

IBM Tivoli Change and Configuration Management Database (CCMDB) V7.2.1 Implementation Guide

Understand the CCMDB architecture and its implementation

Experiment with the CCMDB V7.2 and V7.2.1 new features

Get started using Tivoli CCMDB



Yatresh K Agarwal, Bill Cary
Sandy Cash, Leandro Cassa
Brian Demartini, Chris Duplantis
Alisson Negrisolo de Godoi
Douglas Barranqueiros Gomes, Vasfi Gucer
Murat Kipel, Antonio Orlando Neto
Caio Saad, Ghufraan Shah
Peterson Domingos Tamarindo
Krishnan Venkitasubramanian



International Technical Support Organization

**IBM Tivoli Change and Configuration Management
Database (CCMDB) V7.2.1 Implementation Guide**

December 2010

Archived

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

First Edition (December 2010)

This edition applies to IBM Tivoli Change and Configuration Management Database (CCMDB) Version 7.2.1.

© Copyright International Business Machines Corporation 2010. All rights reserved.

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

Contents

Notices	ix
Trademarks	x
Preface	xi
The team who wrote this book	xi
Now you can become a published author, too!	xv
Comments welcome	xv
Stay connected to IBM Redbooks publications	xvi
Part 1. Introduction to CCMDB V7.2.1 and its components	1
Chapter 1. IBM Tivoli Change and Configuration Management Database overview and architecture	3
1.1 Overview	4
1.1.1 CCMDB features	4
1.1.2 Benefits of ITIL compliant Configuration Management Database	5
1.2 Architecture overview	6
1.2.1 CCMDB Data Layer and Process Layer	6
1.2.2 User interface layer	8
1.2.3 CCMDB discovery server: TADDM	8
1.2.4 Tivoli's process automation engine	11
1.2.5 CCMDB Process Manager Products	13
1.2.6 Integration Composer and Integration Adapter for TADDM	19
1.3 What is new with CCMDB V7.2 and V7.2.1	20
1.3.1 New features	22
1.3.2 Additional resources	32
Part 2. Process Layer	33
Chapter 2. Process flow technology	35
2.1 Change Management process	37
2.2 Technology components	39
2.2.1 Components overview	41
2.2.2 Workflows	41
2.2.3 Action and action groups	45
2.2.4 Work plans and work orders	46
2.2.5 Job plans	48
2.3 An end-to-end example	55
2.3.1 Request for a Change	56

2.3.2 Change Request process	56
2.3.3 Configuration process	78
2.4 Summary	81
Chapter 3. Process Managers	83
3.1 Overview of Process Managers	84
3.1.1 Process Manager role	85
3.1.2 How Process Managers work	86
3.1.3 Job plan	87
3.2 Configuration Management Process Manager	94
3.2.1 Relationships	95
3.2.2 Configuration Management roles	96
3.2.3 Configuration item lifecycle	102
3.2.4 CI lifecycle management	103
3.2.5 Discovering configuration items	109
3.2.6 Authorized configuration items	110
3.2.7 Controlling and updating the CI process	117
3.2.8 Verifying and auditing the CI process	127
3.2.9 Auditing	133
3.2.10 Interaction with other processes	162
3.2.11 CI baselines	163
3.3 Change Management Process Manager	167
3.3.1 Change Management overview	167
3.3.2 Change process step-by-step within CCMDB	177
3.3.3 Functions applicable to Change Management	209
3.3.4 Accepting or rejecting a Request for Change	210
3.3.5 Assessing a Change	212
3.3.6 Change impact analysis	215
3.3.7 Change Management Schedule	224
3.3.8 Change Window	238
3.3.9 Tracking the progress of a Change	242
3.3.10 Interaction with other processes	246
3.4 Release Process Manager	253
3.4.1 IBM Tivoli Release Process Manager overview	254
3.4.2 Phases of Release process	257
3.4.3 The Change and Release process interaction	258
3.4.4 The process flow in Release Manager V7.1.2	260
Chapter 4. Customizing Change in Tivoli Change and Configuration Management Database	301
4.1 Process role identification	303
4.1.1 Individuals and roles	303
4.1.2 Process role considerations	304

