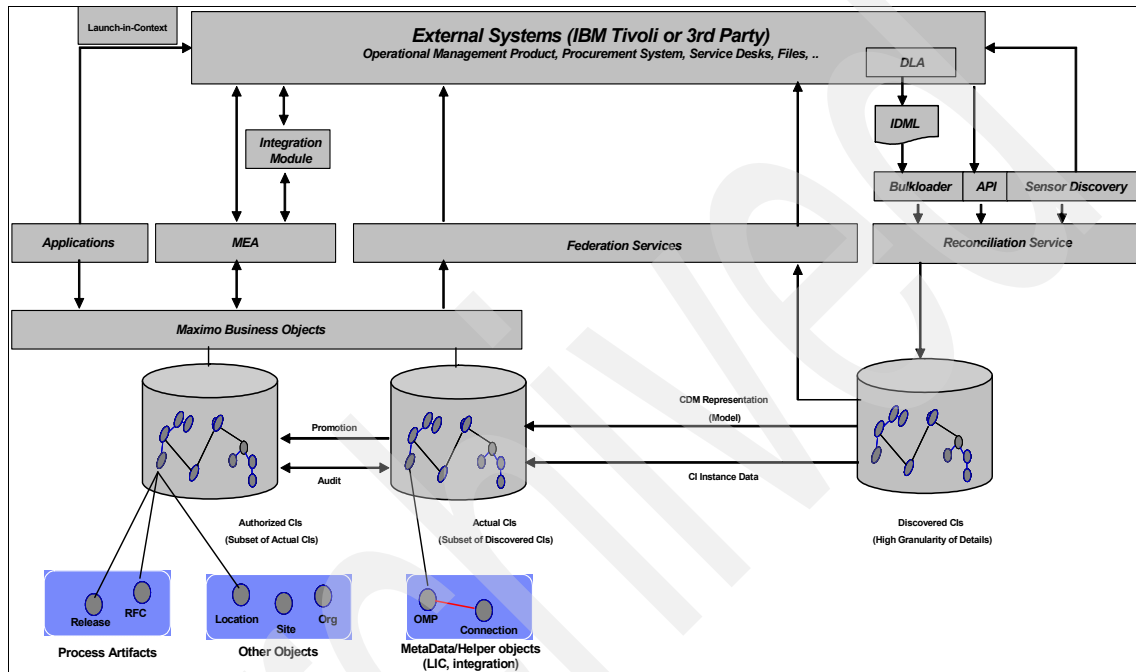


Figure 7-1 Interface technology overview

In Figure 7-1, the reconciliation service is not an interface technology on its own. Reconciliation is a process that tries to avoid duplicates of CI instances. Usually there are multiple data sources that need to be combined in order to provide a complete set of data for a CI. Discovery sensor technology, batch imports, or manual data entries have to be consolidated in order to get a unified data representation that reflects the real IT environment. Data for CI “A” that gets discovered through sensor technology and data for CI “A” imported from an external system using DLA technology need to be combined. Otherwise, there are at least two different representations of CI “A” in the CCMDB.

The CCMDB reconciliation logic is built on naming rules. Each CI type is identified and classified by one or more sets of attributes that need to exist in order to instantiate a CI of this class type. If a CI instance already exists in the CCMDB, a subsequent data import, regardless if the data is imported by sensor,

batch, or manual techniques, compares the new data based on the naming rules as well as the attribute content in order to verify if a new CI instance needs to be created or if an existing set of data needs to be refreshed. In CCMDB V7.1, the reconciliation logic to determine what needs to happen while importing data can be influenced by writing reconciliation plug-ins.

The reconciliation service is within the discovery component of the CCMDB solution. In other words, if there is more than one source of data for a CI, you have to feed the data into the CCMDB through the discovery environment using one of the available technologies (sensor, DLA batch import, API, or manual data entry).

Important: The reconciliation terminology within the CCMDB solution is frequently used within two different contexts. Besides its primary intention to avoid duplicates while importing CI data, the comparison of Actual and Authorized CIs is often referred to as reconciliation as well. The latter should be referred to as Auditing rather than reconciliation. We use the term reconciliation when referring to the process of avoiding duplication of CI data. Technically speaking, the reconciliation engine is within the discovery environment of the CCMDB V7.1 solution.

In this section, we give an overview of the following major integration technologies provided by the overall CCMDB solution:

- ▶ Discovery Sensor
- ▶ Discovery Library Adapter / IdML Files
- ▶ TADDM Application Programming Interface
- ▶ Federation Services
- ▶ Maximo Enterprise Adapter Integration Framework
- ▶ Integration Modules and Logical Management Operations
- ▶ Launch in Context

We explain what their primary domain of usage is, provide examples, and highlight which tooling supports the usage of the respective technology. Please refer to Figure 7-1 on page 146 while reading the following sections.

7.1 Discovery Sensor

The CCMDB discovery environment provides a high number of default sensors to support the discovery of various IT technologies. One of the enhancements in CCMDB V7.1 is the ability to develop pluggable sensors independent from a product development build. This gives you the prerequisites to the capability to extend the CDM class model, add new attributes, and define naming rules for the new class types that are inherited from existing default classes.

Developing new sensors during the V7.1 time frame is reserved to IBM Development and Services. The Eclipse based development environment is not made publicly available.

7.2 Discovery Library adapter and IdML files

While discovery sensors provide a way to collect detailed CI data from a target environment in a synchronous manner when executing a discovery run, the Discovery Library provides a way to share information between an external system and the CCMDB in an asynchronous way.

Given an external system that maintains its own data that you want to physically synchronize into the CCMDB, the Discovery Library facilitates a way to exchange resource and relationship data between the systems.

The Discovery Library includes a specification that prescribes the format of the data to exchange. This XML based schema definition is aligned with the Common Data Model of the CCMDB. The XML schema specification is called Identity Markup Language (IdML). In other words, an XML file that adheres to the IdML specification can be imported into the discovered CI data space of the CCMDB solution. The import is facilitated by the CCMDB bulkloader (a program called *loadidml*) and honors the reconciliation logic while batch importing the data in order to guarantee uniqueness of CI data.

Usually only a subset of data is extracted from the external system and written to the IdML file. The program code that extracts the data from the external system and writes it to an IdML file is referred to as a Discovery Library Adapter (DLA).

In order to support the creation of a DLA, a toolkit and programming environment are provided. Various operations to extract the data from the source and write it to the IdML file (also referred to as a DLA book) are provided within an API by the programming environment. The IdML book file is written to a directory called the Discovery Library File Store (DLFS) from where it is picked up in order to bulkload the XML file into the Discovered CI space. The process of importing an

IdML book is also referred to as “reading a book”, while the process of creating an IdML file is referred to as “writing a book”. The CCMDB solution also provides a way to export data from the Discovered CI data space in an IdML format in case there is a consumer to read the exported book. An example is the interaction between the CCMDB and the Tivoli Business Service Manager V4.1 (TBSM V4.1). TBSM provides a Discovery Library Reader to import discovered data into its own data stores to support the creation of service models.

The Discovery Library facility is one of the primary ways to exchange data between various IBM Tivoli Systems Management products and the CCMDB. There are also various DLAs available from Business Partners. Examples of available DLAs for IBM Tivoli Systems Management products are IBM Tivoli Monitoring, Tivoli Configuration Manager, Tivoli Provisioning Manager, Tivoli Storage Manager, Tivoli Storage Productivity Center, or Tivoli Composite Application Manager for RTT.

Please refer to the Open Process Automation Library Web Site for the latest list of publicly available DLAs, found at:

<http://catalog.lotus.com/wps/portal/tpm>

In essence, if you have data in an external system that needs to be reconciled with existing data in the CCMDB (which means your data does not provide a single source for this information), if you want to leverage capabilities like change history, or if you want your CI data visualized within the topology viewer or you need your external data to be transferred to the Actual CI data space in order to allow an audit of the actual to the authorized data space, then you should consider importing your external data using the Discovery Library Adapter integration technology.

7.3 TADDM application programming interface

The CCMDB V7.1 TADDM discovery environment provides an API to read, export, update, and import data from the Discovered CI data space. Importing and exporting data is based on an internal XML file format, which is different from the IdML format discussed in 7.2, “Discovery Library adapter and IdML files” on page 148.

You would usually use the API for *ad hoc* queries or changes to the CCMDB data rather than regular bulkload data exchanges with an external system.

In addition, if you are intending to write your own application to regularly communicate with the TADDM environment, you would leverage the TADDM APIs to write your own client implementation.

The TADDM Software Development Kit provides a Java API, an EJB™ API, a SOAP API, and a Command-Line Executable that can be used to implement your client implementation to interface with the discovery component of the CCMDB V7.1 solution.

Please note that the usage of the API is encapsulated by the Tivoli Directory Integrator (TDI) toolkit, which is part of the CCMDB V7.1 solution. Using the toolkit hides some of the complexity of using the API natively.

7.4 Federation services

Referring to Figure 7-1 on page 146, federation technologies can be used within the discovery as well as the process environment of the CCMDB V7.1 solution. The technology behind the federation capabilities is the same for both use cases, but there are essential differences that are important to understand.

Federation is all about linking CI data that is kept inside the CCMDB to external data sources without physically copying the external data into the CCMDB. IBM provides federation capabilities within the DB2 subsystem in case you are federating to another DB2 or Informix® database. In case your external data source is different from DB2 or Informix, you have to install the WebSphere Federation Server component on top of the DB2 system that is keeping your CCMDB data. The WebSphere Federation Server is part of the CCMDB V7.1 package. In case your CCMDB implementation is relying on an Oracle RDBMS system, you have to acquire the appropriate federation capabilities from Oracle.

In case you want to enrich your discovered CI data within the TADDM reporting facility (Domain Manager reports), you can leverage the federation capabilities to link physically persisted CI data from within the TADDM data store to external managed data. This is the primary use case of using federation capabilities within the discovery environment.

Important: You cannot transfer CI data that you have federated within the TADDM discovery environment through the ITIC adapter into the Actual CI data space because the data is not physically persisted in the Discovered CI data space.

In case you want to leverage data from remote data sources within the process environment of the CCMDB V7.1, you can make the federated data available to existing applications like the Configuration Items application or processes like Change or Configuration Management.

Please refer to *IBM Tivoli CCMDB Implementation Recommendations*, SG24-7567 for an example of how to use federation from within the process environment. There are several things that need to be considered in case you want to federate data from an external data source within the CCMDB process environment:

- ▶ Federating data from an external data source is a read-only operation. The ownership of the remote data stays with the owner of the remote data source.
- ▶ You cannot use federated data within an Audit process as part of Configuration Management.
- ▶ You should consider the importance of federated data from a service management process perspective. Is your process execution relying on the federated data? If yes, can you rely on the network and availability of the external data source to guarantee the remote data access?
- ▶ Because federation is a way to link to external data sources rather than importing data, no reconciliation process is applied to the federated data.
- ▶ Federated data can be made available in reports using the BIRT technology as part of the CCMDB solution.
- ▶ Federation is ideal in case you have data in an external data source that is not discoverable and is intended to display additional attributes dynamically at run time within the CCMDB application environment.

You can extend each data object (MBO) or application within the CCMDB process environment with federated data. The federation capabilities are not restricted to configuration items objects. For example, if you want to enrich RFC data, you can do so the same way as extending CI data. The major steps, as outlined in more detail in *IBM Tivoli CCMDB Implementation Recommendations*, SG24-7567, are the following:

- ▶ Set up the Federation at the Data Source Layer using DB2, Oracle, or WebSphere Federation Server techniques.
- ▶ Create a new MBO representing your remote Data Source. Use the Database Configuration application to do so.
- ▶ Relate an existing MBO that represents the CCMDB application that you want to extend with additional federated data to your newly created MBO. Use the Database Configuration application to do so.
- ▶ Enhance the application that you want to extend with additional data fields pointing to federated data using the Application Designer application.

In essence, you need to think about if federation is the right concept to use compared to importing data physically into the CCMDB data layer. You also have to think about what purpose is behind your federation setup, that is, whether you

want to expose federated data within reports or expose it to Service Management processes and applications.

7.5 Maximo Enterprise Adapter Integration Framework

The Integration Framework is made up of various applications within the CCMDB environment that help to integrate the process environment with external applications. The Integration Framework is leveraging the Maximo Enterprise Adapter (MEA) technology.

The Integration Framework allows you to expose any object (MBO, which encapsulates database structures) in order to exchange data with external systems.

The integration with external systems can have the following characteristics:

- ▶ Outbound or inbound.
- ▶ Synchronous or asynchronous.
- ▶ Bidirectional or unidirectional.
- ▶ The data synchronization can be real time based on a trigger event (for example, when the attributes of an authorized CI change) or batch oriented (loading data from XML or flat files).
- ▶ A data import or export can be scheduled using the cron task facility.
- ▶ Various protocols for inbound or outbound communication are supported.

Every MBO can be exposed for external communication. You can define which of the fields of an MBO are required for the communication to the external system, either for an inbound or outbound communication.

Various protocol handlers (also referred to as endpoints) are supported:

- ▶ Reading and writing flat files. The flat file is required to use delimiters to separate the fields within the file contents.
- ▶ XML files. Each object structure can expose its XML schema. Figure 7-2 is an example of an XML schema for the MXAUTHCI (Authorized CI) object structure.

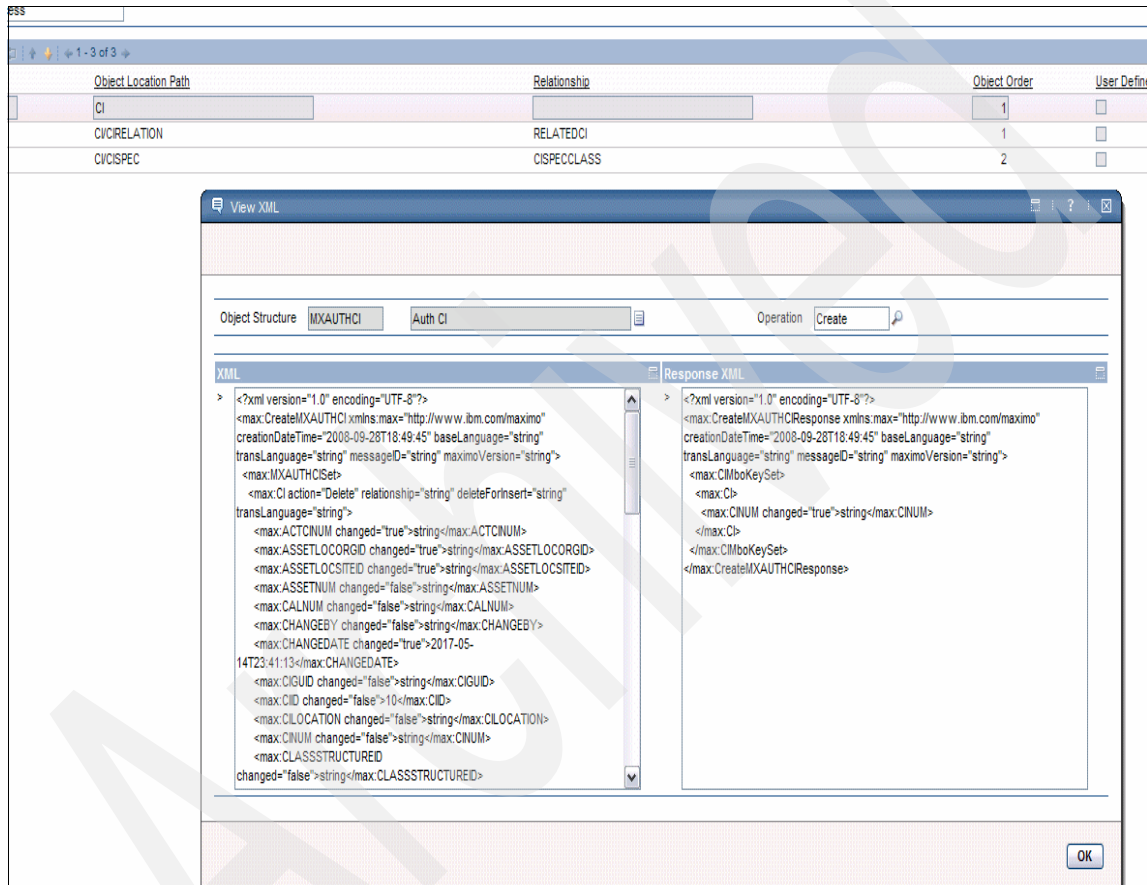


Figure 7-2 MXAUTHCI Object Structure exposed as XML

- ▶ Each object structure can be deployed as a Web Service. Web Services can be used by external applications to query or send transactions to the integration framework. Five operations are supported for each Web service that has been generated from an object structure: Create, Update, Delete, Sync, and Create. You can use the Web Services Library application to create, modify, and delete Web services (Figure 7-3). You also can generate schema and Web Service Description Language (WSDL) files for any Web service that you deploy.

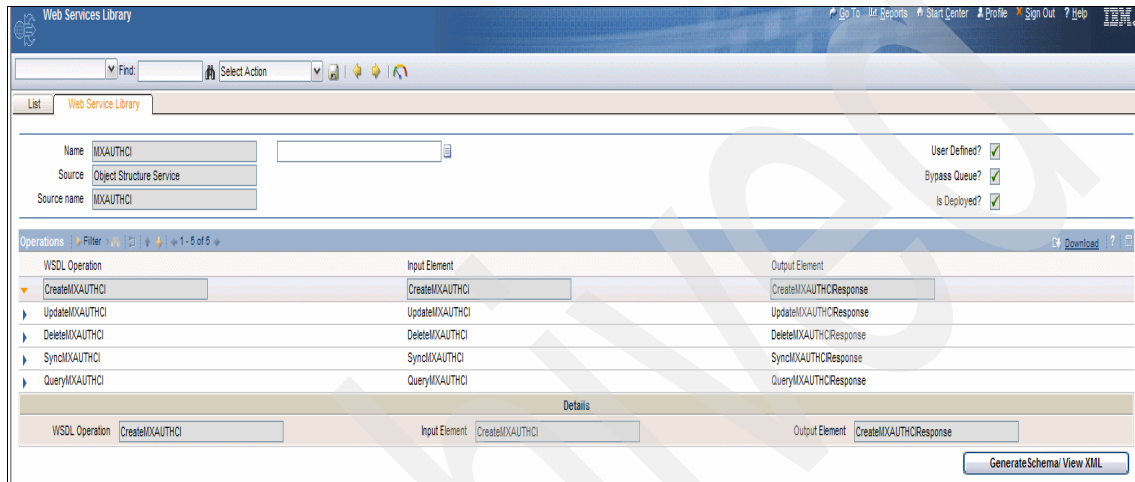


Figure 7-3 Web Services Library application

- ▶ Database Tables, also referred to as Interface tables, can be used to exchange data with external systems. This allows you to “post” outbound oriented data into an intermediate table structure or read inbound oriented data from an interface table.
- ▶ You can use the HTTP(S) protocol, an EJB API, or JMS to communicate with the CCMDB system.

You can define which Endpoint Handler to use in the context of an external system definition within the Endpoint application, as shown in Figure 7-4 on page 155.

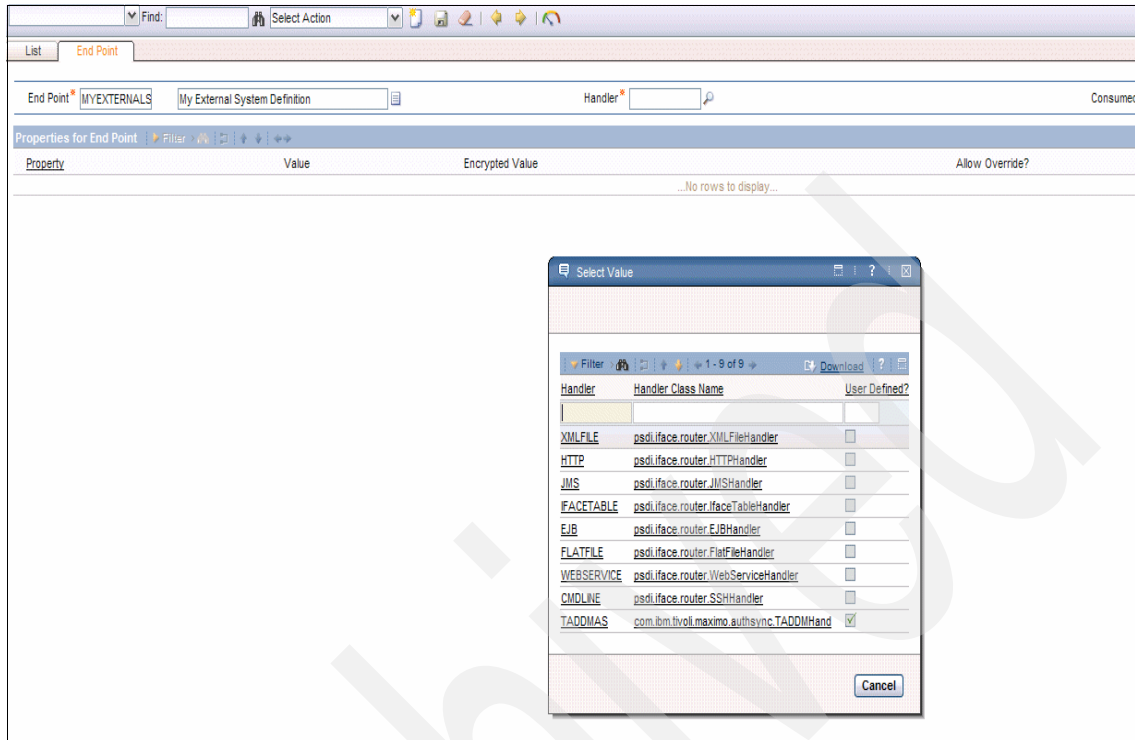


Figure 7-4 Defining Endpoint Handler in Endpoint application

The various integration technologies not only allow you to import or export data into the CCMDB database, for example, to load authorized CIs from a flat file, XML file, or interface table, but also allow you to link your CCMDB system and processes into an overall business flow. For example, you can link your change management process to an external procurement system in case the change requires the purchase of some IT equipment. While the external system is responsible for the procurement, it passes back the result to the change management system to continue its work to execute on the change.

The Integration Framework supports the transformation of data formats between the CCMDB system and the external system using either XSLT or Java.

The MEA Integration Framework is also used for third-party Service Desk Integration scenarios, for example, to interface with BMC Remedy and HP Peregrine based solutions. The MEA Integration Framework will be encapsulated by a single Maximo specific Tivoli Directory Integrator (TDI) connector in order to access any object structure. TDI connectors for Remedy and Peregrine are provided in the solution as well. Exemplary data flows are provided through TDI

assembly lines, for example, to load CI data from Remedy into the CCMDB data space.

7.6 Integration Modules and Logical Management Operations

An Integration Module (IM) is a specialized implementation of the MEA integration framework. An Integration Module is specialized code that interfaces with an external system in order to call an operation within an external system and receive a synchronous or asynchronous response. An IM acts as an adapter between CCMDB and a remote OMP server.

For example, instead of launching to an external software deployment solution to distribute or install software from within a Change Management process, the implementation of an Integration Module would allow the user to push a button or select an action from within the CCMDB application user interface that passes required input data to the external system and trigger the operation remotely without switching the user interface context.

In essence, the IM provides a mechanism for a PMP, such as Change or Release, to invoke an external operation on an Operational Management Product (OMP). The Tivoli Provisioning Manager (TPM) or the Tivoli Configuration Manager (TCM) products are examples of an OMP. You can call an OMP, such as TPM, to perform a Logical Management Operation (LMO), such as Deploy Software.

The Integration Module is a component of the integration framework. An IM has the ability to communicate with the external OMP for specific tasks or operations. The specific operations are referred to as Logical Management Operations (LMOs). An LMO defines an abstract action, such as Deploy Software, which a PMP can execute by calling an IM. Integration Modules provide a method for calling an OMP to execute an implementation of an LMO. For example, IM “A” for Software Distribution solution “A” would provide a different implementation of the LMO “Distribute Software” than IM “B” would provide for Software Distribution solution “B”.

There are two types of IMs: external and internal.

An Internal IM runs within the Maximo/CCMDB framework and knows how to translate between LMOs and MBOs on the CCMDB side, and an OMP server. The Internal IM communicates with the OMP either through an external IM, or through another interface provided by the OMP server. You can implement an internal IM as a Java class or an Invocation Channel. Both implementations,

either Java class or Invocation Channel, can access data stored in the CCMDB database. An Invocation Channel is a data transmission path with an optionally configured endpoint. The endpoint employs a communication protocol or Handler that determines how an IM exchanges information with its target. The integration framework supports standard Handlers, such as Web Services, HTTP, and EJB.

An external IM is an optional component that interacts with an OMP. Additionally, an external IM can manage an OMP that requires multiple execution steps. If you are developing a new interface for an OMP, writing an external IM can be helpful. The external IM can be a Web service that will then be available to applications other than CCMDB as well. Most of the logic can be in the external IM, and the internal IM can just have the Maximo-specific logic.

Within the CCMDB V7.1 solution, the Integration Modules and Logical Management Operations applications are used to set up and configure the integration to external systems using the IM / LMO technique. This does not eliminate the need to write your integration code, either for an internal or external IM.

Please refer to the Integration Module Development document as part of the ISM Toolbox. The document is part of the ISM Toolbox API.

7.7 Launch in Context

In some cases, a physical data exchange or federation setup with an external system is not required or is not the preferred choice of integration. If you, for example, in the context of a Service Management process require a person to be automatically linked to the user interface of an external system to analyze some specifics within the external system, you should think about the Launch in Context facility.

The Launch in Context application lets you define launch points to external system consoles, while you can leverage the launch points inside specific tasks of your process flow definitions.

If your change impact analysis, for example, requires you to analyze some status data within the monitoring or business Service Management solution, you could define a task within your Change Management process to link the appropriate subject matter expert into the monitoring system automatically from within the Service Management environment. Another example would be to link a person participating in the Change Management process to an external monitoring system in order to verify (as part of the Post Implementation Review) the status of a component that has been changed throughout the change process.

7.8 Summary

TADDM discovery sensors perform discovery while the DLAs provide a mechanism for loading additional data that conforms to the CDM. On the other end of the spectrum are the Authorized CIs, which are the entities that you explicitly build that are subject to tightly managed control. You can create Authorized CIs without having a Discovered (and Actual) CI. So where should you load CIs that you wish to put into the CCMDB?

This section provides an overview of various interface technologies for the overall CCMDB solution, including the most relevant use cases, and why and when to use the respective integration solution.

If it is IT data, or discoverable in any way, then it can be loaded into the CCMDB by using the TADDM discovery capabilities, or DLAs. Having data come in through TADDM discovery allows for the use of a key set of discovery services, such as naming and reconciliation and the ability to specify unique identification of a resource. As subsequent discoveries occur, changes are captured so that a change history for a resource exists, which can be valuable for problem analytics. TADDM also provides a graphical topology viewer to visualize what can be complex application dependency maps. And, of course, with periodic discoveries, we can copy Discovered CIs into the Actual CI space that form the basis for the CI auditing between Authorized CIs and Actual CIs. If there is a need for any one of these discovery-related capabilities or any of the others not mentioned here that are available with TADDM, then the CI data should be brought into the CCMDB through TADDM.

It is possible that you have a situation where you have CIs for which there is only a single source (you do not have a need to reconcile these objects with some other source) and you do not have a need to perform an audit between an authorized version of the CI and what would currently be in a “discovered” environment. In this case, it is feasible that you could create these CIs directly in the Authorized CI space in the CCMDB. An example of this type of data would be a logical hierarchy of business services and business applications that you have defined in an external system that you wish to be able to bring under change control, so you decide to use the MEA Integration Framework for the Authorized CI object structure to create Authorized CIs directly in the Authorized CI space without bringing them in through the discovery services.

It is reasonable that you will actually use both of these methods as you continue to build out your CCMDB, using discovery as the workhorse to bring in the discovered environment as well as directly creating Authorized CIs in the cases where you have single-sourced data that you do not wish to bring in through TADDM. It is important to recognize that if you feel that you will at some point need to reconcile these objects you are directly creating in the Authorized CI

space with objects from another data source, or you will come upon a need to audit these CIs for changes, then you should give consideration to whether this data should instead initially be brought in through the TADDM discovery services.

Archived



Part 3

Planning and installation

Archived

CCMDB installation planning

This chapter discusses the planning and customization of a working IBM Tivoli Change and Configuration Management Database V7.1 environment.

8.1 CCMDB components

Though the components of CCMDB have already been described in some detail in Chapter 3, “CCMDB components” on page 29, we include a brief recap here, as the installation process must take the various components that will be installed for a particular deployment into consideration. In addition, we provide additional details here on various products and technologies that are used to implement the components.

This section provides an overview of the CCMDB components:

- ▶ Middleware components
- ▶ IBM Tivoli Application Dependency Discovery Manager
- ▶ IBM Tivoli Change and Configuration Management Database
- ▶ Console - CCMDB WEB Interface
- ▶ IBM Tivoli Integration Composer
- ▶ IBM Tivoli Unified Process Composer (ITUP)

8.1.1 Middleware components

Before you can install the IBM Tivoli Change and Configuration Management Database, there are several prerequisite middleware products that must be installed. Here we explain middleware components and concepts:

- ▶ IBM Rational Agent Controller Version 7.0.3

Rational Agent Controller (RAC) is a daemon process that provides the mechanism by which client applications either launch new host processes or attach to agents that coexist within existing host processes. WebSphere Event Broker uses RAC to provide debugging facilities for message flows deployed to a running broker.

- ▶ DB2 Enterprise Server Edition Version 9.1.2

Configuration for DB2 Enterprise Server Edition

- Provides innovative self-managing and self-tuning capabilities, dramatically reducing time and costs associated with managing database servers.
- Supports integration with other IBM software, such as Lotus for collaboration, Tivoli for management, and WebSphere for dynamic e-business.

- ▶ IBM Tivoli Directory Server Version 6.1

Configuration for IBM Tivoli Directory Server

IBM Tivoli Directory Server is a powerful, security-rich, and standards-compliant enterprise directory for corporate intranets and the Internet. Directory Server is built to serve as the identity data foundation for rapid development and deployment of your Web applications and security and identity management initiatives by including strong management, replication, and security features.

- ▶ WebSphere Application Server ND Version 6.1.0.9

Configuration for WebSphere Application Server ND

IBM HTTP Server Version 6.1

Embedded Security Service Version 6.1

Combines near-continuous availability with automated performance optimization and centralized management and monitoring, for business-critical applications.

8.1.2 Tivoli Application Dependency Discovery Manager

The Tivoli Application Dependency Discovery Manager (TADDM) system consists of several server-systems and network components to be described. These components will make infrastructure scans possible and are the backbone of discoveries.

The main components, described briefly, are as follows:

- ▶ TADDM Server: The central server that run the discoveries. Acquired data is consolidated and topology maps are created.
- ▶ TADDM Database Server: A sever for high-performance data storage and delivery of topology data, change data, and configurations.
- ▶ TADDM Console: Hosts the user interface to operate the software for performing actions such as start of discovery scans, editing of access right and accounts, and other functions.
- ▶ Anchor Host: Acts as an outpost of the TADDM Server when running discoveries across firewalls.
- ▶ Windows Gateway Server: Windows server to scan Windows systems.

8.1.3 IBM Tivoli Change and Configuration Management Database

The IBM Tivoli Change and Configuration Management Database (CCMDB) provides an enterprise-ready platform for storing deep, standardized data on configurations and change histories to help integrate people, processes, information, and technology.

8.1.4 Console - CCMDB Web Interface

Provides the user interface to the CCMDB functions and applications.

8.1.5 IBM Tivoli Integration Composer

IBM Tivoli Integration Composer allows you to take advantage of the data collected by these disparate systems by “fusing” it together into a single actionable repository. Data from different systems management tools, different business units, or distributed locations can be easily consolidated, hence leveraging existing investments.

Integration Composer is a combination of powerful out-of-the-box cartridges for the most prevalent IT system management tools in the market and an easy-to-use, intuitive Java-based application for the setup of out-of-the-box cartridges and creation of new cartridges to meet your specific integration requirements. The ITIC interface allows “drag-and-drop” operations to quickly and easily move inventory data from any systems management tools to Maximo Asset Management for IT. No programming is required.

8.1.6 Integration adapter

To facilitate data migration, the integration adapter for TADDM provides a set of predefined mapping expressions designed to transform data in the TADDM data source to the form required by CCMDB. You can import this mapping into Integration Composer and, with little or no additional programming, execute a mapping to import TADDM data into CCMDB.

8.1.7 IBM Tivoli Unified Process Composer

IBM Tivoli Unified Process (ITUP) Composer V2.1 provides detailed documentation of IT Service Management processes based on industry best practices, enabling users to significantly improve their organization's efficiency and effectiveness. ITUP Composer is the product version of the free IBM Tivoli

Unified Process tool. ITUP Composer provides more detailed content and tooling to enable content customization, extension, and publishing.

8.2 Installation plan

This section provides direction and best practices around areas in which questions typically arise when working with Configuration Items (CIs) and building, deploying, and integrating with a Configuration Management Database (CMDB).

Creating a deployment plan is essential to creating and installing a configuration and tracking environment. The basic considerations for creating a deployment plan are provided in *Planning and Installing Change and Configuration Management Database*. This document covers all the planning considerations and provides scenarios for creating a comprehensive deployment plan. At a minimum, you need to gather the following information before installing any software:

- ▶ Base hardware and software requirements for IBM Tivoli Change and Configuration Management Database V7.1.
- ▶ Whether the computer systems in your distributed network can support this new software, whether these systems can be upgraded to meet your business needs, or whether you will need new systems.
- ▶ Which components to install on which computer systems in your distributed network to support your business needs and whether they have additional third-party software requirements.

The focus of this topic is to describe how we have installed the CCMDDB and its components in our environment and show possible scenarios for implementation in accordance with a customer environment.

8.2.1 Software and hardware requirements

This section provides information about the hardware, software and infrastructure requirements for installation. Table 8-1 shows these requirements.

Table 8-1 Software and hardware requirements

Software	Hardware
IBM DB2 UDB	☐ Minimum 20 GB disk space.
IBM WebSphere Network Deployment	☐ 2 GHz processor ☐ 20 GB disk space ☐ 2 GB RAM
IBM Tivoli Application Dependency Discovery Manager	☐ 2 GHz processor ☐ 20 GB disk space ☐ 2 GB RAM
IBM Tivoli Directory Server	☐ For UNIX, 1 GB of space available in the /opt directory

For hardware requirements related to software components not listed above, refer to the product documentation provided with that product.

Note: If you are installing all middleware onto a single machine, the RAM requirement increases to 4 GB.

Database

This topic provides information about the database and its environment infrastructure for installation, as shown in Table 8-2 on page 169.

Table 8-2 Database and environment infrastructure requirements

Software	Environment
The following products can serve as the database components of a CCMDB V7.1 deployment. ☐ DB2 UDB V9.1 FP 2 (installed by the Tivoli middleware installer) or V8.2 FP 14 ☐ Oracle 9i V2, Oracle 10g Rel1, or Oracle 10g Rel2 ☐ Microsoft SQL Server 2005, Standard or Enterprise version	☐ Windows Server® 2003 SP2 ☐ Red Hat Enterprise Linux V4 (Intel) ☐ IBM AIX 5L™ V5.3 ML level 5300-06 For Oracle 9i v2, and Oracle 10g Rel1, only Windows is supported. Microsoft SQL Server 2005, only Windows is supported. For DB2 on UNIX systems, ensure you have a minimum of 8 gigabytes (binary) free of space in the DB2 database instance home directory (/home/ctginst1) in order to meet the default table space disk space requirements of the DB2 install. For DB2 on Windows, ensure you have a minimum of 8 gigabytes of free space in the DB2 installation directory.

J2EE application server

This topic provides information about J2EE and its required environment for installation, as shown in Table 8-3.

Table 8-3 J2EE environment requirements

Software	Environment
The following product can serve as the J2EE application server component of a CCMDB V7.1 deployment: ☐ IBM WebSphere Network Deployment V6.1 FP 9 This is where CCMDB runs.	☐ Windows Server 2003 SP2 ☐ Red Hat Enterprise Linux V4 (Intel) ☐ IBM AIX 5L V5.3 ML level 5300-06

User WEB Interface Console

This topic provides information about User WEB Interface Console and its environment infrastructure for installation, as shown in Table 8-4.

Table 8-4 WEB Interface console requirements

Software	Environment
The following product can serve as integration options for a CCMDB V7.1 deployment: ☐ IBM Integrated Solutions Console V7.1 IBM Integrated Solutions Console V7.1 is installed as part of IBM WebSphere Network Deployment V6.1 FP9: ☐ IBM WebSphere Portal Server V6.0	☐ CCMDB integration components can be run on any operating system supported by the integration software.

HTTP server

This topic provides information about the HTTP server and its environment infrastructure for installation, as shown in Table 8-5.

Table 8-5 HTTP server requirements

Software	Environment
The following product can serve as the HTTP server component of a CCMDB V7.1 deployment: ☐ IBM HTTP Server V6.1 FP9	☐ Windows Server 2003 SP2 ☐ Red Hat Enterprise Linux V4 (Intel) ☐ IBM AIX 5L V5.3 ML level 5300-06

Directory server

This topic provides information about the Directory server and its environment infrastructure for installation, as shown in Table 8-6 on page 171.

Table 8-6 Directory server requirements

Software	Environment
The following products can serve as the Directory server component of a CCMDB V7.1 deployment. ☐ IBM Tivoli Directory Server V6.1 ☐ Microsoft Windows Server 2003 SP2 Active Directory Microsoft Active Directory Application Mode (ADAM) is not supported.	☐ Windows Server 2003 SP2 ☐ Red Hat Enterprise Linux V4 (Intel) ☐ IBM AIX 5L V5.3 ML level 5300-06

Administrative Console

This topic provides information about Administrative Console and its environment infrastructure for installation, as shown in Table 8-7.

Table 8-7 Administrative Console requirements

Software	Environment
The following product can serve as the administrative system component of a CCMDB V7.1 deployment. ☐ CCMDB administrative system	☐ Windows Server 2003 SP2 (32-bit)

Application Dependency Discovery Manager

This topic provides information about TADDM and its environment infrastructure for installation, as shown in Table 8-8.

Table 8-8 Application Dependency Discovery Manager requirements

Software	Environment
The following product can serve as the Application Dependency Discovery Manager component of a CCMDB V7.1 deployment. ☐ IBM Tivoli Application Dependency Discovery Manager (TADDM) V7.1	☐ IBM AIX V5.3 and V5.2 ☐ Red Hat Enterprise Linux V4 and V3 (Intel) ☐ Red Hat Enterprise Linux 4.0 for (System z™) ☐ Solaris V10 and V9 SPARC ☐ SUSE Linux Enterprise Server V10 and V9 (Intel) ☐ SUSE Linux Enterprise Server V10 and V9 (System z) ☐ Windows 2000 Professional ☐ Windows 2000 Advanced Server ☐ Windows 2000 DataCenter Server ☐ Windows Server 2003 DataCenter ☐ Windows Server 2003 Standard Edition SP2 ☐ Windows Server 2003 Enterprise Edition ☐ Windows Server 2003 Enterprise x64 Edition AMD64 and EM64T ☐ Windows Server 2003 Standard x64 Edition AMD64 and EM64T ☐ Windows XP Professional Refer to the <i>IBM Tivoli Application Dependency Discovery Manager Planning and Installation Guide</i> for support details.

Data migration

This topic provides information about the data migration and its environment infrastructure for installation, as shown in Table 8-9 on page 173.

Table 8-9 Data migration requirements

Software	Environment
The following product can serve as the data migration component of a CCMDB V7.1 deployment. ☐ IBM Tivoli Integration Composer V7.1	☐ Windows Server 2003 Standard (32- and 64-bit) ☐ Windows Server 2003 Enterprise Edition (32- and 64-bit) ☐ Windows Server 2003 DataCenter (32- and 64-bit) ☐ IBM AIX 5L V5.3 (32- and 64-bit) and AIX 5L technologies ☐ Red Hat Linux 3 ☐ Sun™ Solaris 10 ☐ HP-UX 11i

8.2.2 Planning for the deployment of CCMDB

The following sections describe various topics that should be taken into account when planning a deployment of CCMDB.

Planning for security

You must configure security for your data as well as manage access to operations, in both the Configuration Discovery and Tracking and Process Management and Integration Platform features. Managing security for Change and Configuration Management Database involves identifying persons and groups who have the authority to work with particular configuration items and perform specific operations. You can find the information you need about security topics at the product support page. Also, refer to Chapter 6, “CCMDB security architecture” on page 105.

Planning users and roles

Access to CI data stored in the CMDB is controlled using roles. Each user must be assigned to a role, which governs the user's access to CIs.

You can control which of your users can work with different groups of configuration items stored in your CMDB.

You will use these concepts in controlling access:

- ▶ Access collection: A group of CIs to which access is controlled. By creating access collections, you can assign different user roles to work with different sets of CIs.

- ▶ **Role:** A collection of permissions for different sets of CIs.
- ▶ **User:** A specific individual, who is given controlled access to access collections of CIs by assigning roles to the user.

You can set up access collections, roles, and users in the Configuration Discovery and Tracking console after you have completed the installation process. If you also install the Process Management and Integration Platform feature, you will need to synchronize the lists of users and roles between the two features.

Note: For more details about planning and installing IBM Tivoli Change and Configuration Management Database V7.1, refer to Chapter 1, “CCMDB overview” on page 3.

Planning for CCMDB middleware worksheet

This topic provides information about the middleware and its environment for users, groups, and platform upon installation, as shown in Table 8-10.

Table 8-10 CCMDB middleware worksheet

User	Group	Description	Platform
db2admin	☐ DB2USERS ☐ DB2ADMNS	DB2 administrator. Windows Service User ID. This user will be created by the Tivoli middleware installer if it does not already exist.	☐ Windows
idscmdb	☐ Users ☐ Administrators	ITDS user. This user will be created by the Tivoli middleware installer if it does not already exist	☐ Windows v AIX ☐ Linux
maximo	☐ Users ☐ Administrators	Used for Maximo database configuration. This user will be created by the Tivoli middleware installer if it does not already exist.	☐ Windows v AIX ☐ Linux

User	Group	Description	Platform
ctginst1	☐ Users ☐ Administrators ctginst1 must be a member of db2grp1 with the secondary groups of staff and dasadm1.	The system user used as the database instance owner on UNIX platforms. This user will be created by the Tivoli middleware installer if it does not already exist.	☐ AIX ☐ Linux
db2fenc1	☐ db2fgrp1	UNIX system user used as the fence user ID for DB2. This user will be created by the Tivoli middleware installer if it does not already exist	☐ AIX ☐ Linux
wasadmin	Not a system user.	This is a user ID created for use with WebSphere. This user will be created by the Tivoli middleware installer if it does not already exist.	☐ Windows ☐ AIX ☐ Linux

Tivoli middleware installer

This topic provides information about Tivoli middleware and its environment for users, groups, and platform on installation, as shown in Table 8-11.

Table 8-11 Tivoli middleware requirements

Setting	Default
Workspace directory	<user home>\ibm\tivoli\mw\workspace
Middleware images source directory	
Compressed images directory	
Uncompressed images directory	

DB2 configuration

This topic provides information about the DB2 configuration and its environment for users, groups, and platform upon installation, as shown in Table 8-12.

Table 8-12 DB2 configuration

Setting	Default
Installation directory	▶ Windows: <SystemDrive>\Program Files\IBM\SQLLIB ▶ Linux: /opt/ibm/db2/V9.1 ▶ AIX: /opt/IBM/db2/V9.1
DAS user	▶ Windows: db2admin ▶ Linux: dasusr1 ▶ AIX: dasusr1
Fenced user	▶ Linux: db2fenc1 ▶ AIX: db2fenc1
Fenced user group name	▶ Linux: db2fgrp1 ▶ AIX: db2fgrp1
Fenced user home directory	▶ Linux: /home/db2fenc1 ▶ AIX: /home/db2fenc1
Instance name	ctginst1.
Port	50005.
Instance user name home directory	▶ Linux: /home/ctginst1 ▶ AIX: /home/ctginst1
Database instance user ID	▶ Windows: db2admin ▶ Linux: ctginst1 ▶ AIX: ctginst1
DB2 administrators group	▶ Windows: DB2ADMNS ▶ Linux: db2grp1 ▶ AIX: db2grp1
DB2 users group	Windows: DB2USERS.
Use the same user name and password for the remaining DB2 Services.	YES.
Configure Tools Catalog	NO: This value is relevant for reuse scenarios only.
Enable O/S Security for DB2 objects	YES: This value is relevant for reuse scenarios only.
DB2 instance port	

Oracle configuration

This topic provides information about the Oracle configuration and its environment for users, groups, and platform upon installation, as shown in Table 8-13.

Table 8-13 Oracle configuration

Setting	Default
Installation directory	▶ Windows: <SystemDrive>\oracle\product\10.2.0\oradata. ▶ Linux: /opt/app/oracle/product/10.2.0/oradata. ▶ AIX: /opt/app/oracle/product/10.2.0/oradata.
Administrator User ID	sys
Oracle Software Owner ID	▶ Windows: Administrator. ▶ Linux: oracle. ▶ AIX: oracle.
Instance Location	▶ Windows: On Windows, this value might be C:\oracle\product\10.2.0\oradata. ▶ Linux: On Linux, this value might be /opt/app/oracle/product/10.2.0/oradata. ▶ AIX: /opt/app/oracle/product/10.2.0/oradata.

SQL server configuration

This topic provides information about the SQL server configuration and its environment for users, groups, and platform upon installation, as shown in Table 8-14.

Table 8-14 SQL server configuration

Settings	Default
Installation directory	<ProgramFiles>\Microsoft SQL Server\90
Named instance	maximo
SQL Server administrator	sa
SQL Server administrator password	Provided by administrator
Port	1433
Database name	maxdb71

Settings	Default
User ID	maximo
User ID password	Provided by user
Data file name	maxdb71_dat
Log file name	maxdb71_log

WebSphere configuration

This topic provides information about the WebSphere configuration and its environment for users, groups, and platform upon installation, as shown in Table 8-15.

Table 8-15 WebSphere configuration

Settings	Default
Install location	▶ Windows: C:\Program Files\IBM\WebSphere\AppServer ▶ Linux: /opt/IBM/WebSphere/AppServer ▶ AIX: /usr/IBM/WebSphere/AppServer
WebSphere Administration user name	wasadmin.
Deployment Manager profile name	ctgDmgr01.
Application server profile name	ctgAppSrv01.
Profile directory	▶ Linux: /opt/IBM/WebSphere/AppServer/profiles ▶ AIX: /usr/IBM/WebSphere/AppServer/profiles
Cell name	ctgCell01.
Deployment Manager node name	ctgCellManager01.
Application server node name	ctgNode01.
HTTP server install location	▶ Windows: C:\Program Files\IBM\HTTPServer ▶ Linux: /opt/IBM/HTTPServer ▶ AIX: /usr/IBM/HTTPServer
HTTP port	80. Note: On Windows, this port might already be in use. Ensure that you either free up this port, or use another port that is unassigned.

Settings	Default
HTTP admin server port	8008.
HTTP plug-in profile name	ctgAppSvr01.

IBM Tivoli Directory Server configuration

This topic provides information about the IBM Tivoli Directory Server configuration and its environment for users, groups, and platform upon installation, as shown in Table 8-16.

Table 8-16 IBM Tivoli Directory Server configuration

Settings	Default
Install location	▶ Windows: C:\Program Files\IBMLDAP\V6.1 ▶ Linux: /opt/IBM/ldap/V6.1 ▶ AIX: /opt/IBM/ldap/V6.1
Administrator distinguished name	cn=root
Organizational unit	ou=SWG
Organization and country suffix	o=IBM,c=US
Directory server port	389
Directory server secure port	636
Administration port	3538
Administration secure port	3539
Database name	security
Instance name	idscmdb
Instance port	50006
Instance user name	idscmdb

Microsoft Active Directory configuration

This topic provides information about the Microsoft Active Directory configuration and its environment for users, groups and platform upon installation, as shown in Table 8-17.

Table 8-17 Microsoft Active Directory configuration

Settings	Default
Directory server port	389
LDAP base entry	DC=ism71,DC=com
User suffix	CN=Users,DC=ism71,DC=com
Group suffix	DC=ism71,DC=com
Organization container suffix	DC=ism71,DC=com
Bind distinguished name	CN=Administrator,CN=Users,DC=ism71,DC=com

IBM Agent Controller configuration

This topic provides information about the IBM Agent Controller configuration and its environment for users, groups, and platform upon installation, as shown in Table 8-18.

Table 8-18 IBM Agent Controller configuration

Settings	Default
IBM Agent Controller installation location	▶ Windows: C:\Program Files\IBM\AgentController ▶ Linux: /opt/IBM/AgentController ▶ AIX: /opt/IBM/AgentController

8.2.3 Planning for CCMDB worksheet

This topic provides information about CCMDB and its environment for the settings for a custom installation, as shown in Table 8-19.

Table 8-19 Planning for CCMDB worksheet

Settings	Default
Installation directory	C:\IBM\CCMDB.
TADDMM host name	

Settings	Default
RMI port	9433.
TADDM user ID	▶ Windows: administrator ▶ Linux: administrator ▶ AIX: administrator
TADDM Web server port	9430.
DB2 host name	
DB2 port	50005.
Maximo database name	maxdb71.
Maximo database instance	ctginst1.
Maximo database user ID	maximo.
DB2 installation directory	▶ Windows: C:\Program Files\IBM\SQLLIB ▶ Linux: /opt/ibm/db2/V9.1 ▶ AIX: /opt/IBM/db2/V9.1
DB2 instance administrator user ID	▶ Windows: db2admin ▶ Linux: ctginst1 ▶ AIX: ctginst1
Windows DB2 service user ID	db2admin.
Oracle installation directory	▶ Windows: C:\oracle\product\10.2.0\oradata ▶ Linux: /opt/app/oracle/product/10.2.0/oradata ▶ AIX: /opt/app/oracle/product/10.2.0/oradata
Oracle administrator user ID	sys.
Oracle software owner user ID	oracle.
SQL installation directory	<CProgramFiles>\Microsoft SQL Server\90.
Data table space name	maxdata.
Data table space size	▶ DB2 :Medium (5000 Mb) ▶ Oracle: Medium (1000 Mb) ▶ SQL Server (Initial data file size): Medium (1000 Mb)
Temporary table space name	maxtemp.
Temporary table space size	1000 Mb.
WebSphere host name	

Settings	Default
WebSphere SOAP port	8879.
WebSphere server home directory	▶ Windows: C:\Program Files\IBM\WebSphere\AppServer ▶ Linux: /opt/IBM/WebSphere/AppServer ▶ AIX: /usr/IBM/WebSphere/AppServer
WebSphere admin user ID	wasadmin.
WebSphere profile name	ctgDmgr01.
Web server port	9081.
Web server name	webserver1.
Node name	ctgNode01.
Cluster name	MAXIMOCLUSTER.
Application server	MXServer. This value cannot be changed.
JMS datasource name	
JMS database name	maxsibdb.
JMS server name	
Database server port	50000.
Database user ID	maxadmin.
Directory server host name	
Directory server port	389.
Directory server administrator DN	cn=root.
Bind password	
Maximo installation folder	C:\IBM\maximo Note: Maximo can only be installed from the CCMDB administrative system, which must be deployed on a Windows system.
SMTP server	
Workflow Admin E-Mail	
Admin E-Mail	

IBM Tivoli Application Dependency Discovery Manager configuration

This topic provides information about TADDM and its environment for the settings for a custom installation, as shown in Table 8-20.

Table 8-20 IBM Tivoli Application Dependency Discovery Manager configuration

Settings	Default
Host name	
RMI Port	9433
User ID	
Web server port	9430

8.2.4 CCMDB topologies

Use this information to determine the best deployment option for your environment and business needs. There are two primary strategies to deploy CCMDB within your enterprise:

- **Single-server**

The single-server topology consists of loading all CCMDB components onto one machine. This would be done typically for proof-of-concept purposes, as a demonstration, or as a learning environment. For managing enterprise assets and processes, you would typically implement a multi-server topology.

- **Multi-server**

The multi-server topology consists of splitting CCMDB components across several different machines. This is beneficial because it optimizes resource use and decreases each system's workload. This type of deployment would be typical for production use within an enterprise.

A typical deployment life cycle might begin with a single-server topology that would move through phases of demonstration, functional proof-of-concept, and testing integration within the existing environment, and then gradually move towards a pilot multi-server environment before finally implementing a production deployment within the enterprise.

Single server deployment

A topology consisting of deploying IBM Tivoli Change and Configuration Management Database on a single machine is frequently used as a proof-of-concept, educational, or demonstration configuration.

While the CCMDB middleware components, configuration item acquisition components, process managers, administration, and integration software can be installed on systems running non-Windows operating systems, CCMDB must be installed by way of the CCMDB administrative system, which must be hosted on a Windows-based system.

Multiple server deployment

CCMDB should be deployed on multiple machines in order to provide load balancing, availability, reuse, and redundancy. This is the recommended deployment topology for a production environment.

When contemplating your deployment strategy, you must determine if it will include systems already established in your network. Implementing CCMDB by installing all new components using the CCMDB middleware and CCMDB installation programs will simplify the deployment. If you plan to reuse or migrate resources that already exist in your network, make adjustments to your rollout plan to allow time for such things as bringing the existing resources to version levels that are compatible with CCMDB.

In our installation, we have choose to use a structure consisting of two servers, as shown in Figure 8-1.

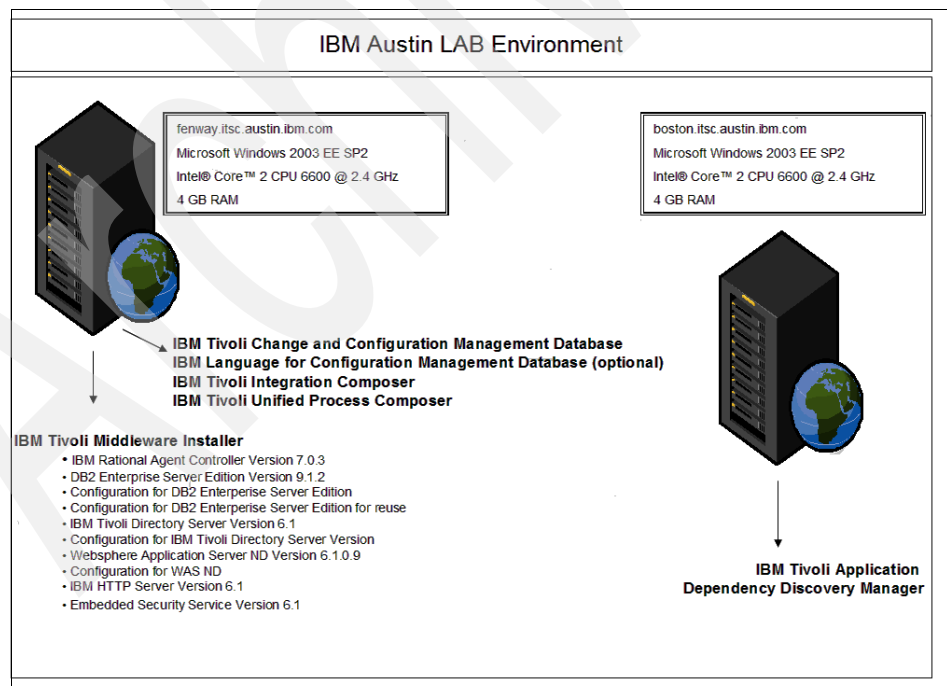


Figure 8-1 Multiple server deployment option

Summary of our lab Installation

We use the same hardware environment for Microsoft Windows 2003 Server Enterprise Edition and for Red Hat Enterprise Linux 4 Server.

We start with System Operation Installations and apply the updates packages to prepare the servers fenway.itsc.austin.ibm.com and boston.itsc.austin.ibm.com for start the CCMDB installation.

The first CCMDB components to be installed are the middleware components. We install the middleware on fenway.itsc.austin.ibm.com and we use the defaults options on this installation. This activity in our environment LAB has taken around two hours.

After the middleware installation is done, we decide to install the TADDM on a stand-alone server named boston.itsc.austin.ibm.com in order to get higher performance in the discovery process. The installation is carried out in accordance with the Planning and Installing IBM Tivoli Change and Configuration Management Database V7.1 and we use the defaults options on this installation. This activity in our lab environment takes around two hours and 30 minutes.

With the prerequisites for installing the CCMDB installed (middleware components and TADDM), we start the CCMDB installation on fenway.itsc.austin.ibm.com and we use the defaults options on this installation. This activity in our environment takes about three hours.

Note: For more details, see *Planning and Installing Change and Configuration Management Database*.

Installation

This chapter describes how IBM Tivoli Change and Configuration Management Database CCMDB V.7.1 (CCMDB V7.1) is installed and configured on a multiple server deployment on Windows and Linux.

The first part will give an overview on how the Windows and Linux topology is set up, followed by an installation flowchart to clarify the installation steps. The most important steps to take care of during installation is covered in the rest of this chapter, which will include additional information not described in *Planning and Installing Change and Configuration Management Database*.

9.1 Topology overview

There are two topologies described in this chapter. One is a pure Windows multiple server deployment and one is a mixed Linux-Windows multiple server deployment, because the CCMDB Administration Workstation is only available on Windows-based systems.

9.1.1 Windows multiple server deployment topology

The setup of a multiple server deployment on Windows machines is shown in Figure 9-1. Server B hosts the TADDM application, including the CCMDB database. Server A provides the CCMDB middleware components, such as Tivoli Directory Server (LDAP), Maximo Database, the J2EE (WebSphere), the Change and Configuration process applications, and the IBM Tivoli Integration Composer, which can be used for mapping TADDM data into CCMDB data. This server also serves as the CCMDB administrative system.

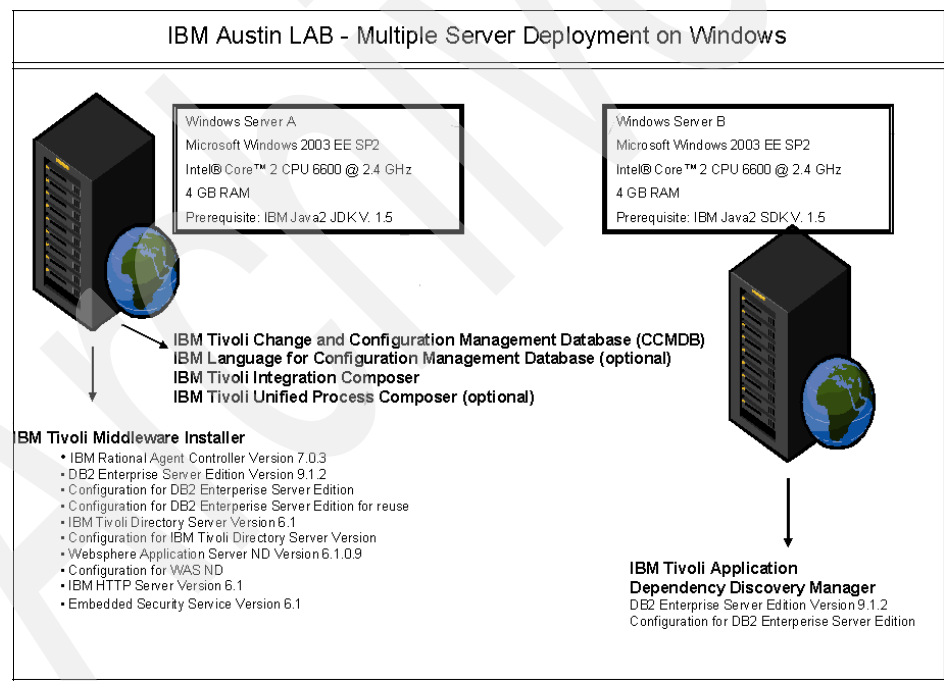


Figure 9-1 Our Windows-based lab environment

9.1.2 Red Hat Linux server multiple server deployment topology

The setup of a multiple server deployment on Red Hat Enterprise Linux (RHEL) machines is shown in 9.2, “Installation flow” on page 190. Because CCMDB must be installed using the CCMDB administrative system, which is only hosted on Windows-based systems, the Linux environment consists of three systems: Linux server B, which hosts the TADDM application, including the CMDB database, Linux server A, and the CCMDB middleware components, such as Tivoli Directory Server (LDAP), Maximo Database, the J2EE (WebSphere), and the Change and Configuration process applications. The IBM Tivoli Integration Composer (ITIC), which can be used for the mapping TADDM data into CCMDB data, also resides on that system. For administrative purposes, there also has to be a Windows system.

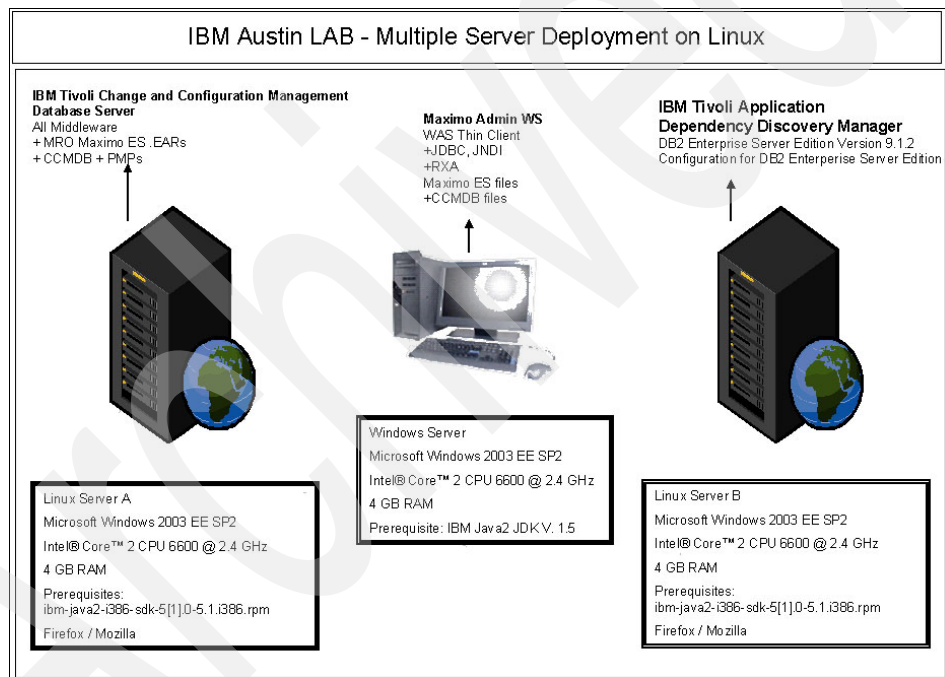


Figure 9-2 Our Linux-based lab environment

9.2 Installation flow

The CCMDB launchpad enables automatic installation and configuration (refer to 9.4, “The CCMDB launchpad” on page 193 for more information). This option is available on Windows and Linux for all CCMDB components. On AIX, it is mandatory to install and configure the middleware components manually. We highly recommend following the order of the installation steps, as shown in Figure 9-3.

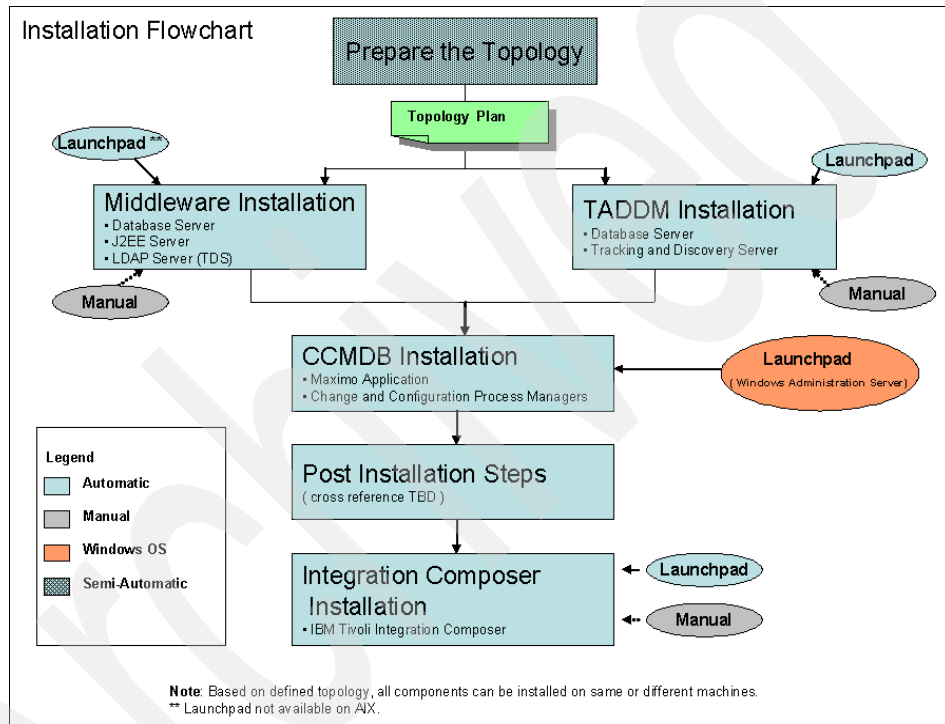


Figure 9-3 Installation flowchart

As you can see in Figure 9-3, most of the installation steps can be done using the launchpad interface. However, the launchpad interface for the middleware installation is only available on Windows and Linux. On AIX, you have to do the installation manually.

For further information, refer to *Planning and Installing Change and Configuration Management Database*.

The first step of the process is preparing the topology, which is done semi-automatically. The deployment engine will create a topology plan that

depends on the deployment selection you make. An XML file will be created containing the software installation and configuration information (see 9.5, “Middleware installation” on page 194 for more information). The next step would be the installation of all the required middleware components and TADDM. These two steps are required before installing the CCMDB component. Once all these components are successfully installed, you must go through the post installation steps in order to create a base organization structure. Finally, you can install the ITIC, which will map data from the TADDM database into the CCMDB database.

9.3 What you should do before you begin

This section describes steps that one must take care of before installing the CCMDB components and also gives an overview of the launchpad interface.

Before you start the installation, we highly recommend that you follow these steps to ensure a simple and smooth installation of CCMDB V7.1:

- ▶ **Backup of systems:** You should make a backup of your system before doing the installation of any CCMDB V7.1 component on your system, as there is no automated uninstall feature supplied with IBM Tivoli CCMDB V7.1. If the installation fails at any point, you need to restore your system from the backup or reinstall the respective OS on your machine.
- ▶ **JDK/JRE:** We recommend having IBM Java V5.1 installed on your system.
- ▶ **Internet Browser:** CCMDB V7.1 only supports Firefox and Mozilla internet browsers on Linux and Internet Explorer® on Windows.
- ▶ **Delete the TEMP and TMP user environments on the Windows systems.** The existence of TEMP and TMP user variables on Windows machine can cause errors with the installation of DB2.
- ▶ **Verify the rpm-build required package on the Linux machine:** The rpm-build package should be installed on the Linux machine before you run the Tivoli middleware installer.

To verify that the rpm-build package is installed, run the following command:

```
$ rpm -qa | grep build
```

If the command returns a value such as rpm-build-4.3.3.-18_nonptl, the package is installed. If nothing returns, you must install the rpm-build package, which is located on disk 3 (of 5) of the Red Hat Enterprise Advance Server Version 4 installation CDs.

- ▶ Setting the **ulimit** on the Linux machine.

For Linux systems, you must set the **ulimit** for the system prior to using the Tivoli middleware installer. From the command line, run the following command:

```
$ ulimit -f unlimited  
$ ulimit -n 8192
```

For AIX, refer to the *AIX Administrator Guide* or *Planning and Installing Change and Configuration Management Database*.

- ▶ Swap space on the Linux/AIX machine.

CCMDB is a resource-intensive application. We recommend that you tune your system for maximum performance. The swap size on Linux/AIX systems should be equivalent to twice the amount of physical RAM in the machine.

- ▶ Setting shared memory on the Linux/AIX machine

On Linux systems, the value of minimum shared memory should be 268435456 (256 Mb).

To set the minimum shared memory value, run the following command:

```
$ sysctl -w kernel.shmmax=268435456
```

Update the `/etc/sysctl.conf` by adding the following line:

```
kernel.shmmax=268435456.
```

- ▶ Set the **PATH** variable for the **JAVA** bin directory on the Linux/AIX machine.

On the Linux/AIX machine, you should set the **PATH** variable for the **JAVA** bin directory before doing the installation. For example:

```
PATH=$PATH:/opt/IBM/java/jre/bin
```

- ▶ Enabling remote configuration.

If you plan to take the advantage of CCMDB installation program (runs from the Windows administration workstation) feature that automates the configuration of CCMDB middleware, you must enable a Remote Execution and Access (RXA) service for each system on which you intend to install CCMDB middleware. RXA requires that the target system enable at least one of the protocols supported by RXA, which include rsh, rexec, SSH, and Windows SMB. Before you start the CCMDB installation program from the Windows administration workstation, ensure that one of these protocols is running and will accept remote logins. If the remote machine is Windows, then you must configure RXA to work over SMB.

- ▶ Microsoft Windows 2003 Enterprise Edition x86-32 Service Pack 2 or higher.

9.4 The CCMDB launchpad

IBM Tivoli CCMDB V7.1 comes with a launchpad application that serves as a centralized interface for launching a collection of installation programs and product information. It is quite important to understand what this interface can do and how to use it during the installation process.

This launchpad application will assist you in choosing the product installation programs and also indicates the order in which they should be installed.

Use the CCMDB launchpad to:

- ▶ Access the CCMDB product Web site.
- ▶ Access the information used to plan the CCMDB installation and deployment.
- ▶ Invoke the middleware installer for the installation of middleware components.
- ▶ Invoke the IBM Agent Controller installation program.
- ▶ Invoke the IBM TADDM installation program.
- ▶ Invoke the CCMDB installation program (to install the Maximo application and the Change and Configuration Management process managers).
- ▶ Invoke the Integration Composer installation program.

To start the CCMDB launchpad, complete the following steps:

- ▶ Log on to an account with system administration privileges (or root) on the computer where CCMDB components will be installed.
- ▶ To install the CCMDB installation program, you need to have a Windows Administration Station from where you should launch the launchpad interface to install the Maximo application and the Change and Configuration Management process managers on any remote machine.
- ▶ Start the launchpad from the CCMDB V7.1 DVD. As shown in Figure 9-8 on page 200, navigate to the root directory of the product disc or the downloaded installation image, and run the following commands:
 - On Windows: **1aunchpad.exe**
 - On UNIX: **1aunchpad.sh**

Figure 9-4 shows the launchpad interface that would give you the options to install the CCMDB components.

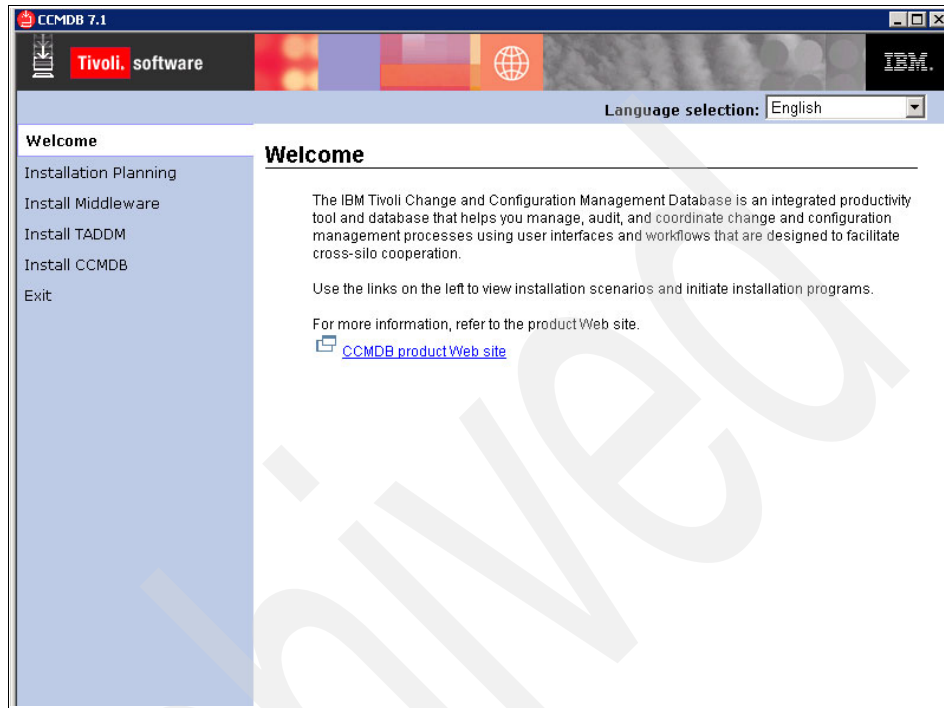


Figure 9-4 Launchpad Initial window

9.5 Middleware installation

Before installing the actual CCMDB, the middleware components should be installed first, as shown in Figure 9-3 on page 190.

We highly recommend following these steps to ensure a simple and smooth installation and configuration of CCMDB V7.1. For more information about this topic, refer to 9.3, "What you should do before you begin" on page 191.

To install the middleware components, you can use the launchpad interface, but remember that it should be invoked from the same system where you want to install the middleware components. On the launchpad interface, click **Middleware Installation** (Figure 9-4), which will take you to the middleware installer interface.

For further installation steps, refer to the *Planning and Installing Change and Configuration Management Database*.

Attention: There is no middleware installer available on AIX. The middleware installation should be done manually.

The middleware installation installs and configures the following components for:

- ▶ IBM Rational Agent Controller Version V7.0.3
- ▶ DB2 Enterprise Server Edition Version V9.1.2
- ▶ Configuration for DB2 Enterprise Server Edition
- ▶ Configuration for DB2 Enterprise Server Edition for reuse
- ▶ IBM Tivoli Directory Server Version 6.1
- ▶ Configuration for IBM Tivoli Directory Server Version 6.1
- ▶ WebSphere Application Server ND Version 6.1.0.9
- ▶ Configuration for WebSphere Application Server ND
- ▶ IBM HTTP Server Version 6.1
- ▶ Embedded Security Service Version 6.1

Note: The installation needs to create a workspace folder for deployment. The process will check automatically the available space on disc.

The workspace contains the following items:

- ▶ Deployment Plan
 - The deployment plan is a collection of installation steps, configuration parameters for those steps, and target machine information. It is generated through the Tivoli middleware installer and it resides in the workspace directory.
 - When deployment steps are changed, the existing deployment plan is deleted and replaced with the new deployment plan.
- ▶ Topology File

The topology file is a properties file that describes the configuration parameters of the CCMDB middleware deployment. This file is created and then updated after every deployment or undeployment. If you have not defined a workspace that is centrally located and accessible to all the systems that will be receiving CCMDB middleware, this file will have to be copied to the workspace of each machine where CCMDB middleware is being deployed. The contents of this file can be used by the CCMDB installation program to populate its panels with meaningful default values.

► Logs

Log files that contain information about the deployment can be found in the workspace directory. In addition, log files native to the CCMDB middleware itself are also contained in this directory.

After this step, you can select the features to deploy on the local machine. We recommend that you analyze your environment before proceeding to choose which distribution is best suited for its environment.

In Figure 9-5, you can see two different environments in which you can deploy the middleware installation.

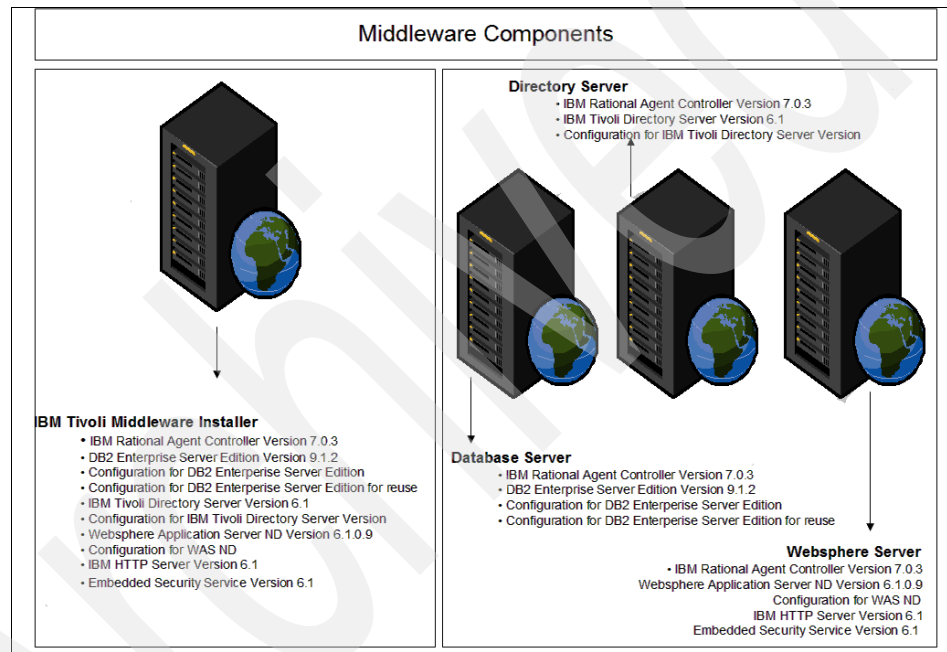


Figure 9-5 Middleware components

We also recommend following the installation process using the default system users and default applications ports for CCMDB V7.1 in order to avoid future problems with the internal structure.

There are two ways to start the installation:

- Copy the middleware install images from the source media to a specified directory.

Select this option to copy the CCMDB middleware images from the product media to a directory that you will specify.

- Specify a directory containing all the required middleware install images.
Select this option if you intend to specify a file system directory that already contains all of the CCMDB middleware installation images.

Note: The middleware installation on the same server takes about two hours when **Specify a directory...** is chosen.

For Windows: After the middleware installation is finished, you must change the startup option of these services, as shown in Figure 9-6:

- DB2 - DB2COPY1 - CTGINST1-0
- IBM Tivoli Directory Admin Daemon V6.1
- IBM Tivoli Directory Server Instance V6.1

Then reboot the system.

DB2 - DB2COPY1 - CTGINST1-0	Allows appl...	Started	Manual	.\\db2admin
DB2 - DB2COPY1 - DB2-0	Allows appl...	Started	Automatic	.\\db2admin
DB2 - DB2COPY1 - IDSCCMDB	Allows appl...	Started	Manual	Local System
DB2 Governor (DB2COPY1)	Collects st...		Manual	.\\db2admin
DB2 License Server (DB2COPY1)	Monitors D...		Manual	Local System
DB2 Management Service (DB2COPY1)	Manages D...	Started	Automatic	Local System
DB2 Remote Command Server (DB2COPY1)	Supports r...	Started	Automatic	.\\db2admin
DB2 Security Server (DB2COPY1)	Authentica...	Started	Automatic	Local System
DB2DAS - DB2DAS00	Supports lo...	Started	Automatic	.\\db2admin
DCOM Server Process Launcher	Provides la...	Started	Automatic	Local System
DHCP Client	Registers a...	Started	Automatic	Network S...
Distributed File System	Integrates ...		Manual	Local System
Distributed Link Tracking Client	Enables cli...	Started	Automatic	Local System
Distributed Link Tracking Server	Enables th...		Disabled	Local System
Distributed Transaction Coordinator	Coordinate...	Started	Automatic	Network S...
DNS Client	Resolves a...	Started	Automatic	Network S...
Error Reporting Service	Collects, st...	Started	Automatic	Local System
Event Log	Enables ev...	Started	Automatic	Local System
File Replication	Allows files...		Manual	Local System
Help and Support	Enables He...	Started	Automatic	Local System
HTTP SSL	This servic...		Manual	Local System
Human Interface Device Access	Enables ge...		Disabled	Local System
IBM ADE Service	Provides a...	Started	Automatic	Local System
IBM HTTP Administration 6.1	IBM_HTTP...	Started	Automatic	Local System
IBM HTTP Server 6.1	IBM_HTTP...	Started	Automatic	Local System
IBM Rational Agent Controller		Started	Automatic	Local System
IBM Tivoli Directory Admin Daemon V6.1 - idscmdb		Started	Manual	Local System
IBM Tivoli Directory Server Instance V6.1 - idscmdb		Started	Manual	Local System
IBM WebSphere Application Server V6.1 - ctgCellManager01	Controls th...	Started	Automatic	Local System

Figure 9-6 Windows services startup options

After the reboot, the same services have to be started step by step manually. Once all services are started, start the following WebSphere components:

► Start Node

```
<WAS_HOME>\profiles\ctgAppSrv01\bin\startNode.bat
```

► Start MXServer

```
<WAS_HOME>/profiles/ctgAppSrv01/bin/startServer.bat MXServer  
-username <username> -password <password>
```

For Linux only:

After the installation, you would find that all the WebSphere components and databases are running.

Note: When you reboot the Linux machine, the LDAP (TDS) server process and databases may not start automatically.

After the reboot of the Linux machine, follow these steps:

1. Start the databases with the following commands:

```
su - idscmdb -c db2start  
su - ctginst1 -c db2start
```

2. Start the LDAP server process with the following command:

```
<LDAP directory>/ldap/V6.1/sbin/ibmslapd
```

3. Start the WebSphere processes with the following commands:

– Start Manager

```
<WAS_HOME>/profiles/ctgDmg01/bin/startManager.sh
```

– Start Node

```
<WAS_HOME>/profiles/ctgAppSrv01/bin/startNode.sh
```

– Start webserver1

```
<WAS_HOME>/profiles/ctgAppSrv01/startServer.sh webserver1
```

– Start MXServer

```
<WAS_HOME>/profiles/ctgAppSrv01/startServer.sh MXServer
```

To verify that all Middleware components are running, log in to the WebSphere Application Server Administrative Console, as shown in Figure 9-7 on page 199.

To start the administrative console, open a browser window and enter the following URL:

`http://<machine_name>:9060/ibm/console`

Enter wasadmin as the administrative user ID and the password to log in, if one is required.

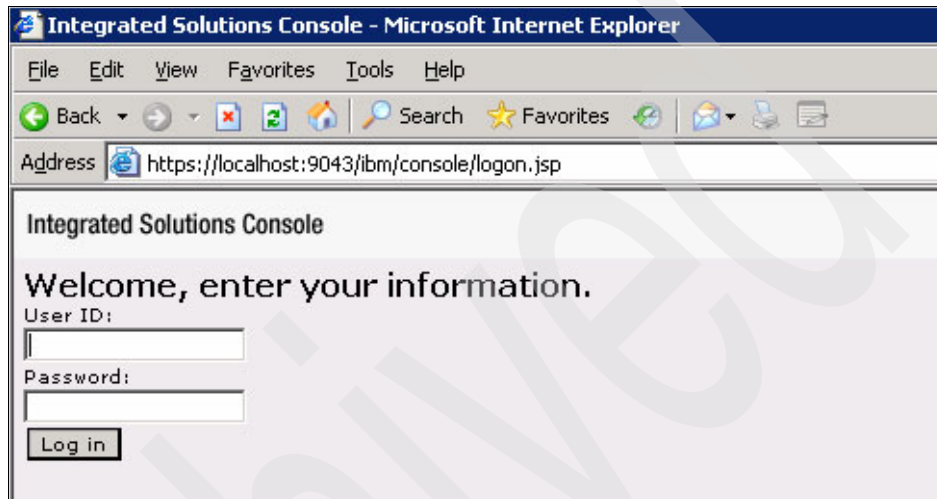


Figure 9-7 Administrative Console login

9.6 Tivoli Application Discovery and Dependency Manager Installation

If you plan for your CCMDB deployment to include IBM Tivoli Application Dependency Discovery Manager, it should be installed prior to running the CCMDB installation program (see Figure 9-3 on page 190).

The TADDM installation program must be launched directly from the system on which it will be installed. Remote installation is not supported.

We recommend installing the database software and instance before installing the TADDM application.

To launch the TADDM installation program from the CCMDB launchpad, log in as Administrator on Windows or root on UNIX systems.

To start the launchpad (Figure 9-8) from the DVD titled CCMDB V7.1, navigate to the root directory of the product disc or the downloaded installation image, and run the following commands:

- ▶ On Windows: **1aunchpad.exe**
- ▶ On UNIX: **1aunchpad.sh**

For further installation steps, refer to the *Planning and Installing Change and Configuration Management Database*.

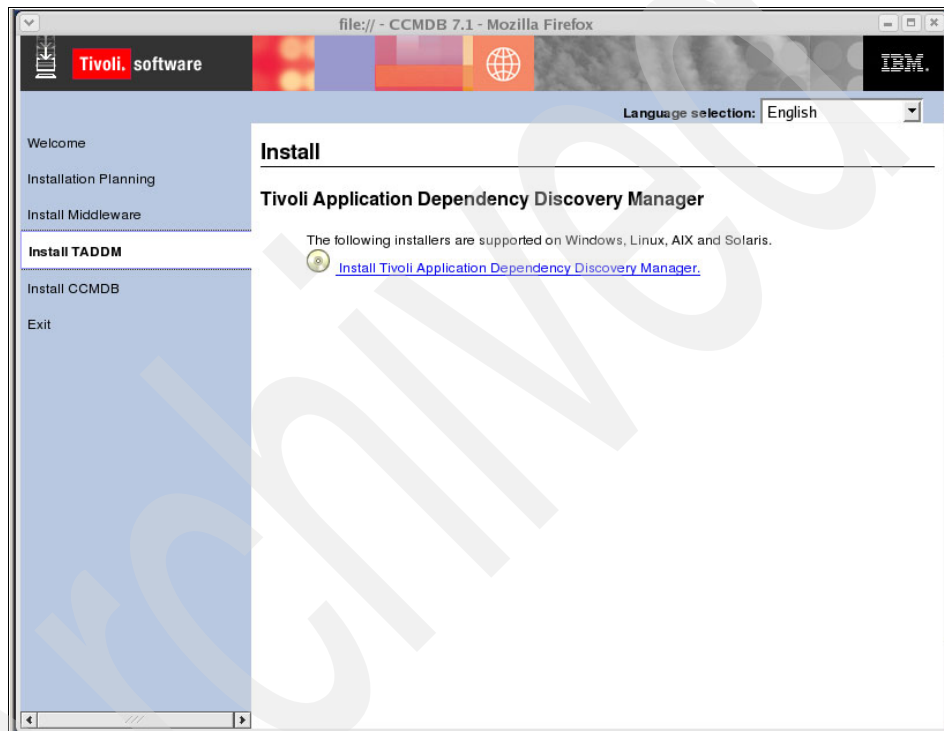


Figure 9-8 TADDM installation initial window

9.7 Change and Configuration Management Database installation

Before installing the CCMDB components, ensure that the middleware and TADDM are successfully installed and running as shown in Figure 9-3 on page 190. We advise having worksheets describing the TADDM and middleware components that were installed. Start the installation by using the launchpad Interface, as shown in Figure 9-9 on page 201.

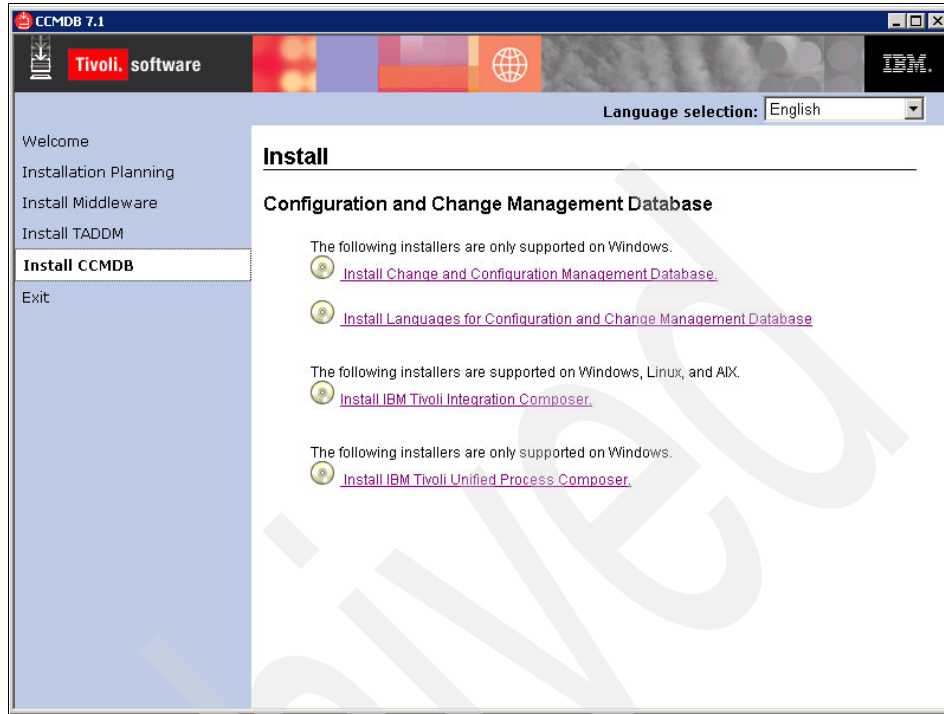


Figure 9-9 CCMDB installation

This interface provides you with the options to install CCMDB (Maximo Database and Change and Configuration Management process), Language Pack, Integration composer, and ITUP Composer.

Note: Installation of CCMDB, Language Pack, and ITUP can be installed through a Windows administration workstation only. The installation of the Language Pack and ITUP is intuitive and simple, so this section does not contain these installation instructions. You may refer to the *Planning and Installing Change and Configuration Management Database*.

Note: To install the CMDB components, follow the installer instructions. The following section describes only the important steps that you would be doing while installing the CMDB components.

The CCMDB installation program provides an interface for installing and deploying CCMDB, which includes the Maximo application and the Change and Configuration Management process managers. Follow these steps to do the installation:

1. The Import Middleware Configuration Information window gives you the option to import the middleware configuration information automatically. In order to do so, you need to provide information about the host name where the middleware installation workspace is located, as shown in Figure 9-10.

IBM Tivoli base services

Import Middleware Configuration Information

Check this box to import the configuration information recorded by the Tivoli middleware installation program. Provide the information for the machine where the Tivoli middleware installation workspace is located. The user ID must have remote access privileges to the machine.

☒ Import middleware configuration information

Host name

User ID

Password

Confirm password

Workspace location

InstallAnywhere by Macrovision

Cancel Previous Next

Figure 9-10 Import middleware configuration

2. The Choose Deployment window gives you the option to decide whether the installation is to be done on a single or on a multiple server deployment scenario, as show in Figure 9-11 on page 203.
 - Simple
Select **Simple** if you want to deploy all CCMDB components on a single system. This deployment option is typically only used for demonstration, proof-of-concept, or training purposes.
 - Custom
Select **Custom** if you want to deploy CCMDB components across several systems. This deployment option is typically used in a production environment.

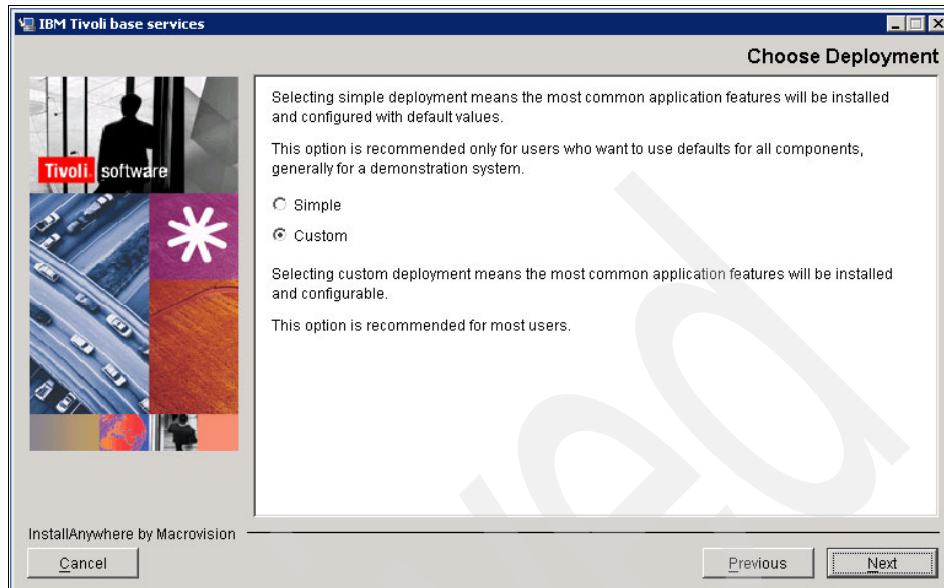


Figure 9-11 Choose deployment option

3. In the Choose Install Folder window, you need to specify the path of the local Windows administration workstation where the CCMDB client will be installed, as shown in Figure 9-12.

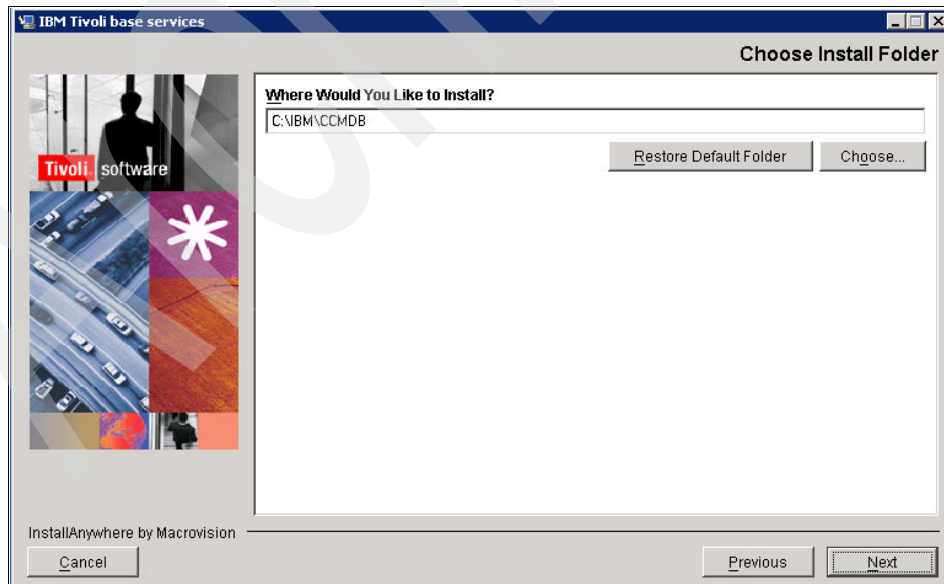
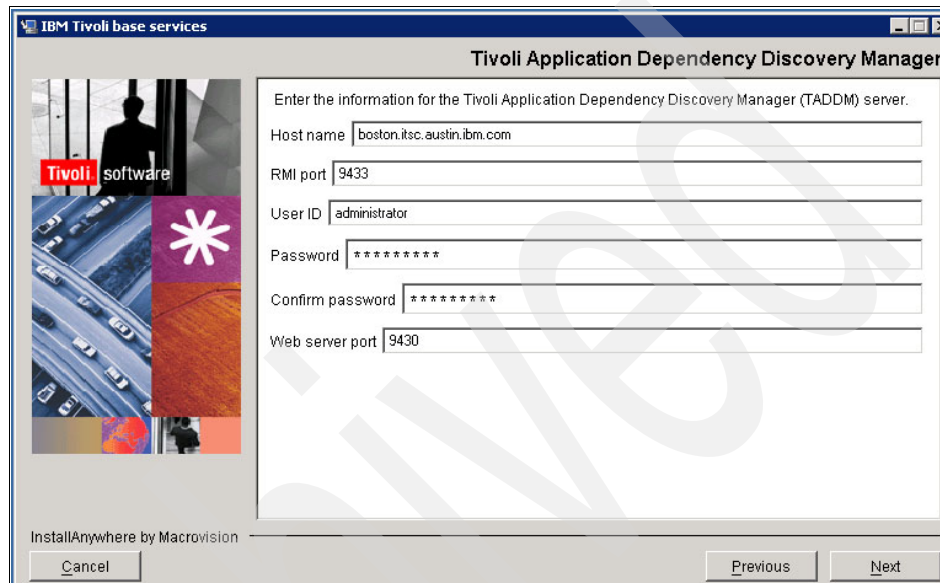


Figure 9-12 Choose installation folder

In the Tivoli Application Dependency and Discovery Manager window, specify the information regarding the TADDM server, as shown in Figure 9-13.

Tip: The default password of the user administrator on TADDM is collation.



IBM Tivoli base services

Tivoli Application Dependency Discovery Manager

Enter the information for the Tivoli Application Dependency Discovery Manager (TADDM) server.

Host name

RMI port

User ID

Password

Confirm password

Web server port

InstallAnywhere by Macrovision

Figure 9-13 Setting the TADDM password

This section does not show all the windows that you would see during the CCMDB installation. For the windows not shown here, you can specify the default values or values used during the middleware and TADDM installation. For further information, refer to *Planning and Installing Change and Configuration Management Database*.

4. In the Maximo window, you need to specify the path of the local Windows administration workstation where the Maximo client component would be installed (Figure 9-14).

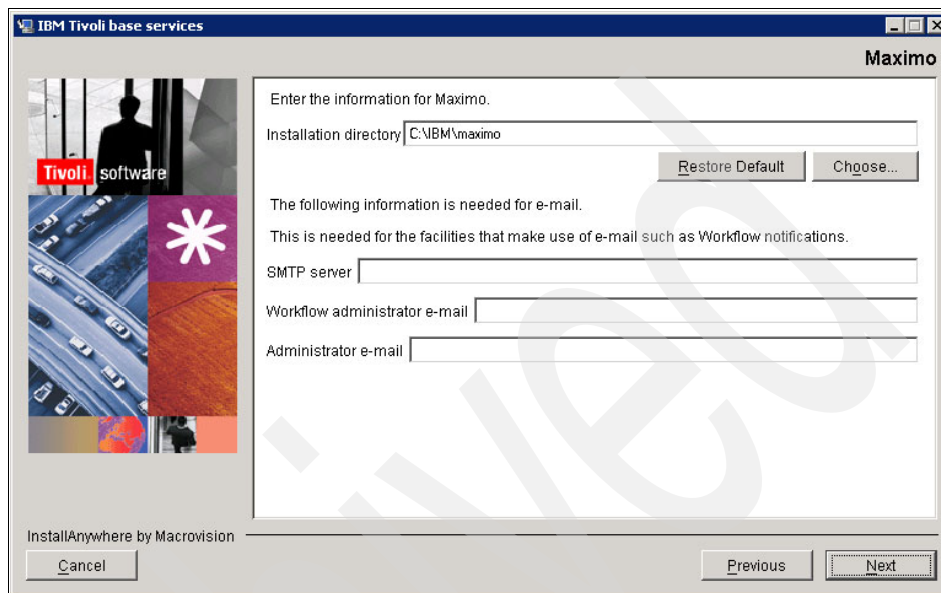


Figure 9-14 Maximo installation directory

5. After completing all the installation steps, you will see the Installing IBM Tivoli Base Services window (Figure 9-15).

Attention: It may take more than an hour to complete the whole installation.



Figure 9-15 Installation status

9.8 CCMDB post installation steps

Once you have successfully installed and configured the CCMDB components, there are several data configuration tasks you must complete prior to using CCMDB.

The following post installation steps have to be completed to set up an initial organization structure:

- ▶ Initial data configuration
- ▶ Data collecting and importing
- ▶ Synchronization data
- ▶ CCMDB language pack installation program overview

9.8.1 Sign in with the default user ID

User management is managed through the directory server that is configured to use CCMDB. When first installed, CCMDB contains the default user IDs shown in Table 9-1, which are members of the specified security group.

Important: Before you begin this procedure, ensure that you have the users and groups shown in Table 9-1 created in the LDAP repository.

Table 9-1 CCMDB users and groups

User	Groups
wasadmin	maxadmin
maxadmin (maxadminuser for MS Active Directory)	maxadmin
mxintadm	maxadmin
maxreg	maxadmin

The default password for each user ID is the same as the user name (for example, maxadmin is both the user name and default password).

Note: User names and passwords are case sensitive. The default user names and passwords are lowercase

Signing in using a default user ID

To sign in into Maximo, complete the following steps:

1. Open a browser window.
2. Enter the URL `http://<host name>/maximo`.
3. Enter the user name maxadmin (lower case).
4. Enter the password maxadmin (lower case) and press Enter.

The application displays an empty start center, as shown in Figure 9-16.

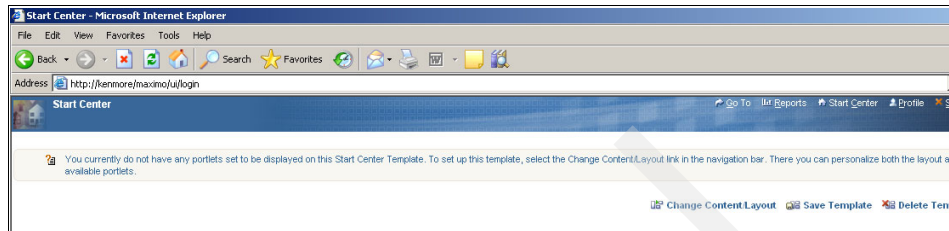


Figure 9-16 Initial start center

Important: Always use the Maximo icons to log out. Do not use the original browser icons.

9.8.2 Granting universal access to the MAXADMIN group

The MAXADMIN user ID belongs to the MAXADMIN security group. By default, this group has very limited security access. You must grant the MAXADMIN group universal security access to all applications and options in order to configure the software.

To grant universal access, complete the following steps:

1. Open the Security Groups application by selecting **Go To** → **Security** → **Security Groups**, as shown in Figure 9-17.

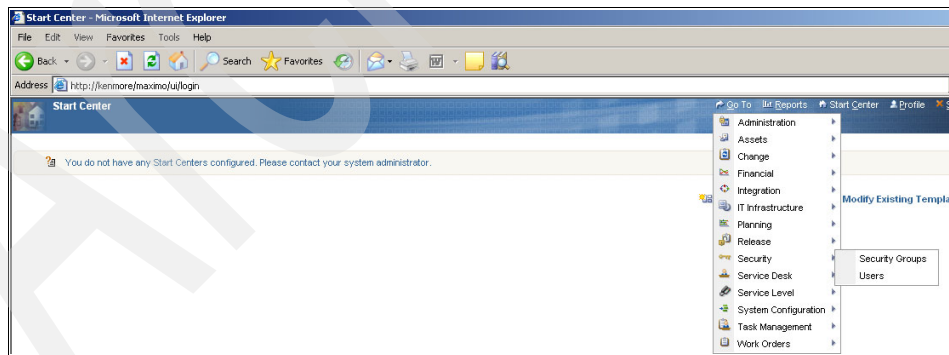


Figure 9-17 Opening Security groups application

2. On the List tab, in the Group Filter field (default cursor position), press Enter to display the default security groups, as shown in Figure 9-18 on page 209.

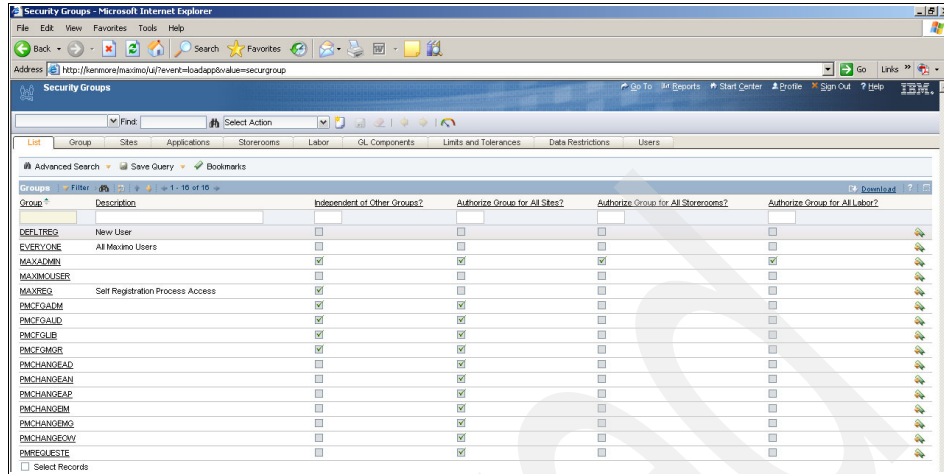


Figure 9-18 Listing default security groups

3. Click the **MAXADMIN** group (Figure 9-19).

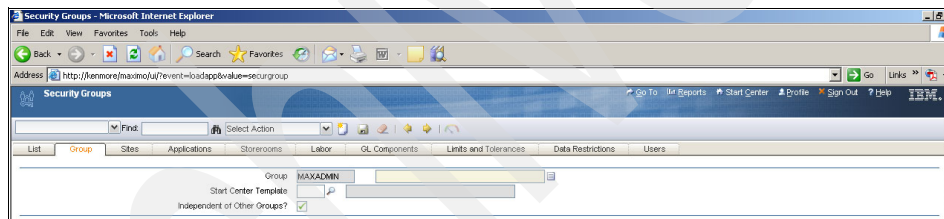


Figure 9-19 Selecting MAXADMIN group

- Click the **Applications** tab (Figure 9-20).

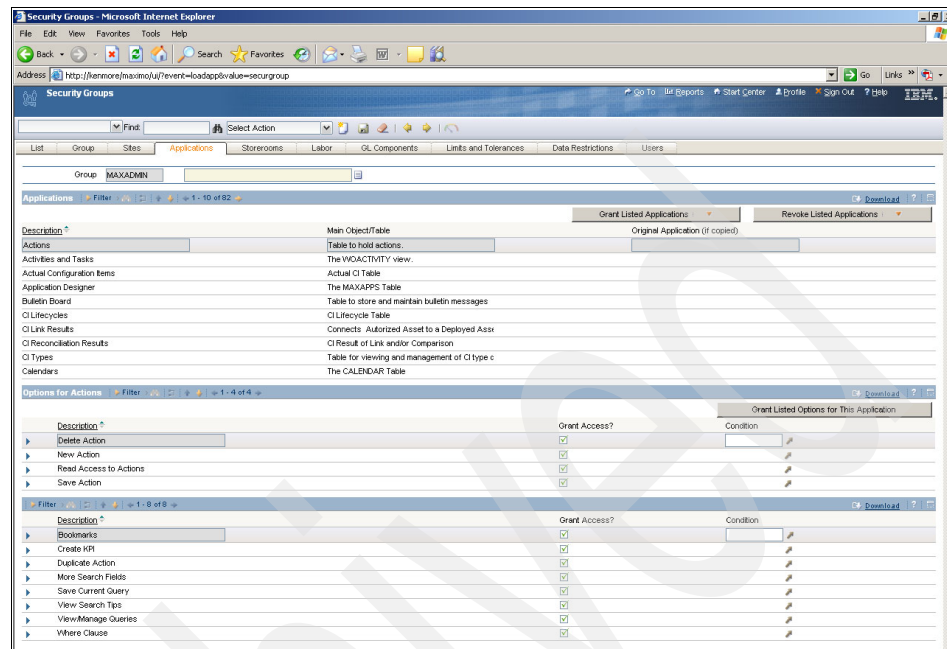


Figure 9-20 Applications tab

- Click **Grant Listed Applications**.
- Click **All Above**.
- Click **OK** and **Accept Grant all access for listed applications...** (Figure 9-21).

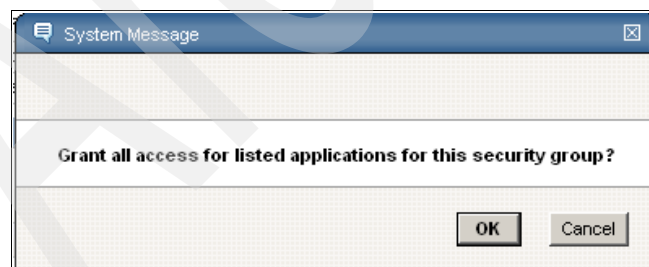


Figure 9-21 Granting access

- Under the Options of the Chart of Accounts section, click **Grant List Options for This Application**.

9. Click **OK** and **Accept Grant listed options for Actions...** (Figure 9-22).

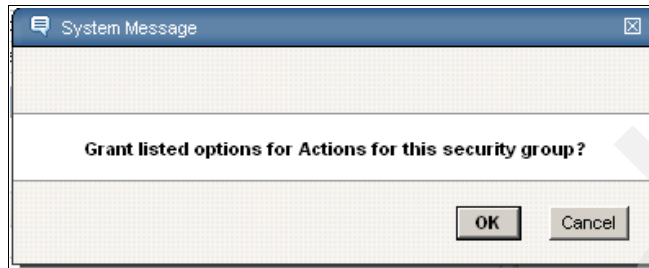


Figure 9-22 Accepting the grant

10. Save the group (Figure 9-23).

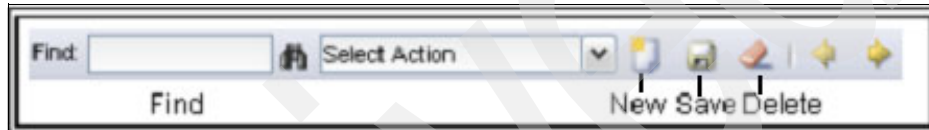


Figure 9-23 Save the group

11. Sign out (Figure 9-24).

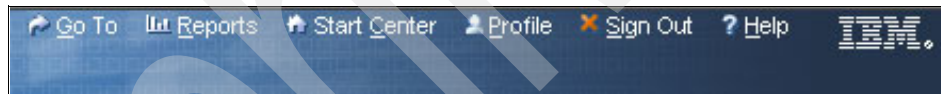


Figure 9-24 Sign out

12. Restart MXServer using a command-line interface or WebSphere UI:

- Command-line interface:

```
Cd <WebSphere-Path>\AppServer\profiles\ctgAppSrv01\bin\  
stopserver.bat MXServer -username <WebSphere-admin> -password  
<password>  
startserver.bat MXServer -username <WebSphere-admin> -password  
<password>
```

– WebSphere UI:

Stop and start the MXServer using the WebSphere Admin Console, as shown in Figure 9-25.

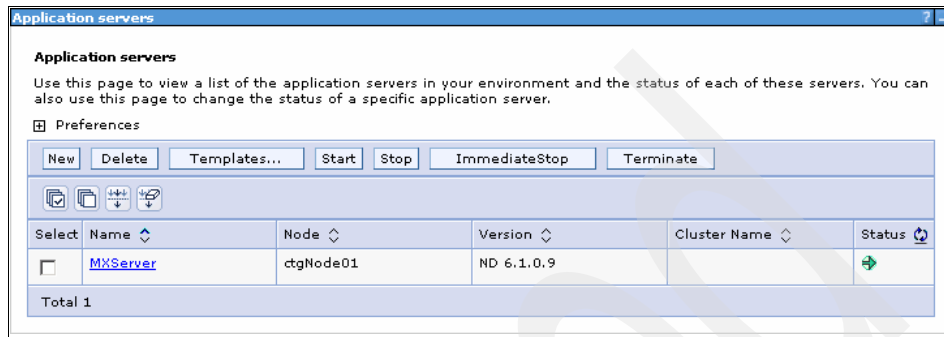


Figure 9-25 WebSphere Admin Console used to stop and start MXServer

9.8.3 Update User Security to view inactive organizations and sites

You must ensure that the maxadmin user has permission to view inactive organization and sites.

To grant maxadmin permission to view inactive organization and sites, complete the following steps:

1. Open the Security application for Users by selecting **Go To → Security → Users**. The window shown in Figure 9-26 will display.

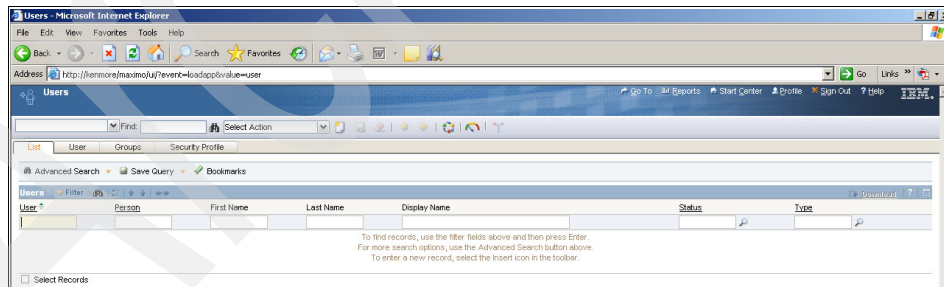


Figure 9-26 User security application

2. Search for the maxadmin user. Press Enter in the User field to get the User list (Figure 9-27 on page 213).

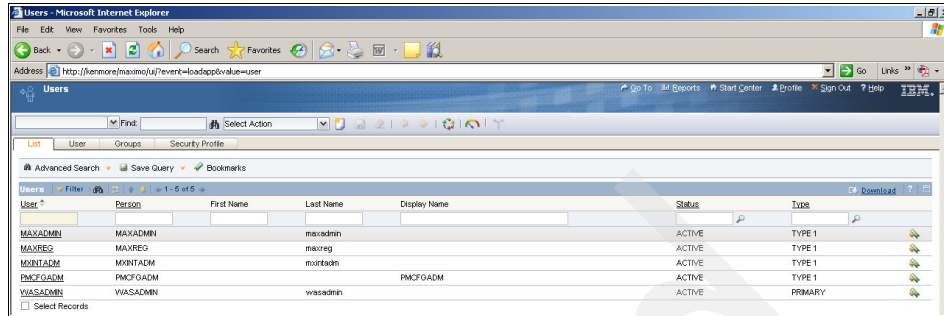


Figure 9-27 User list

3. Click the user name maxadmin to view detailed information (Figure 9-28).

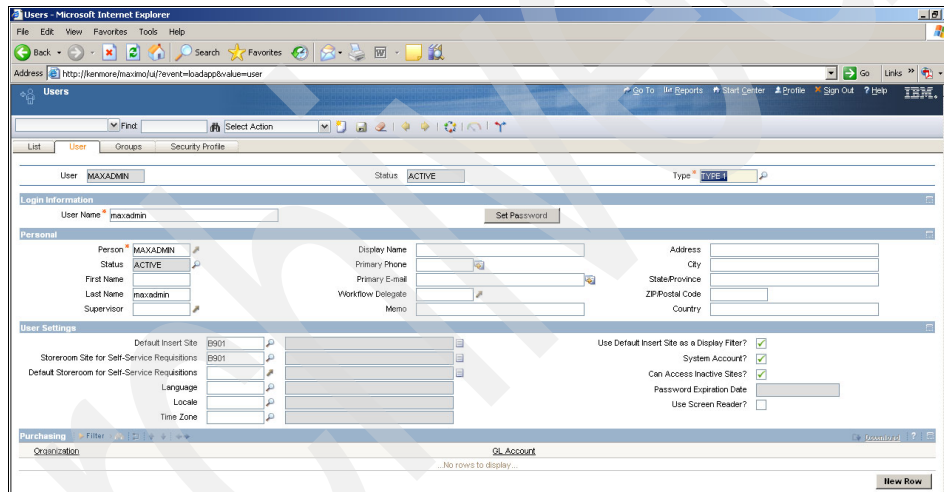


Figure 9-28 Viewing user detail

4. Select the **Can Access Inactive Sites?** check box.
5. Click **Save**.

9.8.4 Create currency codes

You must define a currency code for an organization.

To define a currency code for an organization, complete the following steps:

1. Open the Currency Code application for Users by selecting **Go To** → **Financial** → **Currency Code**. The window shown in Figure 9-29 will display.

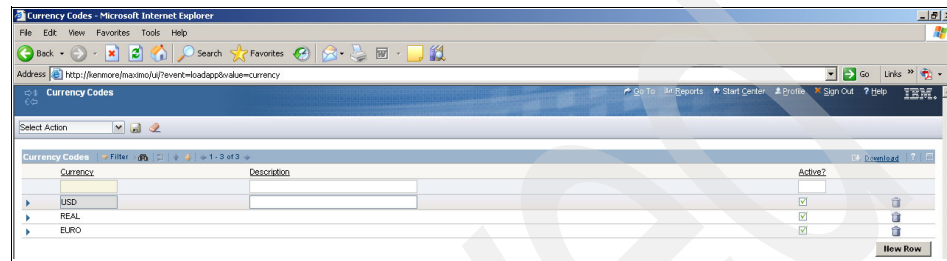


Figure 9-29 Accessing Currency Code application

2. Click **New Row** and enter a currency name, for example, USD or EUR, as shown in Figure 9-30.

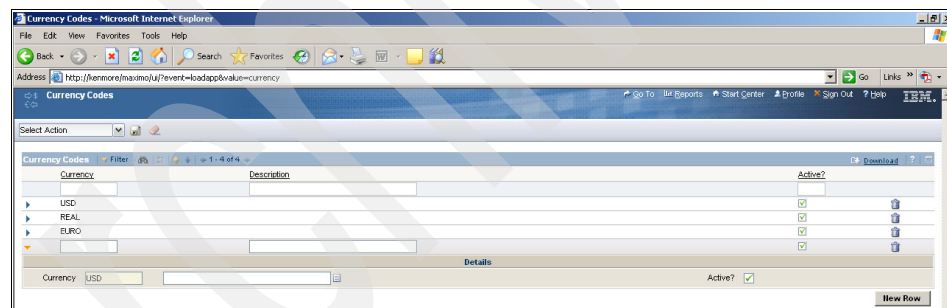


Figure 9-30 Selecting currency

3. Click **Save**.

9.8.5 Create item and company sets

You must define item and company sets for an organization. To define a currency code for an organization, complete the following steps:

1. Open the Sets application for Users by selecting **Go To** → **Administration** → **Sets**.
2. Click **New Row**.

3. Enter the company set name, for example, IT Items, and enter ITEM in the Type field, as shown in Figure 9-31.

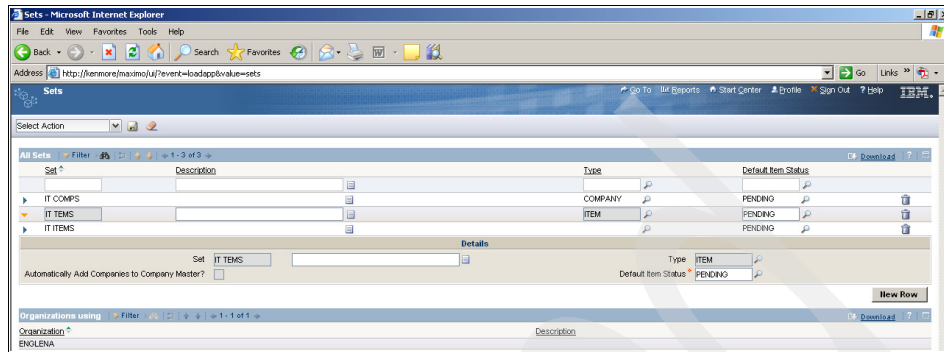


Figure 9-31 Entering IT items

4. Click **New Row** again.
5. Enter company set name, for example, IT Comps, and enter COMPANY in the Type field, as shown in Figure 9-31.
6. Click **Save**.