4.2 Change Management workflows	305
4.2.1 The PMCHGITLV3 workflow	305
4.2.2 Determining Change priority	308
4.2.3 Modifying the PMCHGCALPR workflow	309
4.3 Standardizing Changes with job plans	313
4.3.1 Identifying common activities and tasks	314
4.3.2 Creating the job plan	315
4.4 Configuration item considerations for Changes	327
4.4.1 Change Window calendars and blackout periods	328
4.4.2 Specifying a CI Owner	337
Part 3. Data Layer	339
Chapter 5. Configuration item hierarchy	341
5.1 CCMDB Data Layer	342
5.2 Thinking about your business processes	343
5.3 Discovered configuration items	343
5.4 Actual configuration items	345
5.4.1 Actual CI filter	345
5.4.2 Example of CI types	345
5.5 Authorized configuration items	348
5.5.1 Creating authorized configuration items	349
5.5.2 Other data artifacts in the CCMDB Data Layer	351
5.6 CIs in the CDM	351
5.7 How CIs are represented in the CCMDB	352
5.7.1 Classification objects and relationship objects	353
5.7.2 CDM flags used in CCMDB	353
5.8 Working with CIs in CCMDB	357
Chapter 6. CI space development	359
6.1 What a CI space is	360
6.2 Designing your CI space with Deployer's Workbench	361
6.2.1 Setting up a project with the Deployer's Workbench	361
6.2.2 Configuring your own set of authorized CI classifications	366
6.2.3 Defining a promotion scope	372
6.3 Setting up Tivoli Integration Composer according to the Deployer's Workbench	377
6.4 Promotion at a glance	378
Chapter 7. Importing configuration items	383
7.1 Integration Composer overview	384
7.2 Import process	385
7.3 Installing Integration Composer	386
7.4 Configuring Integration Composer	397

7.4.1 Mapping data sources	401
7.4.2 Importing CI types	407
7.4.3 Importing configuration items	409
7.4.4 Logging	415
7.5 References	415
Chapter 8. CI promotion	417
8.1 Various states of CIs	418
8.2 Promoting actual CIs to authorized CIs using authorized CI hierarchies	419
8.2.1 Determining authorized CI classifications	420
8.2.2 Determining authorized CI attributes	421
8.2.3 Determining authorized CI relationships	422
8.2.4 Putting authorized CI classifications into hierarchies	423
8.2.5 Promoting CIs from actual CIs to authorized CIs	423
8.2.6 Promotion rules	424
8.2.7 Best practices for using business applications	425
8.2.8 Step by step procedure to promote CIs	426
8.3 Summary	441
Chapter 9. Implementing federation	443
9.1 Introduction	444
9.1.1 Scope	445
9.1.2 Vendor related technology	445
9.2 Federation scenario	446
9.3 Setting up federation at the database layer	449
9.3.1 Catalog node and database	451
9.3.2 Creating a wrapper	459
9.3.3 Registering the server	462
9.3.4 Creating a user mapping	465
9.3.5 Creating a nickname	467
9.4 Creating a Maximo Business Object (MBO)	470
9.5 Generating the object in the CCMDB database	474
9.6 Defining a relationship	478
9.7 Creating a new application	483
9.8 Using the new application	490
9.9 Summary	492
Part 4. Miscellaneous CCMDB topics	493
Chapter 10. Virtualization support in CCMDB	495
10.1 Virtualization overview	496
10.2 Virtualization scenarios	497
10.2.1 First proposed scenario	498
10.2.2 Second proposed scenario	499