9.8.6 Create an organization

You must define at least one organization for CCMDDB.

To define an organization, complete the following steps:

1. Open the Organization application by selecting **Go To → Administration - → Organizations**.
2. Click the **New Organizations** icon in the toolbar.

3. Configure the following parameters, as shown in Figure 9-32:
 - a. Organization Field - ENGLINA.
 - b. Base Currency 1 and 2 as you defined.
 - c. Item set as you defined.
 - d. Company set as you defined.
 - e. Enter the default item status of PENDING in the Default Item Status field.

The screenshot shows the 'Organizations' web application in Microsoft Internet Explorer. The 'Organization' tab is selected, and the 'ENGLINA' organization is being configured. The form includes the following fields:

Organization	ENGLINA
Base Currency 1	USD
Base Currency 2	REAL
Item Set	IT ITEMS
Company Set	IT COMPS
Default Item Status	PENDING
Clearing Account	
Active?	☒

Figure 9-32 Creating an organization

4. Click the Site tab and finish the following steps:
 - a. Click **New Row**.
 - b. Enter a site name in the Site field, for example, B901, as shown in Figure 9-33.
 - c. Click **Save**.

The screenshot shows the 'Organizations' web application in Microsoft Internet Explorer. The 'Sites' tab is selected, and the 'ENGLINA' organization is selected. The 'Sites' table is displayed with the following columns: Site, Description, Site to Address Code, Site to Address Code, and Active?. A new row is being added with the site name 'B901'.

Site	Description	Site to Address Code	Site to Address Code	Active?
B901				☐

A 'New Row' button is visible at the bottom right of the table.

Figure 9-33 Entering a Site name

9.8.7 Create a general ledger account component

You must create a general ledger account component for CCMDB.

To create a general ledger account component, complete the following steps:

1. Open the Database Configuration application by selecting **Go To → System Configuration → Platform Configuration → Database Configuration**.
2. Select **GL Account Configuration** from the Select Action drop-down menu, as shown in Figure 9-34.

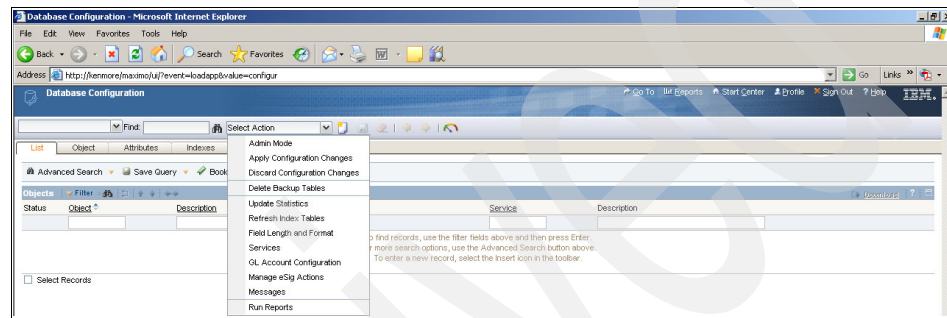


Figure 9-34 General Ledger application

3. Enter the following parameters, as shown in Figure 9-35:
 - a. Click **New Row**.
 - b. Enter the component Name MYCOMPONENT in the Component field.
 - c. Numerical length for the component: 5.
 - d. Type of component: ALN.
 - e. Click **OK**.

GL Account Configuration

GL Account Configuration Filter 1 - 1 of 1 Download

Component	Length	Type	Required?	Screen Delimiter
MYCOMPONENT	5	ALN	☐	

New Row

OK Cancel

Figure 9-35 Account configuration

9.8.8 Create a general ledger account

You must create a general ledger account for CCMDDB.

To create a general ledger account, complete the following steps:

1. Open the Chart of Accounts application by selecting **Go To → Financials → Chart of Accounts**, as shown in Figure 9-36.

Chart of Accounts - Microsoft Internet Explorer

Address: http://memore/maximolui/?event=loadapp&value=chrtacct

Chart of Accounts

Select Action: Update Database, Financial Periods, GL Component Maintenance, Organization Default Accounts, Company-Related Accounts, External Labor Control Accounts, Resource Codes, Validation Options, Run Reports

Description	Type	Active?
		☒
MYGLC		☒

New Row

Figure 9-36 Chart of Accounts application

2. Click the name of your defined Organization to select it, for example, ENGLENA.
3. Select **GL Component Maintenance** from the Select Action drop-down menu.
4. Add a GL Component value and then click **OK**, for example, MYGLC, as shown in Figure 9-37.
5. Click **New Row**.
6. Select your general ledger account.
7. Click **Save**.

Organization: ENGLENA

Components: Filter 1 - 1 of 1 Download

Component
MYCOMPEN

GL Component Values for MYCOMPEN: Filter 1 - 1 of 1 Download

GL Component Value	Description	Active?
MYGLC	MYGLC	☒

New Row

OK Cancel

Figure 9-37 General ledger component

8. Open the Organization Application by selecting **Go To → Administration → Organizations**, as shown in Figure 9-38.

."/>

Organizations - Microsoft Internet Explorer

Address: http://[remotehost]/event=loadappvalue=multisite

Organizations: Go To Reports Start Center Profile Sign Out Help

Find: Select Action

Organization	Base Currency 1	Base Currency 2	Item Set	Company Set	Default Item Status	Clearing Account	Active?
ENGLENA	USD	REAL	IT EMS	IT COMPS	PENDING	MYGLC	☒

Figure 9-38 Organization application

9. Click the organization name you created, for example, ENGLENA.
10. From the Clearing Account Field, select the General Ledger Account you just created, that is, **MYGLC**.
11. Select **Active**.
12. Click **Save**.

9.8.9 Create default insert site

You must create a default insert site for CCMDB.

To create a default insert site, complete the following steps:

1. Open the Users application by selecting **Go To → Security → Users**.
2. Search for MAXADMIN and then select it to open the record for MAXADMIN.
3. Enter a site you created earlier in the Default Insert Site field, for example, B901, as shown in Figure 9-33 on page 216.
4. Enter a site you created earlier in the Storeroom Site for self-Service Requisitions field, for example, B901, as shown in Figure 9-33 on page 216.
5. Click **Save**.

The screenshot shows the IBM Tivoli CCMDB Users application interface. The user record for MAXADMIN is displayed. The 'User Settings' section is expanded, showing the 'Default Insert Site' field set to 'B901'. Other fields like 'Storeroom Site for Self-Service Requisitions' and 'Default Storeroom for Self-Service Requisitions' are also set to 'B901'. The 'Personal' section shows the user's name as 'MAXADMIN' and status as 'ACTIVE'. The 'Login Information' section shows the user name 'maxadmin' and a 'Set Password' button. The 'Purchasing' section shows a 'Filter' button and a 'No rows to display' message.

Figure 9-39 Create default site

6. Restart the MXServer.

9.8.10 Create a Work Type

You must create a Work Type for CCMDB.

To create a Work Type, complete the following steps:

1. Open the Organization application by selecting **Go To → Administration → Organizations**, as shown in Figure 9-40.

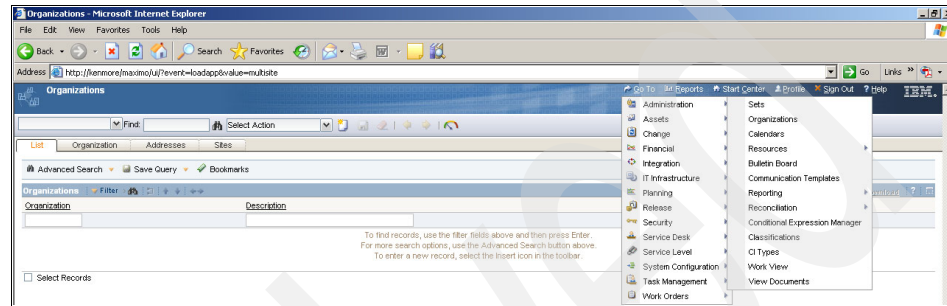


Figure 9-40 Starting the organization application

2. Search for the organization you created, for example, ENGLENA.
3. Click the name of the organization to pen the record for that organization.
4. Select **Work Order Options → Work Type** from the Select Action drop-down menu, as shown in Figure 9-41.

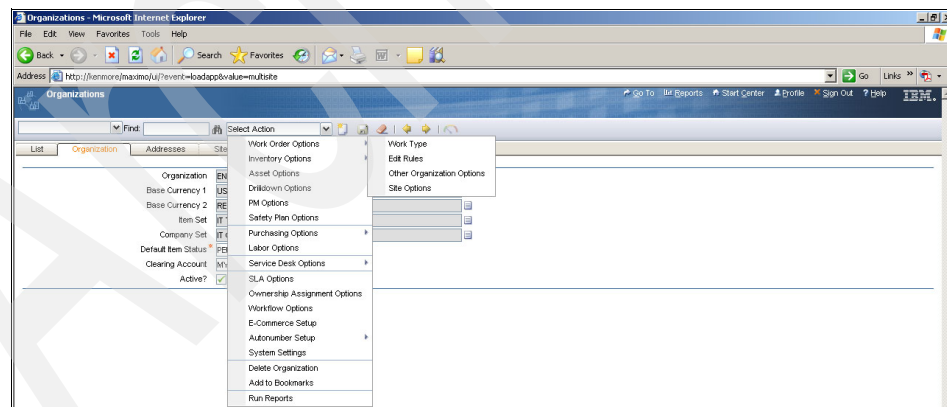


Figure 9-41 Accessing Work Type definition

5. Click **New Row**. The window shown in Figure 9-42 should appear.
6. Select **PMCFGWO** as the Work Order class.
7. Set the Work Type as AR.
8. Set Start Status as INPRG.
9. Set Complete Status as COMP.

Figure 9-42 Setting Work Type information

10. Click **OK**.
11. Click **Save**.
12. Sign out as MAXADMIN and sign in as MAXADMIN.

The post installation steps are finished.

9.8.11 CCMDB post installation steps: Solution Installer Command-Line Interface

The Process Solution Command-Line Interface enables you to query, install, upgrade, and uninstall process solution packages, which can consist of process modules and integration modules.

You find a detailed description in Chapter 11 of the *Planning and Installing Change and Configuration Management Database*.

In this case, we want to check if all the installation packages are available for CCMDB. Our focus is limited to the command-line interface.

The collection of supported parameters for the Process Solution Command-Line Interface are described in Table 9-2.

Table 9-2 Process Solution Command-Line Interface parameters

Parameter name	Description
-action	Specifies the function or software life cycle operation to perform.
-dbpwd	Specifies the password of the database user ID that is used to access the Maximo database.
-dbuser	Specifies the database user ID that is used to access the Maximo database.
-fixid	Specifies the unique identifier of an interim fix or patch that you want processed.
-force	Specifies whether to continue on with a deployment operation even if there are one or more unsatisfied requirements associated with the package being processed.
-license	Specifies if you want to automatically accept the license agreement or be prompted for the acceptance or rejection of the license agreement by using one of the following values: accept or prompt.
-loadlanguages	Specifies whether optional Language Support files for the package should be loaded into the Maximo database.
-loadsampdata	Specifies whether to load sample or demonstration data associated with the package being processed.

9.9 IBM Tivoli Integration Composer installation

The IBM Tivoli Integration Composer connects asset inventory and system management tools to the asset repository, allowing organizations to benefit from existing inventory data collection tools.

This solution aggregates, integrates, and normalizes all data into one central repository, streamlining enterprise IT management reporting and decision support functions.

Data from different systems, management tools, business units, or distributed locations can be easily consolidated and updated, ensuring that the most current hardware and software resource information is available to asset managers and service desk operations.

Tivoli Integration Composer combines intuitive application administration tools with powerful out-of-the-box adapters.

To install the ITIC component, follow these instructions. The following section of this chapter describes only the important steps that you would be doing while installing ITIC component:

1. In the Choose Install Folder window, you need to specify the path on the remote server where the ITIC component will be installed. See Figure 9-43. For example, on Linux, this value might be `/root/Integration_Composer`.

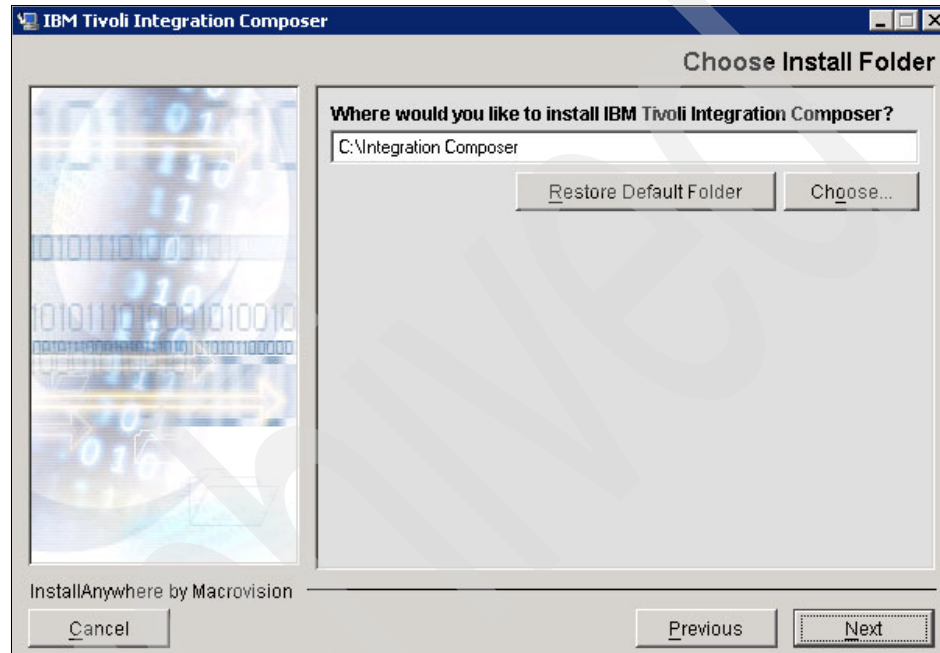


Figure 9-43 ITIC installation

- In the Choose Install Folder window, you need to specify the path on the remote server where the ITIC component will be installed. See Figure 9-44 on page 226.

IBM Tivoli Integration Composer

Database Login Information

The following information is needed to connect to the database. Please enter all of the database information requested below.

At this time, you may choose to skip this step and enter the Database information manually by editing the fusion.properties file.

Database Server Name:	gamma.itsc.austin.ibm.com
Port Number:	50005
Database Name:	maxdb71
Database Username:	maximo
Database/Schema Owner:	maximo

InstallAnywhere by Macrovision

Figure 9-44 ITIC database information

This section does not show all the window that you would see during the ITIC installation. For the windows not mentioned here, you can specify the default values. For further information, refer to the *Planning and Installing Change and Configuration Management Database*.

After completing all the installation steps, you see the Installing IBM Tivoli Base Services window, as shown in Figure 9-45.

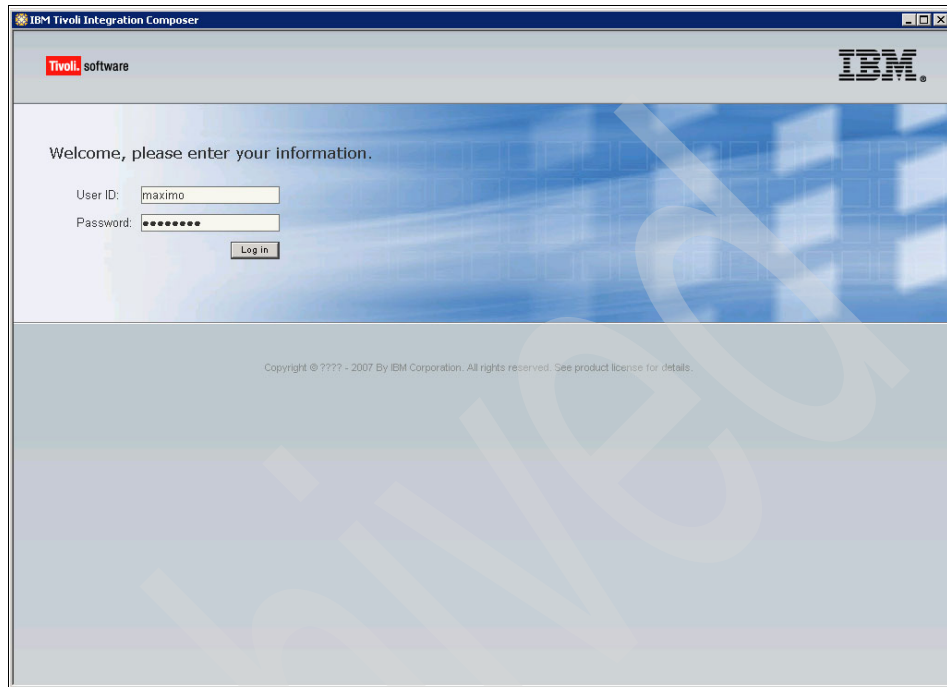


Figure 9-45 ITIC login window

To set up the ITIC adapter, refer to Chapter 10, “TADDM and Process Layer integration” on page 231.



Part 4

Getting started

Archived

TADDM and Process Layer integration

This chapter outlines the steps involved in migrating Configuration Items from a Discovery Engine like TADDM to the CCDBVM Process Layer. We introduce the main concepts of Integration Composer and Adapter and explain the usage and difference between the two. In addition, the chapter will walk you through the end-to-end integration scenario from configuration item (CI) data imported through Discovery Library Adapter (DLA) in the Tivoli Application Dependency Discovery Manager (TADDM) database and later migrated to the process layer database through Integration Composer and Adapter.

The migration process is achieved through Integration Composer and Adapter. Integration Composer is able to migrate configuration items from several Operational Management Products (OMPs).

These are the main topics covered in this chapter:

- ▶ Integration Composer overview
- ▶ IBM Tivoli Integration Adapters
- ▶ Configuring the Integration Composer
- ▶ Configuring the TADDM Adapter
- ▶ Importing OMP data through DLA into TADDM
- ▶ Mapping data sources
- ▶ Validating migrated data in CCMDB

Note: For details on how to navigate and work with Integration Composer, refer to the *IBM Tivoli Integration Composer System Administrator's Guide*.

10.1 End-to-end data discovery and migration

Figure 10-1 shows the end-to-end data migration plan.

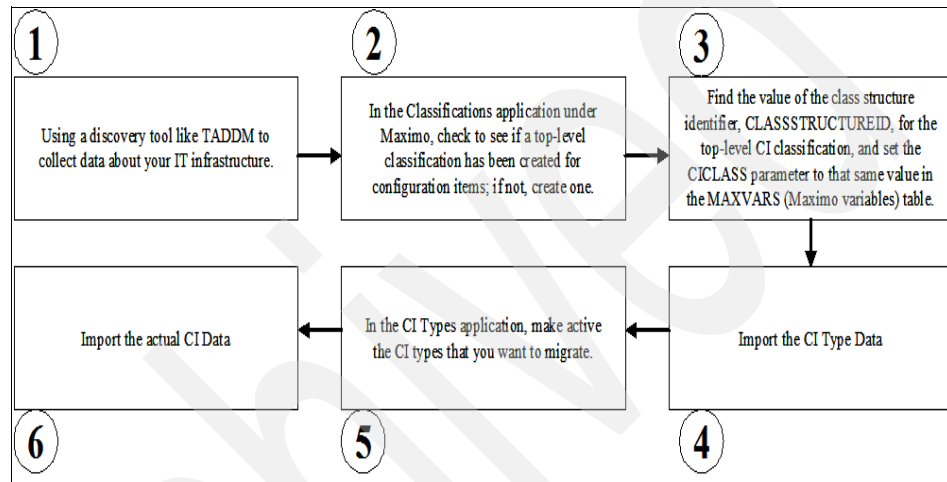


Figure 10-1 End-to-end data migration plan

Performing the complete end-to-end collection and importing of configuration item data involves the use of multiple software tools. In addition to Integration Composer and the integration adapter for TADDM, these software tools include TADDM itself (for discovery), as well as the Process Layer components of CCMDB. Although all the basic steps for collecting and importing data are listed here, only the steps involving the importing of data by Integration Composer are described in detail in this chapter.

For more information about the other steps, refer to the documentation or help provided with TADDM, CCMDB, and the applicable CCMDB applications.

10.2 Integration Composer overview

Integration Composer is a stand-alone integration application that migrates data from source to target. With Integration Composer, an enterprise can aggregate data collected by discovery tools and integrate it into a target database, creating a central repository for enterprise IT asset management, reporting, and decision support.



Figure 10-2 Integration Composer application

To gather data, a discovery tool, such as IBM Tivoli Configuration Manager or Microsoft SMS, scans computers, network devices, and network printers deployed in an enterprise and records information about the hardware and software installed on those assets (see Figure 10-3 for more details).

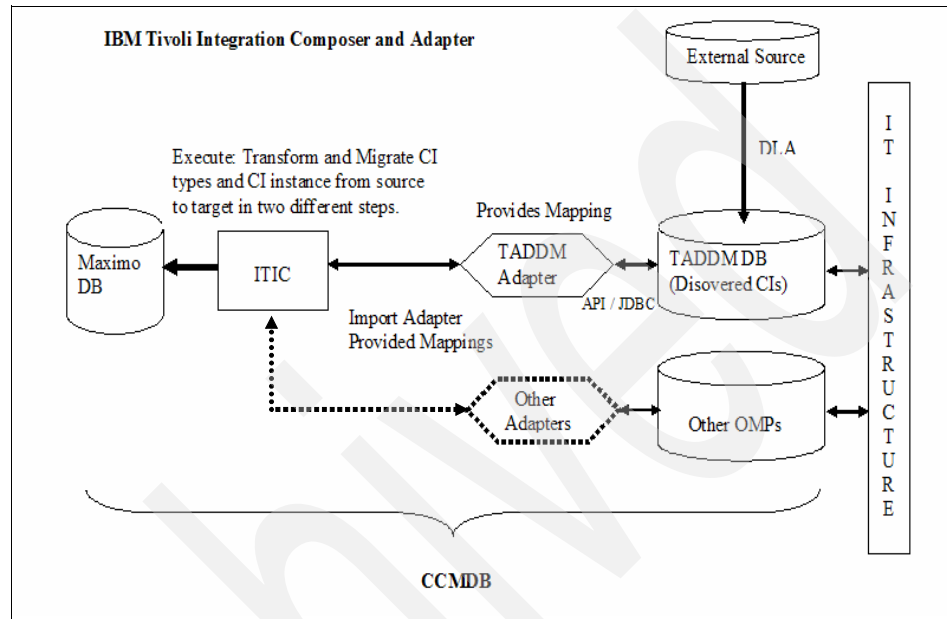


Figure 10-3 Integration Composer and Adapter overview

The Integration Composer is a separately installed software. For detailed Integration Composer installation instructions, refer to 9.9, “IBM Tivoli Integration Composer installation” on page 224.

The steps shown in Figure 10-4 on page 235 must be taken in Integration Composer before source data can be imported into a target database.

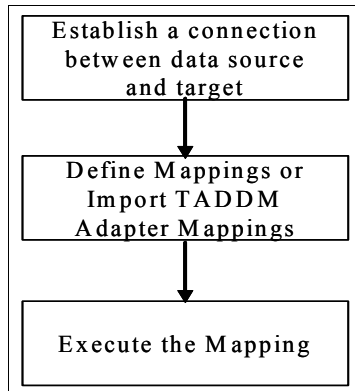


Figure 10-4 Integration steps

A connection is established by defining data source parameters for both a source and target database. In our case, the original data source is TADDM and the destination data source is the Process Layer database. At a high level, CCMDB is comprised of these two databases as well as all other external data sources that capture the configuration and asset inventory of an IT infrastructure.

Before you can bring the configuration item (CI) data from other OMPs such as TADDM, you need to provide schema support in CCMDB or specifically in Maximo, one of the components of CCMDB that acts as the target database for all CIs to be managed centrally from one location.

A mapping is a set of expressions that tells Integration Composer how to create data in the target using information from a source. For each property that you want to import, you define an expression that specifies how to transform the data for that property when Integration Composer imports the data from the source to the target.

The IBM Change and Configuration Management Database (CCMDB) TADDM adapter provides a default mapping between TADDM and Maximo databases. This means that you will be able to import CI data types and instances without any further mapping and modifications.

When you execute a mapping, Integration Composer transforms the collected data and imports it into the target.

Integration Composer connects to data sources using either Java Database Connectivity (JDBC) drivers or an application programming interface (API). In our case, Integration Composer uses TADDM API to connect to the source database as well as a JDBC driver to connect to the Maximo database.

10.2.1 System requirements

The hardware and software requirements for Integration Composer vary according to operating system, database platforms, and site configuration. For information about the minimum and recommended configurations for the several components used in running Integration Composer, refer to *Planning and Installing Change and Configuration Management Database*.

Product version supported

Integration Composer V7.1 works only with IBM Tivoli Change and Configuration Management Database V7.1.

10.2.2 Integration Composer components

Figure 10-5 gives an overview of the Integration Composer components.

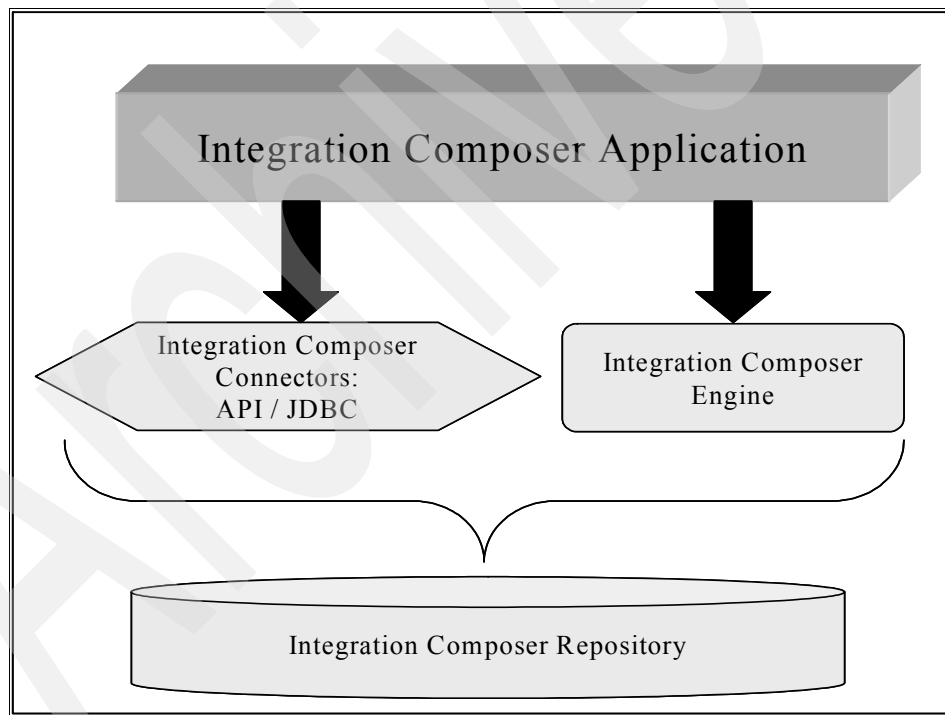


Figure 10-5 Integration Composer components

Integration Composer is comprised of the following components:

- ▶ **Integration Composer Application:** This is comprised of a user interface that lets you do the following:
 - Designates a data source and establishes a connection to that data source.
 - Creates new mappings that define how to transform and migrate data from a data source to a target database.
 - Uses existing mappings to import and migrate data into a target database.
 - Customizes existing mappings.
 - Executes mappings to transform data and import it into the target database.
 - Monitors the status of mappings as Integration Composer executes them.
 - Creates new data schemas to use with Integration Composer.
- ▶ **Integration Composer Engine:** The Integration Composer engine processes mapping expressions that transform data from a source database and integrates it into a target data source.
- ▶ **Connection Methods:** Integration Composer uses JDBC drivers or an API to establish connections to the source data and target database. Integration Composer includes the following JDBC drivers:
 - IBM DB2 JDBC Driver.
 - i-net OPTA JDBC Driver for Microsoft SQL Server 7/2000/2005.
 - Oracle JDBC Thin driver. This driver supports Oracle 10g and earlier versions (including 8.0, 8i, and 9i).

Integration Composer also includes the IBM Configuration Discovery and Tracking API.

- ▶ **Integration Composer Repository:** The Integration Composer resides in the Maximo database and contains the following Integration Composer data:
 - Metadata for read-only data schemas delivered with Integration Composer. This metadata defines the structure of the data. The repository contains data schemas for the following products:
 - Altiris Inventory Solution
 - Centennial Discovery
 - Hewlett Packard (HP) Configuration Management (CM) Inventory Manager (formerly HP Inventory Manager using Radia)
 - IBM Tivoli Configuration Manager
 - Maximo Deployed Assets

- Maximo Discovery
- Microsoft SMS
- Maximo MainControl i.collect
- Metadata for data schemas that users create in Integration Composer.
- Data source definitions that provide database connection parameters.
- Mappings that define how to transform data and migrate it from a source to a target.
- The time stamp of the most recent scan for top-level objects in the source data, if such a last-scan time stamp exists.

10.3 IBM Tivoli Integration Adapters

To facilitate data migration, IBM offers IBM Tivoli Integration Adapters to transform and import data provided by the various asset and systems management tools, such as IBM Tivoli Application Dependency Discovery Manager. The adapters provide mapping expressions that let you transform data and transfer it from a database created by a discovery tool into the target database. You can import an adapter mapping into Integration Composer and, with little or no additional programming, execute the mapping to import data into a target database.

IBM provides integration adapters for the following discovery tools:

- ▶ Altiris Inventory Solution
- ▶ Centennial Discovery
- ▶ Hewlett Packard (HP) Configuration Management (CM) Inventory Manager
- ▶ MainControl i.collect
- ▶ Maximo Discovery
- ▶ Microsoft SMS
- ▶ NetCool/Precision for IP Networks
- ▶ Tivoli Application Dependency Discovery Manager
- ▶ Tivoli Configuration Manager
- ▶ Tivoli License Compliance Manager
- ▶ Tivoli License Compliance Manager for z/OS®
- ▶ Tivoli Provisioning Manager

To integrate data from other discovery tools, you can use the Create Data Schema feature in Integration Composer to create a data schema for the discovery tool. After creating the data schema, you can create a mapping based on the new data schema and use the mapping to import data into the target database.

10.4 TADDM adapter installation

The integration adapter for IBM Tivoli Application Dependency Discovery Manager (TADDM) is used together with IBM Tivoli Integration Composer. Integration Composer (formerly Maximo Fusion) is an integration tool that imports information technology (IT) data into a target database. To gather the data, a discovery tool like TADDM scans computers and other devices connected to a network and records information about their installed hardware and software. In conjunction with the integration adapter for TADDM, Integration Composer transforms the collected data and imports it into the target database. The integration adapter makes the user's job easier by automating the data-mapping part of the transforming task, a part that the user would otherwise have to do manually.

The TADDM Adapter consists of few dbscripts, schema, and mapping files, that are provided to support the importation of two types of discovered data: CI type data and Actual CI data. CI is an abbreviation for configuration item. A configuration item is anything in an IT environment subject to configuration management, or change control: a computer, a printer, a router, and so on.

The TADDM Adapter needs to be installed on the same machine where you have installed Integration Composer and requires the same prerequisites needed for Integration Composer.

Installing the TADDM Adapter is just a matter of unzipping the adapter and placing the appropriate files in the right directories under Integration Composer. Later on, these files are used to import schemas and mappings into Integration Composer that in turn make changes in the database. Some of the files need to be executed directly on databases using a utility, such as the DB2 Control Center, which allows you to run scripts.

Before proceeding, set the database instance owner variable with the following steps:

1. Open a DB2 command window and set the DB2INSTANCE variable value to CTGINST1.
2. Add the OS environment variable DB2INSTANCE=CTGINST1 (or the DB2 instance owner of the MAXdb71 database for your installation).
3. Open up a DB2 command window and connect to the MAXdb71 database as the instance owner:

```
db2 connect to MAXdb71 user CTGINST1
```

10.4.1 Integration adapters for CI Type

This section provides an overview of various scripts and schemas that relate to source and target CI information.

Defining a data schema structure for source CI Type

These data schema scripts define the structure that organizes and classifies the CI Type source data for IBM DB2, Oracle, and SQL Server databases, respectively:

- ▶ createTADDM71CITypeDataSchema.db2
- ▶ createTADDM71CITypeDataSchema.ora
- ▶ createTADDM71CITypeDataSchema.sqs

Defining a data schema structure for target CI Type

This data schema file defines the structure that organizes and classifies the CI Type target data.

- ▶ CCMDB71Classification.schm

Qualifying a data schema structure for target CI Type

These scripts modify the way Integration Composer handles CI Type target data when it performs a data import to IBM DB2, Oracle, and SQL Server databases, respectively.

- ▶ qualifierCCMDB71Classification.db2
- ▶ qualifierCCMDB71Classification.ora
- ▶ qualifierCCMDB71Classification.sqs

Creating a mapping from CI Type to classification

This mapping file provides predefined expressions that you can use to transform CI Type data from the source formats to the target formats.

- ▶ TADDM71CITypeToCCMDB71Classification.fsn

10.4.2 Integration adapters for Actual CI (CI Instances)

This section provides an overview of various scripts and schemas that relate to the Actual CI data.

Defining a data schema structure for source Actual CIs

These data schema scripts define the structure that organizes and classifies the actual CI source data for IBM DB2, Oracle, and SQL Server databases, respectively.

- ▶ createTADDM71ActualCIDataSchema.db2
- ▶ createTADDM71ActualCIDataSchema.ora
- ▶ createTADDM71ActualCIDataSchema.sqs

Defining a data schema structure for target Actual CI

This data schema file defines the structure that organizes and classifies the actual CI target data.

- ▶ CCMDB71ActualCI.schm

Qualifying a data schema structure for target Actual CI

These scripts modify the way Integration Composer handles actual CI target data when it performs a data import to IBM DB2, Oracle, and SQL Server databases, respectively.

- ▶ qualifierCCMDB71ActualCI.db2
- ▶ qualifierCCMDB71ActualCI.ora
- ▶ qualifierCCMDB71ActualCI.sqs

Creating a mapping from Actual CI data to target data

This mapping file provides predefined expressions that you can use to transform actual CI data from the source formats to the target formats.

- ▶ TADDM71ActualCIToCCMDB71ActualCI.fsn

10.4.3 Adding the TADDM Adapter to Integration Composer

Add the adapter files to the Integration Composer installation directory. To do so, copy the CI type-related data schema, qualifier, and mapping files provided with the integration adapter to the Integration Composer server, as follows:

1. Copy the schema files and qualifier files for CI types to `installation_dir\data\dataschema` (where `installation_dir` is the directory where Integration Composer was installed).

Note: You can put these files in another location; however, when you import them into Integration Composer, by default Integration Composer looks for the files in the data schema folder. If you put the files in a different location, you will have to browse to that location and select the files.

- The schema files for CI types are:
 - createTADDM71CITypeDataSchema.db2 (for source data)
 - createTADDM71CITypeDataSchema.ora (for source data)
 - createTADDM71CITypeDataSchema.sqs (for source data)
 - CCMDB71Classification.schm (for target data)
 - The qualifier files for CI types are:
 - qualifierCCMDB71Classification.db2
 - qualifierCCMDB71Classification.ora
 - qualifierCCMDB71Classification.sqs
2. Copy the mapping file for CI types to installation_dir\data\mappings (where installation_dir is the directory where Integration Composer was installed).

Note: You can put these files in another location; however, when you import them into Integration Composer, by default Integration Composer looks for the files in the data schema folder. If you put the files in a different location, you will have to browse to that location and select the files.

- The mapping file for CI types is:
 - TADDM71CITypeToCCMDB71Classification.fsn
 - TADDM71ActualCItoCCMDB71ActualCI.fsn

10.5 Configuration for TADDM and Maximo integration

To access Integration Composer, first install the application on a server. For the hardware, software, and other requirements to run Integration Composer and for installation instructions, refer to Chapter 9, “Installation” on page 187.

To access Integration Composer, log in using the Maximo database user ID (in our case, maximo) and password. Only users with database administrative rights have access. Integration Composer stores the database user IDs that you enter when defining connectivity to the source and target data sources, but it does not store the passwords.

To open Integration Composer, complete the following steps:

1. From the Start menu on the desktop, select **Programs** → **IBM Tivoli** → **Integration Composer** → **IBM Tivoli Integration Composer**. The Integration Composer log-in window opens (Figure 10-6 on page 243).

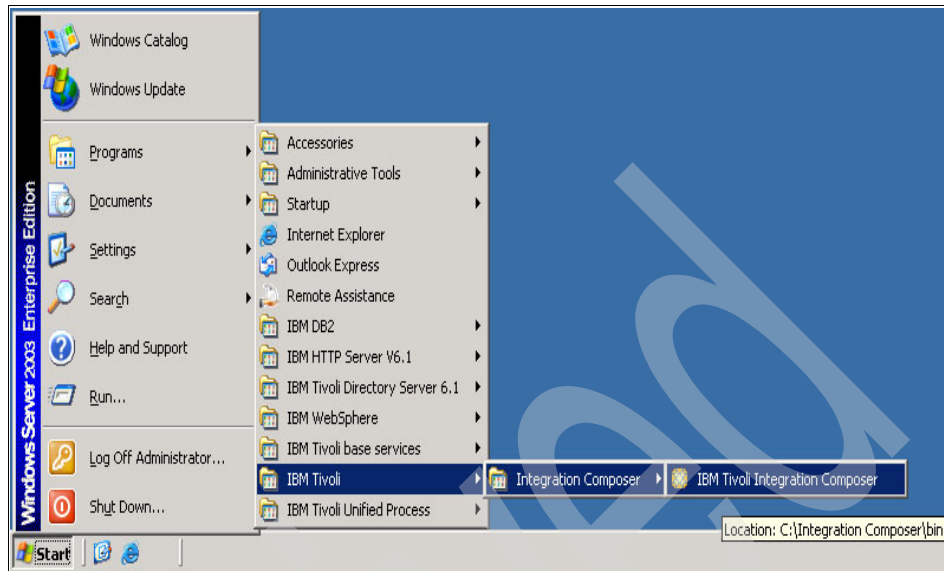


Figure 10-6 Starting Integration Composer

2. In the User Name field, type a user name. In the Password field, type a password (Figure 10-7).

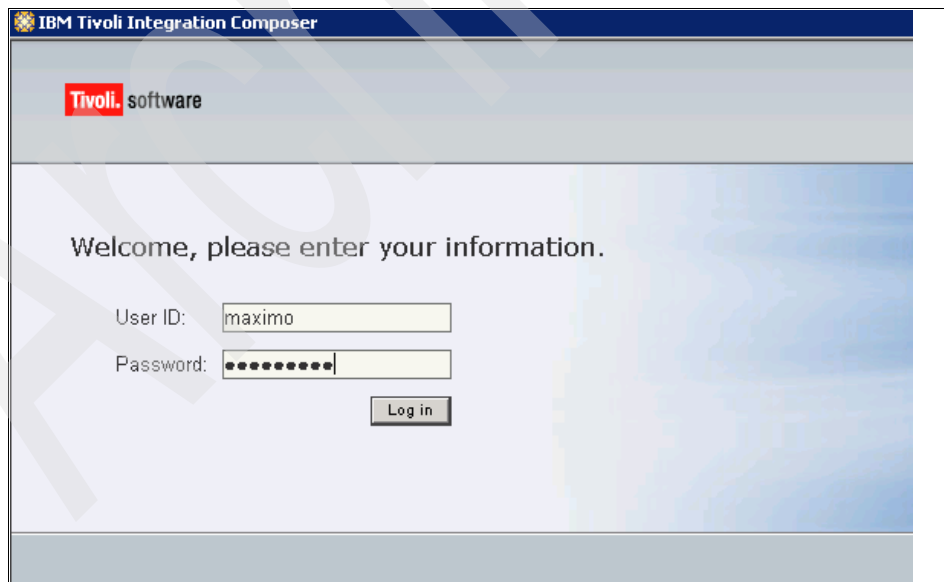


Figure 10-7 Integration Composer login window

3. Click **Log in**. Integration Composer displays the IBM Tivoli Integration Composer window (Figure 10-8).

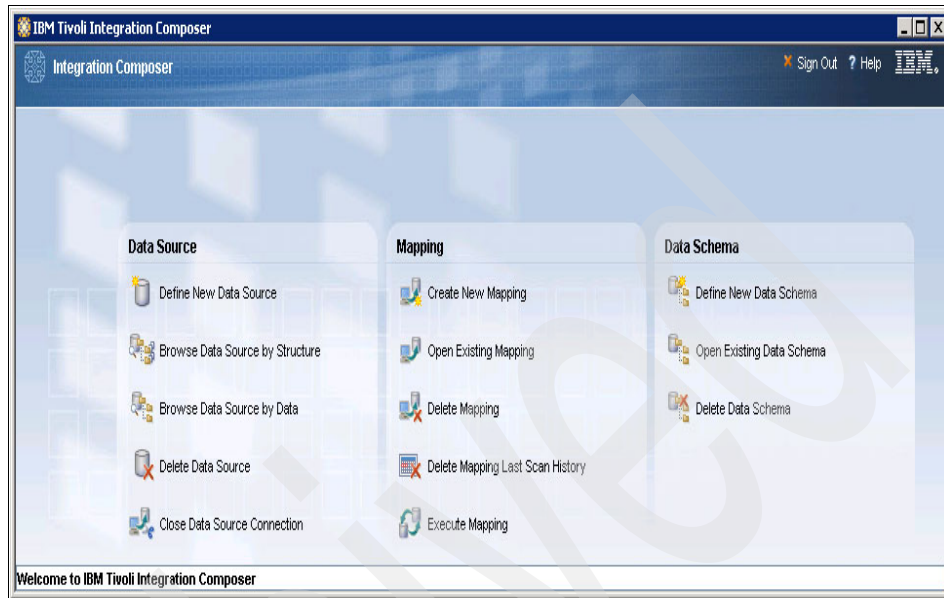


Figure 10-8 Integration Composer main application

10.5.1 Setting up the TADDM integration adapters

The integration adapter is supplementary software designed for use with Integration Composer. These two software tools are used together to take source data (that was previously discovered and saved in a source database) and import it into the Maximo database. Integration Composer works hand in hand with the adapter to import the discovered data and migrate it to a target (in this case, the Maximo database) that serves as a central repository for data from potentially numerous sources. Using this central repository, one enterprise application can then make use of all the data migrated there, regardless of source. This section explains how to add the components of the integration adapter, which consists of two sets of schema and mapping files, to Integration Composer. These integration adapter files are then set up through the process of installing data schemas and creating mappings with Integration Composer. The process must be done twice: once for setting up for CI type imports, and once for setting up for actual CI imports.

After your adapter setups are complete, you can run the mappings that you created (for CI type data and Actual CI data) and in so doing import the

classification data and the data gathered during discovery into the target database.

Integration Composer requires a particular subset of integration adapter files, depending on what kind of data is being imported. The adapter comes with some files to use when importing CI type data, and other files to use when importing Actual CI data, as mentioned in 10.4, “TADDM adapter installation” on page 239

Create the TADDM root class

In the Classifications application, which is part of CCMDB V7.1, make sure there is a top-level CI classification for configuration items. If it does not exist, you will have to create it. This will be the root classification for all the classes from TADDM. In order to create the top-level CI classification in Maximo, perform the following steps:

1. Log in to Maximo and select **GoTo** → **Administration** → **Classification**, as shown in Figure 10-9.

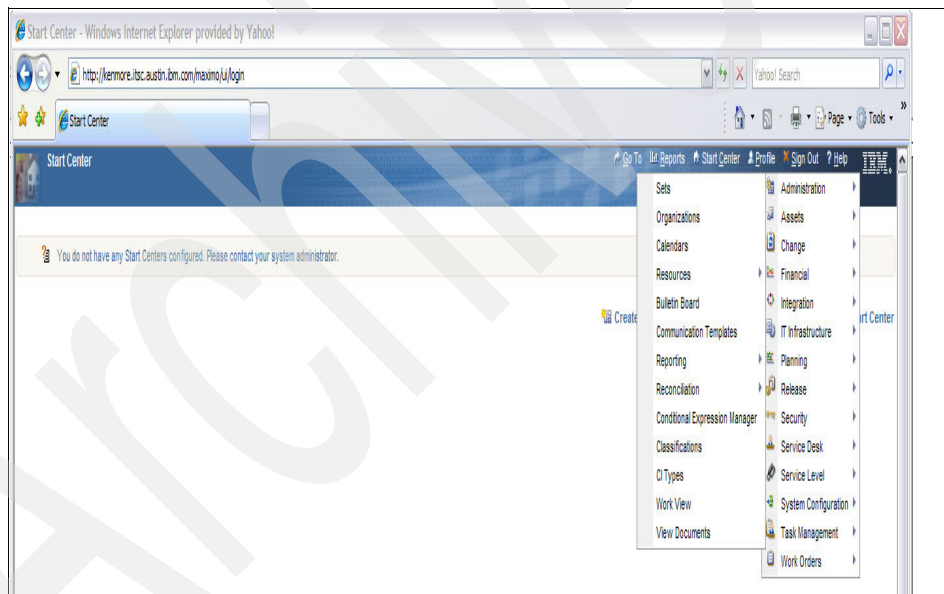


Figure 10-9 Maximo Classification window

2. In the Classification application, create a new classification called 'TOPCICLASS'. This will be the root classification for all the classes from TADDM (Figure 10-10).

The screenshot shows the 'Classifications' application in a web browser. The 'List' tab is active, displaying a table with columns: Classification, Classification Path, Parent Classification, Generate Description?, Use Classification?, Organization, and Site. The 'Classification' field is set to 'TOPCICLASS', 'Classification Path' is 'TOPCICLASS', and 'Parent Classification' is empty. The 'Generate Description?' checkbox is checked. The 'Use Classification?' checkbox is also checked. The 'Organization' and 'Site' fields are empty. Below the table, there are sections for 'Use With Object', 'Children', and 'Attributes', each with a 'New Row' button.

Figure 10-10 Creating TOPCICLASS in Maximo

Updating the MAXVARS table

Using the appropriate database utility (select **Start** → **Programs** → **IBM DB2** → **DB2COPY1** → **Command Line Tools** → **DB2 Command Editor**), run the following query against the Maximo database (Figure 10-11 on page 247). This database query sets the value of your top-level (root) CI class, 'TOPCICLASS', in the Maximo Variables table (MAXVARS table) to the current CLASSTRUCTUREID value:

```
update maxvars set varvalue=(select classtructureid from
classtructure where classificationid='TOPCICLASS') where
varname='CICLASS';
```

Note: DB2 Command Editor is provided as an example. You can use any other tool. Make sure you *type* the above **update** command in the DB2 Command Editor; copying and pasting the command may embed hidden invalid characters in the DB2 Command Editor.

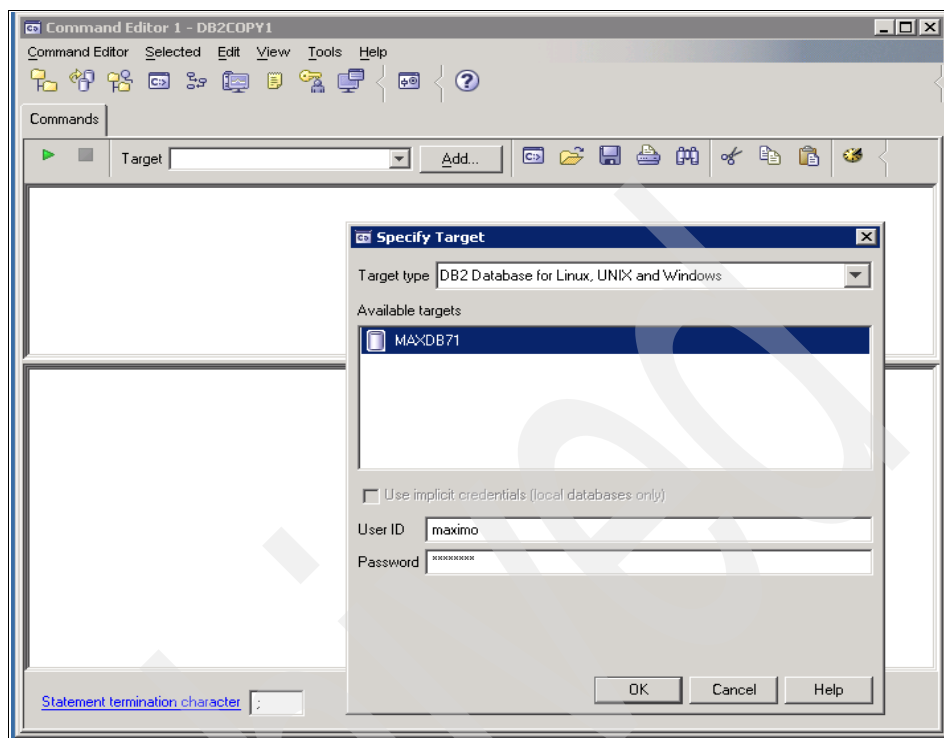


Figure 10-11 Adding a database connection

3. Execute the **commit;** command (see Figure 10-12).

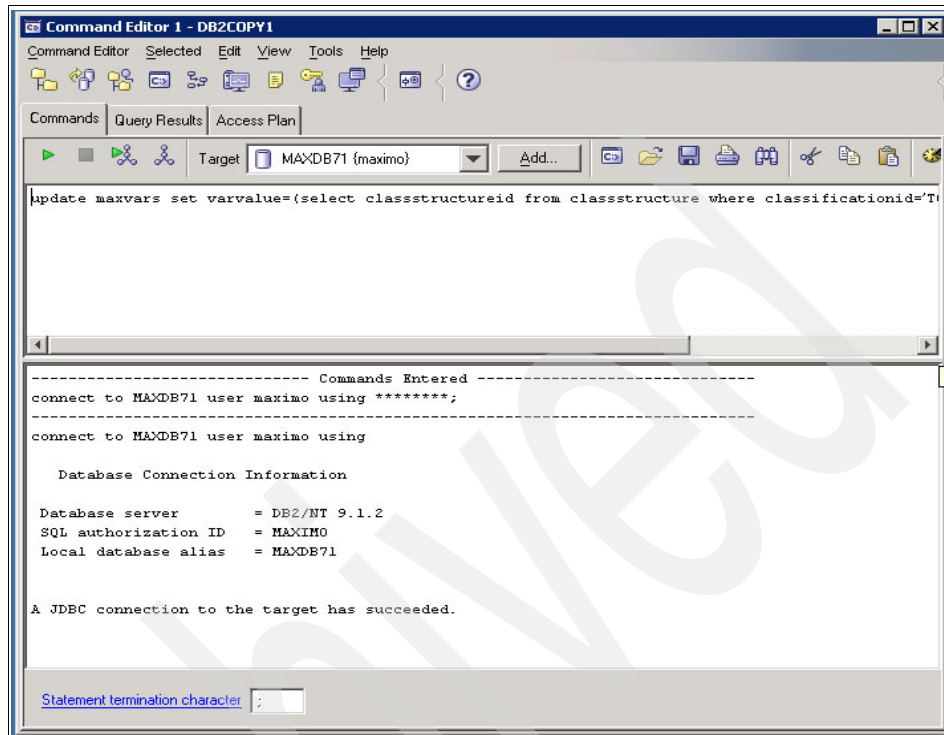


Figure 10-12 Executing the update command in DB2 Command Editor

Note: If you do not create the database record as described in this section, the subsequent import process will stop without warning. No error messages are provided.

Verifying the appropriate level of depth for Actual CI data imports

The level of depth describes a property associated with the CI trees in your data source for actual CIs. It notifies Integration Composer about the number of class relationships to be traversed and then migrated to the target when importing actual CI source data. This verification task is to ensure that the level of depth for the CI trees you intend to import is appropriate for your needs. The value for the level of depth is preset to 3, which is the default, but you can change it by editing the initialization file `fusion.properties`. The Integration Composer initialization file is located in `installation_dir\data\properties\fusion.properties`, where

installation_dir is Integration Composer on Windows operating systems and Integration_Composer (with underscore) on UNIX-based operating systems.

By subtracting 1 from the depth level currently specified in your initialization file, you can calculate the number of relationships that will be traversed in your CI trees during data importing, if no further change is made:

Depth level specified – 1 = Number of relationships traversed

For example, if your depth level is currently set to the default, 3, the three classes shown in Table 10-1 will be imported.

Table 10-1 Depth level imported

Depth level	Class	
1	Computer Systems	
2	Operating System	
3	Software	

In the example, with a depth level of 3, two levels of relationships are traversed: one between the Computer System and Operating System classes, and another between the Operating System and Software classes.

- To check the current value for the level of depth:

Look at the mxe.db.queryDepthLevel property in the initialization file (installation_dir\data\properties\fusion.properties):

```
// Number of levels to be recursively traversed for a query.  
// Default is 3  
mxe.db.queryDepthLevel=3
```

- To change the current value for the level of depth, do the following steps:
 - a. If Integration Composer is running, close all open windows and sign out of the application.
 - b. Edit the initialization file installation_dir\data\properties\fusion.properties.
 - c. Modify the value of the mxe.db.queryDepthLevel property. (Note that increasing the default value of 3 could reduce system performance.)
 - d. Save the file.
 - e. Restart Integration Composer to activate your property change.

Verifying the cache properties for reference classes

The cache properties are properties that configure reference classes related to the integration adapter for TADDM (Figure 10-13). The cache properties need to be included in the Integration Composer initialization file, `fusion.properties`. Among other things, these properties specify the data schema class to be cached, the initial size of the cache, and, for entries in the cache, which type of properties are to be used as lookups to the cache; in this case, the alternate key set is specified. This verification task ensures that all the required cache properties are included in the initialization file. If they are not there already, you must add them.

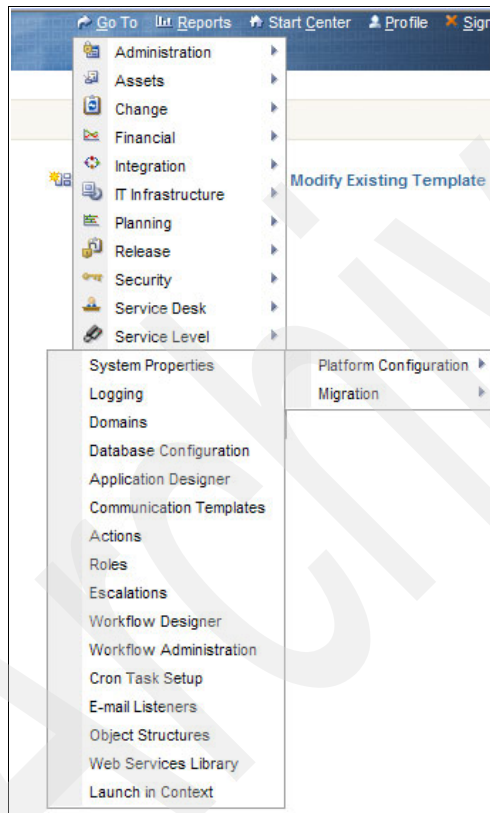


Figure 10-13 Reference classes

The Integration Composer initialization file is located in `installation_dir\data\properties\fusion.properties`, where `installation_dir` is the path to the Integration Composer on Windows operating systems and `Integration_Composer` (with underscore) on UNIX-based operating systems.

To check for the required cache properties:

1. Scroll to the bottom of the Fusion Mapping Execution Properties section of the initialization file (installation_dir\data\properties\fusion.properties).
2. Look for the following lines:

```
mxe.fusion.referencecache.Source_CI=1000,Actcinum,ALTERNATE_KEY  
mxe.fusion.referencecache.Target_CI=1000,Actcinum,ALTERNATE_KEY  
mxe.fusion.referencecache.Relationship_Type=10,Relationnum,  
ALTERNATE_KEY  
mxe.fusion.referencecache.OMP=100,Ompguid,ALTERNATE_KEY  
mxe.fusion.referenceaction.Source_CI=UPDATE  
mxe.fusion.referenceaction.Target_CI=UPDATE  
mxe.fusion.referencecachesameas.Source_CI=Target_CI
```

3. Do one of the following:
 - If the lines are present, close the file and you are done.
 - If the lines are not present:
 - i. Add them to the bottom of the Fusion Mapping Execution Properties section of the file.
 - ii. Save the updated initialization file.
 - iii. Restart Integration Composer to activate your property changes.

10.6 Set schemas, define mapping, and run execution

Before we can bring the actual source (TADDM) data into the target (Maximo), we need to define the data model (schema) in the target database that supports the source data model (schema). Once we achieve the common data model (CDM) in both source and target, we need to do the following:

1. Import the CI Types into the target DB that supports the source (TADDM) CI data.
2. Migrate the source actual CI data into the target DB (Maximo).

The data model in the source (TADDM) is referred to as CI Type while the data model in the target (Maximo) is referred to as CI Classification.

In order to achieve the first step (importing the CI Types into target DB), the following steps need to be performed:

- ▶ Define the source (TADDM) CI Type in the target database (Maximo).
- ▶ Define the target CI Classification in the target database (Maximo).

- Execute the mappings from the source CI Type to the target CI classification.

Figure 10-14 gives an overview of selecting the source CI data model.

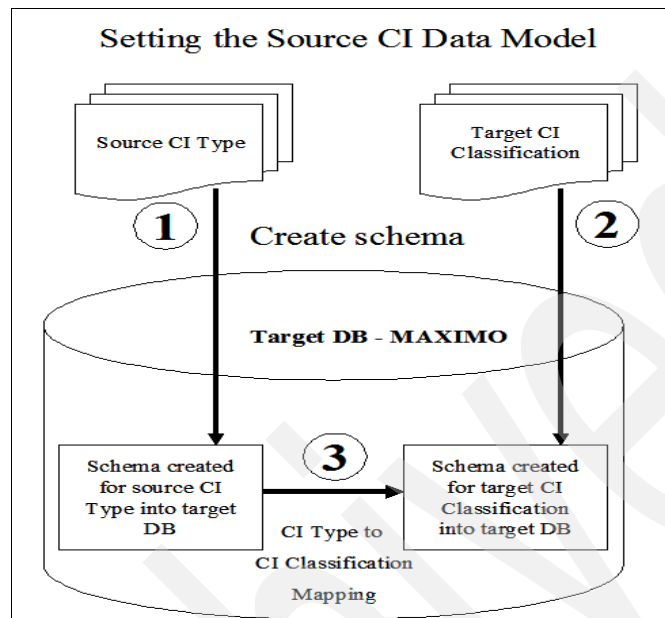


Figure 10-14 Selecting the source CI data model

In order to achieve the second step (importing the actual CI into the target DB), the following steps need to be performed:

- Configure and execute TADDM Adapter for CI Type mapping.
- Configure and execute TADDM Adapter for Actual CI mapping.

The following sections describe how to perform these steps in detail.

10.6.1 Configure and execute TADDM Adapter for CI Type mapping

Follow the detailed steps below to configure and execute the TADDM Adapter for CI Type mapping.

Defining the source (TADDM) CI Type into the target (CCMDB)

To set up for defining CI Types, the integration adapter files related to CI Type data are used. Working with Integration Composer, you begin by creating a data schema for source CI Type and target CI Classification followed by the mapping of your CI Type data to CI Classification all in your target database, in our case, the CCMDB database. Because CI Type data is the data used to classify Actual

CI data, the CI Type data is imported first. The schema and mapping files provided by the integration adapter are designed to help expedite this process. Therefore, using the integration adapter reduces the manual tasks (and time) required of you.

Figure 10-15 shows an overview of creating a TADDM CI Type schema in a Maximo DB.

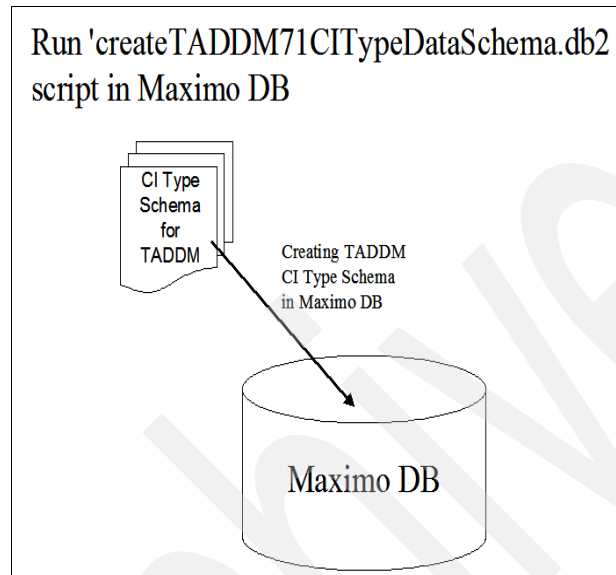


Figure 10-15 Creating a TADDM CI Type schema in a Maximo DB

Before you can use the integration adapter to create a mapping and migrate data, you need to create the source data schema for CI Type data in the Integration Composer repository in the Maximo database.

To create the CI Type data schema for the TADDM data source, use one of the following scripts:

- ▶ createTADDM71CITypeDataSchema.db2
- ▶ createTADDM71CITypeDataSchema.ora
- ▶ createTADDM71CITypeDataSchema.sqs

Since we are using DB2, we will execute the createTADDM71CITypeDataSchema.db2 script using the DB2 Command Editor to create the TADDM CI Type schema in Maximo.

Defining Target CI Classification into Target DB

Before you can use the integration adapter to create a mapping and migrate data, you need to define the data schema for target CI Classification data in the Integration Composer repository in Maximo database.

To define the data schema, use the CI Classification data file that the integration adapter provides, CCMDB71Classification.schm.

To add the data schema, complete the following steps:

1. If you are not already signed in, sign in to Integration Composer using a valid user ID and password. The IBM Tivoli Integration Composer window is displayed.
2. From the IBM Tivoli Integration Composer window, select **Define New Data Schema**. Integration Composer displays the Data Schema field in the Define a New Data Schema window.

Note: To review or change the previous selections, click **Back**. To cancel this procedure and return to the main Integration Composer window, click **Cancel**.

3. In the Data Schema field, type CCMDB71Classification as the name of the new data schema (data schema names are case sensitive).

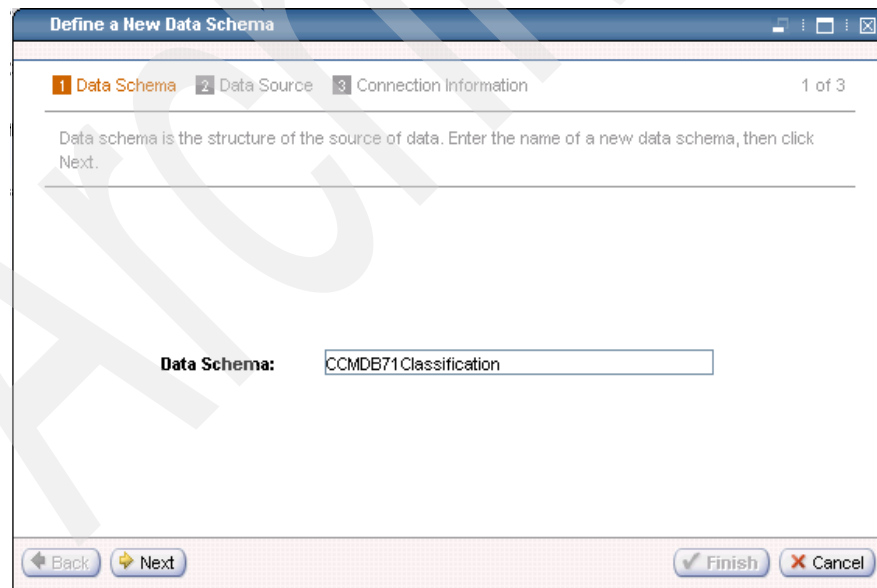


Figure 10-16 Define New Data Schema in Integration Composer

Note: You can type a different name for the new data schema. However, if you do, you will have to change the name CCMDB71Classification in the qualifier script (that you run in “Finalizing the target data schema for CI Type data” on page 262) in order to match the alternative name that you typed here.

4. Click **Next**. The Data Source field is displayed.
5. In the Data Source field, enter a name for the data source (the name must have at least two characters and is case sensitive) and click **Next**. The connection information fields are displayed.

The screenshot shows a Windows-style dialog box titled "Open an Existing Data Schema". It has a tabbed interface with three tabs: "1 Data Schema", "2 Data Source" (which is selected and highlighted in orange), and "3 Connection Information". In the top right corner, it says "2 of 3". Below the tabs, there is a descriptive text: "Data source is the name of the source of data and its related data schema. Choose from the list of data sources for the chosen data schema, then click Next." Below this text is a large text input field containing the text "Maximo_TargetDS". At the bottom of the dialog, there are four buttons: "Back" (with a left arrow), "Next" (with a right arrow), "Finish" (with a checkmark), and "Cancel" (with an X).

Figure 10-17 Data Source for Data Schema

6. In the Connection Method drop-down list, select a connection method (Figure 10-18).

The screenshot shows a window titled "Define a New Data Source" with three tabs: "1 Data Schema", "2 Data Source", and "3 Connection Information" (which is active). The "3 of 3" indicator is in the top right. Below the tabs, the "Connection Method" is a drop-down menu showing "IBM DB2 JDBC Driver". Below this are six text input fields: "Host Name" (gamma.itsc.austin.ibm.com), "Host Port" (50000), "Database" (maxdb71), "User Name" (maximo), "Password" (masked with asterisks), and "Table Owner" (maximo). At the bottom right are "Clear" and "Test Connection" buttons. At the bottom left are "Back" and "Next" buttons. At the bottom right are "Finish" and "Cancel" buttons.

Figure 10-18 New Data Source to Maximo target database

7. Enter the connection parameters, as required. The connection method that you selected in step 6 determines the fields that Integration Composer displays.
8. Optional: If appropriate, click **Test Connection** to test the connection to the data source. Integration Composer displays a Test Connection dialog box. The text in the dialog box indicates whether the test was successful. To respond to the dialog box, select one of the following options:
 - If Integration Composer establishes a connection, it displays a confirmation message. Click **OK**. Integration Composer closes the Test Connection dialog box. Go to step 9.
 - If Integration Composer cannot establish a connection, it displays an explanatory message. Click **OK**. Integration Composer closes the Test Connection dialog box. Review the values for the connection parameters and retry the connection.

Note: The Test Connection feature lets you test only the connection without invoking any additional Integration Composer processes. JDBC drivers that do not comply with JDBC 2.0 probably do not support this feature.

9. On the Connection Information page, click **Finish**. The Data Schema window is displayed (Figure 10-19). Integration Composer displays the root class in red because it has no properties associated with the class.

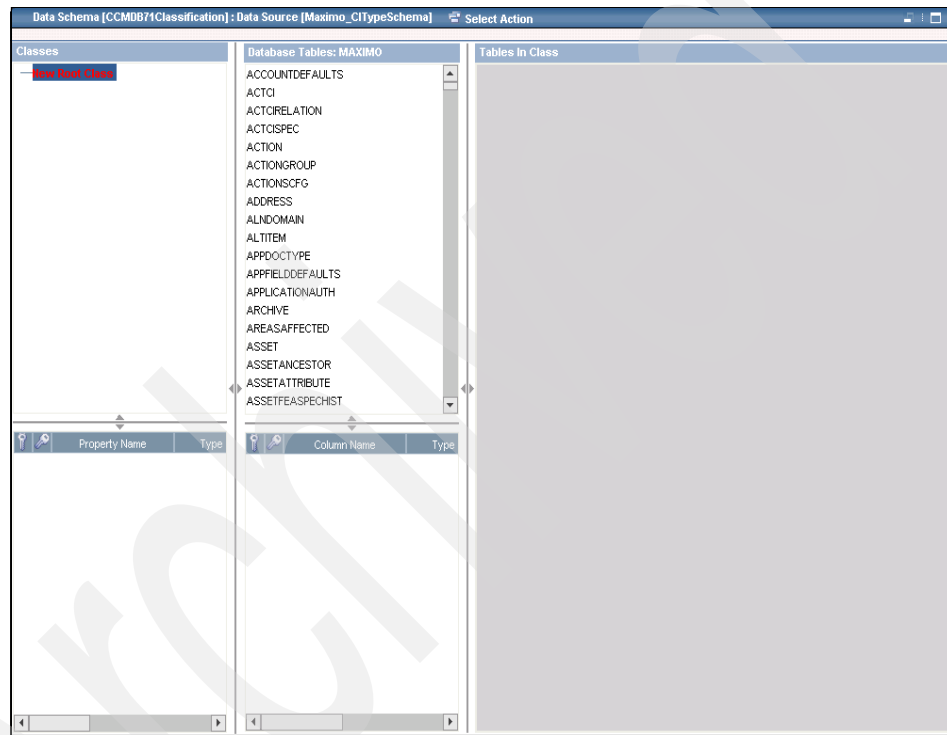


Figure 10-19 Data Schema window

Note: The display properties that you set for your computer can affect colors. The color displayed on your computer can vary. From this window, you can import a data schema file provided with this integration adapter.

10. To import the data schema file provided with the integration adapter, from the Select Action menu in the title bar of the Data Schema window, select **Import Data Schema**. The Import Data Schema dialog box is displayed (Figure 10-20). The Import Data Schema dialog box lists the data schemas that you copied to the data schema folder.

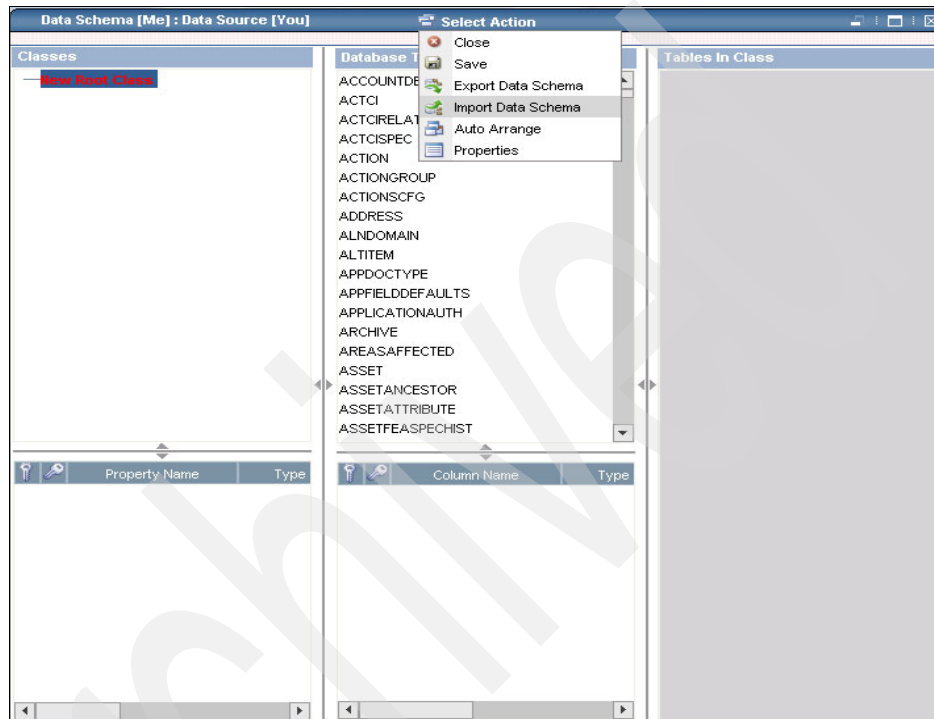


Figure 10-20 Import CI Classification.schm

11. In the Import Data Schema dialog box, select the data schema file that you want to import. For CI type target data, select **CCMDB71Classification.schm**. Integration Composer populates the File name field with the selected file name (Figure 10-21).

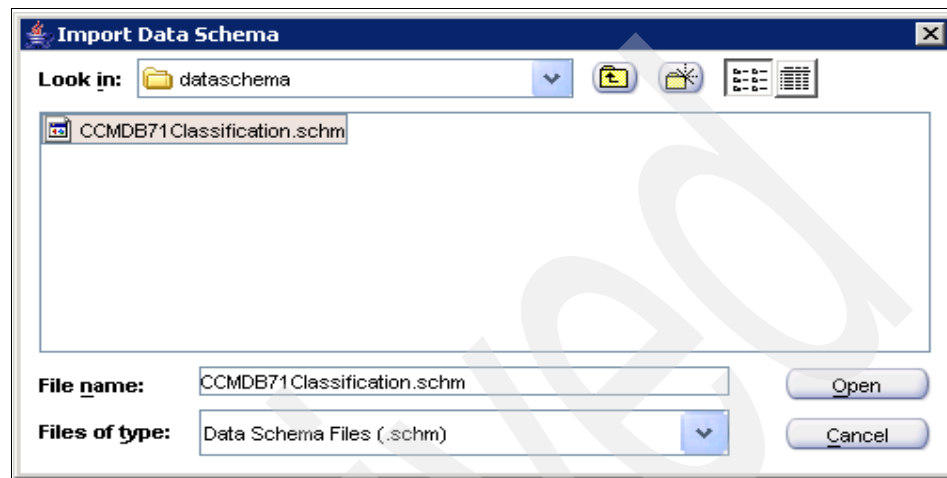


Figure 10-21 Import Data Schema classification

12. Click **Open**. Integration Composer imports the data schema file. If discrepancies exist between the data source and the data schema, the Data Schema Analysis window is displayed (Figure 10-22). This window lists discrepancies between the data schema and the corresponding data source. Integration Composer displays errors that it can correct with a green check.

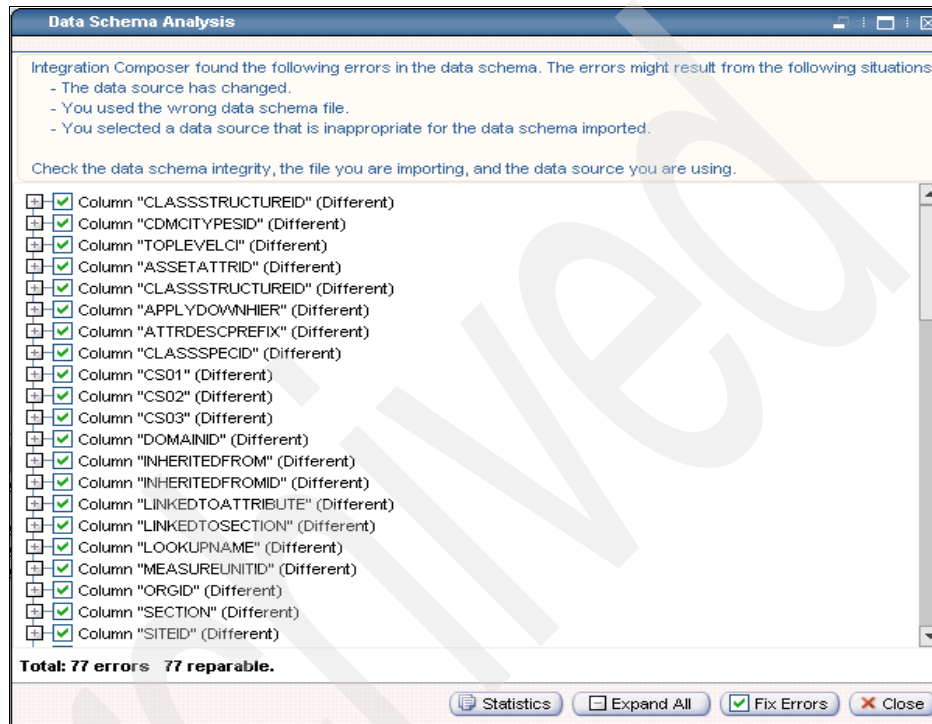


Figure 10-22 Error analysis

13. Review the errors in the Data Schema Analysis window and select one of the following options:

- To repair the errors, click **Fix Errors**. Integration Composer repairs the errors and opens the Data Schema window.

Note: You cannot clear the check boxes. You can either fix all the errors indicated or fix none of the errors.

- To expand all nodes in the tree to display information about inconsistencies between the data schema and the data source, click **Expand All**.
- To view statistics for table and column errors, click **Statistics**.

- To close the dialog box without repairing the errors, click **Close**. A Data Schema Analysis warning window is displayed. In the Data Schema Analysis warning window, select one of the following options:

To make the data schema match the source database, complete the following steps:

- i. In the Data Schema Analysis warning window, click **Yes**. Integration Composer repairs the errors, closes the warning window, closes the Data Schema Analysis window, and displays an Import dialog box indicating that the import is finished.
- ii. In the Import dialog box, click **OK**. The Data Schema window is displayed (Figure 10-23).

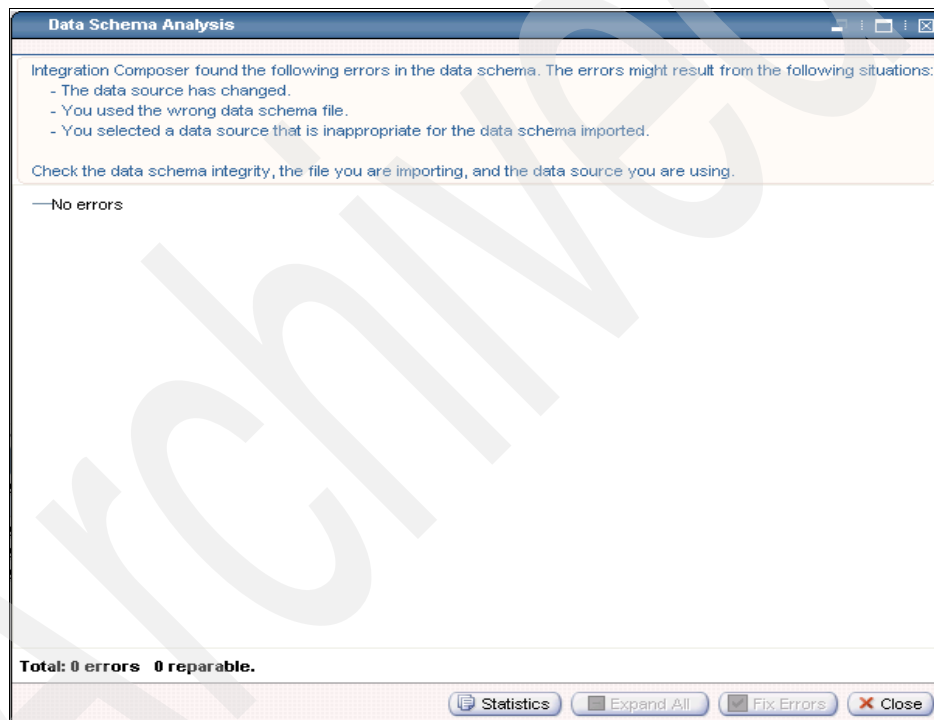


Figure 10-23 Errors fixed

To close the window without changing the data schema, click **No**. Integration Composer imports the data schema as is and displays the Data Schema window.

To cancel the action, click **Cancel**. Integration Composer closes the warning window and displays the Data Schema Analysis window.

14. After you import the data schema file, you can modify the data schema. For more information about working with data schemas, see the book *IBM Tivoli Integration Composer System Administrator's Guide* or see the Integration Composer help. In our example, we did not modify any mappings and used the default TADDM Adapter mappings.
15. After you finish working with the data schema, select **Save** from the Select Action menu. Integration Composer saves the data schema.

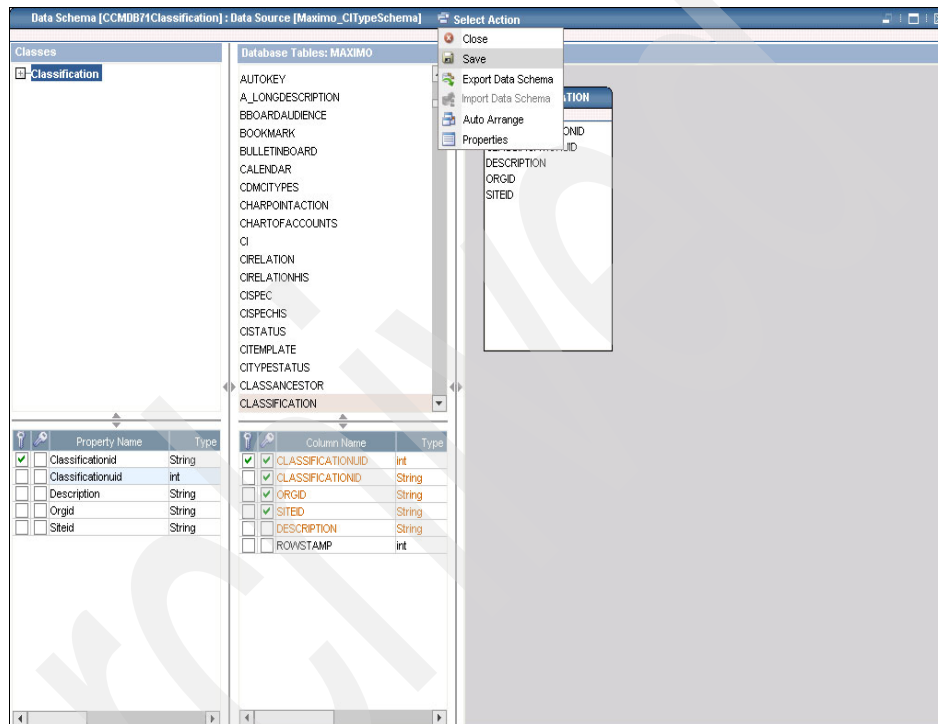


Figure 10-24 Importing data schema for CI Classification

16. Select **Close** from the Select Action menu to close the data schema. Integration Composer closes the data schema and displays the IBM Tivoli Integration Composer window.
17. Sign out of Integration Composer.

Finalizing the target data schema for CI Type data

Before you can use the integration adapter to create a mapping, you need to finalize the target data schema in the Integration Composer repository in the Maximo database. Finalizing the data schema allows Integration Composer to delete certain class records, in addition to merely inserting and updating them,

when executing a mapping. By performing this finalization task, you are modifying the way Integration Composer handles data when it performs a data import.

To perform this task, use one of the following scripts, as appropriate for your database:

- ▶ `qualifierCCMDB71Classification.db2`
- ▶ `qualifierCCMDB71Classification.ora`
- ▶ `qualifierCCMDB71Classification.sqs`

To finalize the target data schema, complete the following steps:

1. Sign out of Integration Composer, if you have not already done so.
2. Optional: If you named the new data schema something other than `CCMDB71Classification` in step 3 on page 17, change the name `CCMDB71Classification` in the qualifier script to match the alternative name that you provided.
3. Using an appropriate database query tool, execute the qualifier script (in our case `qualifierCCMDB71Classification.db2`) for your target data.
4. Check for database script errors and resolve any errors. For error information, refer to the log files located in the log folder in your Windows or UNIX installation directory.

Creating a new data source for the source (TADDM) database

Integration Composer uses a JDBC driver or an API to connect to data sources. Before you create a mapping, you must have a data source connection defined for the data source (in this case, the TADDM database) and the target (the Maximo database).

The data source connection to target Maximo database has already been established in “Defining Target CI Classification into Target DB” on page 254.

In this section, we will create a new data source to source TADDM database through API.

After you define the connection parameters for the data source, Integration Composer saves them in its repository and displays them whenever you attempt to connect to the data source. The only parameter that Integration Composer requests from you in real time is the password for the data source (that is, either a database password for database users, or, if using the Configuration Discovery and Tracking API, a password associated with the user login account).

To define a data source, complete the following steps:

1. Sign in to Integration Composer using a valid user ID and password. The IBM Tivoli Integration Composer window is displayed (Figure 10-25).
2. Select **Define New Data Source**. The Data Schema page is displayed in the Define a New Data Source window. The Data Schema page lists data schemas that were installed with Integration Composer plus any data schemas that you created using database scripts or the data schema feature in Integration Composer.

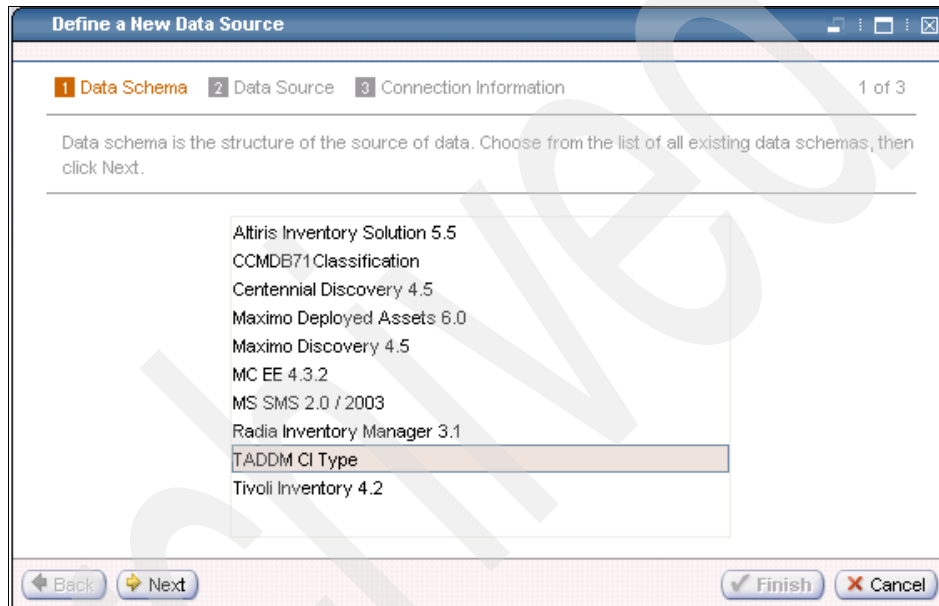


Figure 10-25 Define TADDM data source

3. Select a TADDM Type data schema and click **Next**. The Data Source page is displayed (Figure 10-26 on page 265).
4. In the Data Source field, enter a name for the data source (the name must have at least two characters and is case sensitive) and click **Next**.

Define a New Data Source

1 Data Schema 2 **Data Source** 3 Connection Information 2 of 3

Data source is the name of the source of data and its related data schema. Type in a new data source name, then click Next.

Data Source: TADDM_SourceDS

Back Next Finish Cancel

Figure 10-26 Naming a data source

The connection information fields are displayed.

Note: Make sure the TADDM sever is up and running before you try to connect to the TADDM database.

5. In the Connection Method drop-down list, select a connection method, as shown in Figure 10-27.
6. Enter the connection parameters, as shown in Figure 10-27.

The screenshot shows a window titled "Define a New Data Source" with three tabs: "1 Data Schema", "2 Data Source", and "3 Connection Information" (which is selected). The "Connection Method" is set to "IBM Configuration Discovery and Tracking API". Below this, there are several input fields: "Host Name" with the value "boston.itsc.austin.ibm.com", "Host Port" with "9530", "Use SSL" with an unchecked checkbox, "Trusted Location" which is empty, "User Name" with "administrator", and "Password" which is masked with asterisks. At the bottom right, there are "Clear" and "Test Connection" buttons. At the bottom left, there are "Back" and "Next" buttons. At the bottom right, there are "Finish" and "Cancel" buttons.

Figure 10-27 TADDM Data Source Connection parameters

7. Optional: If appropriate, click **Test Connection** to test the connection to the data source. Integration Composer displays a Test Connection dialog box. The text in the dialog box indicates whether the test was successful. To respond to the dialog box, select one of the following options:
 - If Integration Composer establishes a connection, it displays a confirmation message. Click **OK**. Integration Composer closes the Test Connection dialog box. Go to step 8.
 - If Integration Composer cannot establish a connection, it displays an explanatory message. Click **OK**. Integration Composer closes the Test Connection dialog box. Review the values for the connection parameters and retry the connection.

Note: The Test Connection feature lets you test only the connection without invoking any additional Integration Composer processes. JDBC drivers that do not comply with JDBC 2.0 probably do not support this feature.

8. On the Connection Information page, click **Finish**. Integration Composer displays a Save dialog box.

9. Click **OK**. Integration Composer returns you to the IBM Tivoli Integration Composer window.
10. Close the Integration Composer.

Note: If Integration Composer does not save the data source successfully, it displays one or more error messages. Click **OK**. Integration Composer closes the error message dialog box. Resolve any errors and try defining the data source again. For error information, refer to the log files located in the log folder in your Windows or UNIX installation directory.

Note that in an Integration Composer session, if you connect to one of the defined data sources, Integration Composer keeps the data source connection open throughout the session unless you complete one of the following steps:

1. Run a mapping for the data source.
2. Use the Close Data Source Connection option in the Data Source menu in the IBM Tivoli Integration Composer window to close the connection.
3. Delete the open data source.

Creating a mapping for CI Type data

Before you can migrate CI Type data, you must create a new adapter mapping for the data you want to migrate and then import the appropriate mapping file (TADDM71CITypeToCCMDB71Classification.fsn) into the mapping that you create.

Note: If you import a mapping into a mapping that contains expressions, the imported mapping has the following effect on the original mapping:

- ▶ If an expression exists in both mappings, Integration Composer replaces the existing expression with the imported expression.
- ▶ Integration Composer adds any new expressions in the imported mapping to the original mapping.

Note: Before creating your mapping, make sure you have installed the data schemas for both your source and target CI type data, in accordance with the instructions described in “Creating a new data source for the source (TADDM) database” on page 263.

Before you do the following steps, make sure that the MAXVARS table is updated correctly in the Maximo database.

To create a mapping, complete the following steps:

1. Sign in to the Integration Composer application using a valid user ID and password. Integration Composer displays the IBM Tivoli Integration Composer window.
2. Select **Create New Mapping**. The New Mapping window is displayed (Figure 10-28).

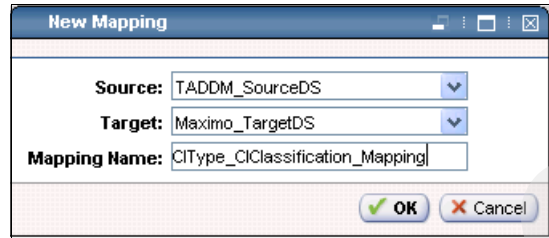


Figure 10-28 Create New Mapping

3. From the Source drop-down list of available data sources, select the data source, in our case, TADDM_SourceDS.
4. From the Target drop-down list of available data sources, select the target database, in our case, Maximo_TargetDS.
5. In the Mapping Name field, enter a new mapping name (maximum of 255 characters). Mapping names are case sensitive; for example, taddm is not the same as TADDM.
6. Click **OK**, and select one of the following options:
 - If Integration Composer opens the Mapping window, go to step 7.
 - If Integration Composer opens the connection information fields in the Open Source Data Source window, use the following procedure to complete the connection information:
 - i. In the Open Source Data Source window, accept the defaults established during the last connection to the data source or update the fields.
 - ii. Enter the password for accessing your source data. (Database users enter a database password, while Configuration Discovery and Tracking API users enter a password associated with the user login account).
 - iii. Click **Finish** to establish the connection to the source.
 - iv. In the Open Target Data Source window, accept the defaults established during the last connection to the data source or update the fields as necessary.

- v. Enter the (database or user login account) password for accessing your target data.
 - vi. Click **Finish** to establish the connection to the target. Integration Composer opens the Mapping window.
7. From the Select Action menu in the Integration Composer Mapping window, select **Import**. The Import Mapping dialog box is displayed.

Note: When you select **Import** from the Select Action menu, by default Integration Composer points to the mappings folder. If you store the .fsn file in a different location, you can use the browse features in this window to locate the file. The .fsn extension identifies Integration Composer files.

8. In the Import Mapping dialog box, select the mapping file for CI Type data, TADD71CITYPEToCCMDB71Classification.fsn. Integration Composer populates the File name field with the selected file name.
9. Click **Open**. Integration Composer imports the mapping file.
10. After you finish the mapping, select one of the following options:
 - To save the mapping with the existing name, select **Save** from the Select Action menu.
 - To save the mapping with a new name, select **Save As** from the Select Action menu. Integration Composer opens a Save Mapping As window. In the Mapping Name field in this window, type a new name for the mapping and click **OK**. Integration Composer saves your changes with the new mapping name.
11. To close the Mapping window, select **Close** from the Select Action menu. A Close Mapping dialog box is displayed.
12. In the Close Mapping dialog box, click **Yes**. Integration Composer closes the Mapping window and displays the IBM Tivoli Integration Composer window.
13. Close Integration Composer.

Execute the mapping

When you create a mapping, you define a set of expressions that specify how to transform instances from a source to a target. To transform the data and migrate it from the source into a target, you execute the mapping. Executing a mapping transforms the data and imports it from the source to the target database.

You can execute a mapping using the Integration Composer interface, or you can execute a mapping from a command line. For performance reasons, it is more efficient to execute a mapping as a stand-alone process from a command line. For more information about how to execute a mapping from command line, refer

to *IBM Tivoli Integration Composer System Administration Guide*. Here in this section, we will execute a mapping from the Integration Composer interface.

To execute a mapping using the Integration Composer interface, complete the following steps:

1. In the IBM Tivoli Integration Composer window, select **Execute Mapping**. Integration Composer displays the Execute Mapping window (Figure 10-29). The Execute Mapping window displays a table that lists the mappings currently available in Integration Composer.

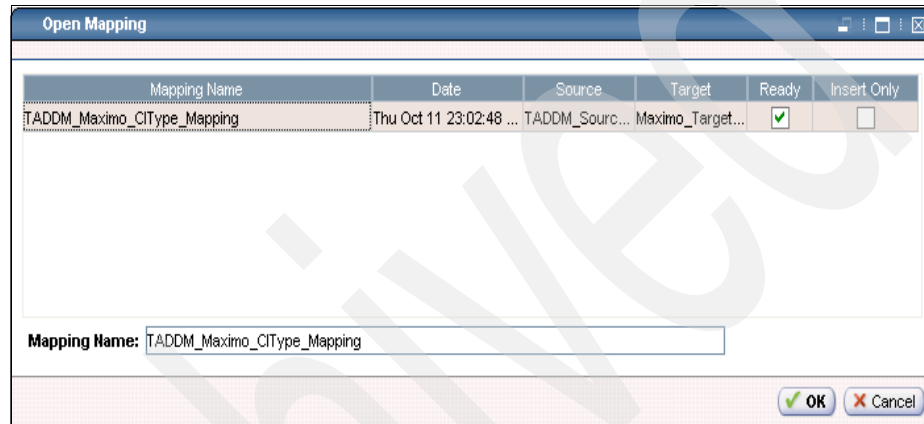


Figure 10-29 Execute the mapping from Integration Composer interface

2. In the list of mappings, verify that Integration Composer has selected the **Ready** check box for the desired mapping and then select one of the following options:
 - If Integration Composer has not selected the Ready field, click **Cancel**. Review the mapping.
 - If Integration Composer has selected the Ready field, the mapping has no syntax errors and is ready to be executed. Go to step 3.
3. To select the mapping to execute, click the row for that mapping in the list of mappings. Integration Composer displays the name of the mapping selected in the Mapping Name field.
4. In the Execute Mapping dialog box, click **OK**. Integration Composer displays the Open Source Data Source window.
5. On the Connection Information page in the Open Source Data Source window, accept the settings established for the last connection to the data source or update the field appropriately. Enter the password.

6. In the Open Source Data Source window, click **Finish**. Integration Composer displays the Mapping Execution Progress dialog box (Figure 10-30 and Figure 10-31).

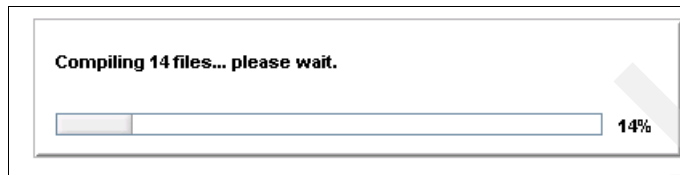


Figure 10-30 Mapping Execution Compiling

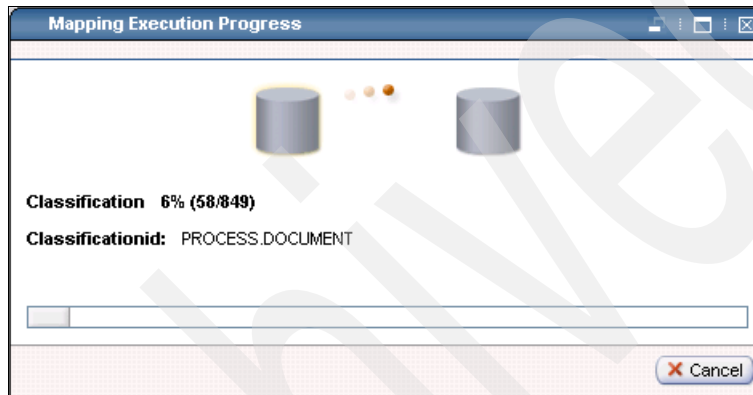


Figure 10-31 Mapping Execution in Progress

If the mapping execution is successful, Integration Composer displays a confirmation message.

Note: If you need to cancel a mapping execution, in the Mapping Execution Progress dialog box, click **Cancel**.

7. In the confirmation dialog box, click **OK**. Integration Composer displays the IBM Tivoli Integration Composer window.

10.6.2 Configuring executing TADDM adapter for Actual CI mapping

To set up for importing Actual CIs, the integration adapter files related to Actual CI data are used. Working with Integration Composer, you begin by creating a data schema and mapping for your Actual CI data. (Note that because CI Type data is used to classify the Actual CI data, the Actual CI data cannot be imported until after you import the CI Type data.).

Defining a source Actual CI schema into a target

1. In the Maximo database, execute the script createTADDM71ActualCIDataSchema.db2.

Before you can use the integration adapter to create a mapping and migrate data, you need to install the source data schema for actual CI data in the Integration Composer repository in the Maximo database.

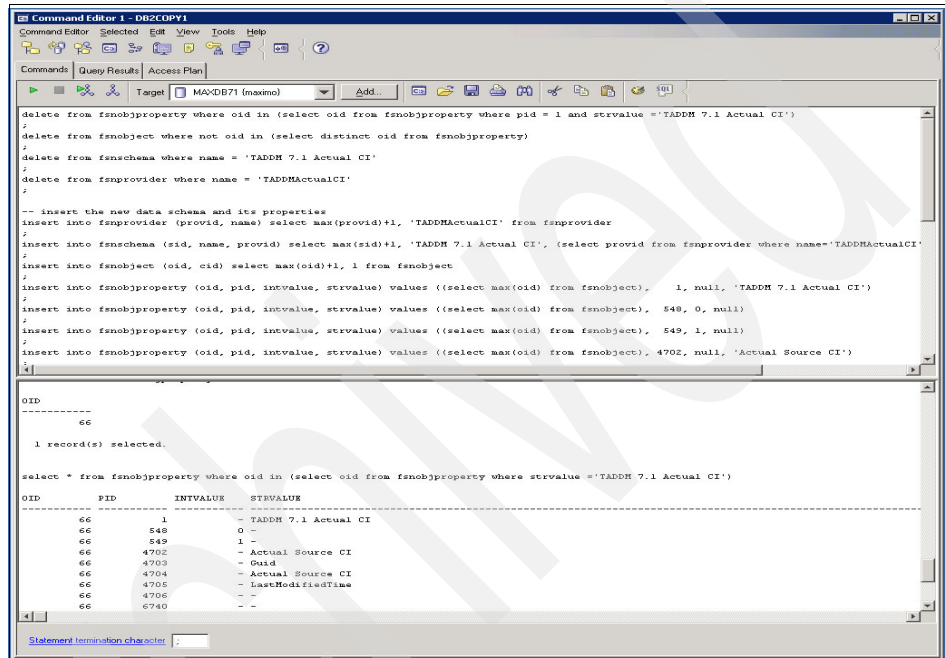


Figure 10-32 Create TADDM71 actual data schema

To install the actual CI data schema for the TADDM data source, use one of the following scripts:

- createTADDM71ActualCIDataSchema.db2 (for DB2)
- createTADDM71ActualCIDataSchema.ora (for Oracle)
- createTADDM71ActualCIDataSchema.sqs (for SQL)

To add the data schema, complete the following steps.

- a. Using an appropriate database query tool, execute the appropriate database script for your source data.
- b. Check for database script errors and resolve any errors. For error information, refer to the log files located in the log folder in your Windows or UNIX installation directory.

Creating a new data source for a source (TADDM) database

Open Integration Composer and create a new data source using the TADDM 7.1 Actual CI data schema.

Integration Composer uses a JDBC driver or an API to connect to data sources. Before you create a mapping, you must have a data source connection defined for the data source (in this case, the TADDM database) and the target (the Maximo database).

After you define the connection parameters for the data source, Integration Composer saves them in its repository and displays them whenever you attempt to connect to the data source (more on this topic is provided in the book *IBM Tivoli Integration Composer System Administrator's Guide*). The only parameter that Integration Composer requests from you in real time is the password for the data source (that is, either a database password for database users, or, if using the Configuration Discovery and Tracking API, a password associated with the user login account).

To define a data source, complete the following steps:

1. Sign in to Integration Composer using a valid user ID and password. The IBM Tivoli Integration Composer window is displayed.

2. Select **Define New Data Source**. The Data Schema page is displayed in the Define a New Data Source window (Figure 10-33). The Data Schema page lists data schemas that were installed with Integration Composer plus any data schemas that you created using database scripts or the data schema feature in Integration Composer. Select **TADDM 7.1 Actual CI**.

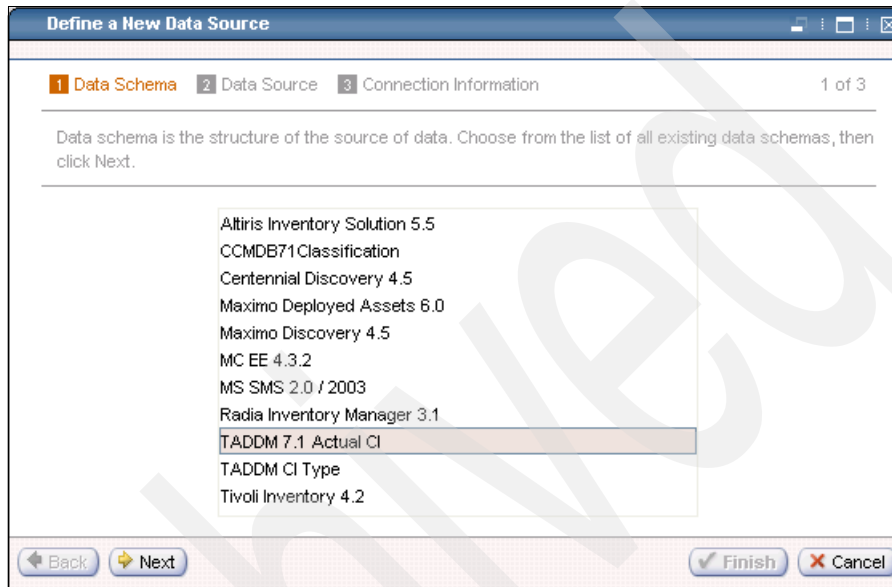


Figure 10-33 Defining a data source using TADDM7.1 Actual CI

3. Select a data schema and click **Next**. The Data Source page is displayed (Figure 10-34 on page 275). In the Data Source field, enter a name for the data source (the name must have at least two characters and is case sensitive) and click **Next**. The connection information fields are displayed. We selected the name **TADDM_Actual_CI**.

Figure 10-34 Naming the data source

4. In the Connection Method drop-down list, select a connection method. Enter the connection parameters, as required. The connection method that you selected determines the fields that Integration Composer displays. Since we are defining a data source connection to TADDM, we pick **IBM Configuration Discovery and Tracking API**. Provide the connection parameters to TADDM:
 - Host name: <Fully_Qualified_Hostname>
 - Port: 9530
 - User name: administrator
 - Password: *****

5. Optional: If appropriate, click **Test Connection** to test the connection to the data source. Integration Composer displays a Test Connection dialog box. The text in the dialog box indicates whether the test was successful.
 - On the Connection Information page, click **Finish**. Integration Composer displays a Save dialog box (Figure 10-35).
 - Click **OK**. Integration Composer returns you to the IBM Tivoli Integration Composer window.

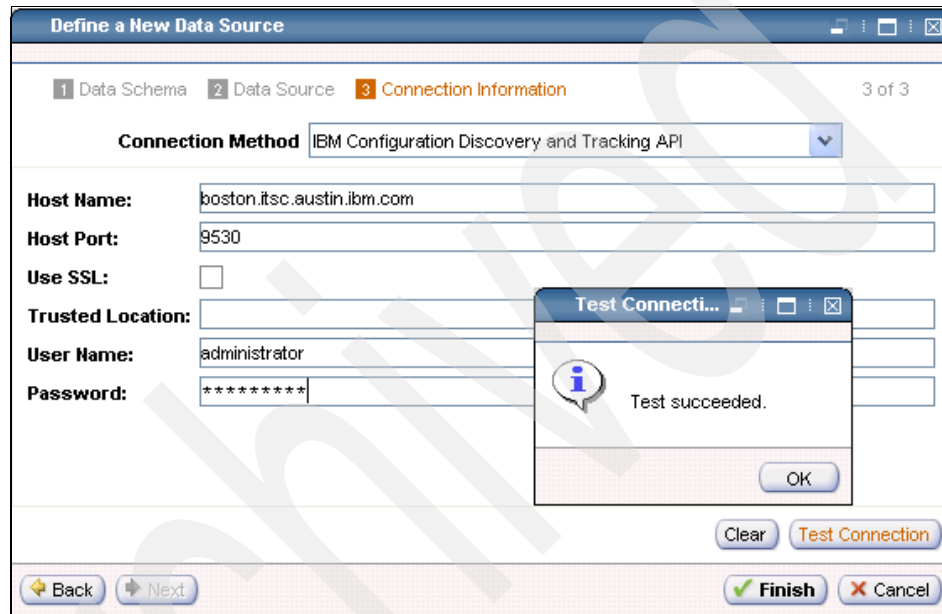


Figure 10-35 Connection parameters to TADDM DB and testing connection

Note that in an Integration Composer session, if you connect to one of the defined data sources, Integration Composer keeps the data source connection open throughout the session unless you complete one of the following steps:

- a. Run a mapping for the data source.
- b. Use the Close Data Source Connection option in the Data Source menu in the IBM Tivoli Integration Composer window to close the connection.
- c. Delete the open data source.

Defining a data schema for a target actual CI into a target DB

Open Integration Composer and create a new data schema and data source and name both of them CCMDB71ActualCI.

Note: It is important that you name the data schema CCMDB71ActualCI or you will have to later make changes in the qualifier DB script.

Before you can use the integration adapter to create a mapping and migrate data, you need to install the data schema for the target Actual CI data in the Integration Composer repository in the Maximo database.

To install the data schema, use the actual CI data file that the integration adapter provides (CCMDB71ActualCI.schm).

To add the data schema, complete the following steps:

1. If you are not already signed in, sign in to Integration Composer using a valid user ID and password. The IBM Tivoli Integration Composer window is displayed.
2. From the IBM Tivoli Integration Composer window, select **Define New Data Schema**. Integration Composer displays the Data Schema field in the Define a New Data Schema window.

Note: To review or change your previous selections, click **Back**. To cancel this procedure and return to the main Integration Composer window, click **Cancel**.

3. In the Data Schema field, type CCMDB71ActualCI as the name of the new data schema (data schema names are case sensitive).

Note: You can type a different name for the new data schema. However, if you do, you will have to change the name CCMDB71ActualCI in the qualifier script (that you run in “Finalizing the target data schema for CI Type data” on page 262) in order to match the alternative name that you typed here.

4. Click **Next**. The Data Source field is displayed (Figure 10-36).

Define a New Data Schema

1 Data Schema 2 Data Source 3 Connection Information 1 of 3

Data schema is the structure of the source of data. Enter the name of a new data schema, then click Next.

Data Schema:

Back Next Finish Cancel

Figure 10-36 Data schema for Actual CI

5. In the Data Source field, enter a name for the data source (the name must have at least two characters and is case sensitive) and click **Next**. The connection information fields are displayed.

Define a New Data Schema

1 Data Schema 2 **Data Source** 3 Connection Information 2 of 3

Data source is the name of the source of data and its related data schema. Type in a new data source name, then click Next.

Data Source:

Back Next Finish Cancel

Figure 10-37 Data Source to target DB (Maximo) for Actual CI

6. In the Connection Method drop-down list, select a connection method. In this case, the connection method would be IBM DB2 JDBC Driver.
7. Provide the connection parameters to connect to the Maximo database:
 - Host name: <Fully_Qualified_Hostname>
 - Host port: 50005
 - Database: maxdb71
 - User name: maximo
 - Password: *****

Note: The above connection parameters values could be different depending how you have set up your environment.

8. Optional: If appropriate, click **Test Connection** to test the connection to the data source. Integration Composer displays a Test Connection dialog box. The text in the dialog box indicates whether the test was successful.

9. On the Connection Information page, click **Finish**. The Data Schema window is displayed (Figure 10-38). Integration Composer displays the root class in red because it has no properties associated with the class.

Note: The display properties that you set for your computer can affect colors. The color displayed on your computer can vary.

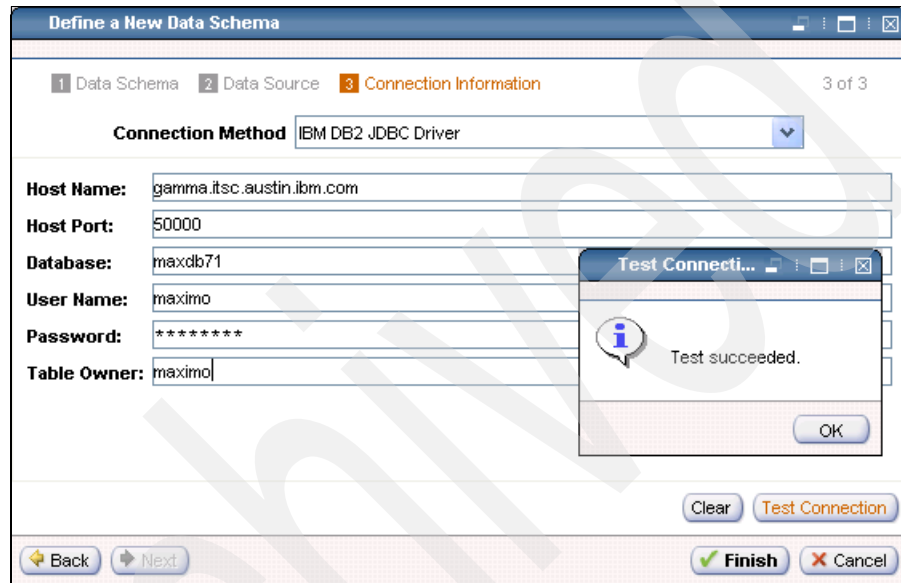


Figure 10-38 Connection parameters to target Maximo DB

10. From this window, you can import a data schema file provided with this integration adapter. To import the data schema file provided with the integration adapter, from the Select Action menu in the title bar of the Data Schema window, select **Import Data Schema**, as shown in Figure 10-39 on page 281. The Import Data Schema dialog box is displayed (Figure 10-40 on page 281). The Import Data Schema dialog box lists the data schemas that you copied to the data schema folder.

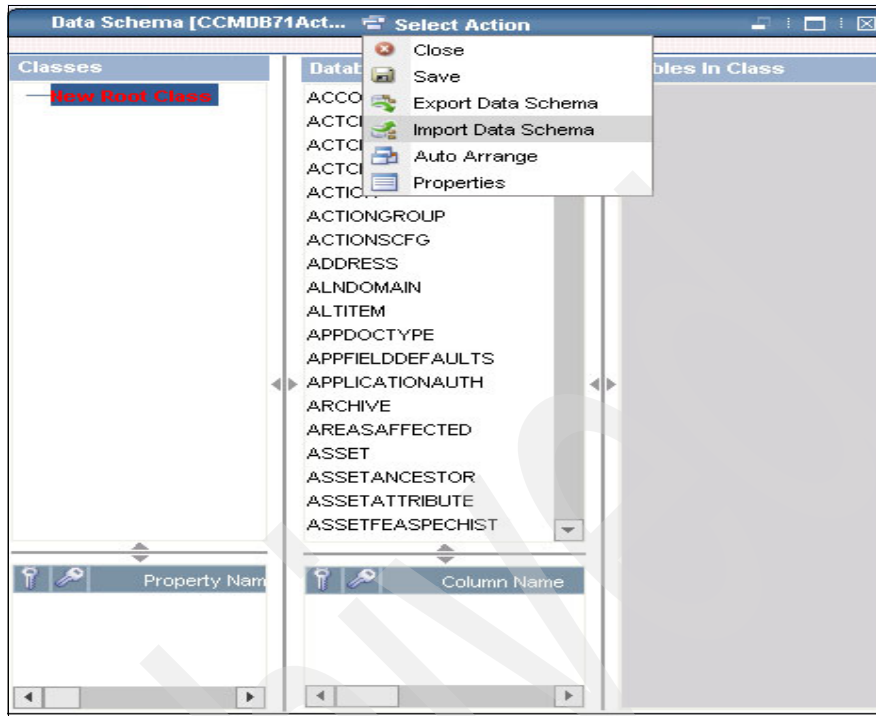


Figure 10-39 Import Actual CI schema in target DB

11. In the Import Data Schema dialog box, select the data schema file that you want to import. For the Actual CI target data, select **CCMDB71ActualCI.schm**. Integration Composer populates the File name field with the selected file name.

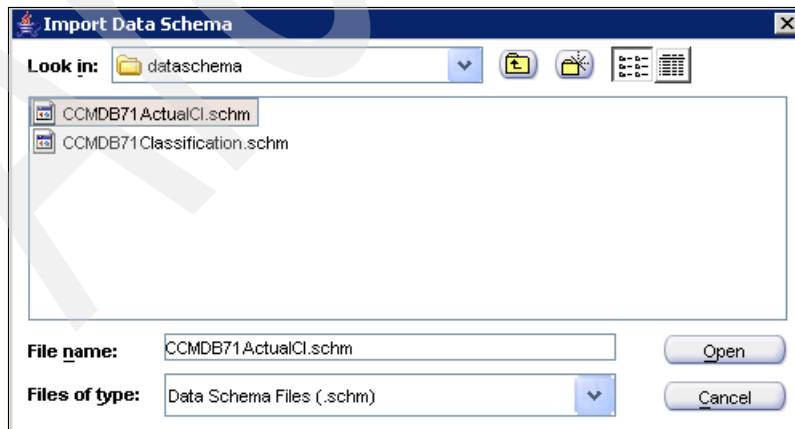


Figure 10-40 Import schema CCMDB V7.1 Classification

12. Click **Open**. Integration Composer imports the data schema file. If discrepancies exist between the data source and the data schema, the Data Schema Analysis window is displayed (Figure 10-41). This window lists discrepancies between the data schema and the corresponding data source. Integration Composer displays errors that it can correct if you check the error.
13. Review the errors in the Data Schema Analysis window and click **Fix Errors**. Integration Composer repairs the errors and opens the Data Schema window (Figure 10-41).