10.2.3 Third proposed scenario	500
10.2.4 Fourth proposed scenario	502
10.3 Information Technology Infrastructure Library (ITIL) V3 workflow	504
Chapter 11. Launch In Context	507
11.1 Purpose of new Launch In Context	508
11.2 Components of the new LIC technology	509
11.3 Naming and Reconciliation Service	510
11.3.1 Reconciliation functions	511
11.3.2 Context Menu Services	511
11.4 Deployment and configuration of DIS/CMS	512
11.4.1 Deployment components	512
11.4.2 Options for creating the CMS and DIS tables	513
11.4.3 Deployment scenarios	514
11.5 CMS Registry Loader externals	517
11.5.1 Syntax	518
11.5.2 Supported actions	518
11.6 LIC using TADDM	518
Chapter 12. CCMDB performance tuning	523
12.1 Performance tuning	524
12.2 The network	525
12.2.1 Multi-device network accelerators	526
12.2.2 Single-device network accelerators	526
12.2.3 Application Virtualization and Delivery Software	527
12.3 Hardware	528
12.4 Operating systems	529
12.5 Services	530
12.5.1 HTTP Servers	530
12.5.2 Application Servers	530
12.5.3 Java Virtual Machines (JVMs)	531
12.5.4 Databases	532
12.5.5 BIRT reporting	533
12.6 IBM Tivoli Change and Configuration Management Database (CCMDB) tuning	533
12.6.1 Properties	533
12.6.2 The mxe.crontask.donotrun property	535
12.6.3 Implementation choices	536
Appendix A. A primer on Deployer's Workbench	541
Deployer's Workbench	542
CI space design tool	542
Partial-sandbox environment	543
Deployer's Workbench user interface	543

Creating a new project in the Deployer's Workbench	551
Synchronizing the CISPACe with the CCMDb	556
Default authorized CI space definitions	560
Editing the authorized classification tree	570
Editing the promotion scopes	573
Editing the relationship rules	576
Data validation	579
Related publications	581
IBM Redbooks publications	581
Online resources	581
How to get Redbooks publications	582
Help from IBM	582
Index	583

Notices

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

IBM might 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 might be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right might 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 might 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 might 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 might 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM might 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 might 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 might 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:

AIX®	IBM®	Redbooks (logo)  ®
DB2®	Informix®	Service Request Manager®
Domino®	InfoSphere™	Tivoli®
DRDA®	Lotus®	WebSphere®
Foundations™	Maximo®	
IBM Process Reference Model for IT™	Rational®	
	Redbooks®	

The following terms are trademarks of other companies:

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

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

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

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation 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 might 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) V 7.2.1 product. It covers the new features available with CCMDB V7.2 and CCMDB V7.2.1, as well as details about the underlying components of the product and utilizing the product to support robust IT processes such as Change and Configuration Management. It also focuses on the details of the data model, process engine, and the Change and Configuration Management Process Management Programs (PMPs).

This book provides a reference for IT Specialists and IT Architects working with the CCMDB V7.2.1 product.

The team who wrote this book

This book was produced by a team of specialists from around the world.

Yatresh K Agarwal is a Software Specialist at India Software Labs. He holds a Bachelor of Technology degree in Computer Science from NITK Suratkal, India. He is working for the Tivoli Lab Services team on Security and Service Management products and is ITIL certified with expertise in service management and ITIL processes. He has implemented Tivoli products for various customers in India and is involved in both post sales and pre sales of Tivoli products.

Bill Cary is a graduate of Hawaii Pacific College and joined Project Software in 1996 as an Infrastructure Engineer. In 2000, the company was renamed MRO Software and in 2006, MRO software was acquired by IBM. During his tenure, Bill has been a lead engineer working with customers and teams world wide to resolve performance and stability issues. In his current role, Bill is the senior

support engineer in the IBM Maximo® Support segment of Tivoli and works with development and performance tiger teams to proactively address infrastructure issues and address customer implementation problems.

Sandy Cash is a Certified IT Architect with IBM Global Services, specializing in Operations and Infrastructure. He works primarily in the fields of IT Service Management and Systems Management and has extensive experience in Event Management, Change Management, Service Execution and Orchestration, and Service Request Management. Sandy has been with IBM since 1999.

Leandro Cassa was graduated from the Senac University with a Bachelor of Computer Science degree and has also worked for three years in the Senac institution in web development for distance learning. He started with IBM in 2007 working on Tivoli IBM Service Request Manager® development. Then he joined the ITD team where he designed and developed a large customization for the SPMS team in Canada. Leandro currently works as the CCMDB L3 lead in IBM Brazil. He has a lot of experience designing, developing, and troubleshooting CCMDB and Tivoli Service Request Manager based solutions.

Brian Demartini is an IBM Certified Consulting IT Specialist working in the IBM Tivoli Automation Technical Sales area. He joined IBM in 2005 and has performed proof of concepts and managed customer deployments of Tivoli provisioning products (Tivoli Configuration Manager and Tivoli Provisioning Manager) as well as CCMDB and TADDM. He also specializes in the Tivoli Monitoring of SOA environments with IBM Tivoli Monitoring, as well as ITCAM for Application Diagnostics, ITCAM for Transactions and ITCAM for SOA.

Chris Duplantis is a Service Management Solution Architect at IBM. He has over 12 years of experience developing technical architectures and providing technical support across the Tivoli Software product suite. He joined the Tivoli Software Technical Professional organization in the US Federal Government region in 1998. He is certified as an ITIL Expert in IT Service Management. He holds a degrees in Economics from the University of Louisiana, and Computer Information Systems from Tulane University.

Alisson Negrisol de Godoi is an IT Specialist for IBM Brazil. He worked in the Maximo Brazil Lab developing solutions for Maximo 6x and 7x versions, both for Maximo Core and Industry Solutions. Later, as a member of the Brazil Tivoli Sales and Services team, he implemented and delivered solutions with Tivoli Service Request Management and Tivoli Change and Configuration Management products, focused on achieving IT governance aligned with ITIL methodologies. Some of these solutions have been delivered to large banks in Brazil. Also, he participated in some of the previous Redbooks publications residencies, such as “Tivoli Asset Management” and “Tivoli Asset Management for IT” Certification Guides.

Douglas Barranqueiros Gomes has been providing deployment and support on Tivoli tools (IBM Tivoli Change and Configuration Management Database [CCMDB], IBM Tivoli Application Dependency Discovery Manager [TADDM], IBM Tivoli Business Service Manager [TBSM] and IBM Tivoli Monitoring [ITM]) on Global Cross Services Technology Integration (GCSTI) on Global Delivery Center (GDC) Brazil in the Enterprise Automation for Architecture and Tools Team. He holds a degree in Computer Science (1996) from Carioca University in City of Rio de Janeiro/RJ, Brazil and Master Business Administrator (MBA) in Business Management from Getulio Vargas Foundation (FGV) in City of Campinas/SP, Brazil.

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

Murat Kipel is an IT Specialist at IBM Global Services in Istanbul, Turkey. He has been working with IBM since 2005. His areas of expertise are IBM Tivoli Service management products including CCMDB, Tivoli Service Request Manager, Maximo for Service Providers and TADDM. He is an IBM Certified Deployment Professional - Tivoli Application Dependency and Discovery Manager V7.1. and an IBM Certified Deployment Professional - Tivoli Change and Configuration Management Database V7.1.1.

Antonio Orlando Neto is an IT Specialist with IBM in Brazil, supporting Global accounts. He is an IBM Certified Deployment Professional - Tivoli Application Dependency and Discovery Manager V7.1, and ITIL® V2 Certified. He is experienced with several Tivoli tools such as TADDM and IBM Tivoli Monitoring. He holds a degree in Social Science from University of Campinas (UNICAMP), Brazil.

Caio Elias Saad is a Solutions Architect for SoftWrench Solutions, in the USA. Caio has approximately 4 years of experience primarily in Maximo V6 and V7 gained during his tenure at IBM in Brazil. Caio has a degree in Computer Science from the State University of Campinas - UNICAMP, he holds a number of Maximo/TAMIT certifications, and has co-authored other three Redbooks publications. His areas of expertise include solution design, integration, and implementations based on industry best practices for clients around the world, and he has a strong knowledge of Maximo Mobile.

Ghufran Shah is an IBM Certified Advanced Deployment Professional in Enterprise, Provisioning, and Business Application Management Solutions. He has 15 years of experience in Systems Development and Enterprise Systems

Management, with a degree in Computer Science. His areas of expertise include Tivoli Systems Management Architecture, Implementation, and Tivoli Training, together with Business Process Improvement. He has written extensively about event management, monitoring, and business systems management integration and has taught IBM Tivoli courses worldwide. He is currently at TeamSwift Solutions, a trusted advisor for IT Service Management Solutions with a focus on automation, service provisioning, monitoring, and virtualization.