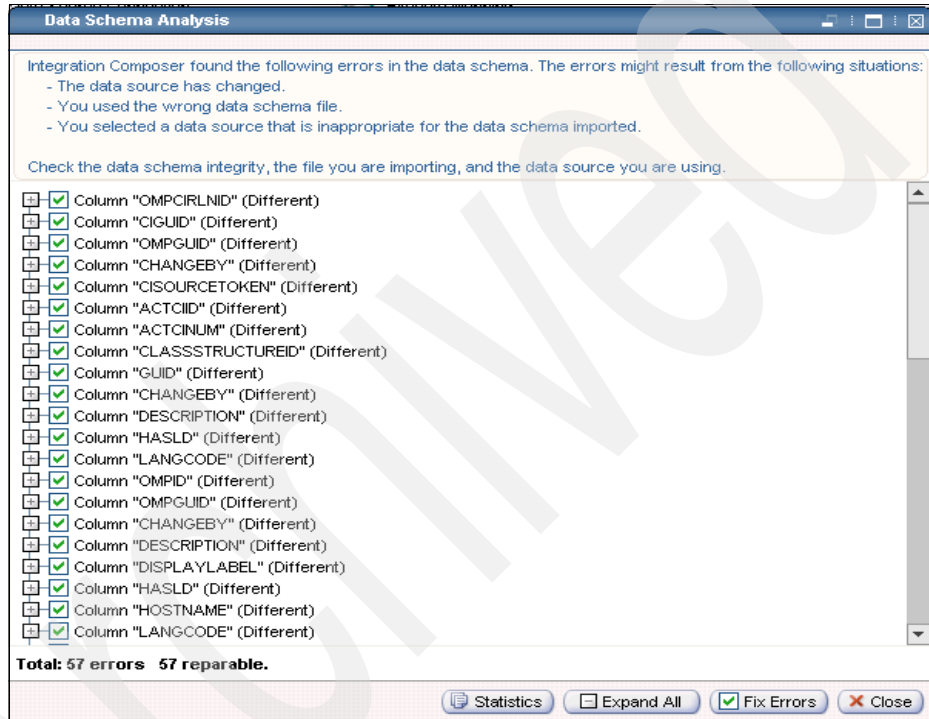


Figure 10-41 Import CCMDB7.1 Actual CI errors

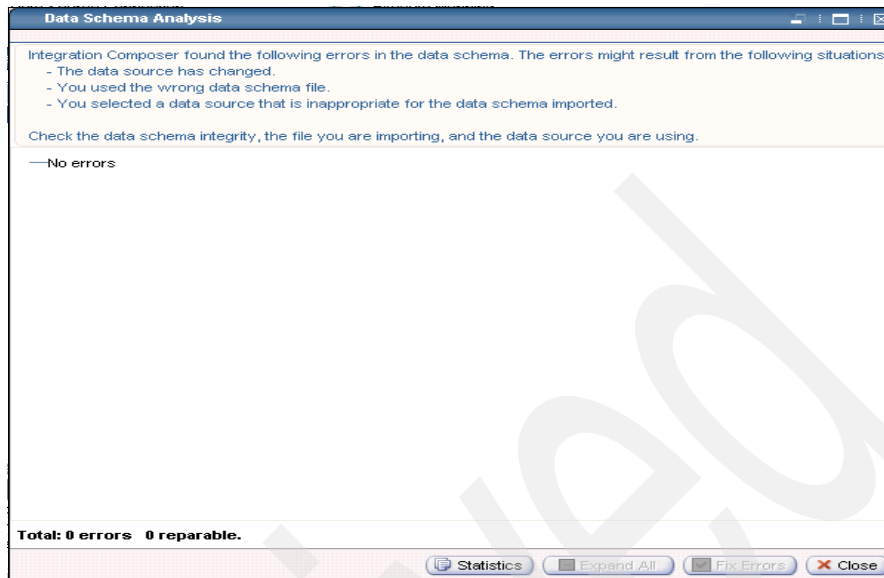


Figure 10-42 Errors fixed

14. After you finish working with the data schema, select **Save** from the Select Action menu (Figure 10-43 on page 284). Integration Composer saves the data schema.

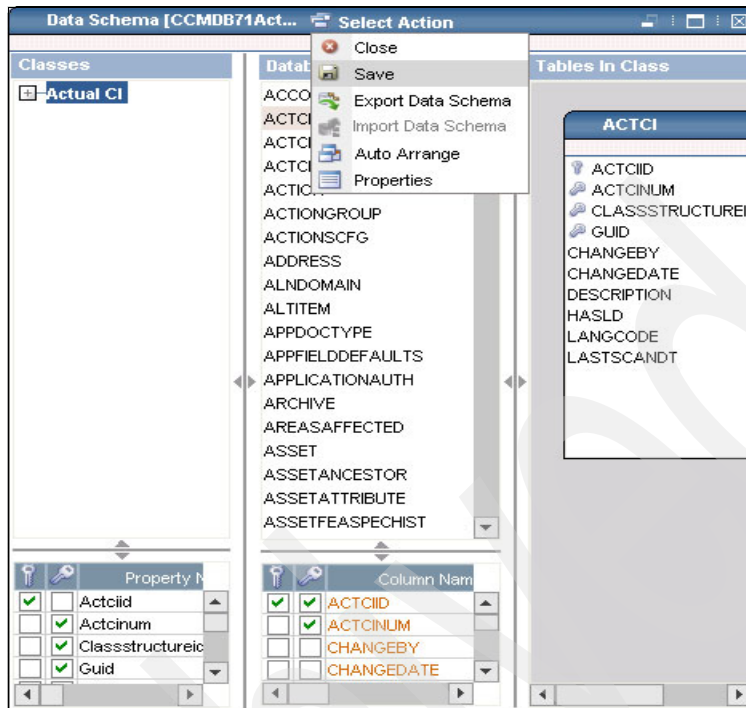


Figure 10-43 Saving the import schema

15. Select **Close** from the Select Action menu to close the data schema.
Integration Composer closes the data schema and displays the IBM Tivoli Integration Composer window.
16. Sign out of Integration Composer.

Finalizing the target data schema for actual CI

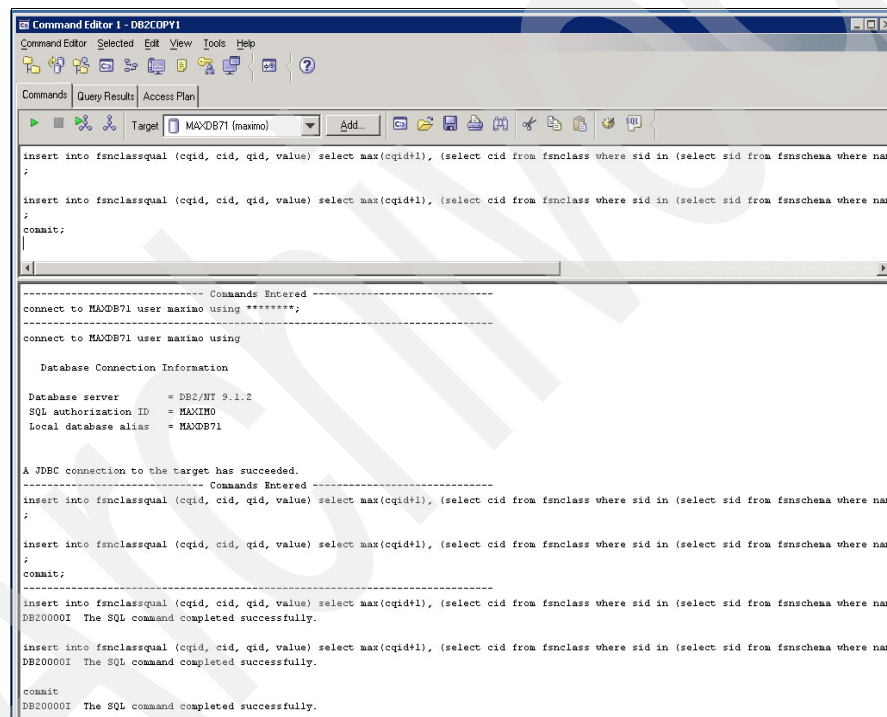
Before you can use the integration adapter to create a mapping, you need to finalize the target data schema in the Integration Composer repository in the Maximo database. Finalizing the data schema allows Integration Composer to delete certain class records, in addition to merely inserting and updating them, when executing a mapping. By performing this finalization task, you are modifying the way Integration Composer handles data when it performs a data import.

To perform this task, use one of the following scripts, as appropriate for your database:

- ▶ qualifierCCMDB71ActualCI.db2 (we are using this script for DB2)
- ▶ qualifierCCMDB71ActualCI.ora
- ▶ qualifierCCMDB71ActualCI.sqs

To finalize the target data schema, complete the following steps:

1. Sign out of Integration Composer, if you have not already done so.
2. Optional: If you named the new data schema something other than CCMDB71ActualCI in “Defining a data schema for a target actual CI into a target DB” on page 277, change the name CCMDB71ActualCI in the qualifier script to match the alternative name that you provided.
3. Using an appropriate database query tool, execute the above qualifier script against the Maximo database for your target data (Figure 10-44).
4. Check for database script errors and resolve any errors. For error information, refer to the log files located in the log folder in your Windows or UNIX installation directory.



The screenshot shows a 'Command Editor 1 - DB2COPY1' window. The top toolbar includes icons for file operations and a 'Query Results' tab. The main text area contains two SQL scripts. The first script is an 'insert into fnsclassqual' statement. The second script is a 'commit;' statement. Below the scripts, the 'Query Results' tab is active, displaying the execution output. The output shows the connection to the Maximo database using the 'MAXIMO' user, followed by the successful execution of the SQL commands. The output text is as follows:

```
----- Commands Entered -----
connect to MAXDB71 user maximo using *****;
connect to MAXDB71 user maximo using

Database Connection Information

Database server      = DB2/NT 9.1.2
SQL authorization ID = MAXIMO
Local database alias = MAXDB71

A JDBC connection to the target has succeeded.
----- Commands Entered -----
insert into fnsclassqual (cqid, cid, gid, value) select max(cqid+1), (select cid from fnsclass where sid in (select sid from fnschema where nam
;

insert into fnsclassqual (cqid, cid, gid, value) select max(cqid+1), (select cid from fnsclass where sid in (select sid from fnschema where nam
;
commit;
-----
insert into fnsclassqual (cqid, cid, gid, value) select max(cqid+1), (select cid from fnsclass where sid in (select sid from fnschema where nam
DB20000I The SQL command completed successfully.

insert into fnsclassqual (cqid, cid, gid, value) select max(cqid+1), (select cid from fnsclass where sid in (select sid from fnschema where nam
DB20000I The SQL command completed successfully.

commit
DB20000I The SQL command completed successfully.
```

Figure 10-44 Execute qualifierCCMDB71ActualCI script

Creating a mapping for Actual CI Type

Before you can migrate actual CI data, you must create a new adapter mapping for the data you want to migrate and then import the appropriate mapping file (TADDM71ActualCIToCCMDB71ActualCI.fsn) into the mapping that you create.

Before creating your mapping, make sure you have installed the data schemas for both your source and target actual CI data, in accordance with the instructions provided in the last sections.

Before you do the following steps, make sure that the MAXVARS table is updated correctly in the Maximo database.

To create a mapping, complete the following steps:

1. Sign in to the Integration Composer application using a valid user ID and password. Integration Composer displays the IBM Tivoli Integration Composer window.
2. Select **Create New Mapping**. The New Mapping window is displayed (Figure 10-45).
3. From the Source drop-down list of available data sources, select the data source.
4. From the Target drop-down list of available data sources, select the target database.

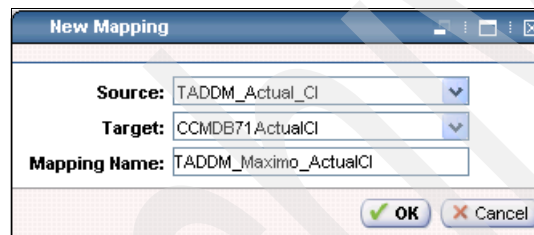


Figure 10-45 Import Actual CI mapping

5. In the Mapping Name field, enter a new mapping name (maximum of 255 characters). Mapping names are case sensitive; for example, taddm is not the same as TADDM.
6. Click **OK**, and select one of the following options:
 - If Integration Composer opens the Mapping window, go to step 9.
 - If Integration Composer opens the connection information fields in the Open Source Data Source window (Figure 10-46 on page 287), use the following procedure to complete the connection information:
 - In the Open Source Data Source window, accept the defaults established during the last connection to the data source.
 - Enter the password for accessing your source data.

Open Source Data Source: TADDM_Actual_CI

1 Connection Information 1 of 1

Connection Method IBM Configuration Discovery and Tracking API

Host Name: boston.itsc.austin.ibm.com

Host Port: 9530

Use SSL: ☐

Trusted Location:

User Name: administrator

Password: *****

Clear Test Connection

Back Next Finish Cancel

Figure 10-46 Log into TADDM DB

- Click **Finish** to establish the connection to the source.
- In the Open Target Data Source window (Figure 10-47 on page 288), accept the defaults established during the last connection to the data source.
- Enter the (database or user login account) password for accessing your target data.

Open Target Data Source: CCMDB71ActualCI

1 of 1

Connection Information

Connection Method: IBM DB2 JDBC Driver

Host Name: gamma.itsc.austin.ibm.com

Host Port: 50000

Database: maxdb71

User Name: maximo

Password: *****

Table Owner: maximo

Clear Test Connection

Back Next Finish Cancel

Figure 10-47 Log into the Maximo DB

- Click **Finish** to establish the connection to the target. Integration Composer opens the Mapping window (Figure 10-48 on page 289).

Mapping: TADDM_Maximo_ActualCI Select Action

Source: TADDM_Actual_CI

Actual Source CI

Key	Relation	Name	Type	Length	Value
		Guid	String	192	
		Classification	String	192	
		Classstructureid	String	192	
		Description	String	192	
		DisplayName	String	192	
		Label	String	192	
		SourceToken	String	192	

Back Next

Target: CCMDB71ActualCI

Actual CI ss: Case Number: 0

Key	Relation	Name	Type	Length	Value
		Actcid	int	10	
		Actcinum	String	384	
		Classstructureid	String	40	
		Guid	String	384	
		Changeby	String	60	
		Changdate	Timestamp	30	

Clear All Expression Builder Back Next

Figure 10-48 TADDM to Maximo Actual CI Mapping window

- From the Select Action menu in the Integration Composer Mapping window, select **Import**. The Import Mapping dialog box is displayed (Figure 10-49 on page 290).

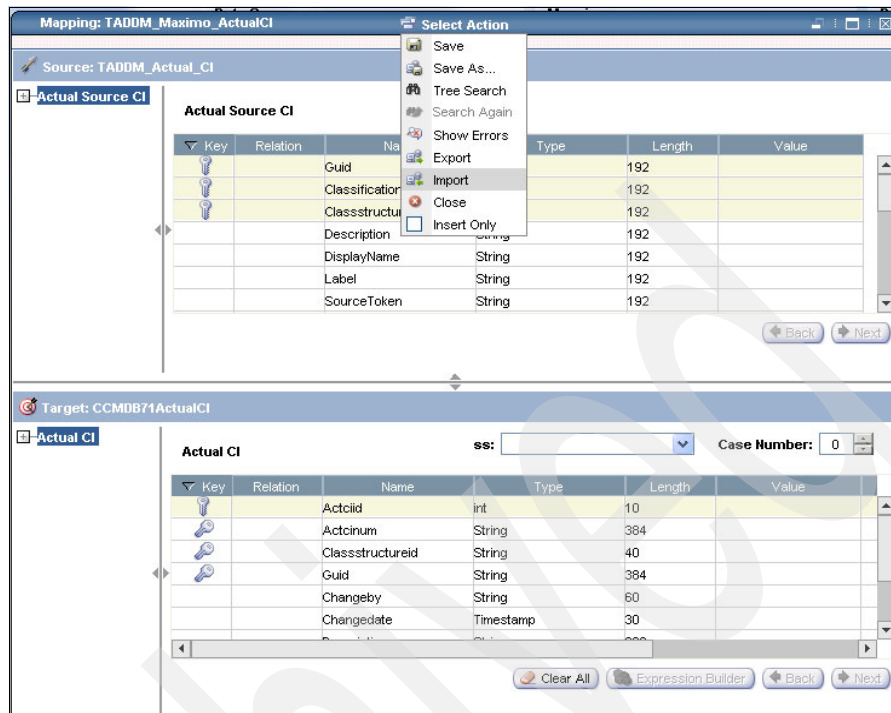


Figure 10-49 Import TADDM mapping file

- In the Import Mapping dialog box, select the mapping file for actual CI data, TADDM71ActualCIToCCMDB71ActualCI.fsn (Figure 10-50). Integration Composer populates the File name field with the selected file name.

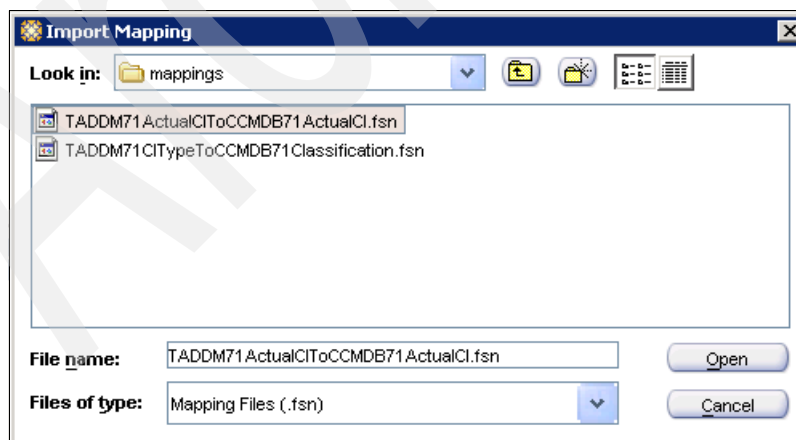


Figure 10-50 Import TADDM dialog box

9. Click **Open**. Integration Composer imports the mapping file.
10. If errors occur when you import the mapping, Integration Composer displays a dialog box that lists the errors and asks if you want to continue to import the mapping. To respond, select one of the following options:
 - Click **No** to cancel the import action without importing the mapping. Integration Composer closes the dialog box and does not import the mapping.
 - Click **Yes** to continue to import the mapping. Integration Composer imports the mapping and displays errors in red. Resolve the errors before saving the mapping.
11. After you finish the mapping, save the mapping with the existing name and select **Save** from the Select Action menu (Figure 10-51).

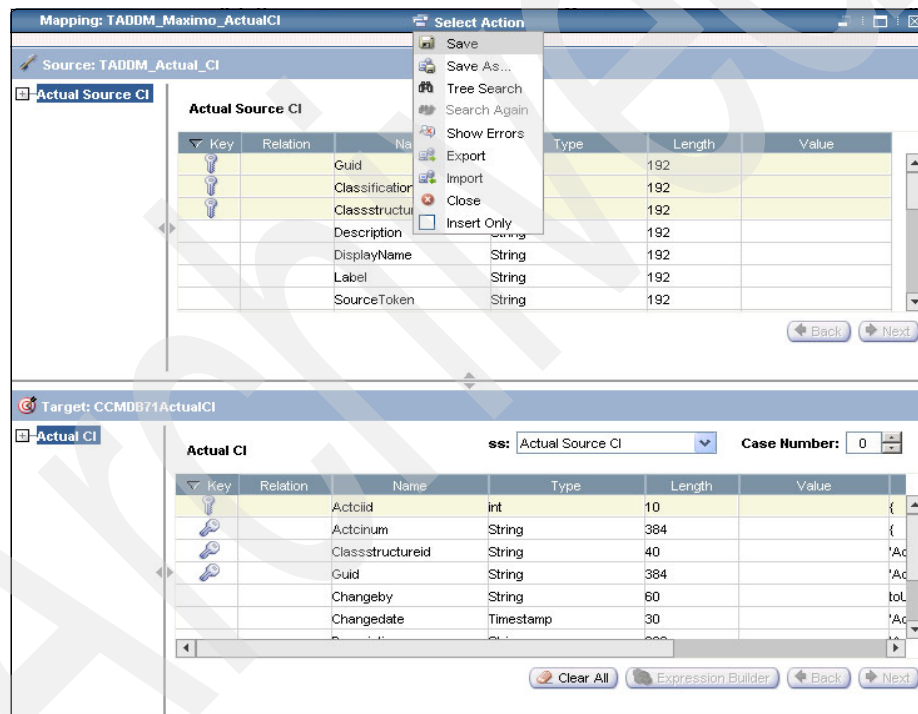


Figure 10-51 Saving mapping

12. To close the Mapping window, select **Close** from the Select Action menu. A Close Mapping dialog box is displayed.
13. In the Close Mapping dialog box, click **Yes**. Integration Composer closes the Mapping window and displays the IBM Tivoli Integration Composer window.

Execute the mapping

To execute a mapping using the Integration Composer interface, complete the following steps:

1. In the IBM Tivoli Integration Composer window, select **Execute Mapping**. Integration Composer displays the Execute Mapping window. The Execute Mapping window displays a table that lists the mappings currently available in Integration Composer. Select the **TADDM_Maximo_ActualCI** mapping.
2. In the list of mappings, verify that Integration Composer has selected the **Ready** check box for the desired mapping and then select one of the following options:
 - If Integration Composer has not selected the Ready field, click **Cancel**. Review the mapping.
 - If Integration Composer has selected the Ready field, the mapping has no syntax errors and is ready to be executed. Go to step 4.
3. To select the mapping to execute, click the row for that mapping in the list of mappings. Integration Composer displays the name of the mapping selected in the Mapping Name field (Figure 10-52).

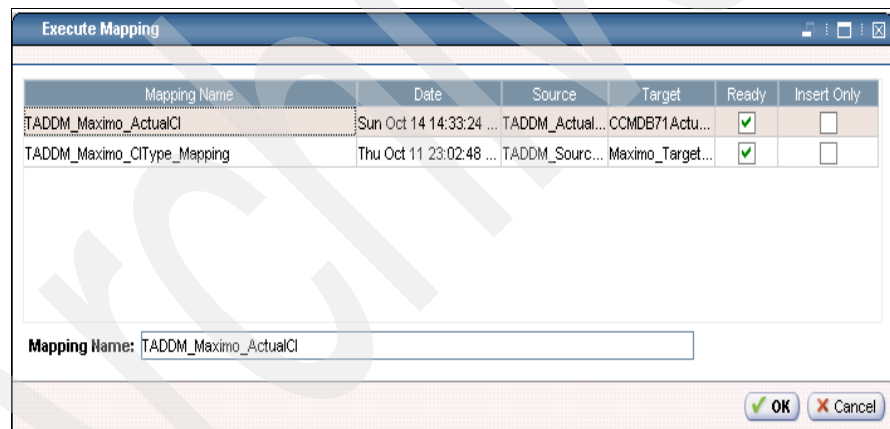


Figure 10-52 Executing Actual CI mapping

4. In the Execute Mapping dialog box, click **OK**. Integration Composer displays the Open Source and Target Source Data Sources window (Figure 10-53 on page 293). On the Connection Information page in the Open Source and Target Source Data Source window, accept the settings established for the last connection to the data source. Enter the password (Figure 10-54 on page 293).

Open Source Data Source: TADDM_Actual_CI

1 of 1

Connection Method IBM Configuration Discovery and Tracking API

Host Name: boston.itsc.austin.ibm.com

Host Port: 9530

Use SSL: ☐

Trusted Location:

User Name: administrator

Password: *****

Clear Test Connection

Back Next Finish Cancel

Figure 10-53 Log into source for executing Actual CI mapping

Open Target Data Source: CCMDB71ActualCI

1 of 1

Connection Method IBM DB2 JDBC Driver

Host Name: gamma.itsc.austin.ibm.com

Host Port: 50000

Database: maxdb71

User Name: maximo

Password: *****

Table Owner: maximo

Clear Test Connection

Back Next Finish Cancel

Figure 10-54 Log into target for executing Actual CI mapping

5. In the Open Source Data Source window, click **Finish**. Integration Composer displays the Mapping Execution Progress dialog box (Figure 10-55).

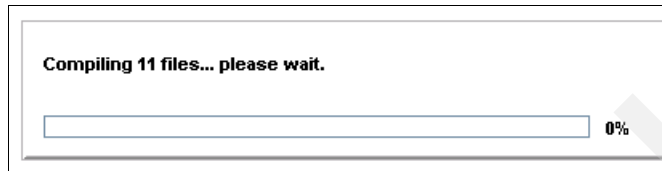


Figure 10-55 Compiling mapping

Note: It might take a while for the Progress Window to appear after you execute the mapping. The Integration Composer engine prepares and reads the data before it can display the progress window. Be patient and wait about five minutes for the progress window to appear and do not close the Integration Composer window.

6. In the confirmation dialog box after mapping completes successfully, click **OK**. Integration Composer displays the IBM Tivoli Integration Composer window (Figure 10-56).

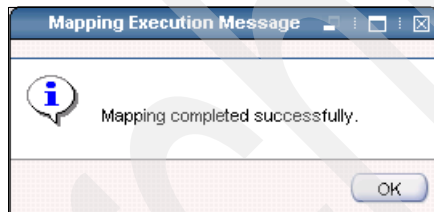


Figure 10-56 Mapping success

10.7 Transfer of new or updated CIs after successful migration

If you discover new CIs or update existing CIs in TADDM after you have successfully migrated TADDM CI Types (common data model) and actual CIs into the CCMDB database, then there are two different approaches you can adopt to migrate your new or updated CI data. Adapter configuration and setup is a one time operation. Later CI updates or additions in TADDM can be migrated using one of the following two strategies:

- ▶ Execute Mapping through Insert
- ▶ Execute Mapping through Updates

As Integration Composer executes a mapping, for each object in the source and target, it creates Java source code and compiles the source code. When integration Composer completes a top-level object, including all its children and reference classes, it commits the data for that object to the database.

10.7.1 Execute mapping through insertion

If you have executed the mapping successfully in the past, then it means Integration composer knows about your last scan date, and you can create a mapping that only inserts records for objects that were not imported in previous runs. To do this, you must select **Insert Only** from the Select Action menu in the Mapping window when you create the mapping. When Integration Composer imports data using a mapping with the Insert Only flag set to true, Integration Composer only creates new records in the target. It does not update existing records. This feature works only if the source data contains a last scan date, meaning in the past you have run the mapping and imported the data from a source successfully.

1. Open Integration Composer and click **Existing Mapping**.
2. The Open Mapping window will appear (Figure 10-57). Select the **Actual CI** mapping and click **OK**. Actual Mapping is responsible for migrating the actual CI instances from TADDM to Maximo.

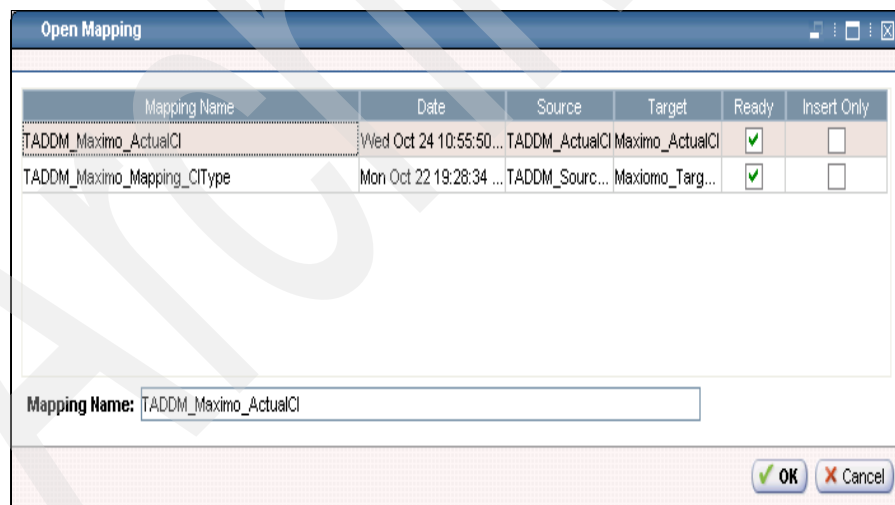


Figure 10-57 Open Mapping window for Actual CI mapping

3. You might get Login prompts to connect to your source (TADDM) and target (Maximo) data source. Fill in the passwords and click **Finish**.

- You will get a Mapping window where you will mark the **Insert** check box under the Action menu. Save the mapping under the Action menu and close the mapping (Figure 10-58).

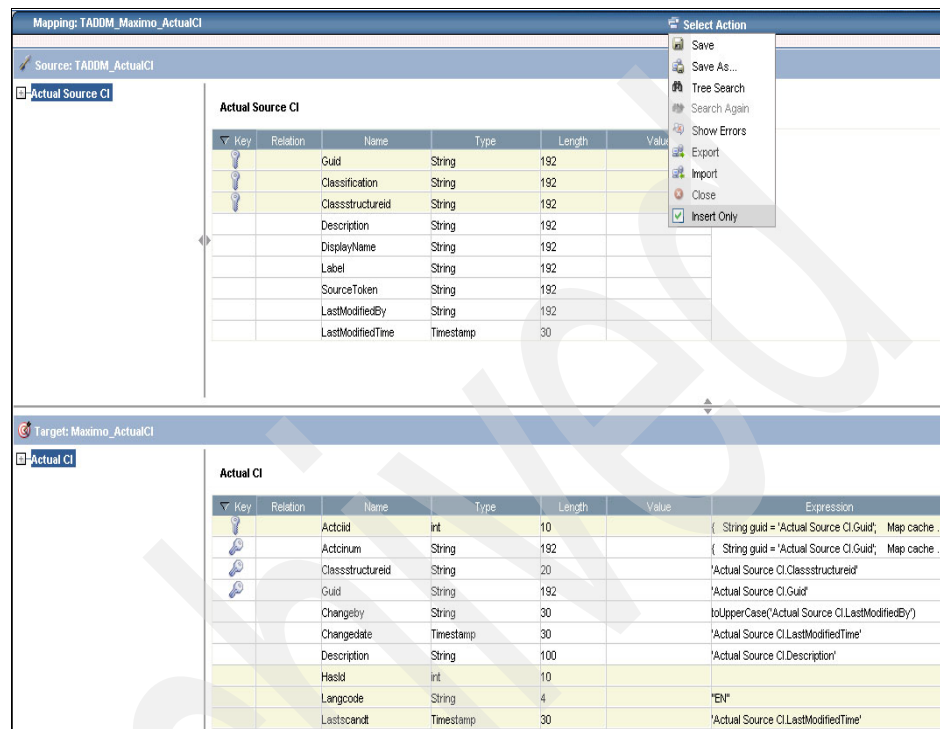


Figure 10-58 Check Insert box in existing mapping

- Once the mapping is saved and the window is closed, click **Execute the Mapping**. You will see the Execute Mapping window has the Insert check box marked (Figure 10-59 on page 297). This means that mapping will now execute on new CIs in TADDM and will not make any updates to existing CIs in TADDM over Maximo.

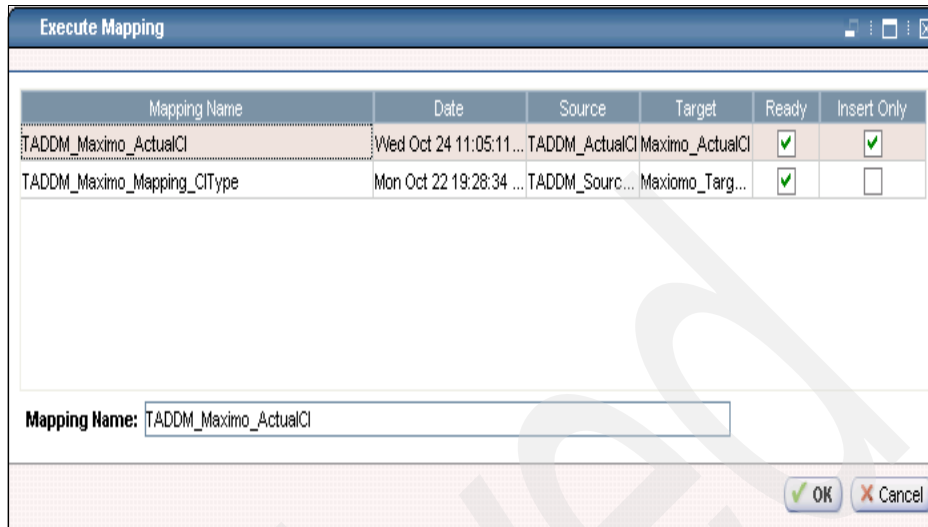


Figure 10-59 Execute mapping with the Insert box checked

- Click **OK**. The migration of only new CIs will start from TADDM to Maximo. You will notice a relatively small number of total CIs (2381) in Figure 10-60.

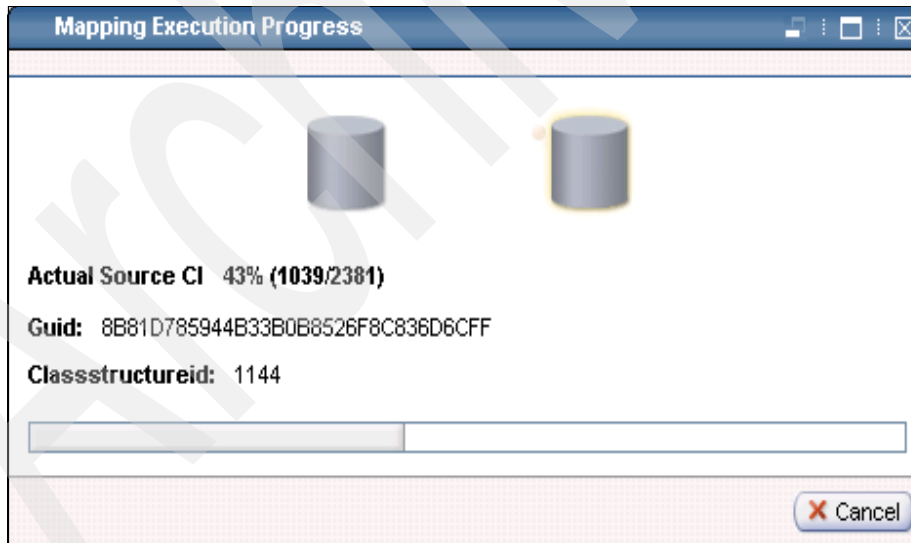


Figure 10-60 Mapping execution progress migration for only new CIs

As shown in Figure 10-61, we have added new CIs in TADDM that we have migrated to Maximo using the Insert Only approach. When you use this approach, no other updates will migrate in Maximo and the existing data in Maximo remains intact. You will find new additions or inserts in Maximo once the migration is completed from TADDM, as shown in Figure 10-62 on page 299.

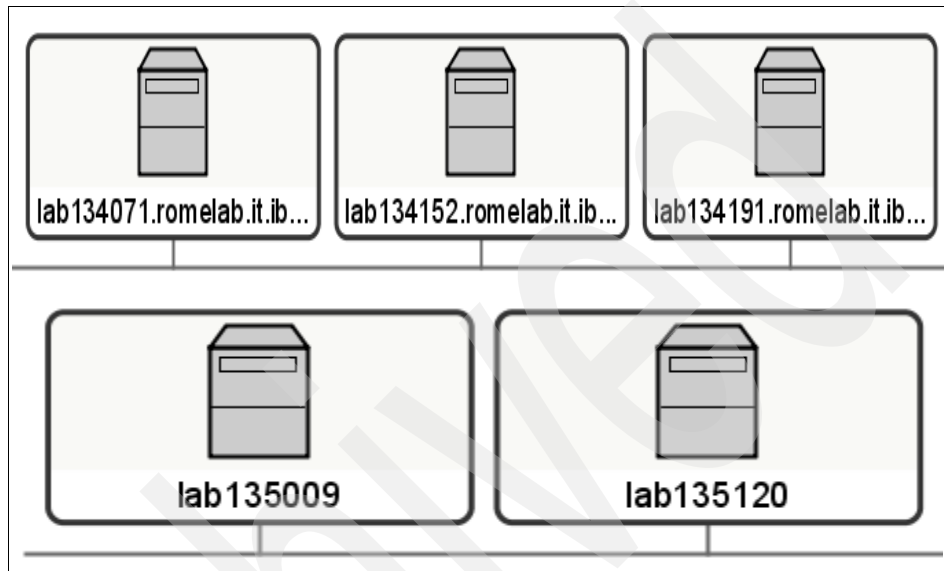


Figure 10-61 TADDM CI loaded through DLA

Actual CI	Description	Classification	Classification
lab0%		sys.com%	
LAB1341071.ROMELAB.IT.BM.COM-9350		SVS.COMPUTERSYSTEM	TOPICLASS\SVS.COMPUTERSYSTEM
LAB134152.ROMELAB.IT.BM.COM-9357		SVS.COMPUTERSYSTEM	TOPICLASS\SVS.COMPUTERSYSTEM
LAB134181.ROMELAB.IT.BM.COM-9356		SVS.COMPUTERSYSTEM	TOPICLASS\SVS.COMPUTERSYSTEM
LAB133008-9357		SVS.COMPUTERSYSTEM	TOPICLASS\SVS.COMPUTERSYSTEM
LAB135120-9330		SVS.COMPUTERSYSTEM	TOPICLASS\SVS.COMPUTERSYSTEM

Figure 10-62 New CI added in Maximo from TADDM with no other updates made to existing CIs in Maximo

10.7.2 Execute mapping through an update

To determine which object or attribute records to update, Integration Composer uses a last scan time stamp. When Integration Composer processes a new object, it records the data for that source object in the Integration Composer repository. Upon subsequent mapping executions, Integration Composer compares the last scan date in the Integration Composer repository with the scan date in the source data source and performs one of the following actions:

- ▶ If the dates match, Integration Composer skips the source instance.
- ▶ If the dates differ, Integration Composer processes the expressions for the source instance and updates the last scan date in the repository.

By doing this task, Integration Composer ensures that it processes only the objects that have changed since the last scan date.

We made some updates in existing CI data in TADDM. Also, until now we have have not brought over all the relationship that a particular CI is carrying. For example, we did not migrate the Installed Software and CI relationship in Maximo. So, if we want to bring all the other relationships of a CI that exists in TADDM but are not yet migrated to Maximo, then we will want to execute the

Update Mapping. With Update Mapping, it will bring only the updates (or remaining relationships or attributes in TADDM) into Maximo without copying all the instances again from scratch.

Note: Remember to uncheck the Insert Only check box in the Execute Mapping if you want to execute updates. You can uncheck the Insert Only check box by opening the existing mapping, Go to Select Actions in the Mapping window, uncheck the **Insert Only** check box, save your changes, close the mapping window, and execute your mapping. This entire process has been shown in detail in 10.7.1, “Execute mapping through insertion” on page 295. Just reverse the same process for unchecking the Insert Only box in the Mapping Window.

Figure 10-63 shows rerunning the mapping with updates.

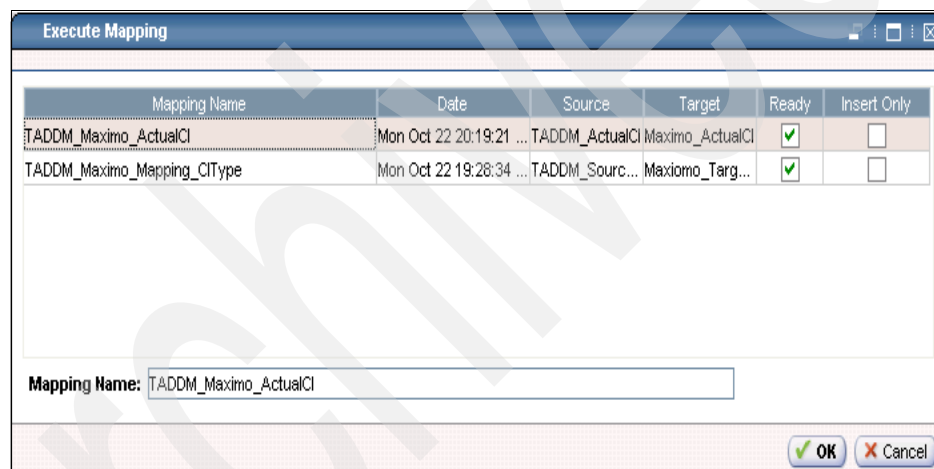


Figure 10-63 Rerun mapping with updates

TADDM CI and relationships

Let us consider a particular CI (lab134071.romelab.it.ibm.com) in TADDM and examine its relationship with other CIs in TADDM, as shown in Figure 10-64 on page 301.

Details				
Items: lab134071.romelab.it.ibm.com				
General	OS	Storage	Interfaces	Packages
Software Installed	Kernel Modules	Config Files	CI Info	Dependencies
Product Name	Version	Manufacturer	Description	Installed Location
4Suite	1.0.3	Red Hat		/usr/
ADMTAGINT	2.2.0	IBM		
ADMTAGIOR	2.2.0	IBM		
Canna	3.7	Red Hat		/etc/./usr/./var/
Canna-devel	3.7	Red Hat		/usr/
Canna-libs	3.7	Red Hat		/usr/lib/
ElectricFence	2.2.2.19	Red Hat		/usr/
FreeVhn	1.10	Red Hat		/etc/./usr/./var/lib/
FreeVhn-devel	1.10	Red Hat		/usr/
FreeVhn-libs	1.10	Red Hat		/usr/lib/
GConf2	2.8.1.1	Red Hat		/etc/gconf/./usr/
GConf2-devel	2.8.1.1	Red Hat		/usr/
HelixPlayer	1.0.1	Red Hat		/usr/
IBM HTTP Server	6.0.0.0	IBM		/opt/IBMHS
IBM Tivoli License Manager	2.2.0.0	IBM		/opt/IBM/TLM
IBMJava2-SDK	1.4.1.2	International Business Machines		/opt/
IBM_db2ad81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2ads81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2cac81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2cc81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2chen81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2ci81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2civ81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2conn81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2conv81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2crt81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2cucs81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2das81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2dc81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/
IBM_db2di81	8.1.0.64	International Business Machines Corp.		/opt/IBMdb2/V8.1/

Figure 10-64 TADDM CI and its associated relationship

We notice a relationship exists in TADDM between the Computer System CI (lab134071.romelab.it.ibm.com) and the Software CIs that are installed on this system, as shown in Figure 10-64.

This relationship does not exist in Maximo because we never activated the software CI Types in Maximo. We will now **ACTIVATE** the software CI Types in Maximo and rerun the mapping in ITIC with only updates to get all the changes and remaining relationship that exists in TADDM (Figure 10-65 on page 302).

Note: Executing the mapping with updates does not bring all the Actual CI data from scratch from TADDM to Maximo. It just brings the changes and any remaining relationships and instance data over to Maximo.

Source Actual CI	Relation	Target Actual CI	Containment?
LAB134071.ROMELAB.IT.BM.COM-6968	RELATION INSTALLED ON	LAB134071.LINUX.2.6.9-5.EL5.WP-4388	☐
LAB134071.ROMELAB.IT.BM.COM-6968	RELATION CONTAINS	000256C7C527.ETHERNET ADAPTER ETH0-6333	☒
LAB134071.ROMELAB.IT.BM.COM-6968	RELATION CONTAINS	9.87.134.71-6959	☐

Figure 10-65 CI Relationship migrated in Maximo

1. We change the status of all SOFTWARE CI Types to **ACTIVATE**, select **Go To → Administration → CI Types** and filter for CI Types related to Software (Figure 10-66).

Classification	Description	Classification Path	Top Level CI Type?	CI Type Status	Date Created
%Software%			☐		
1141	SYS.SOFTWARECOMPONENT	TOPOCLASS SYS.COMPUTERS STEW SYS	☐	ACTIVE	
1142	APP.SOFTWAREINSTALLATION	TOPOCLASS SYS.COMPUTERS STEW SYS	☐	ACTIVE	
1439	SYS.BIOS.BIOSSOFTWARECOMPONENT	TOPOCLASS SYS.BIOS.BIOSSOFTWARECON	☐	ACTIVE	
1588	SYS.SOFTWAREPATCH	TOPOCLASS SYS.SOFTWAREPATCH	☐	ACTIVE	
1602	PROCESS.MANAGEMENTSOFTWARESYSTEM	TOPOCLASS PROCESS.MANAGEMENTSOFT	☐	ACTIVE	
1720	APP.SOFTWAREFIX	TOPOCLASS APP.SOFTWAREFIX	☐	ACTIVE	
1722	APP.SOFTWAREUPDATE	TOPOCLASS APP.SOFTWAREUPDATE	☐	ACTIVE	
1745	APP.SOFTWAREMODULEFUNCTIONALGROUP	TOPOCLASS APP.SOFTWAREMODULEFNC	☐	ACTIVE	

Figure 10-66 Searching Software CI Types in Maximo

2. Select all Software CI Types or the ones you are interested in and change their status to **ACTIVE** (Figure 10-67 on page 303, Figure 10-68 on page 303, and Figure 10-69 on page 304).

The screenshot shows the 'CI Types' application window. At the top, there is a navigation bar with links for 'Go To', 'Reports', 'Start Center', 'Profile', 'Sign Out', and 'Help'. Below the navigation bar, there are buttons for 'Select Action' and 'Change Status'. A search bar is also present. The main area displays a table of CI Types with the following columns: Classification, Description, Classification Path, Top Level CI Type?, CI Type Status, and Date Created. The table contains several rows of data, including 'SYS.SOFTWARECOMPONENT', 'APP.SOFTWAREINSTALLATION', 'SYS.BOS.SOFTWARECOMPONENT', 'SYS.SOFTWAREPATCH', 'PROCESS.MANAGEMENT.SOFTWARESYSTEM', 'APP.SOFTWAREFX', 'APP.SOFTWAREUPDATE', and 'APP.SOFTWAREMODULEFUNCTIONALGROUP'.

Classification	Description	Classification Path	Top Level CI Type?	CI Type Status	Date Created
	%Software%				
1141	SYS.SOFTWARECOMPONENT	TOPICCLASS \ SYS.COMPUTERSYSTEM \ SYS			
1142	APP.SOFTWAREINSTALLATION	TOPICCLASS \ SYS.COMPUTERSYSTEM \ SYS			
1436	SYS.BOS.SOFTWARECOMPONENT	TOPICCLASS \ SYS.BOS.SOFTWARECOMPONENT			
1588	SYS.SOFTWAREPATCH	TOPICCLASS \ SYS.SOFTWAREPATCH			
1602	PROCESS.MANAGEMENT.SOFTWARESYSTEM	TOPICCLASS \ PROCESS.MANAGEMENT.SOFT			
1720	APP.SOFTWAREFX	TOPICCLASS \ APP.SOFTWAREFX			
1722	APP.SOFTWAREUPDATE	TOPICCLASS \ APP.SOFTWAREUPDATE			
1745	APP.SOFTWAREMODULEFUNCTIONALGROUP	TOPICCLASS \ APP.SOFTWAREMODULEFUNC			

Figure 10-67 Activate Software CI Types

The screenshot shows a 'Change Status' dialog box. It has a title bar with a close button. The dialog contains a 'New Status' dropdown menu set to 'Active', a 'Changed Date' text box with a calendar icon set to '10/24/07', and a 'Memo' text area. At the bottom, there are 'OK' and 'Cancel' buttons.

Figure 10-68 Activating Software CI Types

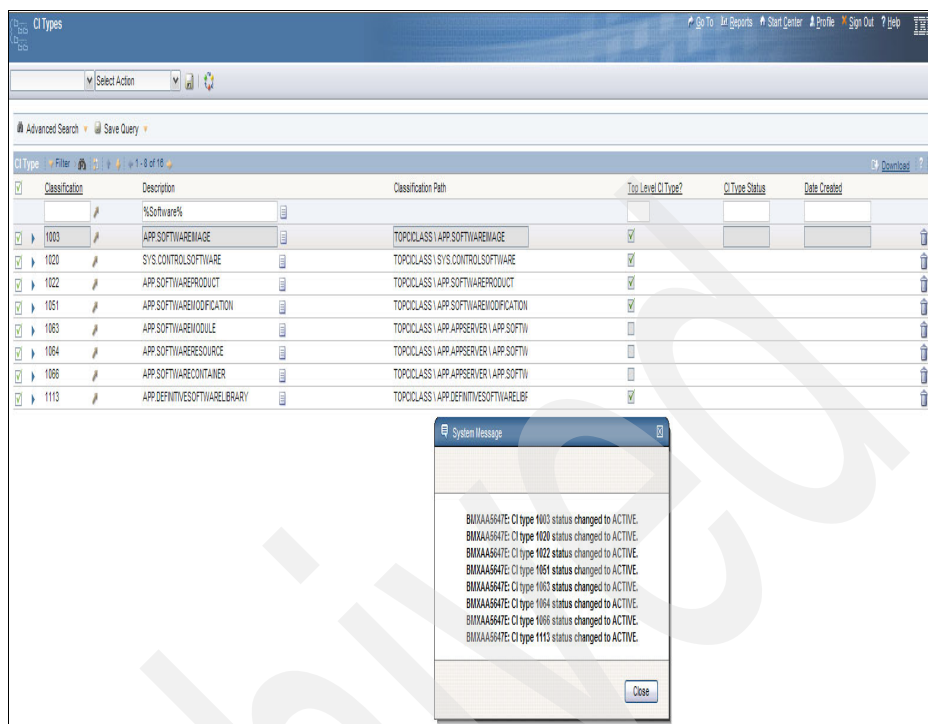


Figure 10-69 Software CI Types activated

Once all the Software related CI Types are activated, go to Integration Composer and rerun your mappings with updates. Remember to execute the mapping with the unchecked Insert Only box is actually running the mapping with updates, as shown in Figure 10-63 on page 300.

10.8 Import CI data through DLA

A Discovery Library Adapter (DLA) is a software program that extracts data from a source application, such as IBM Tivoli Monitoring, IBM Tivoli Business Systems Manager, ITCAM, z/OS, and so on.

The data that is imported by the DLA is provided in an XML file that comes either with the product or has to be created on the target system. First this data is imported into TADDM and then into the CCMDB. To create this XML file, you must use separate programs for each Tivoli product.

The software is installed with the TADDM installation. The command to import the XML file is `loadidm1.bat`. There are several options to run this command. For

example, use -o for overwrite and -f to point to the XML file directory you created first. You can find this program in the <TADDM_HOME>\cmdb\dist\bin.

Here is an example on how to import an XML file into a TADDM environment.

The TADDM server must be up and running, as shown in Figure 10-70.

```
C:\IBM\cmdb\dist\bin>control status
Discover: Started
GigaSpaces: Started
DbInit: Started
Tomcat: Started
Topology: Started
DiscoverAdmin: Started
Proxy: Started
EventsCore: Started

TADDM: Running
```

Figure 10-70 Status of TADDM server

Do the following steps:

1. Copy the XML file to a local directory, for example, C:\temp\zosDLA\.
2. Open a command prompt.
3. Change to the <TADDM-Install>\cmdb\dist\bin directory.
4. Run **loadidm1.bat -o -f <directory of XML file>** (in this case, use -o because it is the second import of that same XML file; you do not have to use -o option for the first import).

This process might take a while depending on the data size and structure.

After finishing this process, the imported CIs are present in the TADDM-UI. Refresh the TADDM UI and check the newly imported systems or applications, as shown in Figure 10-71.

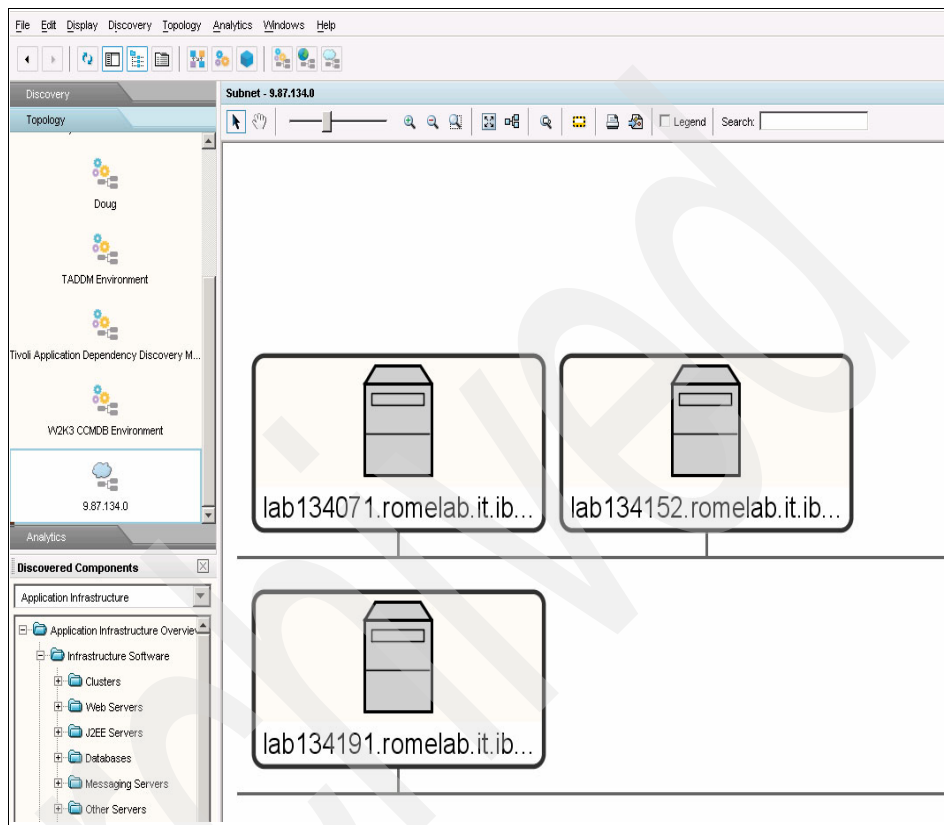


Figure 10-71 TADDM UI showing new CIs

For further information, refer to the *IBM Tivoli Application Dependency Discovery Manager Planning and Installation Guide*.

Launch in Context

The Launch in Context application comes with the base services of CCMDB V7.1. It gives you the ability to launch from the CCMDB Web User interface into different TADDM views (physical infrastructure, application infrastructure, and business application) or to other Operational Management Products like Tivoli Provisioning Manager (TPM), IBM Tivoli Monitoring (ITM), Tivoli Configuration Manager (TCM), and others.

Out of the box, CCMDB V7.1 comes with a set of default launch entries. The Launch in Context user interface can launch in context from the Actual CI and Authorized CI applications in the CCMDB Web interface directly into TADDM.

Predefined launch points determine which TADDM or Change History view is launched into.

The Launch in Context application is a servlet (Java program that runs in the Maximo Web server) that by default provides access to the following TADDM consoles:

- ▶ Product Console
- ▶ Domain Manager (Single domain or eCMDB)

The configuration of the URL and parameters of the launch entry point determines which console is started and what view is displayed.

If the variable for the Global Unique Identifier (GUID) in the CCMDB for each CI is provided in the console URL, the TADDM detail panel of that particular CI is presented as well.

You may choose to launch into a Change History Report view when a valid GUID is provided. If not, the General Change History view is launched for the manual report.

From the CCMDB Web User interface, it is also possible to launch into external operational management products (OMP), for example, IBM Tivoli Monitoring, Tivoli Provisioning Manager, or Tivoli Configuration Manager.

By assigning special launch points to users or groups, you can design the Launch in Context application according to the organization's needs.

Important: Before the Launch in Context application is started, two prerequisites have to be fulfilled in order to profit from the whole CCMDB V7.1 architecture:

- ▶ The TADDM data must be imported into the CCMDB database. This is done through the IBM Tivoli Integration Composer (ITIC). For further information about how to install, configure, and import data with ITIC, refer to Chapter 10, “TADDM and Process Layer integration” on page 231.
- ▶ The single sign-on setup must be activated and configured. For further information about how security integration between the process layer runtime and the discovery environment is implemented and configured, refer to Chapter 6, “CCMDB security architecture” on page 105.

11.1 Launch in Context graphical user interface

The following figures provide a short introduction to the Launch in Context application. The look and feel is exactly like all other applications launched from the Start Center.

Start the Launch in Context application from the Start Center by selecting **Go To → System Configuration → Platform Configuration → Launch in Context**, as shown in Figure 11-1.

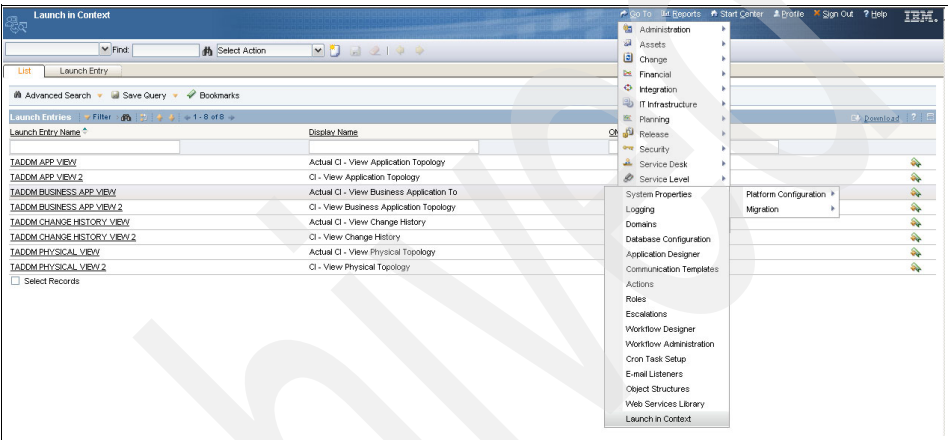


Figure 11-1 Configuration for Launch in Context application

In the List tab, the existing launch points are listed. These launch points are predefined and come with the CCMDB V7.1 installation. See Figure 11-2.

The screenshot shows the 'Launch in Context' application window with the 'List' tab selected. The main area displays a table of launch entries. The sidebar on the right contains a menu with various system configuration options. The 'Launch in Context' option is highlighted at the bottom of the sidebar.

Launch Entry Name	Display Name
TADDM APP VIEW	Actual CI - View Application Topology
TADDM APP VIEW 2	CI - View Application Topology
TADDM BUSINESS APP VIEW	Actual CI - View Business Application To
TADDM BUSINESS APP VIEW 2	CI - View Business Application Topology
TADDM CHANGE HISTORY VIEW	Actual CI - View Change History
TADDM CHANGE HISTORY VIEW 2	CI - View Change History
TADDM PHYSICAL VIEW	Actual CI - View Physical Topology
TADDM PHYSICAL VIEW 2	CI - View Physical Topology

Figure 11-2 List of launch points

In the Launch Entry tab, the URL specification and parameter setting is done, as shown in Figure 11-3.

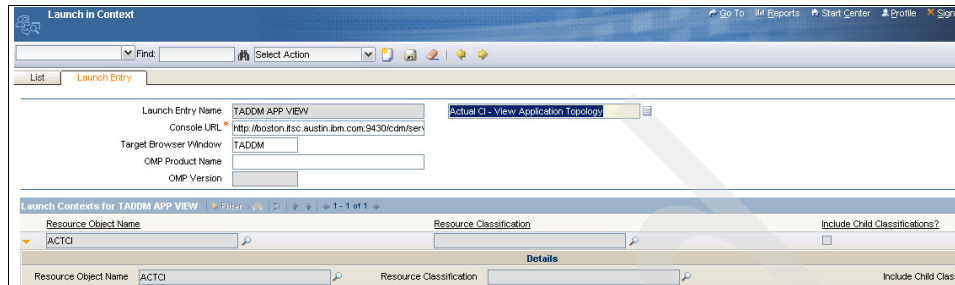


Figure 11-3 Specifying a URL

Figure 11-4 through Figure 11-7 on page 311 show the options on the Launch in Context toolbar.

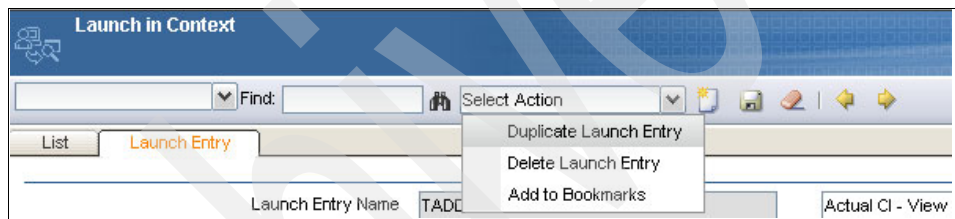


Figure 11-4 Select Action

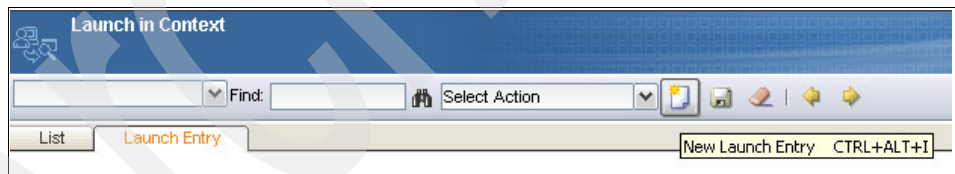


Figure 11-5 New Launch Entry

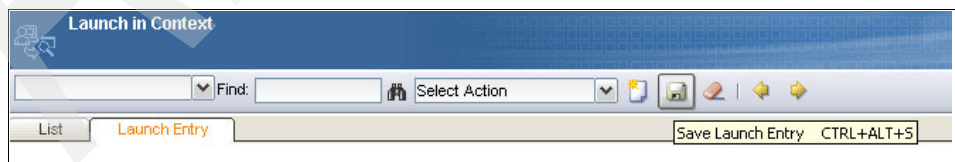


Figure 11-6 Save Launch Entry

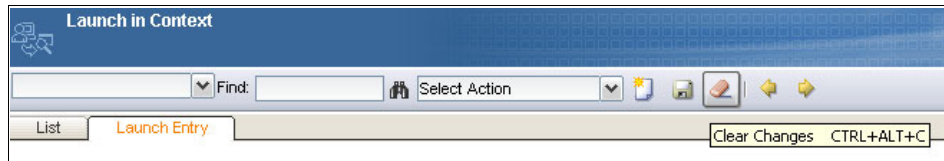


Figure 11-7 Clear Changes

For further information, online help is provided, as shown in Figure 11-8.

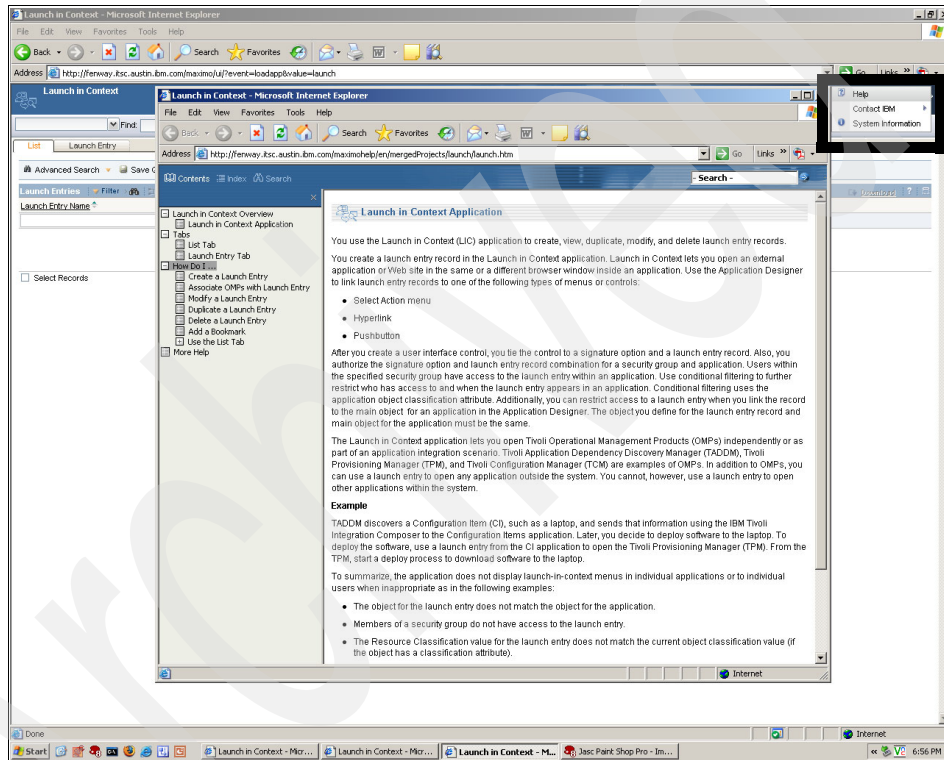


Figure 11-8 Help for Launch in Context configuration

11.2 Launch entry URL specifications and parameters

What the user is allowed to see depends on the URL specifications and parameter settings. To implement a new launch entry, add a URL according to your requirements.

Start the Launch in Context application from the Start Center by selecting **Go To → System Configuration → IT Infrastructure → Configuration Items or Actual Configuration Items**, as shown in Figure 11-9.

The screenshot shows the 'Launch in Context' application window. At the top, there's a title bar with the application name and a search bar. Below the title bar, there's a tabbed interface with 'List' and 'Launch Entry' tabs. The 'Launch Entry' tab is active, showing a form with the following fields: 'Launch Entry Name' (TADDM APP VIEW 2), 'Console URL' (http://boston.itsc.austin.ibm.com:9430/cdm/ser), 'Target Browser Window' (TADDM), 'OMP Product Name', and 'OMP Version'. A 'View Application' button is located to the right of the 'Launch Entry Name' field.

Figure 11-9 Specifying a URL

The URL syntax looks like this:

<Protocol>: //<TADDMHostName>:<TADDMPort>/<ContextRoot>/?<queryString>

Where:

Protocol	http or https Protocol used on TADDM server
TADDMHostName	Host name Host name of TADDM Domain or ECMDB server
TADDMPort	9430 Port where TADDM server is running on
ContextRoot	cdm/servlet/LICServlet URL prefix
queryString	Name1=value1&name2=value2 Syntax
nameValuePair	Name=value syntax
Separators	Between name value pairs ='&' Between name and value ='='

Table 11-1 provides a description of the parameters that can appear in the query string.

Table 11-1 Parameters in query string

Name	Description	Values
Graph	Topology view to be launched into.	physicalinfrastructure applicationinfrastructure businessapplications app_software app_physical bus_svc_software bus_svc_physical collection_relationship collection_physical
GUID	Variable for the General Unique Identifier of Configuration Items.	{GUID} {CIGUID}
Target	A new product console instance is opened or an existing product console is used.	new existing
View	Target view.	Change History View
days_previous	Number of days to go back for change history from current day.	any integer
console	Product console to launch.	web java

The following two tables describe the possible variations of setting these parameters for TADDM. Table 11-2 on page 314 shows how to Launch in Context into the TADDM Product Console (Java client) and Table 11-3 on page 315 shows how to Launch in Context into the TADDM Domain Manager (Web client).

Here is an example URL for an Actual CI launch into Product Console - View Application Topology:

`http://fenway.itsc.austin.ibm.com:9430/cdm/servlet/LICServlet?default.port=9433&default.server=fenway.itsc.austin.ibm.com&graph=applicationinfrastructure&guid={guid}`

Here is an example URL for an Actual CI launch into Web Console - View Business Applications:

<http://fenway.itsc.austin.ibm.com:9430/cdm/servlet/LICServlet?default.port=9433&default.server=fenway.itsc.austin.ibm.com&console=web&graph=businessapplications&guid={guid}>

Here is an example URL for an Actual CI launch into Product Console - Change History View:

<http://fenway.itsc.austin.ibm.com:9430/cdm/servlet/LICServlet?default.port=9433&default.server=fenway.itsc.austin.ibm.com&guid={guid}&view=changehistory>

Table 11-2 Launch in Context into the TADDM Product Console (Java client)

TADDM view to Launch in Context	Graph required	Parameter value	GUID required	Detail panels shown
From TADDM menu: "Physical Infrastructure"	Yes	physicalinfrastructure	No	No
From TADDM menu: "Application Infrastructure"	Yes	applicationinfrastructure	No	No
From TADDM menu: "Business Applications"	Yes	businessapplications	No	No
On Component context menu "Show Physical Infrastructure and Details"	No	GUID	Yes	Yes
On Business Application "Show Software Topology"	Yes	app_software	Yes	Yes
On Business Application "Show Physical Topology"	Yes	app_physical	Yes	Yes

TADDM view to Launch in Context	Graph required	Parameter value	GUID required	Detail panels shown
On Business Service "Show Software Topology"	Yes	bus_svc_software	Yes	Yes
On Business Service "Show Physical Topology"	Yes	bus_svc_physical	Yes	Yes
On Collection "Show Relationship Topology"	Yes	collection_relationship	Yes	Yes
On Collection "Show Physical Topology"	Yes	collection_physical	Yes	Yes
On Component "Change History"	Yes	changehistory	No	Yes

Table 11-3 Launch in Context into the TADDM Domain Manager (Web client)

TADDM view to Launch in Context	Graph required	Parameter Value	GUID required	Detail panel shown
Configuration Items details	No	GUID	Yes	Yes
Physical Infrastructure	Yes	physicalinfrastructure	No	No
Application Infrastructure	Yes	applicationinfrastructure	No	No
Business Applications	Yes	businessapplications	No	No
Change History	Yes	changehistory	Yes	Yes

Note: The GUID is not required for the application views, unless details need to be shown.

11.2.1 Launching the TADDM Product Console within CCMDB V7.1

After the TADDM data is loaded into the CCMDB database using the ITIC adapter, you can launch in context into the TADDM application. All URLs provide the GUID parameter to show the details window.

At first, TADDM will prompt you to download the confignia.jnlp, which contains the TADDM Java client, as shown in Figure 11-10.

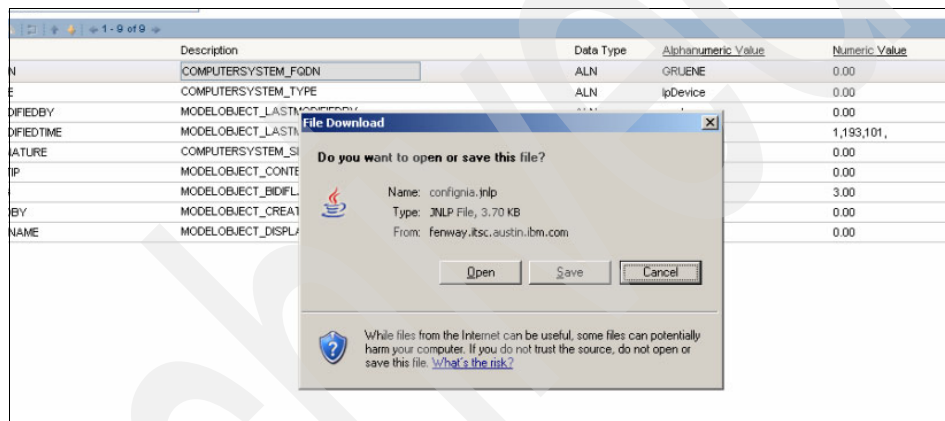


Figure 11-10 Prompt for downloading confignia.jnlp

Open the file. The TADDM application is started, as shown in Figure 11-11.



Figure 11-11 TADDM startup window

Depending on how long it takes for TADDM to build the topology view, the pop0-up window shown in Figure 11-12 might appear. Click **OK**.

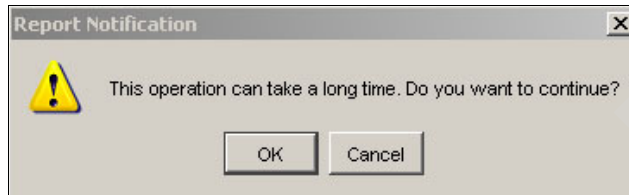


Figure 11-12 Possible pop-up window for slow launch

When single sign-on is set up, the appropriate TADDM view is launched. If no single sign-on is available, the user is prompted to log in.

Two scenarios are described below:

- ▶ Scenario 1: Launch from Actual CI into TADDM Application view
- ▶ Scenario 2: Launch from Actual CI into TADDM Physical view

Scenario 1: Launch from Actual CI into TADDM Application view

Do the following:

1. Start the Launch in Context application from the Start Center by selecting **Go To → System Configuration → IT Infrastructure → Configuration Items** or **Actual → *Select the designated Configuration Items* → *Select the CI from which to launch from* → Select Actions → View Actual CI Topology → Application**, as shown in Figure 11-13 on page 318.

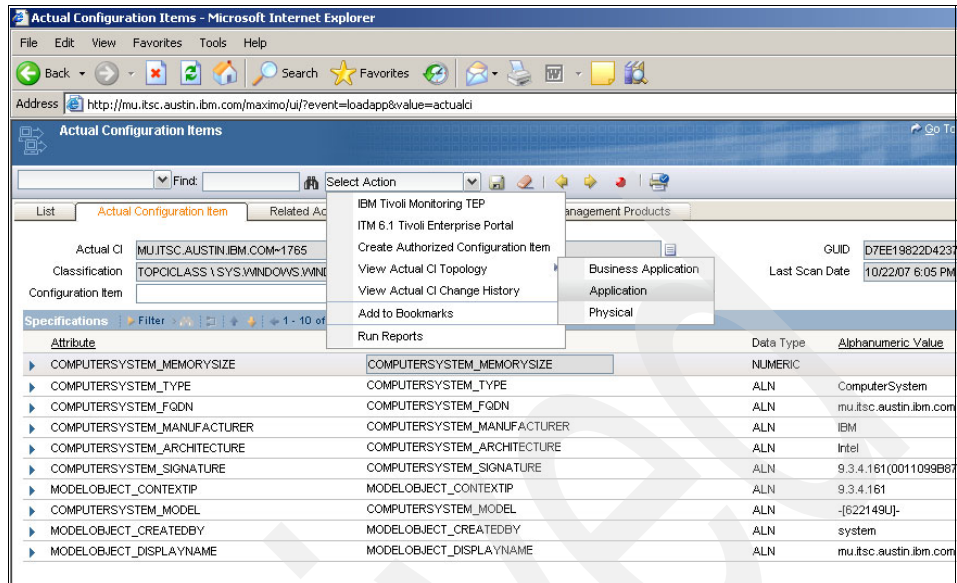


Figure 11-13 Starting Launch in Context from the Actual CI application

2. Enter the Launch in Context Console URL:

`http://fenway.itsc.austin.ibm.com:9430/cdm/servlet/LICServlet?default.port=9433&default.server=fenway.itsc.austin.ibm.com&graph=applicationinfrastructure&guid={guid}`

3. The TADDM Application Infrastructure view is shown in Figure 11-14 on page 319.

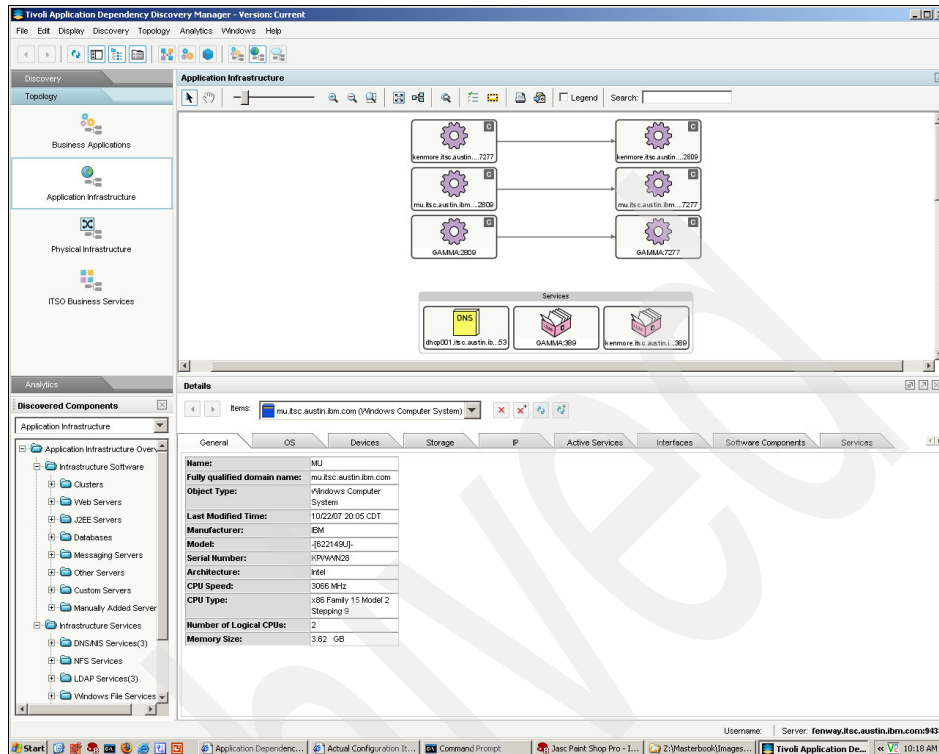


Figure 11-14 TADDM Application Infrastructure view

Scenario 2: Launch from Actual CI into TADDM Physical view

Do the following:

1. Start the Launch in Context application from the Start Center by selecting **Go To** → **System Configuration** → **IT Infrastructure** → **Configuration Items** or **Actual** → **Select the designated Configuration Items** → **Select the CI from which to launch from** → **Select Actions** → **View Actual CI Topology** → **Physical**. The CCMD V7.1 view “Physical Topology” is shown in Figure 11-15 on page 320.

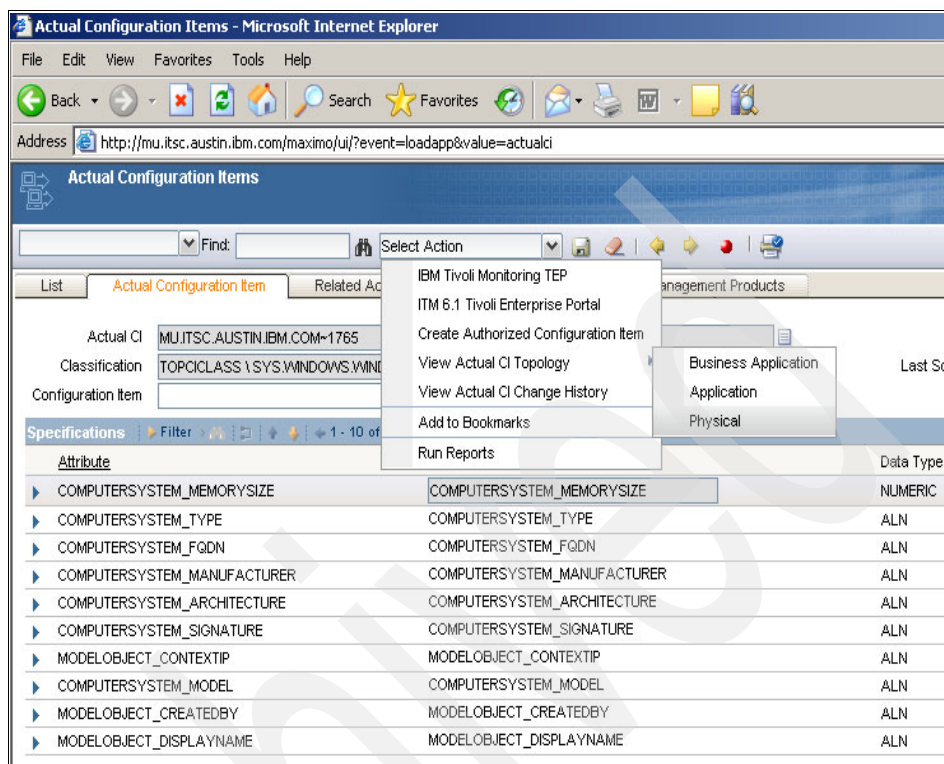


Figure 11-15 CCMDB Physical Topology view

2. The Launch in Context Console URL to enter is:

`http://boston.itsc.austin.ibm.com:9430/cdm/servlet/LICServlet?default.port=9433&default.server=boston.itsc.austin.ibm.com&console=web&graph=physicalinfrastructure&guid={guid}`

3. The TADDM Physical Infrastructure view is shown in Figure 11-16 on page 321.

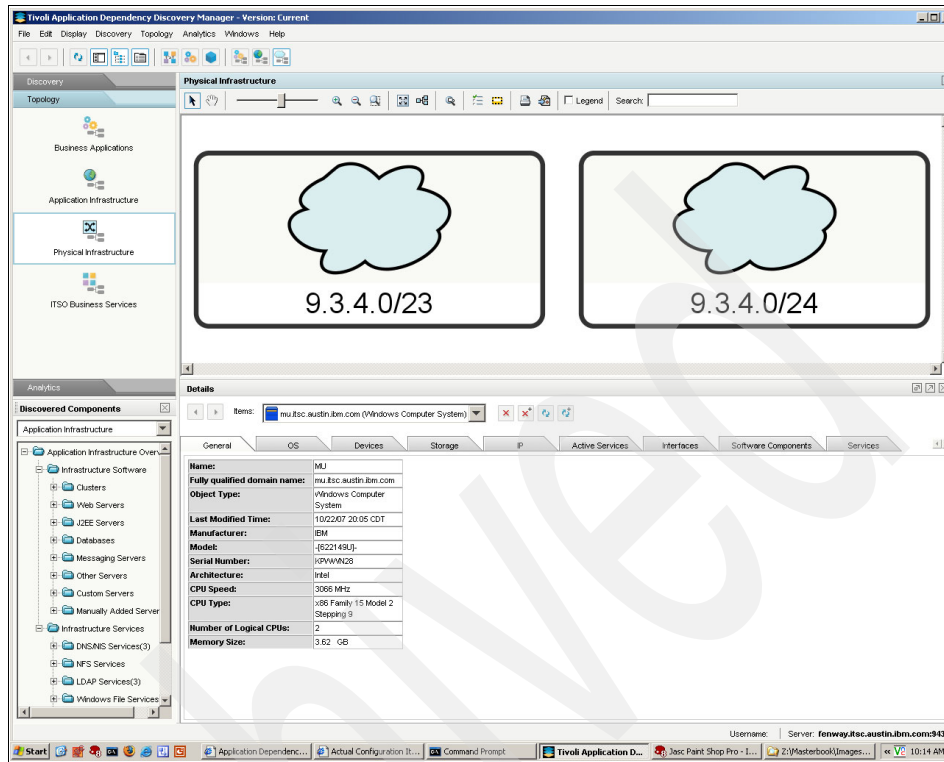


Figure 11-16 Physical Infrastructure view

11.3 Adding a new Launch in Context into CCMDB V7.1

This section describes step by step how a new launch entry is created for an external application, in this case, the Tivoli Enterprise Portal of IBM Tivoli Monitoring V6.1. It is implemented as a Select Action in the Actual Configuration Items application.

To create a new launch in context entry for users, the following steps are required:

1. Define a Launch Entry point.
2. Associate the Launch Entry with a Sigoption.
3. Modify the Select Action menu.
4. Allow access for users or groups by defining security.
5. Verify the new launch entry.

Attention: Before you begin, ensure that the application itself can be launched into by using a URL in a Web browser.

Note: In our example, there is no single sign-on setup for ITM V6.1, so the user is prompted to log in.

11.3.1 Define a launch entry point

The first step in adding a new launch in context application is to create a new launch entry point and define the appropriate URL specifications and parameters. See 11.2, “Launch entry URL specifications and parameters” on page 312. In this case, these entries specify the standard login for the Tivoli Enterprise Portal (TEP) of IBM Tivoli Monitoring V6.1. Select **Go To → System Configuration → Platform Configuration → Launch in Context** and create a new Launch in Context entry. Provide the information shown in Figure 11-17 and save it.

Launch in Context

Find: Select Action

List Launch Entry

Launch Entry Name: ITM 6.1 TEP

Console URL: http://nice.itsc.austin.ibm.com:1920//cnp/rdh/ik

Target Browser Window: _use_new

OMP Product Name: IBM Tivoli Monitoring

OMP Version:

IBM Tivoli Monitoring V

Launch Contexts for Filter

Resource Object Name	Resource Classification
...No rows	

Figure 11-17 Define the launch entry point

11.3.2 Associate the launch entry with a Signature option

Once the URL specification and parameter setting is accomplished, the launch entry must be associated with an application so that it becomes visible in the Maximo Actual CI application. This association is created by a Signature option (sigoption).

Do the following:

1. Select **Go To** → **System Configuration** → **Platform Configuration** → **Application Designer**, as shown in Figure 11-18.

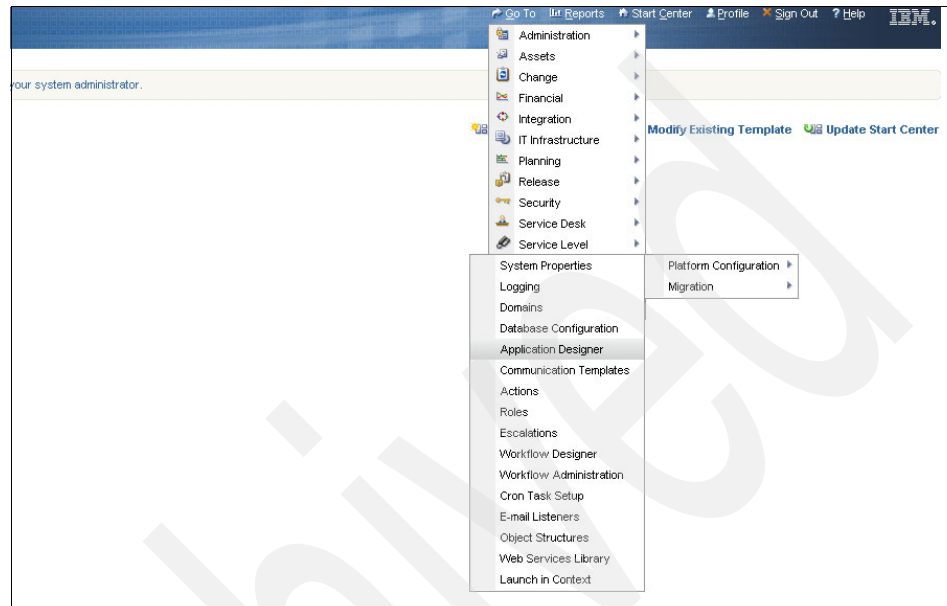


Figure 11-18 Selecting Application Designer

2. In the List tab, select the application where the Launch Entry should be implemented. In this case, the user should be able to launch the TEP from within the Actual Configuration Items, as shown in Figure 11-19.

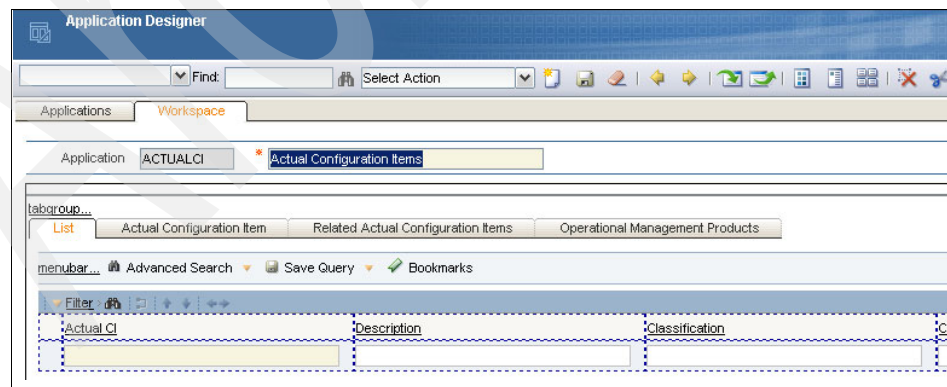


Figure 11-19 Choosing Actual Configuration Items application

3. Select **Go To** → **Select Action** → **Add/Modify Signature Options**, as shown in Figure 11-20.

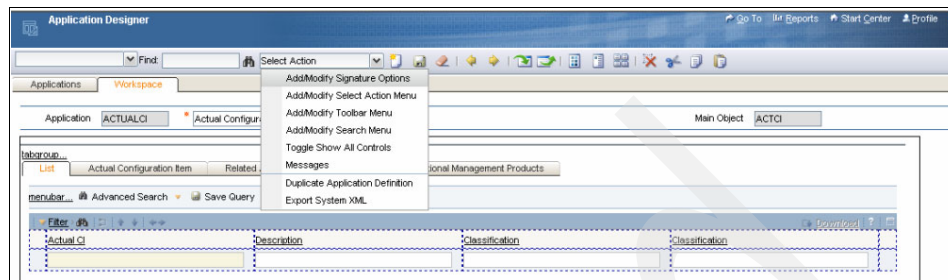


Figure 11-20 Add/Modify Signature options

4. Click **New Row** in the Add/Modify Signature Options, as shown in Figure 11-21 on page 325.

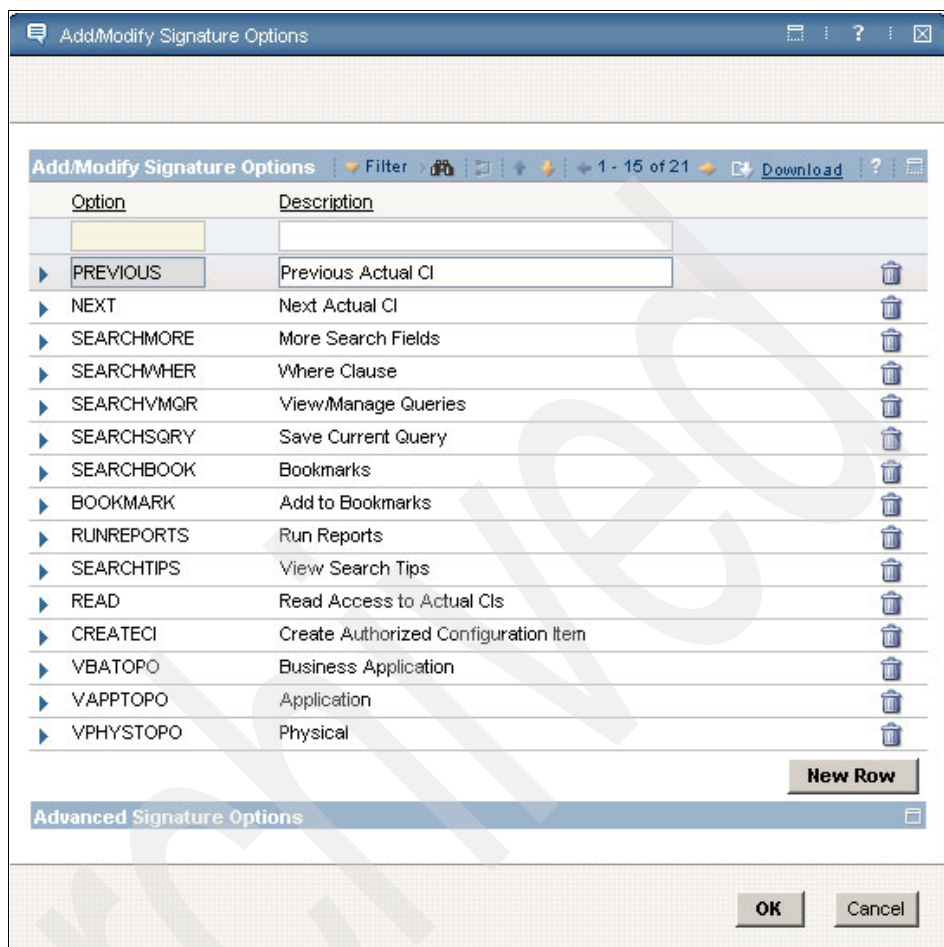


Figure 11-21 Add/Modify Signature options window

5. Provide an option (for example, ITMLE1) and a meaningful description, as shown in Figure 11-22 on page 326. This description will be the entry shown in the Select Action menu.

Add/Modify Signature Options

Filter 2 - 16 of 22 Download

Option	Description
NEXT	Next Actual CI
SEARCHMORE	More Search Fields
SEARCHWHERE	Where Clause
SEARCHVIMQR	View/Manage Queries
SEARCHSGRY	Save Current Query
SEARCHBOOK	Bookmarks
BOOKMARK	Add to Bookmarks
RUNREPORTS	Run Reports
SEARCHTIPS	View Search Tips
READ	Read Access to Actual CIs
CREATECI	Create Authorized Configuration Item
VBATOPO	Business Application
VAPPTOPO	Application
VPHYSTOPO	Physical
TEPLE1	ITM 6.1 Tivoli Enterprise Portal

Option:

Description:

Option Also Grants:

Option Also Revokes:

Prerequisite:

E-signature Enabled? ☐

Visible? ☒

Advanced Signature Options

☐ None
☐ Warning appears when this action is selected from List page where multiple records are shown and no particular record is selected
☐ This is an action that must be invoked by user in the UI
☒ Associate to launch entry to enable launch in context

Launch Entry Name:

OK Cancel

Figure 11-22 Specifying the new option

- Expand the Advanced Signature Option section to insert the appropriate launch entry point. Click the magnifier and select the launch entry name you created, as shown in Figure 11-23 on page 327.

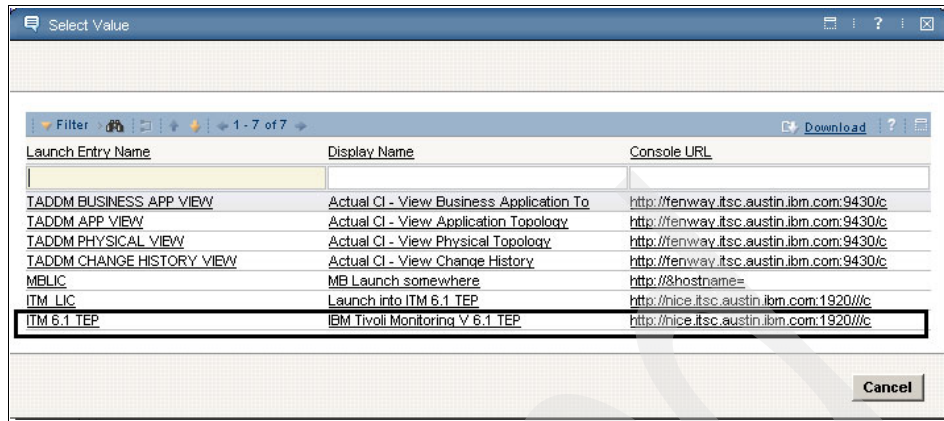


Figure 11-23 Selecting the new option

7. Double click the entry and it will be inserted as the “Launch entry name”, as shown in Figure 11-24 on page 328.
8. Check the option **Associate to launch entry to enable launch in context**.
9. Click **OK**.
10. Save the changes.

Add/Modify Signature Options

Filter: 2 - 16 of 22

Option	Description
NEXT	Next Actual CI
SEARCHMORE	More Search Fields
SEARCHWHERE	Where Clause
SEARCHVMGR	View/Manage Queries
SEARCHSQRY	Save Current Query
SEARCHBOOK	Bookmarks
BOOKMARK	Add to Bookmarks
RUNREPORTS	Run Reports
SEARCHTIPS	View Search Tips
READ	Read Access to Actual CIs
CREATECI	Create Authorized Configuration Item
VBATOPO	Business Application
VAPPTOPO	Application
VPHYSTOPO	Physical
TEPLE1	ITM 6.1 Tivoli Enterprise Portal

Option Details:

Option:
Description:
Option Also Grants:
Option Also Revokes:
Prerequisite:

E-signature Enabled? ☐
Visible? ☒

Advanced Signature Options

☐ None
☐ Warning appears when this action is selected from List page where multiple records are shown and no particular record is selected
☐ This is an action that must be invoked by user in the UI
☒ Associate to launch entry to enable launch in context

Launch Entry Name:

OK Cancel

Figure 11-24 Associating the launch entry

11.3.3 Modify the Select Action menu

In order to add the launch in context for the Tivoli Enterprise Portal, the newly created Sigoption has to be added to the Actual CI application. In this case, the entry is added to the Select Action menu.

Do the following

1. Select **Go To** → **System Configuration** → **Platform Configuration** → **Application Designer**.
2. Select **Select Action** → **Add/Modify Select Action Menu**.
3. Add a row.
4. Provide the entries shown in Figure 11-25 on page 330. The Key Value is the name in the Sigoption entry. The position number is the relative item position of the Select Action menu.

Tip: Whenever there is a magnifier, the entries can be selected from that list.

5. Click **OK**.
6. Save the changes.

Add/Modify Select Action Menu

Filter > 2 - 11 of 13 Download

Key Value	Header Description
BOOKMARK	
CREATECI	
	View Actual CI Topology
VBATOPO	
VAPPTOPO	
VPHYSTOPO	
CHGHIST	
TEPLE1	ITM V 6.1 Tivoli Enterprise Portal

Element Details for TEPLE1:

- Element Type:
- Key Value:
- Header Description:
- Position:
- Subposition:
- URL:
- Image:
- Visible?: ☒
- Access Key:
- Tabs:

New Row **OK** **Cancel**

Figure 11-25 Modifying the Select Action menu

11.3.4 Allow access for everybody by defining security

CCMDB V7.1 provides the opportunity to restrict access for users or groups for certain applications, views, or actions. In this case, we want everybody to be able to launch the Tivoli Enterprise Portal from the Actual CI application. In order to do that, the security must be adjusted.

Do the following:

1. Select **Go To** → **Security** → **Security Groups** → **EVERYONE** → **Applications**, as shown in Figure 11-26.

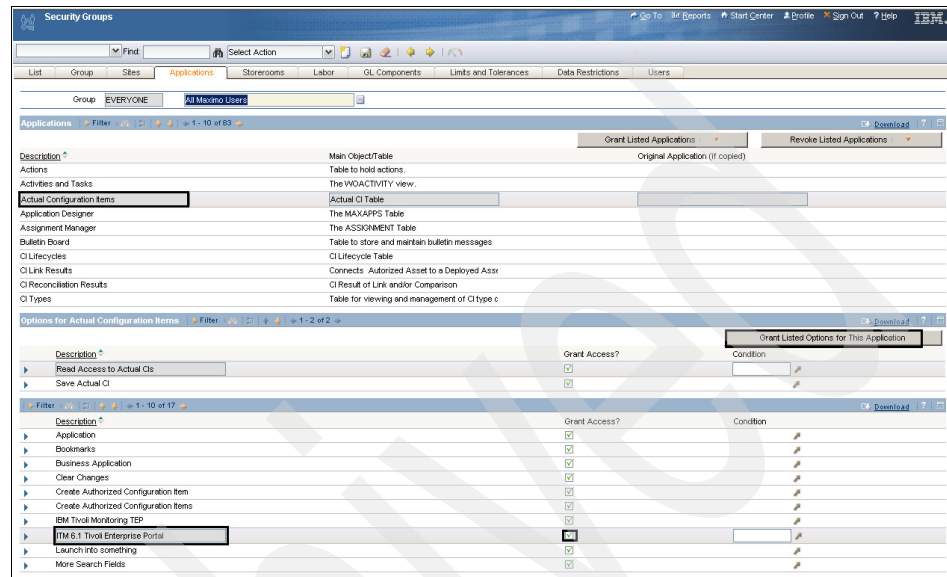


Figure 11-26 Security groups

2. Switch to the Applications tab.
3. In the Applications tab, select the **Actual Configuration Items** application, which is the place where the new launch entry should be.
4. In the “Options for the Actual Configuration Items” section, select the Launch Entry name (refer to Figure 11-20 on page 324).
5. Check the option **Grant Access?**.
6. Click **Grant Listed Options for This Application**.
7. Click **OK**.
8. Save the changes.

Click the **OK** button shown in Figure 11-27 to approve the changes.

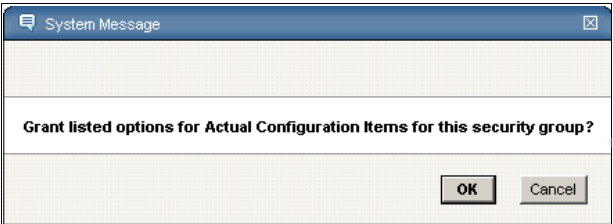


Figure 11-27 Approve the changes

11.3.5 Verify the new launch entry

The new Launch in Context entry will be visible after a new login. Sign out and sign in to the application. Select **Go To** → **IT Infrastructure** → **Actual Configuration Items** → **Register “Actual Configuration Items”** → “**Select Action**” drop-down menu, as shown in Figure 11-28.

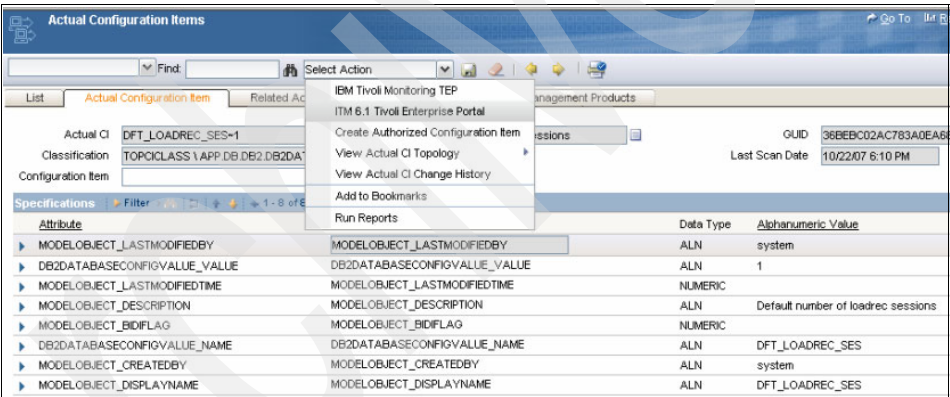


Figure 11-28 New Select Action menu

Launch the ITM 6.1 Tivoli Enterprise Portal. The Java applet download is started in a new browser window and then the user is prompted for the login and the application is launched.

Reporting

CCMDB V7.1 comes integrated with the Eclipse Foundation's Business Intelligence Reporting Tool (BIRT). BIRT is an open source reporting system that integrates with Java/J2EE applications, such as CCMDB V7.1, to produce reports. BIRT utilizes XML report definitions to generate reports in PDF or HTML Output. BIRT uses the data from CCMDB V7.1 and manages and displays it to users in a way that they can immediately take action if necessary. That action may involve drilling down into reports for a specific problem issue, or an analysis of the data for the cost for regulatory purposes.

Note: Refer to the ISM Toolbox for a complete list of out-of-the-box reports IBM ships to you with this product.

12.1 BIRT architecture

The BIRT architecture is made up of various components, as shown in Figure 12-1.

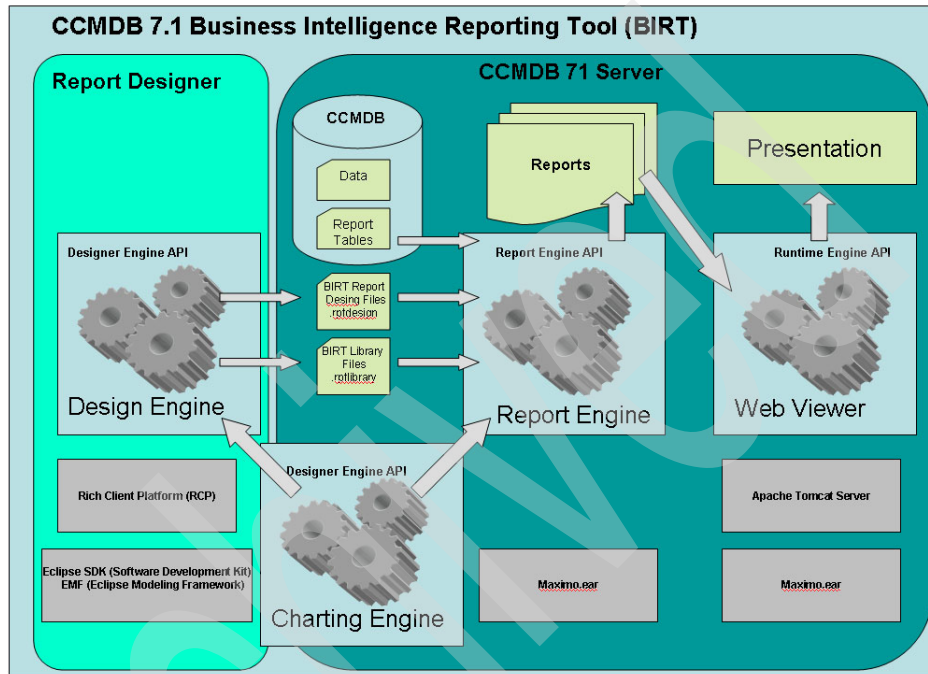


Figure 12-1 BIRT architecture

► BIRT Report Designer

The BIRT Report Designer is a visual tool provided by Eclipse, as a Rich Client Platform (RCP) application. The RCP is available as a set of plug-ins that are installed on an existing Eclipse server or as an all in one package including Eclipse. This tool makes it easier for developers to design reports. It has to be downloaded and installed separately. It is not part of the CCMDB V7.1 installation.

► Design Engine

The Design Engine is responsible for creating and modifying report designs. The created report design is stored in .rptdesign and .rptlibrary files. The Design Engine API (DEAPI) performs a wide range of low-level tasks:

- Reads and writes design files.
- Maintains the command history for undo/redo.

- Provides a rich semantic representation of the report design.
- Provides meta-data about the Report Object Model.
- Performs property value validation.
- Notifies the application when the model changes.

BIRT Report Design files are XML files and have the extension `.rptdesign`. BIRT Reports can contain single or multiple files. The files are categorized as either library files or resource files.

BIRT library files are also XML files and have the extension `.rptlibrary`. BIRT library files can contain code that is used multiple times for items such as font type, size, page numbers, and time stamp.

Resource Files contain items such as images or external files. Resource files can be used by either report design files or library files.

The XML of the BIRT Report details which library files and resource files the report requires. In the XML file, a flag indicates whether the file is a library file. Without these files, the BIRT report does not execute.

For further information, refer to the *Change and Configuration Management Database: Report Developer Guide*.

► Charting Engine

The Charting Engine is used to design and generate charts that are used by the Design Engine and Report Engine in order to deliver Charts. It contains chart models and factory classes. The Charting Engine API (CEAPI) allows a developer to add charting capabilities to their application.

► Report Engine

The Report Engine API (REAPI) generates and renders reports from a report design file. The engine supports the following operations:

- Discover the set of parameters defined for a report.
- Get the default values for parameters.
- Run a report to produce HTML/Paginated HTML or PDF output.
- Fetch an image or chart for a report.
- Export CSV.
- Retrieve TOCs, bookmarks, and so on

The BIRT Report Engine, as part of CCMDB V7.1, stores its data in the CCMDB directory, as shown in Figure 12-2.

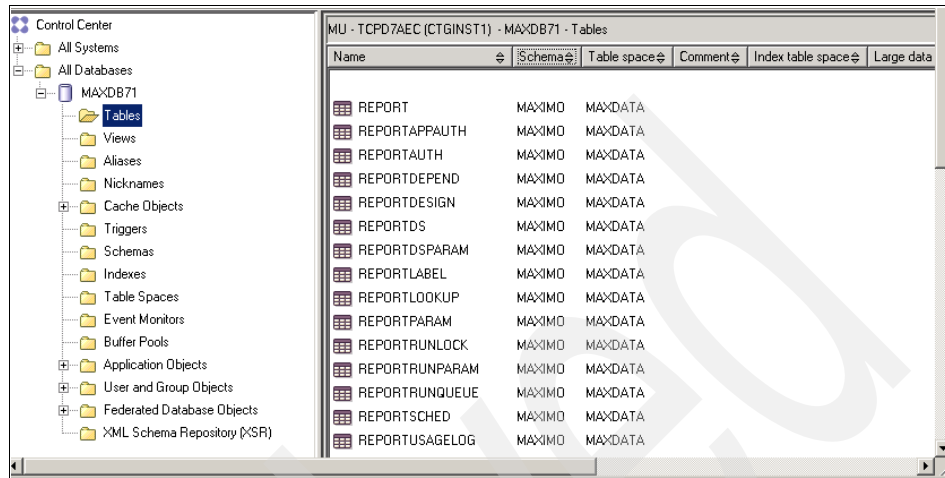


Figure 12-2 Report Engine directory structure

► Web Viewer

The Web Viewer is optimized for use within Eclipse for the preview operation. The Viewer performs three distinct operations, of which one is internal and not visible to your application:

- Creates a frameset based viewer that interacts with the servlet using AJAX to support more features.
- Given a set of report parameter values, runs a report and returns the output as either HTML or PDF.
- Retrieves an embedded image, or an image of a chart within the report (internal operation).

12.2 BIRT reporting process

There are three major tasks that generate BIRT reports:

- Developing: BIRT Report Designer (installed separately)
- Administering: CCMDB V7.1 Report Administration Application
- Running: CCMDB V7.1 Viewer or Run Report Action

12.2.1 BIRT report development

Before starting report development, the BIRT Report Designer application has to be installed. The following software is required:

- ▶ Eclipse
- ▶ BIRT Report Designer
- ▶ Graphics-Editor Framework (GEF)
- ▶ Eclipse Modeling Framework (EMF)
- ▶ Java Software Developer Kit (SDK)
- ▶ itext-1.3.jar

Download the BIRT Report Designer component and all its prerequisites at:

<http://download.eclipse.org/birt/downloads/>

Here you can also find detailed information about how to install the application.

Report development is done in three parts:

- ▶ Create a Report Design File .rptdesign (maximo\reports\birt\reports).
- ▶ Create a Property File .rptlibrary (maximo\reports\birt\libraries).
- ▶ Create / Update reports.xml file (maximo\reports\birt\reports).

With the BIRT Design Engine (Figure 12-3), you can add a rich variety of reports to your application.

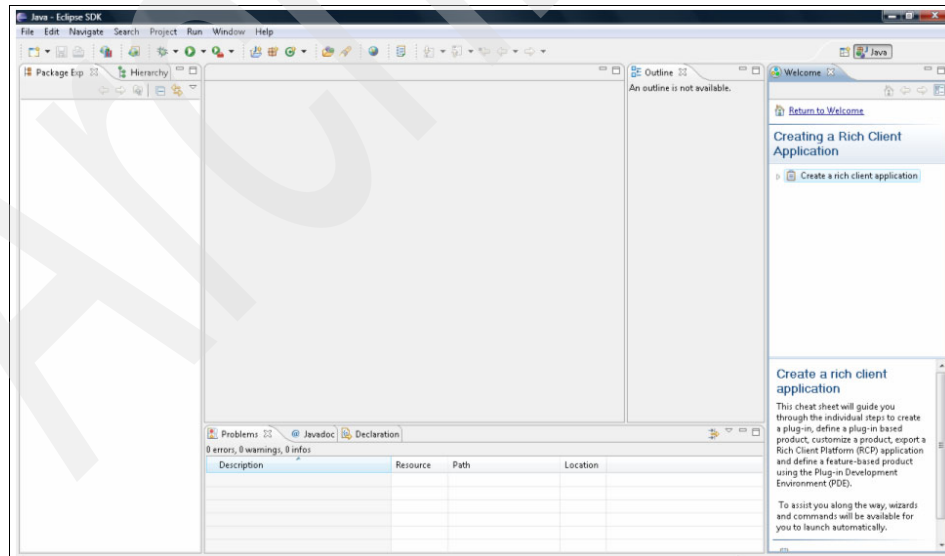


Figure 12-3 BIRT Design Engine

Here are the reports you can add:

- ▶ **Lists:** The simplest reports are lists of data. As the lists get longer, you can group related data together. If your data is numeric, you can easily add totals, averages, and other summaries.
- ▶ **Charts:** Numeric data is much easier to understand when presented as a chart. BIRT provides pie, line, and bar charts, and many more. BIRT charts can be rendered in SVG and support events to allow user interaction.
- ▶ **Crosstabs:** Crosstabs (also called a cross-tabulation or matrix) shows data in two dimensions.
- ▶ **Letters and Documents:** Notices, form letters, and other textual documents can be created with BIRT. Documents can include text, formatting, lists, charts, and more.
- ▶ **Compound Reports:** Many reports need to combine the above into a single document.

For further information about how to develop new reports, refer to the following resources:

- ▶ <http://www.eclipse.org>
- ▶ *Change and Configuration Management Database: Report Developer Guide*

12.2.2 BIRT Report Administration

The BIRT Report Administration is integrated into CCMDB V7.1. As the Report Administrator, you can specify the following for users:

- ▶ The availability of reports and how they open, run, and print
- ▶ The appearance of report titles and headings
- ▶ Report security settings

The following figures provide a short introduction to the Report Administration application. The look and feel is like all other applications launched from the Start Center. There are two ways to open the Report Administration application. The first way is for the initial administration, while the second way can be used when the reports are already defined:

- ▶ Start the Report Administration application from the Start Center by selecting **Go To → Administration → Reporting → Report Administration**, as shown in Figure 12-4 on page 339.

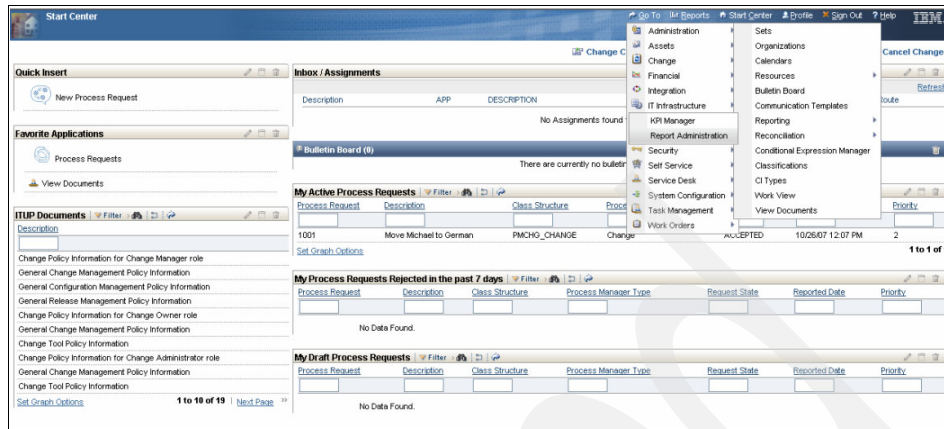


Figure 12-4 Accessing the Report Administration application through Go To

- Start the Report Administration application from the Start Center by selecting **Reports** → **Administration** → **Reporting** → **Report Administration**, as shown in Figure 12-5.

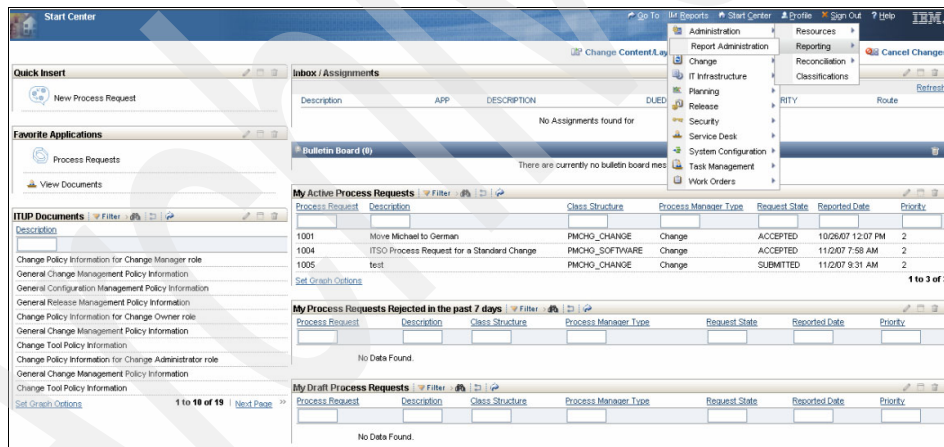


Figure 12-5 Accessing the Report Administration application through the Reports option

The Report Administration has following tabs:

- ▶ List: List of all existing reports (Figure 12-6)
- ▶ Reports: Details of a selected report (Figure 12-7)
- ▶ Security: Set and view Report and Application Security (Figure 12-8 on page 341)
- ▶ Labels: Change Report Labels and settings, like Label Key and Label Value (Figure 12-9 on page 341)

The screenshot shows the 'List' tab of the Report Administration interface. It displays a table with the following columns: Report File Name, Description, Application, and Report Folder. The table lists various reports such as 'actualci_detail.rptdesign', 'actualci_history.rptdesign', 'ci_template.rptdesign', etc. A 'Generate Request Pages' button is located at the bottom right.

Report File Name	Description	Application	Report Folder
actualci_detail.rptdesign	Actual CI Details	ACTUALCI	ACTUALCI
actualci_history.rptdesign	Actual CI History	ACTUALCI	ACTUALCI
ci_template.rptdesign	CI Template	ACTUALCI	ACTUALCI
class_hierarchy.rptdesign	Classification Hierarchy	ASSETCAT	ASSETCAT
forwardedchanges.rptdesign	Forward Schedule of Changes	CHANGE	CHANGE
it_forwardedchanges.rptdesign	Forward schedule of IT changes	CHANGE	CHANGE
it_change_process_history.rptdesign	IT change History grouped by progress	CHANGE	CHANGE
it_change_final_state.rptdesign	Percentage of IT changes in final state	CHANGE	CHANGE
it_change_by_priority.rptdesign	Percentage of IT changes in each priority	CHANGE	CHANGE
it_change_by_classification.rptdesign	IT changes grouped by classification	CHANGE	CHANGE
it_change_avg_complete_by_type.rptdesign	Average time to complete IT changes, by type	CHANGE	CHANGE
it_change_process_request_open_by_type.rptdesign	Open Change Process Requests, by type	CHANGE	CHANGE
it_change_process_request_new.rptdesign	Open Change Process Requests in new state	CHANGE	CHANGE
it_change_target_ci.rptdesign	IT changes associated with a CI	CHANGE	CHANGE
it_change_impacted_ci.rptdesign	IT changes impacting CI	CHANGE	CHANGE
it_active_change_process.rptdesign	Active changes, grouped by progress	CHANGE	CHANGE
it_process_request_final_states.rptdesign	Open Change Process Requests, by state	CHANGE	CHANGE
it_change_window_conflict.rptdesign	Implementation tasks with change window conflict	CHANGE	CHANGE
ci_attribute_history.rptdesign	CI Attribute History	CI	CI

Figure 12-6 Lists of reports

The screenshot shows the 'Reports' tab of the Report Administration interface. It displays the details for the report 'actualci_detail.rptdesign'. The 'Report Type' is 'RPT'. The 'Application' is 'ACTUALCI' and the 'Report Folder' is 'ACTUALCI'. The 'Settings' section includes options for 'Limit Records?' (checked), 'Max Record Limit' (50), 'Use Where Clause?' (unchecked), 'No Request Page?' (unchecked), 'Priority' (empty), 'Browser View?' (unchecked), 'Direct Print?' (checked), 'Direct Print with Attachments?' (checked), 'Toolbar Sequence' (1), 'Browser View Location' (NONE), 'Direct Print Location' (ALL), and 'Direct Print with Attachments Location' (NONE). There are buttons for 'Generate Request Page' and 'Preview'. The 'Parameters' section at the bottom shows a table with columns: Parameter Name, Attribute Name, Sequence, Display Name, and Required?.

Parameter Name	Attribute Name	Sequence	Display Name	Required?
...No rows to display...				

Figure 12-7 Report details

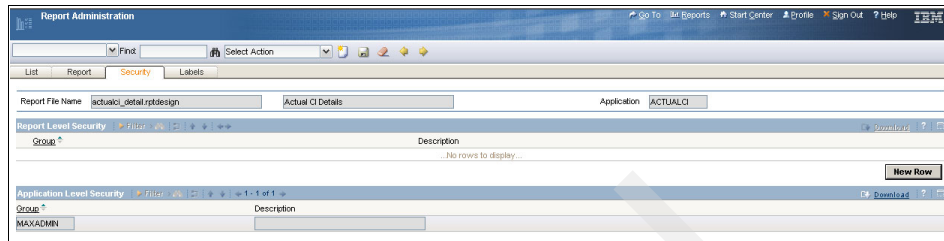


Figure 12-8 Report security

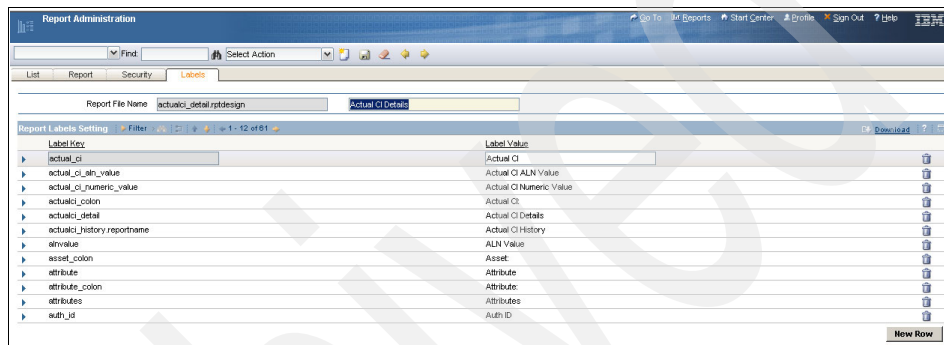


Figure 12-9 report labels

For the List tab, the Select Action only offers general administration tasks like:

- ▶ View Scheduled Reports
- ▶ Set Application Security
- ▶ View Group Security
- ▶ View Library Files
- ▶ Run Reports

Once a specific report is selected, additional Select Action items are available:

- ▶ Import Report
- ▶ Import Library File
- ▶ View Report Dependencies
- ▶ Add to Bookmark
- ▶ Duplicate Report
- ▶ Delete Report

The following sections covers the more complex Select Actions in more detail.

View Scheduled Reports

The View Scheduled Reports dialog box lets you manage scheduled report jobs. You can view the report load and delete scheduled report jobs as necessary, as shown in Figure 12-10.

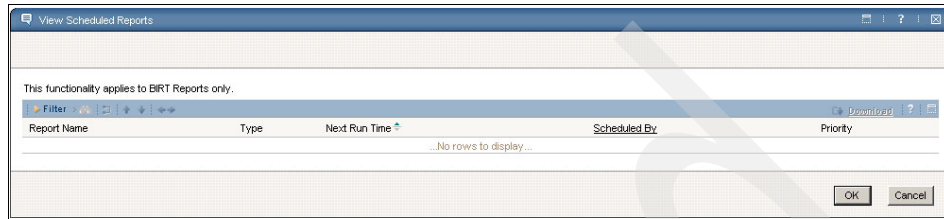


Figure 12-10 View Scheduled Reports

Set Application Security

The Application Security settings let you set group security for all reports in a selected application. The MAXADMIN group has access to all “out-of-the-box” reports. You must set up group or report access for each individual application for new or customized reports, as shown in Figure 12-11.

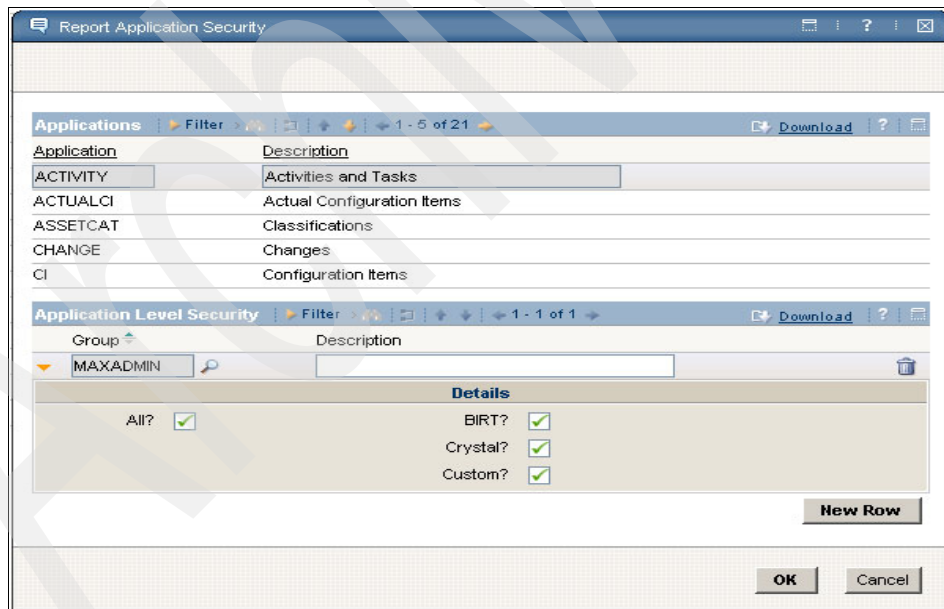


Figure 12-11 Report Application Security

The two different ways of setting security for all reports for an application and for individual reports are shown in Figure 12-12 on page 343 and Figure 12-13 on page 344.

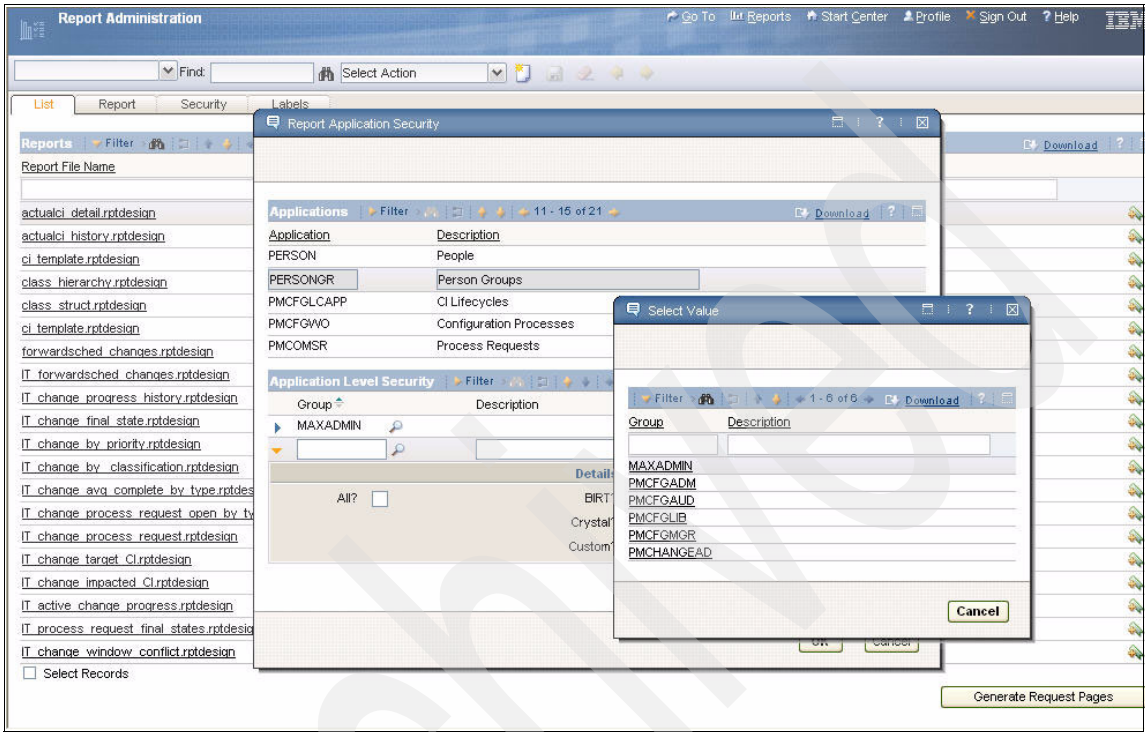


Figure 12-12 Application security settings

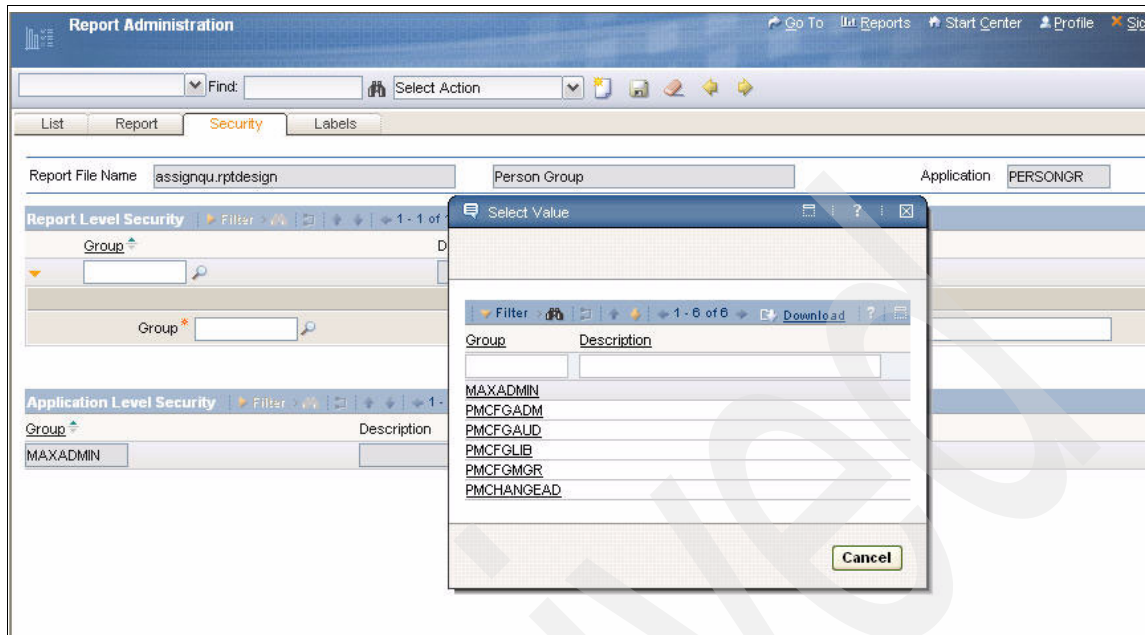


Figure 12-13 Individual security settings

View Group Security

You can manage report security for a group through the Report Administration application. The MAXADMIN group has access to registered out-of-the-box reports. You must set up application access for other groups individually, as shown in Figure 12-14 on page 345.

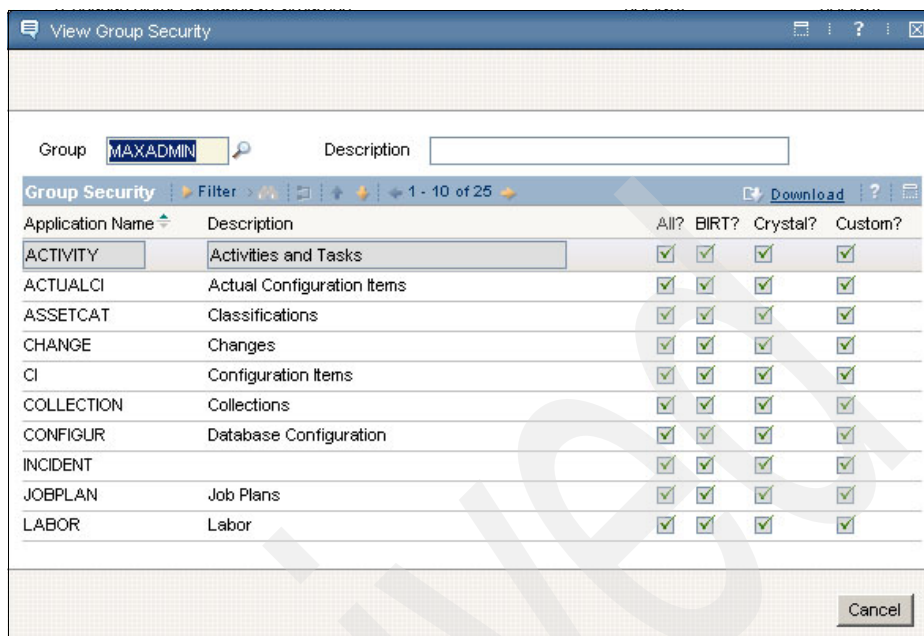


Figure 12-14 View Group Security

View Library Files

Use the View Library File action to determine if the libraries you need for a report already exist in the database, as shown in Figure 12-15.

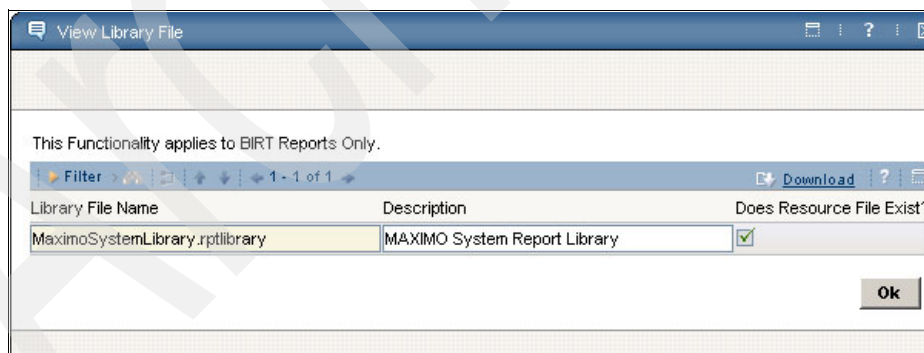


Figure 12-15 View Library Files

Import Report

You perform this action to add a new report to your database or bring an updated version of an existing report into your database, as shown in Figure 12-16.

Before you import the report design file, you must import any associated library files.

This action is available from only the Report tab for the following reasons:

- ▶ If the report is new, you use the Report tab to add the report to the Report Administration application and then import the report to the database.
- ▶ If the report already exists, you import the report from the Report tab to be certain you choose a correct combination of Report Design file and Application name.

To add multiple design files, use the `importreport.cmd` command.

Import Report

This Functionality applies to BIRT Reports Only.

Report Design File

Report Description *

Report Design File *

Report Resource File

Figure 12-16 Import Report

Import Library File

Library files contain components that you can use in one or more report designs to provide consistent behavior and performance. Library files are useful when many reports use the same component multiple times. Use the Report Administration application to import a report library file into the database, as shown in Figure 12-17 on page 347. You import a library file before you import the corresponding reports. In the Report Resource File field, enter the location of

any resource files. Resource files contain items such as images or external files. This field is optional.

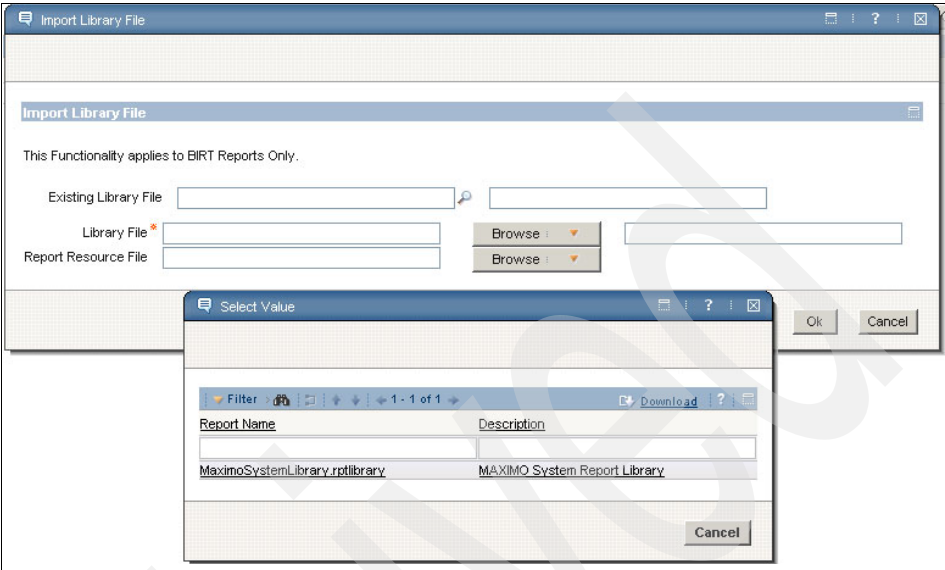


Figure 12-17 Import Resources

View Report Dependencies

Use the View Report Dependencies action to view the libraries that a report design file requires. For each report library, you also can view dependent library files and check for any resource files. You view report dependencies for BIRT reports, as shown in Figure 12-18.

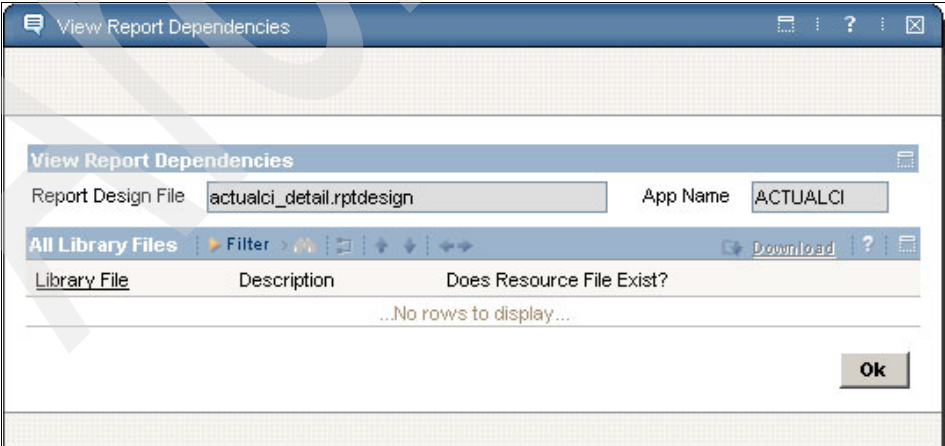


Figure 12-18 View Report Dependencies

Duplicate Report

Among the reasons for duplicating a report are:

- ▶ You create a cloned application. You duplicate the report and save the duplicate to the cloned application.
- ▶ You want to register (add) a report to multiple related applications. You duplicate the report and save it to the related applications.

Delete Report

When you delete a report, you remove the report and its associated files from the database. You also remove any scheduled activities for the report.

12.2.3 BIRT Configure Reports

In the Report tab, there are several options to configure a report. Those with an (*) Asterisk are required parameters (see Figure 12-19).

The screenshot shows the 'Report Administration' application window. The 'Report' tab is selected. The 'Report File Name' is 'actuals_detail.rptdesign' and the 'Report Type' is 'BIRT'. The 'Application' is 'ACTUALCI' and the 'Report Folder' is 'ACTUALCI'. The 'Settings' section includes: 'Limit Records?' (checked), 'Max Record Limit*' (50), 'Use Where Clause?' (unchecked), 'No Request Page?' (unchecked), 'Priority' (empty), 'Browser View?' (unchecked), 'Direct Print?' (checked), 'Direct Print with Attachments?' (unchecked), 'Browser View Location*' (NONE), 'Direct Print Location*' (ALL), and 'Direct Print with Attachments Location*' (NONE). The 'Parameters' table has columns: 'Parameter Name', 'Attribute Name', 'Sequence*', 'Display Name', and 'Required?'. The table is currently empty with the message 'No rows to display'. Buttons for 'Generate Request Page' and 'Preview' are visible.

Figure 12-19 Report configuration

The fields are described as follows:

Report Type

BIRT, Crystal, or Custom. By determining the report type and settings, you register that report in the CCMDB V7.1 database.

Limit Records

The action limits the number of records against which an user can run a report. It prevents users from executing large queries, which can cause negative performance impacts. Use the Report Administration application to set record restrictions on reports. This feature applies to only reports without parameters. This parameter goes along with the Max Record Limit.

Use Where Clause? Enables Current /Selected plus User Inputted parameters.

No Request Page Disables the Request Page for Database Updates.

Priority The numeric field used in the report queuing process.

Browser View and Browser View Location

The Browser View feature lets you create a shortcut. With the shortcut, the user can click an icon once in the application toolbar to open a report directly in the browser. When Browser View is checked, enter a value other than None in the Browser View Location field. This field determines the application tabs that have an active Browser View icon. The following options are available:

All: The Browser View icon is available on all tabs for the selected application.

List: The Browser View icon is only available on the List tab for the selected application.

Main: The Browser View icon is available on all tabs, except for the List tab.

None: The Browser View icon does not appear in the selected application. None is the default.

Click **Save Report** to apply the changes.

Direct Print and Direct Print Location

The Direct Print feature lets you create a shortcut so a user can click an icon once in the application toolbar to print the report. The configuration is the same as for Browser View Location.

Direct Print with Attachments and Direct Print with Attachments Location

The Direct Print with Attached Documents feature lets you create a shortcut so an user can click an application icon once (and select **Yes** in a message dialog box) to print the report and any associated attached documents. The configuration is the same as for Browser View Location.

Generate Request Page

Click **Generate Request Page** if you have not previously configured the report for Browser View. This option is available for all reports or at individual report level.

Preview Report

You can check for the following items:

The correct parameters, if any, appear to the user on the request page.

The generated report opens with the correct data and format.

The Request Page dialog box opens. The parameters that appear depend on the report that you select.

Enter values in any required fields. Required fields have an orange asterisk (*) next to them.

Figure 12-20 shows these fields.

The screenshot displays the 'Report Administration' window. At the top, there's a navigation bar with 'Go To' and links for 'All Reports', 'Start Center', 'Profile', 'Sign Out', and 'Help'. Below this is a search bar with 'Find:' and a 'Select Action' dropdown. The main content area has tabs for 'List', 'Report' (selected), 'Security', and 'Labels'. Under the 'Report' tab, there are fields for 'Report File Name' (sla.rptdesign), 'Report Type' (BIRT), 'Application' (SLA), and 'Report Folder' (SLA). A 'Settings' section contains various checkboxes and dropdowns: 'Limit Records?' (unchecked), 'Max Record Limit' (input field), 'Use Where Clause?' (unchecked), 'No Request Page?' (unchecked), 'Priority' (input field), 'Browser View?' (checked), 'Direct Print?' (unchecked), 'Direct Print with Attachments?' (unchecked), 'Toolbar Sequence' (1), 'Browser View Location' (ALL), 'Direct Print Location' (NONE), and 'Direct Print with Attachments Location' (NONE). There are 'Generate Request Page' and 'Preview' buttons. Below the settings is a 'Parameters' section with a table header: 'Parameter Name', 'Attribute Name', 'Sequence', 'Display Name', and 'Required?'. The table is currently empty, showing 'No rows to display...'. A 'New Row' button is at the bottom right of the parameters section.

Figure 12-20 Report parameters

12.2.4 BIRT Run Reports

Follow these instructions to run a report. After you run a report, you have the options to print, export data, and toggle the table of contents.

1. Open the Reports dialog box through one of the following methods:
 - From the Reports Menu in the application toolbar, select an application.
 - From the Select Action menu, select **Run Reports**. The On Demand Reports tab opens. The Reports to Run table window lists the available

reports for the application. Click the report that you want to run, as shown in Figure 12-21.

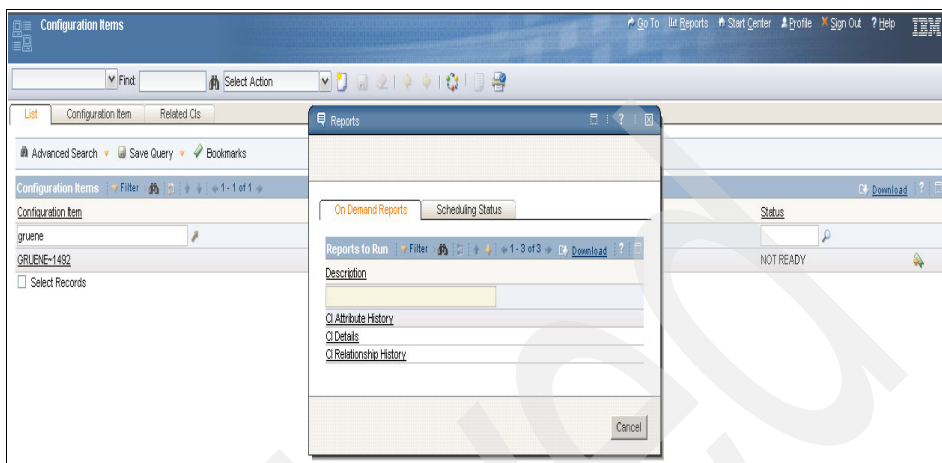


Figure 12-21 Selecting the report to run

- Select the report you want to see, for example, CI Details. Enter any required parameters in the Request Page dialog box, as shown in Figure 12-22 on page 352.

Note: Depending on what application you are in and whether the records are limited, you might not be able to run a report. In order to limit the request, go to a specific record and run the report again.

The screenshot shows a web browser window titled "Request Page". Inside the window, there is a yellow informational box at the top with a question mark icon and the following text: "Fill in the fields in the Parameters section below and select the Submit button to run the report. If no parameters are displayed, the report will execute against the current/selected record set. Optionally, fill out the Schedule and Email Sections to schedule report run times and receive the scheduled report via email." Below this box are three main sections: "Parameters", "Schedule", and "Email". The "Parameters" section is currently empty. The "Schedule" section contains three radio button options: "Immediate" (which is selected), "At this Time" (with an adjacent empty text box and a calendar icon), and "Recurring" (with an adjacent empty text box and a magnifying glass icon). The "Email" section contains three text input fields labeled "To", "Subject", and "Comments". At the bottom right of the form are two buttons: "Submit" and "Cancel".

Figure 12-22 Run request

- Click **Submit** to run the report. The report opens in your browser, as shown in Figure 12-23 on page 353.

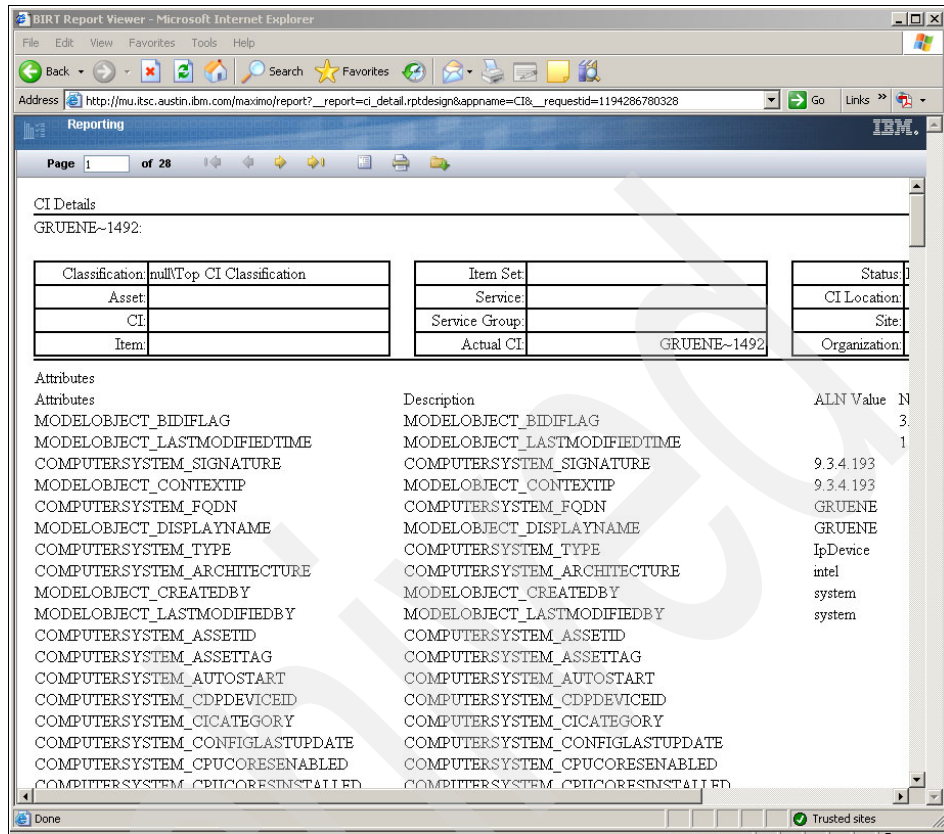


Figure 12-23 Sample report

2. On the Reporting toolbar shown in Figure 12-24 on page 354, perform any of the following actions:
 - Click the **Print Report as PDF** icon to print the report.
 - Click the **Export Data** icon to export the data in .CSV format.
 - Click the **Toggle table of contents** icon to see the table of contents for your report. The report you select determines the table of contents.



Figure 12-24 Available actions on report toolbar

To improve performance and to reduce load on the database server during working hours, it is possible to schedule report runs. Scheduled reports are then e-mailed when completed (see Figure 12-25). The e-mail can be sent to a single user or a group, including subject and comments. The report is e-mailed in PDF format.

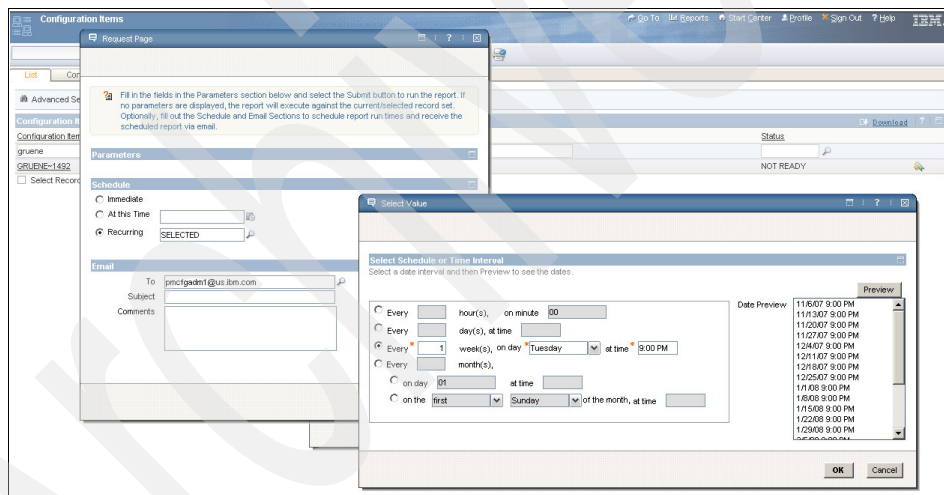


Figure 12-25 E-mailing reports

12.2.5 BIRT Report examples

CCMDB V7.1 comes with several out-of-the-box reports. Here are some report examples.

Job Plan List

Job Plan list, where you can drill down into further reports, as shown in Figure 12-26 on page 355.

Job Plan	Description	Duration	Priority	Organization
APPLREQ	Application Request	0:00	3	
AUDITCI	Audit CIs	0:00	0	
AUDITCI2	Audit CIs Phase 2	0:00	0	
AUTOTRANS	Automatic Transmission Service	3:00	2	EAGLENA
CHG-F1	Assessment	0:00	0	
CHG-F2	Approval	0:00	0	
CHG-F3	Schedule	0:00	0	
CHG-F4	Implement the Change	0:00	0	
CHG-F5	Post Implementation Review	0:00	0	
CHG-F6	Business Assessment	0:00	0	
CHG-F7	Technical Assessment	0:00	0	
CHG-F8	J2EE Implementation	0:00	0	
CHG-P-F1	Standard Change JobPlan	0:00	0	
CHG-P-F2	Pre-Approved JobPlan	0:00	0	
CHG-P-F3	J2EE Application JobPlan	0:00	0	
ENGSESV	ENGINE SERVICE - SP343 DECK	3:00	1	EAGLENA
INS-PC	PC Inspection	1:00	3	EAGLENA
INS-SUBSYS	General Inspection of Major Subsystems	3:00	9	EAGLENA
INS1002	Fire Extinguisher Inspection	0:00	0	EAGLENA
INS11200	HVAC System Inspection	2:30	9	EAGLENA
INS11450	Reciprocating Compressor Inspection	3:00	9	EAGLENA
INS11460	Burner Gas Fired Inspection	3:00	9	EAGLENA
INS11470	Electric Cart/Forklift Inspection	1:00	7	EAGLENA
INS11480	Overhead Crane Inspection	1:00	7	EAGLENA
INS11490	PKG Machine, Elevator & Drailpan Inspection	3:00	9	EAGLENA
INS11500	Inspect and Adjust Westinghouse Overcurrent Relay	1:00	8	EAGLENA

Figure 12-26 Job plan list report

Job Detail Plan

See Figure 12-27 for an example of a Job Detail Plan report.

Task ID	Description	Duration	Nested Job Plan	Meter Name	Owner
10	Financial Approval	0:00			
20	Business Approval	0:00			
30	Security Approval	0:00			
40	Target Approval	0:00			
50	Impacted CI Approval	0:00			
60	Update Change Progress to Approved	0:00			

Figure 12-27 Job Detail Plan report

Service Level Exceptions

Figure 12-28 is an example of a Service Level Exceptions report.

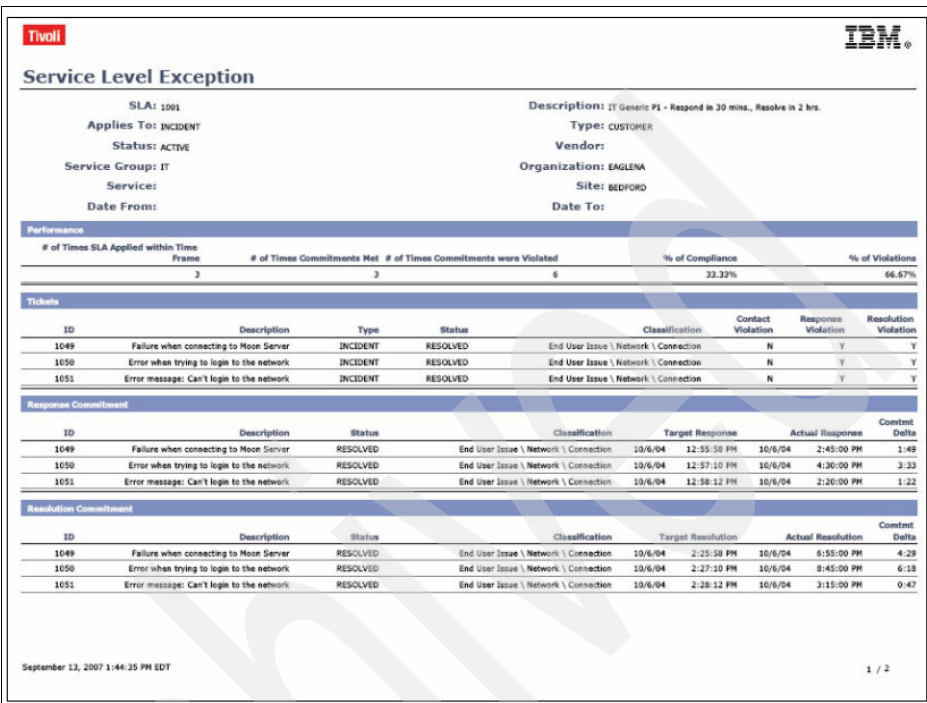


Figure 12-28 Service Level exception report

Release by Classification

Figure 12-29 on page 357 shows a Releases by Classification report.

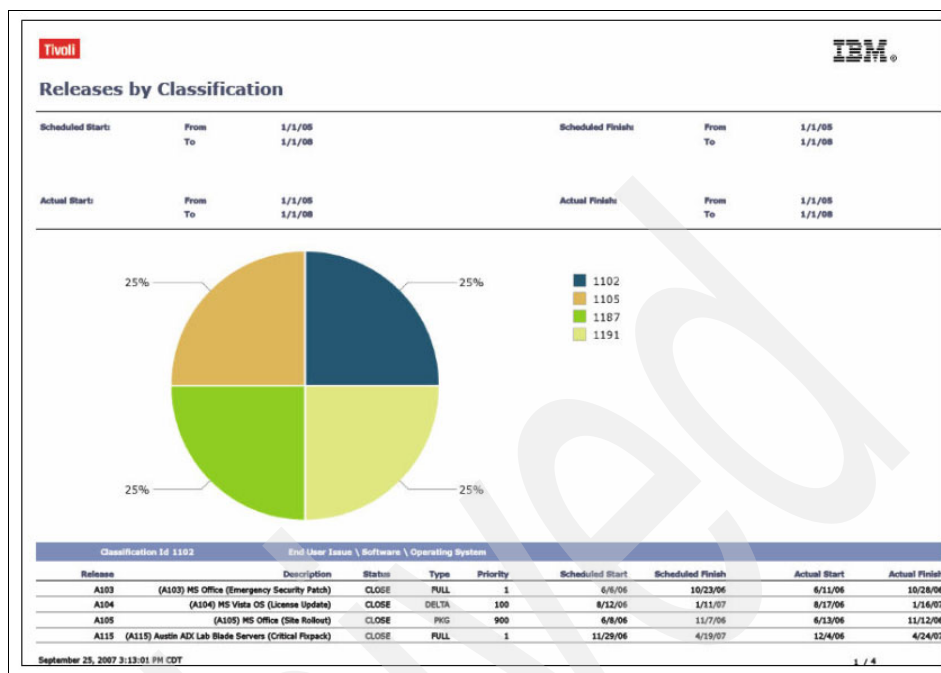


Figure 12-29 Releases by Classification report

12.2.6 BIRT manage reports

CCMDB V7.1 also provides the ability to monitor the usage and execution of BIRT reports. It shows what reports are used, how much time the execution took, whether the report was scheduled, and if it ran successfully.

Start the Report Administration application from the Start Center by selecting **Reports** → **Administration** → **Reporting** → **Report Administration** → **Select Action** → **Run Reports** → **Report Usage**.

Figure 12-30 shows what the report looks like.

Start Date	End Date	Run Time	User	Application	Scheduled?	Success?
11/5/07 12:50:20 PM	11/5/07 12:50:35 PM	0:15	MAXADMIN	CHANGE	N	Y
11/5/07 1:22:12 PM	11/5/07 1:22:14 PM	0:02	MAXADMIN	CI	N	Y
11/5/07 12:57:38 PM	11/5/07 12:57:51 PM	0:13	MAXADMIN	JOBPLAN	N	Y
11/5/07 12:57:27 PM	11/5/07 12:57:28 PM	0:01	MAXADMIN	JOBPLAN	N	Y
11/5/07 12:55:31 PM	11/5/07 12:55:37 PM	0:06	MAXADMIN	JOBPLAN	N	Y
11/5/07 1:02:47 PM	11/5/07 1:02:49 PM	0:02	MAXADMIN	LOCATION	N	Y
11/5/07 1:01:14 PM	11/5/07 1:01:21 PM	0:08	MAXADMIN	LOCATION	N	Y

Figure 12-30 Report Usage report

12.2.7 BIRT report queue

The report queuing function limits the number of jobs that can be executed at any given time. Reports Jobs concurrently processed is configurable by the client. This helps distribute load on the server due to reports, and reduces the amount of wait time for report users who benefit from better performance.

12.2.8 BIRT report localization

In order to set up a reporting language other than English, you must enable reports for localization.

Two specific actions are required for CCMDB V7.1 BIRT reports:

- ▶ Both the language and the locale for the user must be set. They can be set in the User's Default Profile Information, or in the User's Application.
- ▶ The request pages for each of the languages required by a client need to be generated in the report admin app for each of these different languages.

Client needs a report in English, Spanish, and Italian.

Here is an example of report localization. The locale is set to English, as shown in Figure 12-31.

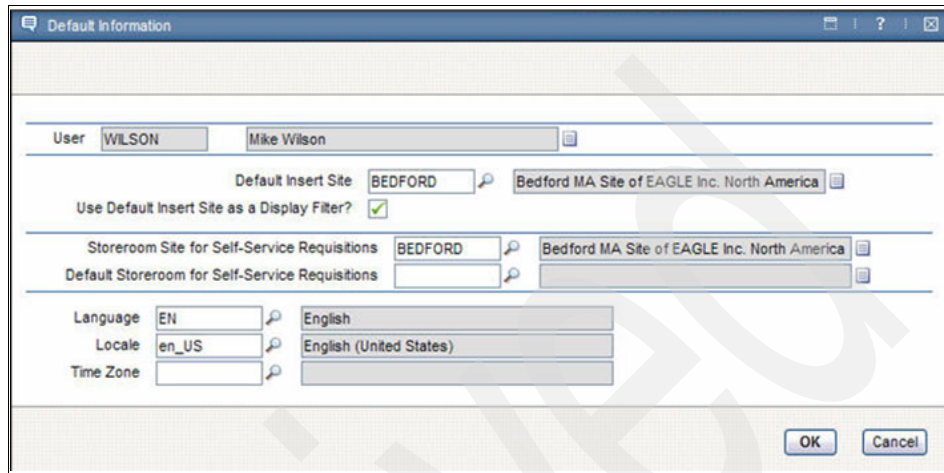
The screenshot shows a 'Default Information' dialog box with several configuration fields. The 'User' field is set to 'WILSON' and 'Mike Wilson'. The 'Default Insert Site' is 'BEDFORD' with a dropdown showing 'Bedford MA Site of EAGLE Inc. North America'. The checkbox 'Use Default Insert Site as a Display Filter?' is checked. The 'Storeroom Site for Self-Service Requisitions' is also 'BEDFORD' with the same dropdown. The 'Default Storeroom for Self-Service Requisitions' is empty. The 'Language' is set to 'EN' with a dropdown showing 'English'. The 'Locale' is set to 'en_US' with a dropdown showing 'English (United States)'. The 'Time Zone' field is empty. At the bottom right are 'OK' and 'Cancel' buttons.

Figure 12-31 Setting report administrator's locale

Do the following:

1. Access Report Admin, and generate the request page XML in English.
2. Sign out, sign back in, and changes the locale to Spanish.
3. Access Report Admin and generate the report request pages in Spanish.
4. Sign out, sign back in, changes the locale to Italian, and generate the report request pages in Italian.

The request pages would then be available in all three languages: English, Spanish, and Italian.

For further information, refer to the *Change and Configuration Management Database: Report Developer Guide*.

12.3 BO Crystal Reports XI Integration (BOC)

The integration of to the Business Objects XI Product Suite enables clients to execute BO/Crystal Reports from within CCMDB V7.1. This integration will be referred to as the BOC Integration. The user does not need to install any Crystal Client tools, and modifications to application specific CCMBD V7.1 files are not required. BOC reports will be deployed as a separate .war file that will be

deployed on the Business Objects Enterprise Server. Its main components include the CCMDB V7.1 Crystal Integration, and the Crystal API and Viewer Libraries.

This section covers the following topics:

- ▶ Requirements
- ▶ Integration with CCMBD V7.1
- ▶ Report development
- ▶ Report administration
- ▶ Security
- ▶ Example windows
- ▶ Report functionality not enabled/supported

12.3.1 Requirements

A report server is required for this integration. The two server components that are supported are either BusinessObjects Enterprise XI or Crystal Reports Server XI.

When BusinessObjects Enterprise XI is used as the server component, Crystal Reports XI is also required for report design. Crystal Reports Server XI includes one report designer license.

An application server separate from the Maximo application server is required. The integration component runs as a Web application on this server, as do several Business Objects applications. Both BusinessObjects XI server products include an optional install of Tomcat, and we recommend that this be used for the integration. However, other application servers are supported by Business Objects.

Note:

- ▶ The integration is developed exclusively for BusinessObjects XI Release 2. No other versions of BOC will be developed or tested.
- ▶ The Business Objects XI Release 2 Version is required due to support for Java 1.5.
- ▶ IBM will not provide any BOC Licenses with Version 7. Clients are responsible for the purchase and maintenance of their own BOC Licenses.

12.3.2 Integration with CCMDB V7.1

Users will access BOC Reports in the same way that they access other reports within CCMDB. This is by:

- Selecting the **Run Report** action within an application (Figure 12-32).

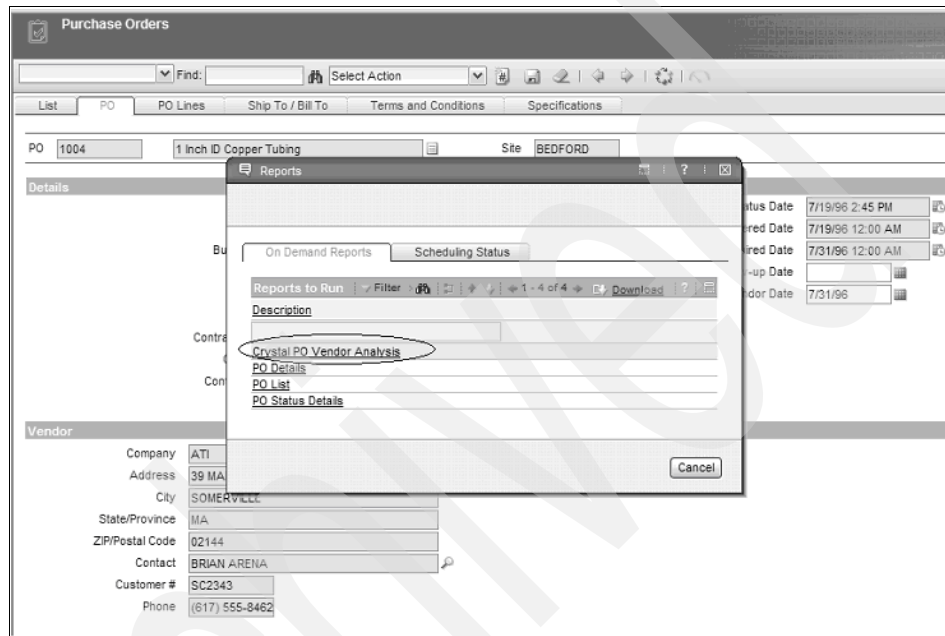


Figure 12-32 Selecting a report through the application

- Through the Reports menu (Figure 12-33).

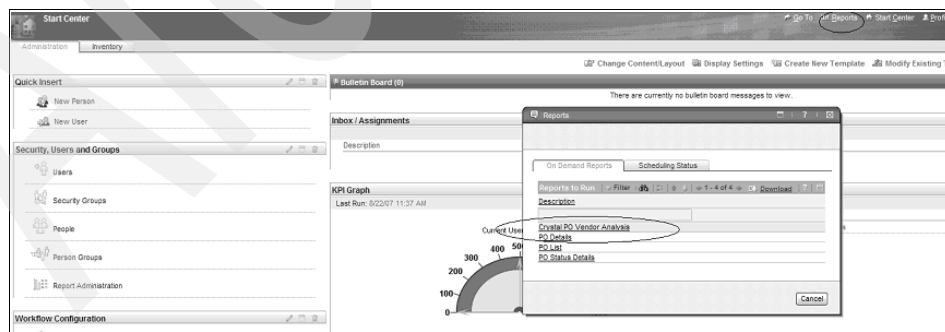


Figure 12-33 Selecting a report through the Report menu

A listing of available BOC Reports will then be visible in the Run Report sub-tab of the Reporting Window.

Once the user submits his report request, the BOC Report will be displayed in a BOC browser session.

The integration will execute against BOC Reports that use either Current/Selected/All Parameters or Bound parameters.

12.3.3 Report development

For the CCMDB V7.1 integration, the BOC Reports must be developed using the Command Table Functionality in the Crystal Report Designer.

Important: Specifically, the Business Object/Crystal Report used in the Maximo 6 Integration must only contain a single command table.

If you try to use both command tables with a combination of database tables, the report will fail because CCMDB V7.1 only updates the database connection for one command table.

Additionally, Business Objects/Crystal Reports does not recommend using command tables in combination with database tables. This is a Business Objects/Crystal Issue, and leads to report performance being slower than expected.

The report performance is slower than expected because command objects have custom SQL statements. When a command object is linked to a table or is linked to another command object, two separate SQL statements are executed on the server. However, the linking is not done on the server.

All the records from both SQL statements are returned to the Crystal Reports Designer. In addition, the join between the command object and tables or the join between the two command objects is performed locally in the Report Designer. These behaviors affect the performance of the report.

To improve the report performance, do not generate a link between the command objects and the tables in Crystal Reports. Instead, use one of the following methods:

- ▶ Modify the command object to use the table within its custom SQL Statement.
- ▶ Create a stored procedure that creates a link between multiple SQL Statements on the server and then returns a single result set to Crystal Reports.

12.3.4 Report administration

BOC Reports must be registered to the CCMDB V7.1 database using the Report Administration Application:

- ▶ A report type of CRYSTAL is used to identify the Crystal Reports in the Database.
- ▶ After registering the BOC Report, the Administrator must use the Create XML functionality to generate the report's Request page.

After registering the report and creating the XML, the security for the report must be set. More details on this are shown in 12.3.5, "Security" on page 363.

12.3.5 Security

There are three levels of Report Security:

- ▶ Does the user have access to Run Reports? (Set in the Security Groups Application for each specific application.)
- ▶ What specific reports can the user run? (Set in the Report Administration Application.)
- ▶ Once the user runs the report, what data will he see? (Passed through the Report Where Clause to Crystal. The report Where Clause includes the Maximo MBO set for security, so whatever the person sees in Maximo, he should see in the report.)

All three Security Levels can be applied to the CCMDB V7.1 BOC Integration. Security Level, that is, what specific reports the user can run, is a new feature for BOC Reports in CCMDB V7.1. This security feature can be set at either the individual report level or at the application level. Prior to this release, all users would have access to all BOC Reports.

To set report security at the report level, access the report in Report Admin. Click the **Security** tab, and select the Security Group that can run that specific report. Only those Security Groups who have Run Report access to the application will be available for selection. See Figure 12-34.

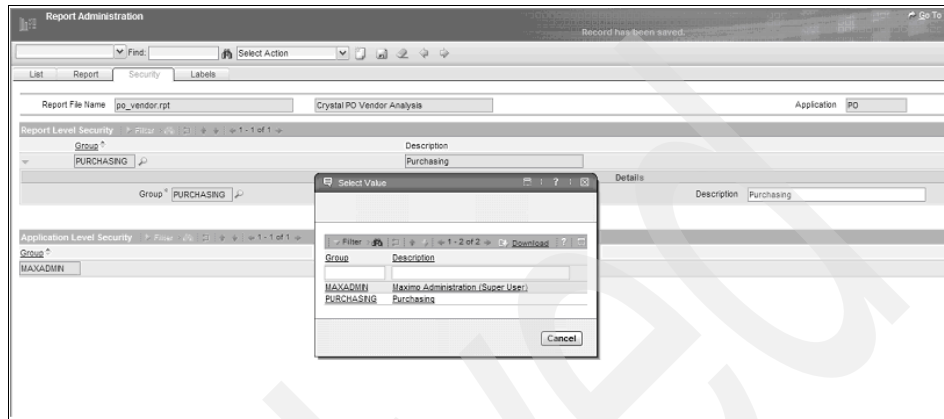


Figure 12-34 Selecting security groups

To set report security at the application level (meaning the security group can see ALL reports registered to that specific application), access Report Admin. From the Action Menu, click **Set Application Security**. Select the application, and then choose the Security Group that should have access. Please note that you can grant report application security to all report types, or just to specific report types like BIRT or Crystal, as shown in Figure 12-35 on page 365.

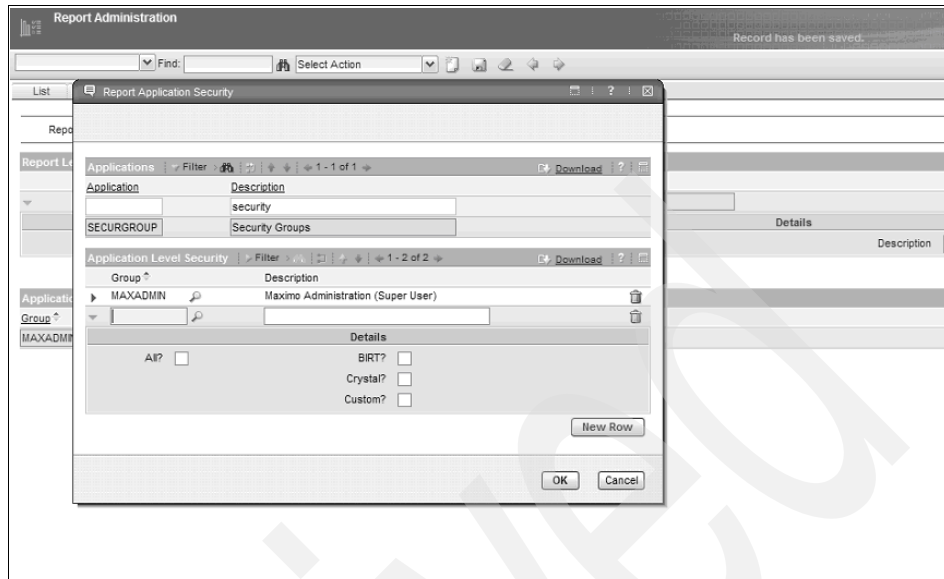


Figure 12-35 Setting reporting application security

Run **BOC Report** from the On Demand Reports sub-tab in CCMDDB V7.1, as shown in Figure 12-36.

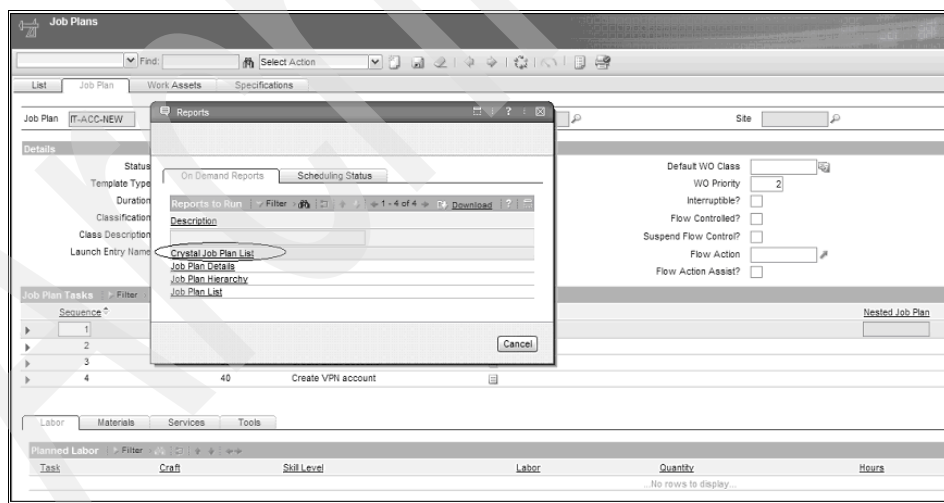


Figure 12-36 On Demand reports sub-tab

When a user selects Crystal Report, a Request Page appears. This is where the user would enter any desired parameter values. See Figure 12-37.

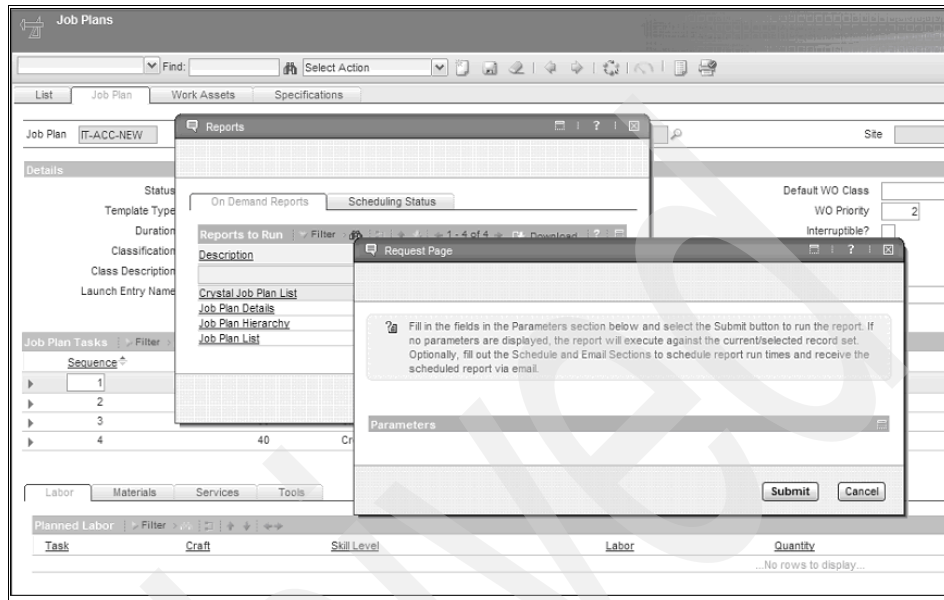
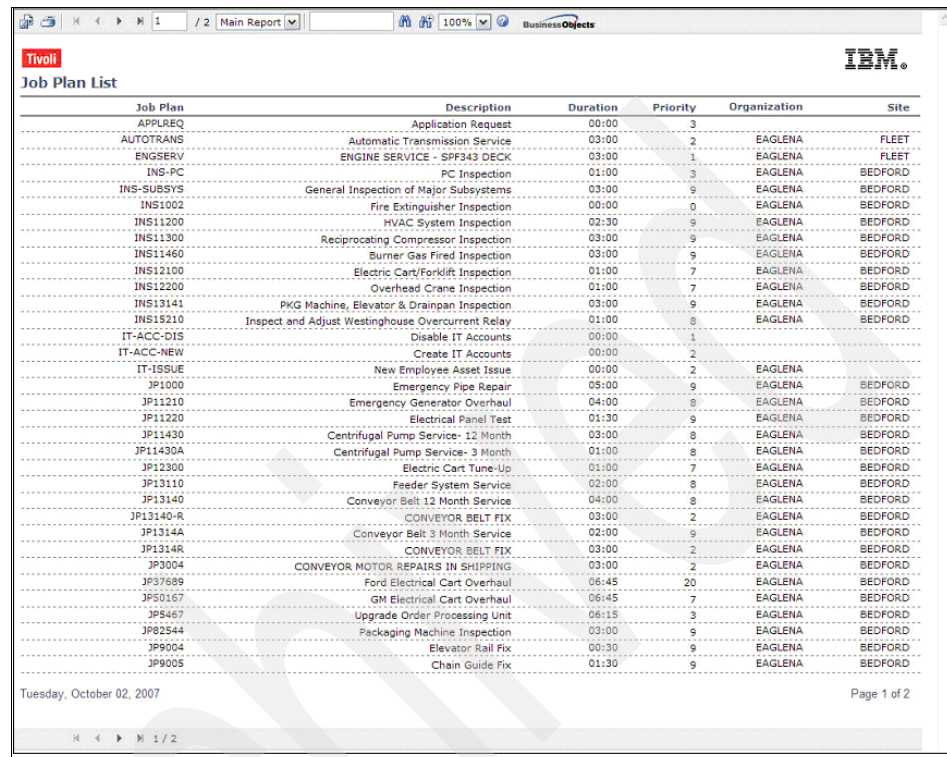


Figure 12-37 Entering report parameters

An example of a CCMDB V7.1 BOC Report Displayed in the BOC Browser is shown in Figure 12-38.



Job Plan	Description	Duration	Priority	Organization	Site
APPLREQ	Application Request	00:00	3		
AUTOTRANS	Automatic Transmission Service	03:00	2	EAGLENA	FLEET
ENGSESV	ENGINE SERVICE - SPF343 DECK	03:00	1	EAGLENA	FLEET
INS-PC	PC Inspection	01:00	3	EAGLENA	BEDFORD
INS-SUBSYS	General Inspection of Major Subsystems	03:00	9	EAGLENA	BEDFORD
INS1002	Fire Extinguisher Inspection	00:00	0	EAGLENA	BEDFORD
INS11200	HVAC System Inspection	02:30	9	EAGLENA	BEDFORD
INS11300	Reciprocating Compressor Inspection	03:00	9	EAGLENA	BEDFORD
INS11460	Burner Gas Fired Inspection	03:00	9	EAGLENA	BEDFORD
INS12100	Electric Cart/Forklift Inspection	01:00	7	EAGLENA	BEDFORD
INS12200	Overhead Crane Inspection	01:00	7	EAGLENA	BEDFORD
INS13141	PKG Machine, Elevator & Drapain Inspection	03:00	9	EAGLENA	BEDFORD
INS15210	Inspect and Adjust Westinghouse Overcurrent Relay	01:00	8	EAGLENA	BEDFORD
IT-ACC-DIS	Disable IT Accounts	00:00	1		
IT-ACC-NEW	Create IT Accounts	00:00	2		
IT-ISSUE	New Employee Asset Issue	00:00	2	EAGLENA	
JP1000	Emergency Pipe Repair	05:00	9	EAGLENA	BEDFORD
JP11210	Emergency Generator Overhaul	04:00	8	EAGLENA	BEDFORD
JP11220	Electrical Panel Test	01:30	9	EAGLENA	BEDFORD
JP11430	Centrifugal Pump Service- 12 Month	03:00	8	EAGLENA	BEDFORD
JP11430A	Centrifugal Pump Service- 3 Month	01:00	8	EAGLENA	BEDFORD
JP12300	Electric Cart Tune-Up	01:00	7	EAGLENA	BEDFORD
JP13110	Feeder System Service	02:00	8	EAGLENA	BEDFORD
JP13140	Conveyor Belt 12 Month Service	04:00	8	EAGLENA	BEDFORD
JP13140-R	CONVEYOR BELT FIX	03:00	2	EAGLENA	BEDFORD
JP1314A	Conveyor Belt 3 Month Service	02:00	9	EAGLENA	BEDFORD
JP1314R	CONVEYOR BELT FIX	03:00	2	EAGLENA	BEDFORD
JP3004	CONVEYOR MOTOR REPAIRS IN SHIPPING	03:00	2	EAGLENA	BEDFORD
JP37689	Ford Electrical Cart Overhaul	06:45	20	EAGLENA	BEDFORD
JP50167	GM Electrical Cart Overhaul	06:45	7	EAGLENA	BEDFORD
JP5467	Upgrade Order Processing Unit	06:15	3	EAGLENA	BEDFORD
JP82544	Packaging Machine Inspection	03:00	9	EAGLENA	BEDFORD
JP9004	Elevator Rail Fix	00:30	9	EAGLENA	BEDFORD
JP9005	Chain Guide Fix	01:30	9	EAGLENA	BEDFORD

Tuesday, October 02, 2007

Page 1 of 2

Figure 12-38 Example of BOC report

Important:

- ▶ A client may want to use the out-of-the-box V7 BIRT Reports, but develop their own custom reports in BOC. This configuration is supported, but we strongly recommend that the Application Server, Database, and Report Server, each have an individual server.
- ▶ All development for the BOC Integration was done on a Processor Based License, which has led to the development of reports under one named user. Optional Crystal Licenses are available, including Named User Licenses. The client must manage any licensing conflicts they encounter directly with BOC.
- ▶ DB2, Oracle, and SQL Server Databases are supported in this integration. DB2 and SQL Server reports will use an ODBC Connection, while Oracle reports will use a Native Oracle Connection.
- ▶ Bound parameters can be used. Bound parameters are parameters with equivalent fields or relationships in the Maximo databases. By using the attribute field lookup in the Report Administration Application, you can easily tell if parameters are bound or not.
- ▶ Unbound parameters, that is, those that do not have relationships with the Maximo database, are not supported for the BOC Integration.
- ▶ For best performance, IBM recommends that your configuration contain dedicated servers for the CCMDB V7.1 Application Server, BusinessObjects Enterprise Server, and Database Server.
- ▶ Business Objects/Crystal is replaced by BIRT as the embedded reporting tool in CCMDB V7.1 and future releases.
- ▶ CCMDB V7.1 runs on an open database systems (Oracle/ MS SQL Server), and customers can run their reporting tool of choice directly against the CCMDB V7.1 database.

12.3.6 Report functions not supported

Table 12-1 on page 369 details the report functions that are not enabled or supported with the BOC integration.

Table 12-1 BOC features not enabled or supported

Feature	Description
Licensing	IBM does not include a BusinessObjects/Crystal License.
Multi Server Configurations	IBM will not support configurations of multiple servers to one BO Enterprise Server.
Report Browser	No customization of the Report Browser will be done to mimic the UI and functionality of the V7 Release. The BO/Crystal Browser will display as it is delivered from BO/Crystal.
Hyperlinks	Report hyperlinks from one BOC Report to another.

Table 12-2 details the report functions that are not enabled or supported with the CCMDB V7.1 integration.

Table 12-2 CCMDB V7.1 reporting functions not enabled or supported

Feature	Description
CCMDB V7.1 Out-of-the-Box Reports	The CCMDB V7.1 Out-of-the-Box reports developed in BIRT will not be rewritten in BOC. However, the Job Plan List will be delivered in BOC for testing purposes.
View Reports	Previously executed BOC reports are not retained, and will not be available for display in CCMDB V7.1.
Schedule Reports	Scheduling of BOC Reports through the Report Request Page. This functionality is hidden for BOC Report Types.
E-mail Reports	E-mailing of BOC Reports through the Report Request Page. This functionality is hidden for BOC Report Types.
Unbound Parameter Reports	An unbound parameter is a parameter that has no relationship to the application's main table or maxrelationships.
Quick Toolbar Access	Direct Access to BOC Reports from the toolbar in Maximo applications.
Print Attached Documents	Not enabled for BOC Reports.

Feature	Description
Report Label Features	The visibility and ability to update report titles and labels through Report Admin is not available.
Report Limits	Configurable record limits are not enabled for BOC Reports.
Priorities	Administrators can input values for priorities for BOC Reports. However, the V7 Integration will not use this functionality; it will be up to the client to extend this functionality if needed.

12.4 External Report Integration (ERI)

External Reporting Integration enables the user to pass the CCMDB V7.1 *Where Clause* (or current/selected/all record set) to an External Reporting System without leaving CCMDB V7.1. Additionally, any number of report parameters can be passed, with or without the Where Clause, to an external report system.

This function is referred to as External Report Integration (ERI.) It is very similar to the BusinessObjects Crystal Reports XI Integration, except it is more flexible in that it is Report System and Report Version independent.

Note: CCMDB V7.1 runs on an open database system (DB2, Oracle, or MS SQL Server). Customers can bypass this integration, and run their reporting tool of choice directly against the CCMDB V7.1 database.

This section covers the following topics:

- ▶ Requirements of ERI
- ▶ Enabling the ERI
- ▶ Registering and running ERI reports
- ▶ Report Admin App
- ▶ Parameters
- ▶ Security
- ▶ Report functionality not supported

12.4.1 Requirements of ERI

The External Report Server must provide a Web-based interface. Maximo will send required report-related information through an HTML form post to the URL specified in the Maximo System Properties. There must be a Web application running at that URL that can receive the information, process it accordingly, and communicate with the Report Server in order to execute a report.

Note: IBM will not provide any additional external Reporting System Licenses with CCMDB V7.1. Clients are responsible for the purchase and maintenance of their own External Report System licenses.

Note: IBM will not support issues with External Reporting Systems related to the setup or maintenance of the Report Server, report development, or debugging.

12.4.2 Enabling the ERI

There are three components to consider for enabling CCMDB V7.1 to integrate with your selected Reporting System:

- ▶ System Properties
- ▶ customreport.jsp
- ▶ Password Encryption

System Properties

First, access the System Properties Application. In the Global Properties table, select **New Row** and enter the two Property Values shown below. Leave the default values except where specified:

Property Name	mxe.report.custom.serverURL.
Description	URL of the custom reporting application.
Global Value	http://<maximoserver>:<port>/maximo/webclient/utility/customreport.jsp.
Global Only	True (checked). The Global Value above can be used to run the provided sample integration JSP™ (described below). Normally, this would be the URL where the integration to the external report server is located.
Property Name	mxe.report.custom. rptServerLogonPass.

Description	Password used for logging on to the external report server administration tool.
Global Value	<enter Password>.
Encrypted	True (checked).
Global Only	True (checked).

Second, depending on your External Reporting System, you may need to pass additional property values from CCMDB V7.1. To do this, you can add more property settings in the System Properties Application.

For example, assume a root folder name for the External Reporting System must be passed. This new property value must be added with the prefix `mxe.report.custom` so it is picked up by CCMDB V7.1.

Therefore, the new property value would be:

- ▶ Property Name: `mxe.report.custom.rootFolder`
- ▶ Description: Root folder for external report server
- ▶ Global Value: `maxreports`
- ▶ Global Only: True (checked)

Any additional properties having names beginning with `mxe.report.custom` will be passed as parameters to the external report server. The parameter name will be the word “custom” prepended to the last segment of the property name. In this example, the property above would have a parameter name of `customrootFolder`. See the sample integration JSP in “`customreport.jsp`” on page 372 for more information.

After the property values have been entered, and you have verified that the information is correct, save the values.

You must restart Maximo for these changes to take effect.

customreport.jsp

Next, the `customreport.jsp` file will be reviewed. This file is located in `<maximo>applications\maximo\maximouiweb\webmodule\webclient\utility`.

The `customreport.jsp` file shows the relevant information that is always sent from CCMDB V7.1 to the report server, and provides instructions on how to access that information. There are additional values sent, but they may only pertain to the CCMDB V7.1 Embedded Report Integration.

Examples of the values in this file that are always passed include:

- ▶ Report File
- ▶ Report Folder
- ▶ Schema
- ▶ Application Name

It will be up to the client and the External Reporting System that they are integrating with to determine if they will use this customreport.jsp file in their integration. The intention of this file is to only highlight the information that is passed from CCMDDB V7.1 to the external reporting System, so that the receiving portion can determine how it will process this information.

These parameters are sent from a hidden form post from the CCMDDB V7.1Server to the External Report Server.

Here is a list showing Standard Values passed as request parameters from CCMDDB V7.1:

Report File Name	Security_analysis.rep.
Report Description	Oracle BI Security Analysis Report.
Report Folder	SECURGROUP.
Report Type	SECURGROUP.
CCMDDB V7.1 User Name	wilson.
CCMDDB V7.1 Password	See "Password encryption" on page 374.
Database User Name	maximo.
Database Password	See "Password encryption" on page 374.
Schema	maximo.
Maximo Where Clause	((independent = 1)).
User Org	EAGLENA.
User Site	BUFFALO.
App Main Table	MAXGROUP.
App Name	SECURGROUP.
Language Code	EN.
Locale	EN_US.
User's Local Timezone	EST.
Report Server URL	http://144.23.45:8090/oraclebi.
Report Server Password	See "Password encryption" on page 374.

Password encryption

CCMDB V7.1 always encrypts the following values before being sending them to the External Reporting System:

- ▶ Maximo Password
- ▶ Database Password
- ▶ Report Server Password

It is up to the client to determine what security will be implemented on the External Reporting System.

If the client chooses to use any of the passwords sent from CCMDB V7.1, the values must be decrypted in the Web application. The `customreport.jsp` file provides examples of how to do so.

The decryption requires a CCMDB V7.1 class file, `CipherPlusBase64.class`, which must be made available to the receiving Web application, for example, by placing it in the Web application directory `WEB-INF\classes\psdi\util`.

`CipherPlusBase64.class` can be obtained from
<maximo>applications\maximo\businessobjects\classes\psdi\util.

12.4.3 Registering and running ERI Reports in CCMDB V7.1

This section will discuss how to register an ERI Report within CCMDB V7.1.

We will use an example of a client who has the corporate reporting tool of Oracle Business Intelligence (BI). They want to use this reporting tool to create a custom CCMDB V7.1 Report called the Oracle BI Custom Security Analysis Report.

After enabling the ERI, the Report Administrator needs to register each Oracle BI Report that will be accessed from CCMDB V7.1. To do this, the Administrator accesses the CCMDB V7.1 Report Administration Application. The following information is required for every report that will be executed from CCMDB V7.1, as shown in Figure 12-39 on page 375:

Report File Name	This is the file name of the report. It is the name that the Report Server uses to identify the report to execute. The Report System determines the value of this field, and whether or not a file extension must be included.
Report Description	This is the description of the report as it appears to the user on the Report's Request Page in Maximo.

- Report Run Type

For ERI, the Report Run Type is always CUSTOM. The Report Run Type of CUSTOM should *always* be used when registering a report that is not a BIRT, or Business Objects/Crystal Report.
- Application

This is the Maximo Application where the Report will be accessible from. In this example, the Oracle BI Custom Security Group Report will be accessible from the CCMDB V7.1 Security Group Application.
- Report Folder

The Report Folder is the folder name where the report is kept on the Report Server. This field defaults to the Application folder name. With the Standard CCMDB V7.1 Report Implementation, Report Folders were created on the Report Server to correspond to each of the Maximo Applications that had reports. Depending on your report server and setup, you may or may not want to set it up this way.

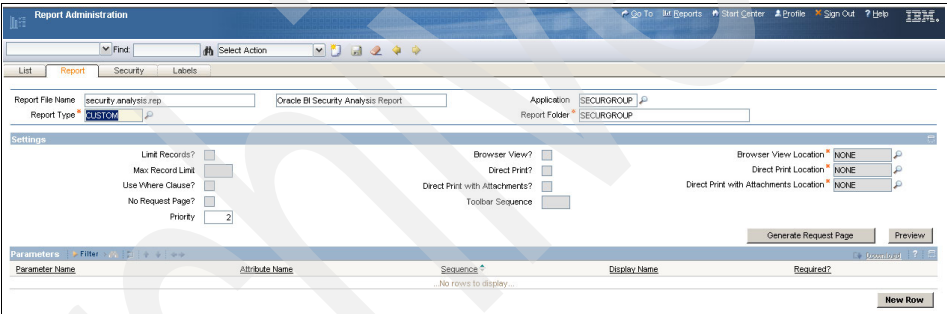


Figure 12-39 Report Administration application

Note:

- ▶ There is no validation of any of these fields to the External Report Server.
- ▶ All but one value in the Report Details Section are not applicable to Reports from External Reporting Systems. These fields are only applicable to CCMDB V7.1 Embedded Reports.
- ▶ The values that are not applicable are greyed out in Figure 12-39 on page 375, and include:
 - Limit Records
 - Use Where Clause
 - No Request Page
 - Toolbar Links to Browser View, Direct Print, and Direct Print with Attachments
- ▶ The only field that is enabled is priority. It is up to the client if they want to use this value in their custom CCMDB V7.1 Report Integration.

Parameters

The ERI enables external reports to include both bound and unbound parameters. Bound parameters are parameters that have either a relationship to the application's main table or exist through the maxrelationships value. Unbound parameters do not have any relationships to the application's main table and they do not exist through the maxrelationship value.

In the custom Security Group Report in 12.4.3, “Registering and running ERI Reports in CCMDB V7.1” on page 374, an example of a bound parameter would be independent (MAXGROUP.INDEPENDENT). An example of an unbound parameter would be Classification Path. Classification Path is unbound because there is no relationship between that variable and the Security Group Application.

If a parameter is bound, its value will be included in the CCMDB V7.1 Where Clause that is passed to the report. If a parameter is unbound, its value will not be included in the CCMDB V7.1 Where Clause.

In the example, no parameters have been defined for this report, so it will execute against the Current/Selected/All Record Set defined in the application.

Finally, note that it is the responsibility of the client to manage the types of report parameters used.

Once the external report and any of its parameters have been registered, the Report Administrator must generate the Request Page for the Report. This is done by clicking the **Generate Request Page** button. When this has been done,

a message will display on the toolbar saying that the Request Page has been generated.

Report security

Once the request page has been generated, security for this report needs to be granted. In CCMDB V7.1, security can be granted at either the Application Level or Individual Report Level.

If report security is set at the Application Level, the selected security groups will have access to all reports stored in that application. This is done through the Set App Security Action available in Report Admin.

To grant access, click a new row and then select the Security Group from the lookup. The lookup only displays those Security Groups who have been granted Run Report Access to the selected Application through the Security Group Application, as shown in Figure 12-40.

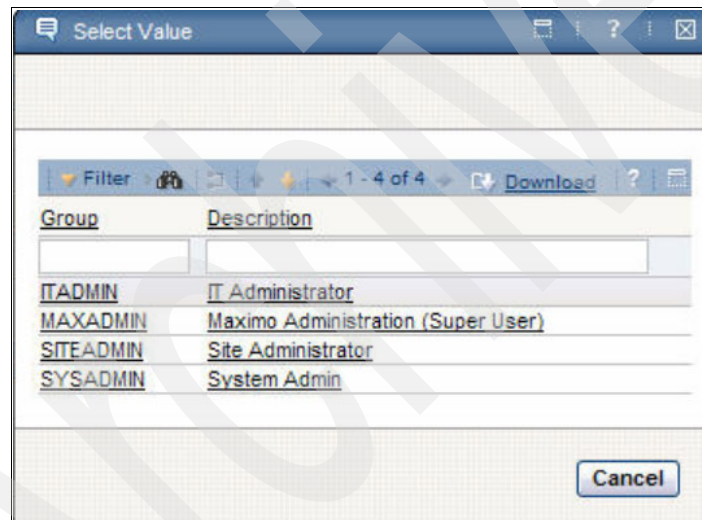


Figure 12-40 Security group selection

The application security can also be set by Report Type, as shown in Figure 12-41 on page 378. The options available are:

All	Grants access to all reports, regardless of type.
BIRT	Grants access to BIRT Reports only.
Crystal	Grants access to BO/Crystal Reports only.
Custom	Grants access to Custom ERI Reports only.

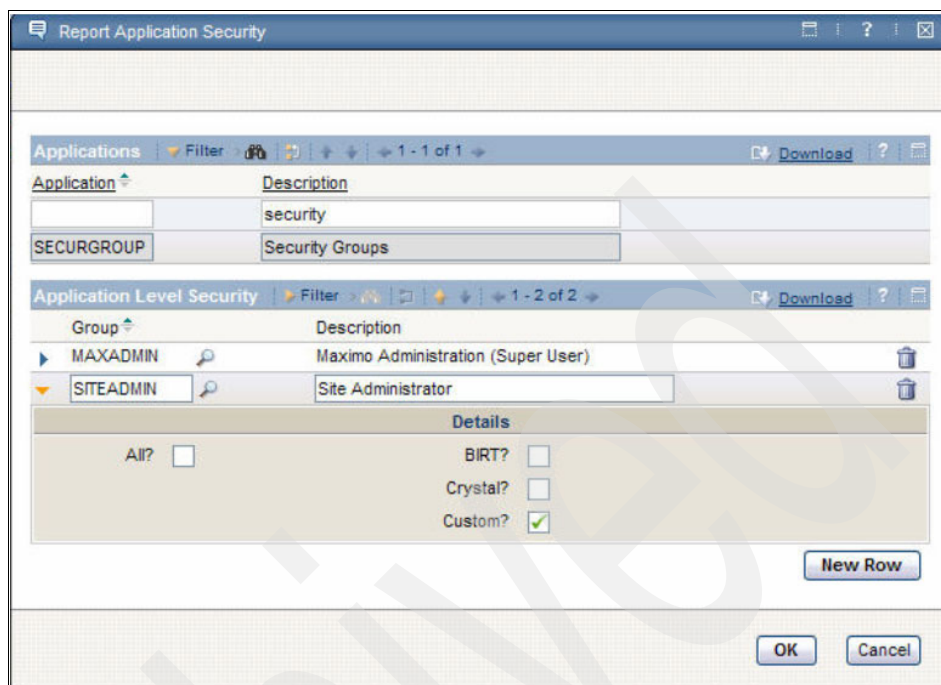


Figure 12-41 Setting security by application

There are no report type limitations on application level security. For example, one group (MAXADMIN) can have access to ALL reports within the Security Group App, while another group (SITEADMIN) can only have access to CUSTOM reports within the Security Group App.

If report security is set at the Report Level, the selected security groups will only have access to the specified report.

To set individual report level security, click the **Security** tab for the report, click **New Row**, select the Maximo Group from the lookup, and click **Save**.

Note: If App level security has already been granted, it will be displayed in the section Application Level Security.

The custom report is now available to the specified Security Groups from the Security Group Application.

To run this custom report, access the Security Group Application. Filter the result set to obtain a selected query. In Figure 12-42 on page 379, a filter of Independent = Yes was input. This results in a selected record set of 10 records.

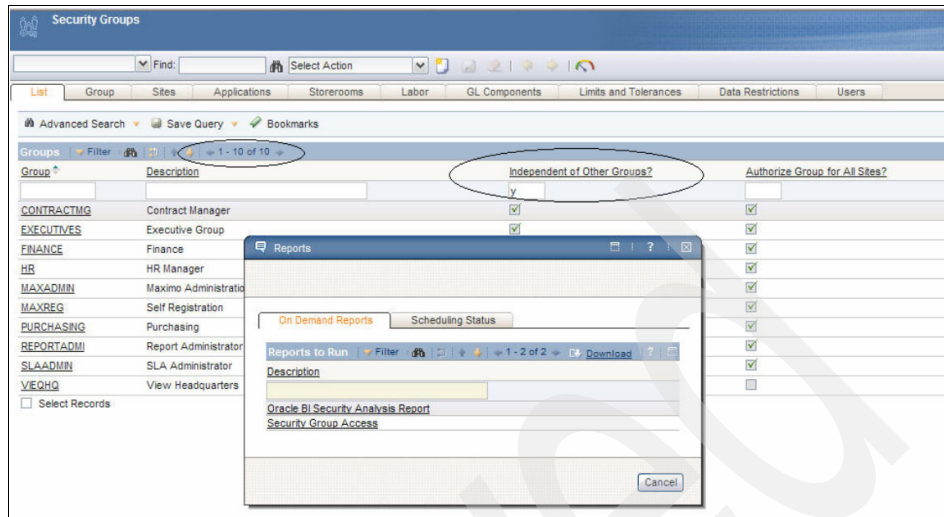


Figure 12-42 Report availability

From the action menu, select **Run Reports**.

A list of all reports that the user (through his security group rights) has access to will display. In this example, both an Oracle BI Report and a BIRT report are displayed.

The user selects the Oracle BI Security Analysis Report and its Maximo Request Page is displayed, as shown in Figure 12-43.

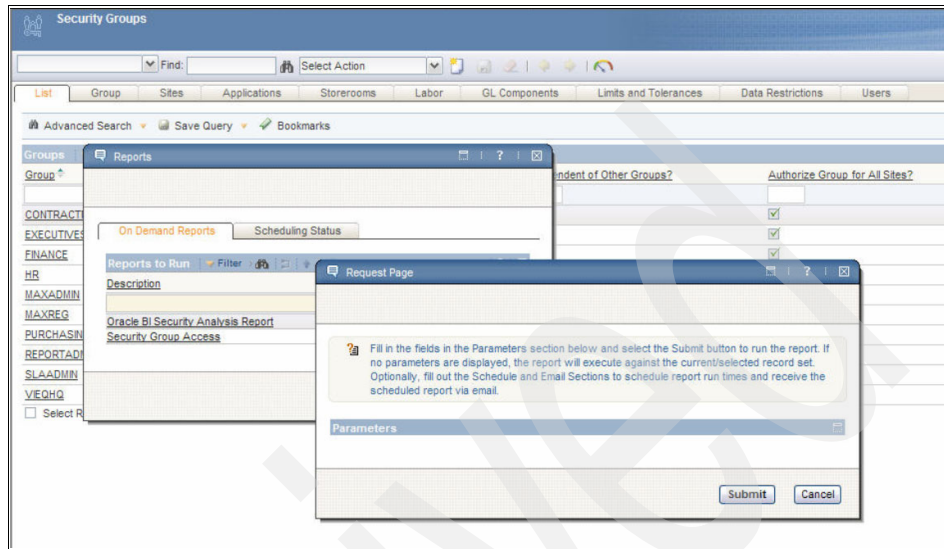


Figure 12-43 Selected Oracle BI Security Analysis Report

The user clicks **Submit**. CCMDB V7.1 will pass the Selected Record Set of 10 Security Group Records to the External Reporting System. A separate browser session is opened, and the external report displays using the selected record set.

Important: A client may want to use the out-of-the-box CCMDB V7.1 BIRT Reports, along with their own custom reports developed in an external reporting system. This configuration is supported; however, we strongly recommend that the Maximo Application Server, Database, and External Report Server have its own box.

DB2, Oracle, and SQL Server Databases are supported in this integration.

For best performance, IBM recommends that your configuration contains dedicated servers for the CCMDB V7.1 Application Server, Report Server, and Database Server.

Report functions not supported

Table 12-3 on page 381 details the report features that are not enabled or supported in the Maximo ERI.

Table 12-3 Report functions that are not supported

Feature	Description
Ccldb V7.1 Out-of-the-box Reports	No ERI Reports will be delivered with CCMDB V7.1.
Multi Server Configurations	IBM will not support configurations of multiple V7 servers to multiple External Report Servers.
Schedule Reports	Scheduling of Reports through the CCMDB V7.1 Report Request Page.
E-mail Reports	E-mailing of Reports through the CCMDB V7.1 Report Request Page.
Application Toolbar Icons	Direct Access to External Reports from the toolbar in CCMDB V7.1 applications.
Printing Of Attached Documents With Report	Not enabled for External Reports.
Report Label Features	The visibility and ability to update report titles and labels through Report Admin is not available.

Glossary

This section provides a list of terminology used within the scope of the IBM CCMDB solution. Instead of providing this glossary in alphabetical order, we have structured this glossary to align the terms to the appropriate CCMDB component to which they primarily belong.

Data Model

Actual Configuration Item A configuration item (CI) in the CCMDB process database (Maximo) created by importing a discovered CI from the CCMDB Discovery Server / TADDM database through the Integration Composer. Actual CIs are read-only. They can be promoted to Authorized CIs.

Attribute A descriptive characteristic of a Configuration Item (CI), such as a make or model number, version number, purchase contract number, or release number. The Common Data Model is extensible to add attributes.

Authorized Configuration Item A configuration item (CI) in the CCMDB process database (Maximo) that is subject to control and modification by the Change Management and Configuration Management processes. An authorized CI can be created by promoting an actual CI, by creating it manually or by using the MEA interface technology.

Bulkloader This is an instance of the CCMDB Discovery Server / TADDM. The bulkloader, also known as the discovery library reader, reads books in the discovery library file store and imports them into the database of the discovery server. During the import, a

book is translated from an IdML format to the Common Data Model representation. The bulkloader uses the `loadidml` command to import a book.

Configuration Item Any component of an information technology infrastructure that is under the control of configuration management.

Configuration Management Database A database that contains details about the attributes and history of each configuration item and the details about the relationships between configuration items. In CCMDB, the CMDB is implemented as the conjunction of the TADDM and Maximo databases, plus federated databases added to either of these.

Common Data Model A logical representation of CMDB entities, relationships, and their semantics.

Discovered Configuration Item A configuration item that has been discovered in the IT environment and stored in the CCMDB Discovery Server / TADDM database. The CIs can be synchronized into the Actual CI representation.

Discovery Library A file system directory that stores discovery library books, which contain IdML representations of data copied from operational management products by discovery library adapters.

Discovery Library Adapter (DLA) A program that copies data from an operational management product, converts it to IdML, and stores it in books in the discovery library.

Discovery Library book An IdML file in the discovery library, containing data copied from an operational management product.

Discovery Library Reader™ Any program that reads configuration item data from IdML books in the discovery library and converts it to the format that is required by a consuming application. The bulk loader is a Discovery Library Reader.

Extended Attribute A configuration item (CI) attribute that is not part of the original Common Data Model, but is added by the customer.

General Unique Identifier (GUID) A system generated identifier to uniquely identify a CI. The GUID is transferred from the discovered CI space to the Actual CI space through the Integration Composer and provides the basis for the Launch in Context application.

Integration Composer A component to import data into the CCMDB process database from external discovery systems. In CCMDB V7.1, the integration composer is used to import data from the discovered CI space in the CCMDB Discovery Server / TADDM database into the Actual CI space.

International Development Markup Language (IdML) The XML format that is used to store data in the discovery library.

Promotion The process of creating an authorized configuration item from an actual configuration item.

Reconciliation The process of avoiding duplicates of configuration items in the CCMDB when discovering or importing data from multiple sources. In CCMDB V7.1, the process of reconciliation is executed in the CCMDB Discovery Server when discovering

or importing data through the Bulkloader or API.

Relationship A description of the dependency or connectivity between configuration items.

Base Services

Access Collection Belonging to the area of security, an access collection is a group of objects, such as configuration items, created for data-level access control. Users can work with objects in an access collection if they belong to a security group assigned to work with that access collection. Collections defined in the process environment are synchronized to the CCMDB Discovery Server / TADDM.

Activity The largest unit into which a process is divided. It gets its technical representation through a Job Plan template definition. An activity can be a container for multiple tasks.

Artifact Any file, object, or other piece of data that is created or used during the execution of a process.

Collection A general-purpose container that has configuration items or other database artifacts as its members. See *Access Collections*.

Communications Template A template to define or customize a communication or notification. For example, when notifying a person or person group by e-mail, the body of the message can be predefined by a template and gets completed at runtime by substituting variables in the communication template definition.

Escalation A condition-action pair that can be run against the CCMDB process database at scheduled times. The condition is an SQL-Where clause.

Job Plan A template consisting of tasks to be performed to get a particular job done. CCMDB V7.1 provides default templates of Job Plans for Configuration and Change Management, such as a template for a Standard or predefined Change. A job plan gets applied, for example, to a change object and gets initiated into a work plan.

Key Performance Indicator A measurement identified as an important metric for how efficiently or effectively a process is being carried out. Role-appropriate indicators are displayed on start centers.

Logical Management Operation (LMO) Interfaces specified by system integration module PortType Web services description language (WSDL) operations and discrete operations that are transformed to operational management product operations and workflows. An example of a logical management operation is Distribute Software, which needs to get implemented by specific code in order to call a specific Software Distribution product to perform this action. The Release PMP provides an implementation for Tivoli Configuration Manager and Tivoli Provisioning Manager to implement this action.

Maximo Business Object (MBO) A Java object with business logic that encapsulates a database table in the CCMDB process database.

Maximo Enterprise Adapter (MEA) A component of the CCMDB process layer that facilitates sending and receiving business objects (for example, process requests or authorized configuration items) to or from

external applications, such as an external service desk or database. The inputs and outputs are typically MBOs.

Notification A way to send a notification to a person or person group. This can either be an e-mail or a message in the Start Center.

Process A series of related activities aimed at achieving a set of objectives in a measurable, usually repeatable manner.

Process Manager Product (PMP) A system of managing the execution of a process. A process manager operates the defined and agreed process, ensuring that it interfaces with all other relevant processes, target setting, process audits, effectiveness and efficiency reviews, and managing the process improvement cycle. Change and Configuration Manager PMPs are delivered with the IBM CCMDB V7.1.

Process Request A trackable and assignable request for work, specifically created for process manager products like Configuration, Change, or Release Management. These entities include RFCs, Release Records, Incidents, or Problems,

Role A set of job responsibilities related to a Service Management process. In CCMDB, each role is implemented as a security group, which gives users with that role access to a set of applications and a start center with role-appropriate information.

Security Group A group defined in a directory server for the purpose of providing access to applications and optionally to collections of data.

Start Center The first page a CCMDB user sees when logging in into the CCMDB Web Interface. Contains links, work queues, performance indicators, and other content appropriate to the user's role.

System Integration Module (SIM) A program that is invoked during a process activity, which interacts with a managed software system.

Task A step in an activity that is part of a defined process. Tasks are defined as part of a Job Plan template definition.

Template (Job Plan Template) A predefined process of activity roadmap that can be applied to specific process workflows and modified to meet the needs of a specific workflow. Templates can be created, edited, cloned, or deleted. They are also referred to as job plan templates.

Workflow Workflows are a mechanism inside the Base Services to help route a record, for example, a Change Object, from one status to the next one, apply conditional checking on the semantics of a record (for example, approval limit), trigger actions, or assign records to appropriate groups of persons. In the context of the CCMDB V7.1 solution, the workflow technology is used from within the end-to-end process flow definition (Job Plans). Job Plans define activities and tasks, and tasks can call actions or workflows. CCMDB V7.1 uses the workflow technology to call short sequences of code logic, for example, to set the state of a Change record or do conditional checking on the attributes of a record.

In CCMDB V7.1, it is also used to support activities in the context of process requests like Accept, Submit, or Reject an RFC.

Work Order Consists of an instantiation of all work plans corresponding to an approved work request.

Work Plan An instantiation of a Job Plan Template when it gets applied to an object like the change object.

Change Management Process Manager

Change Implementation Schedule A view in Change Management that shows the start and end dates for changes and included implementation tasks to selected configuration items. Different views are provided for this Change Calendar-like application.

Change Management The process of planning for and executing changes to configuration items.

Change Window A period of time defined for one or more configuration items, which specifies when the CI can be taken out of service for changes to be made. The Change Window application is used to define maintenance windows for configuration items.

Change Window Conflict A condition that occurs when implementation tasks have been scheduled for a CI outside its defined change window or maintenance window.

Impact Analysis The definition of the impact of a proposed change, including the identification of implementation tasks, targeted and impacted configuration items, and other possible impacts.

Impacted Configuration Item A CI that is not the direct target of a proposed change, but that will be affected by the change. Often identified by relationships with target CIs.

Implementation Task A task required to carry out a change. The performance of an implementation task might cause the targeted and impacted configuration items to be taken out of service. Identification of implementation tasks is often performed during impact analysis.

Related RFCs Other requests that are parent, corequisites, or prerequisites of an RFC. An RFC can have only one parent RFC.

Request for Change A means of proposing a change to any component of the information technology infrastructure or any aspect of an information technology service.

Request for Change Type The kind of change that a request for change (RFC) defines. Each RFC is required to have a type. Change Management process owners can edit or remove the predefined RFC types and create new types. RFCs get classified in order to get a type.

Target CI A configuration item (CI) that is expected to be affected by a proposed change. A target CI can be defined when a request for change (RFC) is created, when an implementation task is defined, or at other points in the change process, especially during impact analysis.

Configuration Management Process Manager

Audit A comparison of the actual and authorized versions of a set of configuration items.

Configuration Management The process of planning for, identifying, controlling, and verifying the configuration items within a service, recording and reporting their status and, in support of Change Management,

assessing the potential impact of changing those items.

Life Cycle State One of a defined set of status values that indicates the current state of a configuration item, such as development, test, production, or decommissioned.

Update Request Provided as part of the Configuration Management process. The Update Request is a specifically classified process request to Configuration Management in order to ensure a controlled way of adding, updating, or deleting data in the CMDB. An update request is usually generated as a final step within a Change Management process.

CCMDB Discovery Server / TADDM

Anchor A component in the CCMDB discovery environment that supports the main discovery server to access data from machines that are separated by a firewall.

Discovery The process of identifying the configuration items present in an IT environment. In the IBM CCMDB environment, the CCMDB Discovery Server / TADDM uses sensors, custom server templates, or DLA / IdML imports to establish its discovered CI representation.

Domain Discovery Server A discovery server that is used within a network of discovery servers, which contains data about a subset of the configuration items in the network, and passes data to an enterprise discovery server. The process of passing the data from the domain discovery server to the enterprise discovery server is also referred to as synchronization.

Enterprise Discovery Server A discovery server that aggregates the data collected by domain discovery servers and presents a view of all the collected configuration item information. If used or implemented in an environment, it is the component that is used by the Integration Composer to transfer data from the discovered CI space to the Actual CI space.

Operational Management Product (OMP) A product in the IT environment that supplies information about configuration items to the discovery environment of the CCMDB by providing a discovery library book. An OMP can also be the target of a launch in context operation.

Sensor A program that executes information from a target system and reads information to create CI information as part of the discovery process.

Windows Gateway A component in the CCMDB discovery environment to support the main discovery server to discover Windows based environments.

User Interface Layer

CCMDB Web Interface This is the Web interface used by different personas involved in Configuration or Change Management.

Domain Manager The Web Interface used for the CMDB Discovery Server / TADDM. In CCMDB V7.1, the Domain Manager also provides topology views. Used when enterprise discovery servers are set up in an environment.

Java Client The Java Interface used for the CMDB Discovery Server / TADDM. Sometimes also referred to as the product console.

Launch in Context A functionality provided by the base services layer of the process environment to launch in context from the CCMDB Web interface to either the discovered CI space in the discovery / TADDM environment or to an external operational management product (OMP). Launching into the TADDM environment requires a GUID as a reference identifier. You can launch into either the Domain Manager or Java Client in either a Domain or Enterprise Discovery Server. As part of the base services, the CCMDB V7.1 provides a Launch in Context application to define the launch points to different target systems and launch reference points.

Optional Components

Business Process Execution Language (BPEL) BPEL provides a language for the formal specification of business processes and business interaction protocols. If you have an environment hosting BPEL flows, for example, based on WebSphere Process Server technology, the CCMDB V7.1 predefined process flows (Change and Configuration Management) provide Web service interfaces in order to call specific activities from an external process engine hosting the BPEL workflow. The detailed task flow is still maintained in the CCMDB process environment.

IBM Tivoli Unified Process (ITUP) ITUP provides detailed documentation of IT Service Management processes based on industry best practices. It is a Web-based, free-downloadable tool that enables you to understand processes, the relationships between processes, and the roles and Tivoli tools involved in the implementation.

IBM Tivoli Unified Process Composer

(ITUPC) The ITUP composer is the tool used to create and publish the ITUP content and is based on the Rational Method composer technology. It can be used to create, document, and publish your organizations operational processes.

Integrated Solutions Console (ISC) The CCMDB Web Interface can be set up to be accessible from the Integrated Solutions Console, also referred to as ISC. The ISC is included with the WebSphere Application Server ND and serves as a common console for multiple IBM and Tivoli applications in specific. The ISC provides a consistent view and common interface tool for Web-based applications access and administration. Accessing the CCMDB through the ISC is optional; the default is to access the CCMDB through the CCMDB Web interface.

Log and Trace Analyzer A troubleshooting tool installed with CCMDB. It enables you to collect and correlate log files from different processes and machines.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 392. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *IBM Tivoli Application Dependency Discovery Manager Capabilities and Best Practices*, SG24-7519
- ▶ *IBM Tivoli CCMDB Implementation Recommendations*, SG24-7567

Online resources

These Web sites are also relevant as further information sources:

- ▶ The following documents can be found on the Web:
 - *Change and Configuration Management Database: Report Developer Guide*
 - *IBM Tivoli Application Dependency Discovery Manager Planning and Installation Guide*
 - *IBM Tivoli Integration Composer System Administrator's Guide*
 - *Planning and Installing Change and Configuration Management Database*

These documents can be found at:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp>

- ▶ Tivoli Product Documentation Information Center

<http://publib.boulder.ibm.com/tividd/td/tdprodlist.html>

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

- Access collection 173
- Active Directory 108
 - configuration 180
- Actual and Authorized CI space
 - required data 145
- Actual CI 83
 - Data Schema 272, 278, 284
 - Integration Adapters 240
 - only source 35
- Actual CI Filter 79
- Actual CI Mapping 271
- Actual CI space
 - computer systems 80
 - Discovered CIs 88
- Actual Configuration Items 78
- adopt and adapt 12
- Agent Controller configuration 180
- Altiris Inventory Solution 238
- Anchor Host 165
- Application Programming Interface (API) 147–149, 154, 157, 235
- application server 39, 52, 90–94, 104, 165, 169, 178, 182, 360, 368, 380
- attribute prioritization 43, 45
- attribute restrictions 129
- Authorized CI 35, 63, 75, 81–85, 87, 136, 145, 147, 152, 155, 158
 - change 158
 - CI auditing 88
 - data 81
 - data space 83–85, 145
 - data structure 85
 - instance 84
 - process environment 136
 - scheme 85
 - space 53, 82, 84, 88, 138, 158
- Authorized CI space 53, 55, 65, 158
- authorized configuration items 82

B

- Base Services 31, 52
- BIRT Report 335–336, 347, 357–358

- Administration 338
- CCMDB V7.1 examples 354, 374
- Designer 334, 336–337
- Designer application 337
- Designer component 337
 - detail 335
- Development 337
- Engine 336
- examples 354
- Localization 358
- Queue 358
- BIRT technology 52, 151
- BOC report 359, 361–363, 365, 367, 369
- BPEL 62
- budget 12
- bulkloader 33, 46, 49, 77
- Business Intelligence (BI) 373–374, 379–380
- Business Intelligence Reporting Tool (BIRT) 333

C

- Capability patterns 21
- CCMDB Base Services 31, 52
- CCMDB Component 164, 183, 185, 191
- CCMDB installation
 - program 184, 192–193, 195, 199, 202
 - time 110
- CCMDB middleware
 - component 184, 188–189
 - deployment 195
 - image 196
 - installation image 197
- CCMDB Middleware worksheet 174
- CCMDB overview 3
- CCMDB solution 29–31, 34, 83, 87–88, 94, 99–100, 105, 117, 120, 124–125, 147–148, 151, 158
 - discovered CI data space 149
- CCMDB system 53, 154–155
 - data formats 155
- CCMDB V7.1
 - Logical Architecture Overview 30
 - process layer 51
- CDM 40–41, 75, 87, 148, 158

- Centennial Discovery 238
- Change Management 3, 7, 11, 16, 18, 31, 52, 54–55, 57, 78, 82–83, 85
- Change Management PMP 55, 67–71
- Change Window 69
- chief information officer (CIO) 7
- CI
 - Synchronizing collections 136
- CI Audit 85
- CI data 34–35, 72, 79, 87, 145, 147–150, 173, 239–241, 244–245, 294, 299, 301, 304
 - Import 248
 - mapping file 290
 - schema 272–273
 - Service Management 79
 - space 76–77, 85–86
 - type 235
- CI Dependencies 78
- CI Filter 79
- CI instance 79, 82, 146, 240
- CI Lifecycle application 64
- CI object
 - structure 83
 - table 131
- CI type 35, 79–80, 87, 100, 146, 239–241, 244–245, 285
 - adapter 35
 - data 72, 240
 - data schema 253
 - import 240, 244, 259
 - information 73
 - Integration Adapters 240
 - mapping file 242
 - qualifier files 242
 - schema files 242
 - source data 240
 - target data 240, 259
- CIs 3
- Classification 57
- Classification Attributes 59
- COBIT 6, 14
- collation.properties file 116
- Collection Restrictions 130
- Collections application 136, 138
- common data model 39
- Common Process Manager Product 54
 - other PMPs leverage 55
- company sets 214
- compliance 8
- computer system 42, 44, 72, 77, 167, 249
 - Authorized CIs 82, 84
 - software components 77
- Concepts 1
- Conditional Expression Manager 130
- configuration item
 - different groups 173
 - end-to-end integration scenario 231
 - top-level CI classification 245
- configuration item (CI) 3–4, 31, 39, 62–63, 68, 95, 100, 129, 138, 145–149, 167, 173, 231–232, 235, 239, 297, 312–313, 315, 317, 319
- Configuration Management 3–4, 11, 15, 31, 33–34, 52–53, 55, 60, 62–64, 67, 78, 82, 85–86, 91–93, 101, 127, 134, 150, 235, 237–239
 - Key Performance Indicator 67
 - process requests 63
- Configuration Management Database
 - ITIL-aligned implementation 4
- Configuration Manager 31, 66
- connection parameter 238, 256, 263, 266
- Content Authoring overview 21
- Context application 62, 157, 307–309, 312, 317
 - new launch 322
- Continual Service Improvement 11
- Control CI 64
- Create New Mapping 268
- currency codes 214
- Custom Server Templates 49

D

- data layer 29–31, 34, 55, 75–76, 85, 89, 146, 151
 - implementation details 75
- data model 35, 40, 43, 72, 251, 294
- data restrictions 128
- data schema 237–238, 240–241, 244, 250, 258, 262
 - analysis warning window 261
 - analysis window 260–261, 282
 - data source 255
 - different name 255, 277
 - feature 238, 264, 274
 - field 254, 277
 - page 257, 264, 274
 - structure 240
 - window 254, 257, 260–261, 277
- Data Schema Analysis 261
- data source 33, 45–46, 52–53, 88, 145, 159, 231,

- 235, 237–238, 242, 248, 253, 263, 267, 295, 299
 - connection parameters 273
- database configuration application 52
- database user
 - ID 181–182, 223
- DB2 configuration 176
- DB2 Enterprise Server Edition 164
- Delivery processes 21
- deployment manager
 - node 92–94, 99
 - node name 178
 - profile name 178
- deployment plan 195
- deployment planning 173
- dialog box 256, 258–259, 261
- discovered CI data space
 - import data 149
- discovered CI space 148
 - Actual CI space 79
 - CI type 79
 - computer system 81
 - Process Manager Product 116
 - XML file 148
- discovered configuration items 77
- discovery environment 34, 47, 72, 90, 97, 100–101, 104–105, 107, 147, 149–150
- Discovery Library
 - Adapter 77, 147–149, 231
 - facility 149
 - File Store 148
 - Reader 149
- Discovery Sensor 147–148
- Domain Manager Topology Graph 50

E

- Eclipse Modeling Framework (EMF) 337
- eCMDB 47
- Embedded Security Service (ESS) 115
- endpoints application 140
- Enhanced Telecommunications Operations Map 14
- enhancements 35
- escalation services 53
- e-Sourcing Capability Model 6
- eTOM 14
- execute the mapping 269, 292
- External Report
 - Integration 370–371, 374, 380

- Server 371–373, 376, 380
- External Report Integration (ERI) 370–371, 374, 376–377
- external system 32, 53, 88, 145, 155
 - CCMDB data rather than regular bulkload data exchanges 149
 - federation setup 157
 - share information 148
 - user interface 157
- external systems application 141

F

- federated data 86, 150–151
 - discovered data 86
- federation of external data 86
- federation services 147, 150
- filter
 - actual CI 79
- fix errors 260
- FQDN 141
- fsn file 269
- full synchronization 47

G

- general ledger account 218
- General Unique Identifier (GUID) 51, 122, 137, 308, 313–316
- governance 8
- Graphics-Editor Framework (GEF) 337

H

- Health Insurance Portability and Accountability Act (HIPAA) 8
- Hewlett Packard (HP) 237–238
- high availability 103

I

- IBM and ITIL 13
- IBM CCMDB
 - Components 30
 - product 4
 - V7.1 solution 146
- IBM DB2
 - JDBC Driver 237, 279
- IBM Directory Server 108
- IBM HTTP Server
 - V6.1 Fix Pack 9 91, 170

- Version 6.1 195
 - WebSphere Application Server Plug-in 91
 - IBM HTTP Server (IHS) 91–92, 97
 - IBM Service Management (ISM) 3, 8
 - IBM Service Management strategy 4
 - IBM Solution Console (ISC) 32
 - IBM Tivoli
 - Application Dependency Discovery Manager 164, 168, 172, 183, 199, 238–239
 - Business Systems Manager 304
 - Change 3
 - Directory Admin Daemon V6.1 197
 - Directory Server configuration 179
 - Directory Server Instance V6.1 197
 - Directory Server V6.1 171
 - Directory Server Version 195
 - Directory Server Version 6.1 195
 - Integration Adapter 231, 238
 - Integration Composer 31, 72, 79, 164, 166, 173, 188–189, 224, 239, 242, 244, 254, 262
 - Integration Composer window 262, 264, 267–268
 - Monitoring 149, 304
 - Systems Management product 149
 - Unified Process 13–14, 17, 164
 - IBM Tivoli Unified Process 4
 - IBM WebSphere Process Server 62
 - IdML File 147–148
 - impact analysis 69
 - Import Data Schema 281
 - inactive organizations 212
 - Information Technology Infrastructure Library (ITIL) 3–4, 9
 - infrastructure complexity 7
 - installation
 - middleware 194
 - installation directory 169, 176–177, 180
 - installation plan 167
 - Integration Adapter 31, 34–35, 72–73, 96, 166, 232, 239, 241, 244–245
 - Integration Composer 31, 33–35, 47, 72, 91, 95–97, 100, 105, 136, 164, 166, 173, 188–189, 193, 201, 224, 231–235, 295, 299, 304
 - adapter mapping 238
 - data migration responsibilities 95
 - main concepts 231
 - main use case 72
 - New Data Schema 254
 - Select Action menu 289
 - software requirements 236
 - Integration Module 62, 140–141, 147, 156–157, 222
 - Integration Module Development document 157
 - Integration services 53
 - integration technologies 145
 - ISMRealm 109
 - ISO IEC 20000 6
 - IT Infrastructure Library 6
 - IT Service Management
 - success factors 12
 - ITIL Version 3 10
 - ITUP Composer 16, 24, 201
- ## J
- J2EE application 99
 - Java Virtual Machine 102
 - JDBC 100, 237
 - JDBC driver 235, 237, 263, 273, 279
 - JMS 154
 - Job Plan 52, 70–71
- ## L
- Language Pack 201
 - launch entry 307, 310, 312, 321–322, 331–332
 - launch in context 32, 157
 - launch in context operation 100, 116, 122
 - launch point 51, 307–309
 - launchpad interface 190–191, 193–194, 200
 - LDAP 108, 189
 - LDAP protocol 99
 - License Compliance Manager 238
 - life cycle 3
 - Lightweight Third Party Authentication (LTPA) 115
 - Linux 97, 187, 189–191
 - logical component overview 30
 - Logical Management Operation (LMO) 62, 147, 156–157
 - Logs 196
 - LTPA token 116, 122
- ## M
- MainControl i.collect 238
 - Managed Software Systems (MSS) 33
 - manual merge 43
 - mapping for CI Type Data 267
 - Mapping window 268–270, 286, 288, 295–296, 300

- maxadmin 112
- MAXGROUP table 113
- Maximo Business Object (MBO) 53, 129, 151
- Maximo database 188–189, 201, 223, 235, 237, 242, 244, 246, 368
 - above qualifier script 285
 - instance 181
 - Integration Composer repository 253, 262, 272, 277, 284
 - name 181
- Maximo Enterprise Adapter (MEA) 83, 88, 139, 147, 152
- MAXUSER table 113
- MAXVARS table 246, 267, 286
- MEA 53
- Method Content 19–22
 - minimal set 23
- method framework 20
- Microsoft Active Directory 108
- Microsoft SMS 234, 238
- Microsoft SQL Server 237
- middleware 188–191
 - installation 194
- middleware component 32, 164, 184–185, 191, 193–194, 198
- middleware configuration 202
- middleware installation 185, 194–197, 202
- middleware installer 91–92, 95, 175, 192–195
- Middleware worksheet 174
- Multiple server deployment 184
- Multiple Server Deployment Topology 188
- Multi-server 183
- MXServer 99, 198

N

- NetCool/Precision for IP Networks 238

O

- Object Restrictions 130
- OMP 156
- Open Process Automation Library Web Site 149
- Operational Management Product
 - configuration items 231
- Operational Management Product (OMP) 9, 33, 51, 61, 77, 231, 307
- operational model 31–32, 89, 97–98, 100
- Oracle 10g 237
- Oracle configuration 177

P

- people application 134
- physical component 29, 77, 91, 97
- pluggable sensors 43
- PMPs 54, 92–93, 134
- Post Implementation Review (PIR) 67
- post installation steps 206
- Preface xix
- PRM-IT 14
- Process Authoring Overview 23
- process environment 17, 34, 51, 53, 72–73, 100, 107, 110, 114, 116–118, 121, 125, 150–151
 - Actual CI applications 100
 - authorization process 125
 - configuration steps 139
 - Launch in Context application 122
 - Launch in Context call 119
 - remote data sources 150
 - Security group 142
- process layer 29–31, 33–34, 86–87, 100, 231–232, 235, 308
 - Actual CI configuration application 51
 - database 52–53, 72–73, 111, 113, 231
 - runtime 47
- Process Library 17
- Process Manager
 - implementation 31
 - product (PMP) 25, 31, 90, 108
- process manager
 - product (PMP) 54, 89
- Process Manager Product (PMP) 31–32, 52–54
- Process Manager Type 56
- Process Reference Model for IT 6, 14
- process request 55–58
- Process Requests Application 56
- process run time 91
- Publish Channels application 140

Q

- Query based Collections 48

R

- Rational Agent Controller (RAC) 164
- Rational Method Composer
 - fundamental principle 19
- Rational Method Composer (RMC) 18–20, 22
- Reconciliation 43
- Reconciliation Plug-In 46

- Redbooks Web site 392
 - Contact us xxiii
- reference classes 250
- Release Management 3
- Request for Change (RFC) 55, 59, 63, 65, 67, 75
- request page 348–349, 351, 358–359, 363, 366
 - user 350
- Rich Client Platform (RCP) 334
- Roles 17, 70
- roles 173
- run time environment 90

S

- Sarbanes-Oxley Act 8
- Scenarios 17
- Secure Token Service (STS) 108, 115
- security 173
- Security Enhancements 47
- security group 66, 125–129, 132–133, 139, 331, 364, 375–377
- security groups application 127
- security module 113
- security profile 132
- security users 132
- sensors 36
- service design 10
- Service Management 5–6, 8–9, 31, 33–34, 52, 78, 82, 84–86, 107
 - fundamental characteristics 6
 - strategy 4
- service operation 11
- service strategy 10
- service transition 11
- single server deployment 183
- single sign-on 47, 105, 120
- single-server 183
- Six Sigma 14
- Software Developer Kit (SDK) 337
- Solution Installer Command-Line Interface 222
- SQL Server configuration 177
- Start Center 52, 66, 128, 309, 312, 317, 319, 338–339, 357
 - Context application 309, 319
 - Report Administration application 338
- statistics 260
- STS authentication client 118
- STS instance 116, 118
- success factors 12

- synchronization 136
- System Configuration
 - module 110

T

- TADDM Access Collections 136
- TADDM CI and Relationship 300
- TADDM Console 165
- TADDM Database Server 165
- TADDM discovery
 - capability 88, 158
 - component 101, 145
 - environment 87, 105, 150
 - sensor 87, 158
 - server 35, 97, 100
 - service 88, 159
- TADDM Domain
 - Manager 123–124, 142–143, 313, 315
 - Manager Interface Miscellaneous Enhancement 48
 - Server 101
- TADDM Domain Manager Interface 48, 50
- TADDM environment 35, 72–73, 87, 96, 100, 104, 116–117, 120–122, 136, 138–139, 149, 305
 - CI collections 136
 - process run time environment 136
- TADDM installation
 - program 199
- TADDM product 41, 122, 313, 316
- TADDM server 33–34, 39, 47, 49, 96, 100, 116, 118–119, 125, 165, 204, 305, 312
 - authentication service client 118, 120
 - complete WebSphere Application Server 39
 - different configuration files 116
 - specific servlet 100
 - WebSphere discovery purposes 119
- TADDM system 33, 40, 42–43, 47, 100, 165
 - instance data 100
 - new classes 42
 - specific view 47
- target database 73, 233, 245
 - CI Type 251
 - discovery tool 238
 - target CI classification 251
- target DB 251–252, 254, 277, 279, 281
 - CI Types 251
- TCM 156
- Tivoli Application Dependency Discovery Manager

31–35, 73, 77, 86–88, 90–91, 96–97, 164–166, 168,
172, 180, 183, 188–189, 191, 193, 199, 231,
238–239, 307–308, 312–314
Tivoli Configuration Manager (TCM) 61, 149, 234,
237–238, 307–308
Tivoli Directory Integrator (TDI) 150, 155
Tivoli Provisioning Manager (TPM) 61, 307
TLCM 238
TLCMz 238
tool mentors 17
topologies 183
topology file 195
topology overview 188
TPM 156

U

Update CI
 Process 66
 process request 68
 Request 60
user 107, 173, 348–349, 359, 361, 363, 370
user interface 32, 34, 50, 52, 83–84, 87, 156–157,
237
 Authorized CIs 83
 behavior 32
 context 156
 extended attributes 87
 look 52
user interface layer 32
user security 212
USERGROUP table 113

V

validating the configuration 121
verify and audit ci 64
Virtual Member Manager (VMM) 94–95, 99, 108
 directory server flow 108
VMM 99

W

Web Service Description Language (WSDL) 154
WebSphere Admin
 group entries 114
 SSO Domain 121
Websphere Admin 91–93, 114, 118, 120, 212
WebSphere Application Server
 Admin 109

instance 100
Plugin 91
security 107, 115
security model 115–116
WebSphere cell
 physical systems 102
Websphere Cell 52, 92, 94–95, 99
Work Products 17
Work Type 221

X

XML file 42, 148–149, 155, 304, 335



Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning

(0.5" spine)
0.475" <-> 0.875"
250 <-> 459 pages



Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning

Understand the CCMDB architecture

Plan for installation

Get started using Tivoli CCMDB

The IBM Tivoli Change and Configuration Management Database (CCMDB) is one of the key components of the IBM Service Management (ISM) strategy. It is the foundation for automating and supporting Change and Configuration Management processes as described by the Information Technology Infrastructure Library (ITIL).

This IBM Redbooks publication provides information that can be used by clients, Business Partners, or IBM field personnel who are looking to engage in an effort to implement Change and Configuration Management processes in an enterprise environment utilizing the IBM Tivoli Change and Configuration Management Database product. This book is divided into four parts:

- ▶ Concepts
- ▶ Components
- ▶ Planning and Installation
- ▶ Getting Started

A companion book, *IBM Tivoli CCMDB Solution Design Guide*, SG24-7567, provides more advanced details about the underlying components of the product and utilizing the product to support robust IT processes, such as Change and Configuration Management. The companion book focuses on the details of the data model, process engine, and the Change and Configuration Management Process Management Programs (PMPs). It provides details on how these can be extended and customized to meet client requirements.

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