Peterson Domingos Tamarindo is a Tivoli Maximo Specialist working for IBM Brazil. He has a degree in Computer Science and has experience on IBM Service Management products, and IBM Maximo Service Provider. Peterson is an IBM Certified Deployment Professional and ITIL IBM Foundations™ Certified IT Specialist.

Krishnan Venkitasubramanian is a Systems Management Consultant with the Tivoli Lab Services team in IBM India Software Labs. He holds a Bachelor of Engineering degree in Computer Science from Anna University, India. He works in a customer facing role and is involved with the design and implementation of various Tivoli products, including Tivoli Service Management and Security related products. He is involved in both the pre sales and post sales activities of these products.

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

Tamikia Barrow, Emma Jacobs, Stephen Smith, Leslie Parham, Margaret Ticknor, Wade Wallace, Yvonne Lyon
International Technical Support Organization

Melody Bienfang, Srinivas Chowdhury, Jim Collins, Scott Dickerson, Bob Dunyak, Ann Marie Fred, Nancy Gore, Brian Jeffrey, Praveen Hirsave, Joseph Niedemeyer, Pierre Padovani, Charles Rich, Adriana Lopez Russell, Susan Wenzinger, Edward Whitehead
IBM USA

Paul O'mahoney
IBM UK

Ivan Henrique Nestorenko, Marcus Vinicius De Arruda Santos
IBM Brazil

Thanks also to the authors of the *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565, published in May 2008: Bart Jacob, Michael Brokmann, Scott Dickerson, Douglas Barranqueiros Gomes, Rainer Hoppen, Arsalan Lodhi, Kapil Madaan, Annelie Meels-Kurz, Rosemeire Oikawa, Tadeu Stellet Teixeira

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

ibm.com/redbooks

- Send your comments in an email to:

redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks publications

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



Part 1

Introduction to CCMDB V7.2.1 and its components

In this part of the book, we introduce CCMDB V7.2.1 and its components. We provide more details about these components in subsequent parts of the book.

IBM Tivoli Change and Configuration Management Database overview and architecture

In this chapter, we introduce the IBM Tivoli Change and Configuration Management Database (CCMDB) V 7.2.1 and discuss the architecture of the product. For more information about CCMDB product features, see the CCMDB 7.2.1 InfoCenter, Product Overview web page at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/index.jsp?topic=/com.ibm.ccmdb.doc_721/overview/c_ccmdb_overview.html.

We cover the following topics:

- ▶ 1.1, “Overview” on page 4
- ▶ 1.2, “Architecture overview” on page 6
- ▶ 1.3, “What is new with CCMDB V7.2 and V7.2.1” on page 20
- ▶ 1.3.2, “Additional resources” on page 32

1.1 Overview

In this topic we present an overview of the CCMDB features and benefits.

1.1.1 CCMDB features

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 such as Change and Configuration or Release Management. These process solutions provide best practice implementations of core ITIL processes.

The CCMDB provides a multi-customer and service provider support infrastructure as well as a set of foundation services used by various ISM process solutions and includes the Change and Configuration 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 lifecycle. 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 various characteristics, a classification or type, attributes that describe the CI depending on its classification, and relationships that describe how a CI is related to other configuration items.

CCMDB defines configuration items (CIs) as managed components of an IT Service. Configuration records within a CMDB contain information about the CI, and are maintained through their lifecycles. Because CIs are managed components, they come under the control of the Change Management process.

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

1.1.2 Benefits of ITIL compliant Configuration Management Database

Before delving into the details of CCMDB, let us discuss the benefits of having an ITIL compliant Configuration Management Database.

Quicker resolution of problems: When you have a repository that shows the relationships between CIs, Problem Management, and IT Support groups, you will be able to resolve problems more efficiently and effectively. These groups are able to perform root-cause analysis activities (an example might be various problems affecting the same CI).

Enhanced Change management and control: By using the information in CMDB, it is possible to reduce the risk of making a Change as a result of the increased configuration knowledge, thus reducing the number of failed Changes and also decreasing the number re-scheduled Changes. Change Approvers can approve the Changes with more confidence and unauthorized Changes can be detected easily. This will reduce the risk on the production environment, and decrease the probability of breaking Service Level Agreements.

More visible software Changes: Because the tool can track the configuration Changes, it is possible to reconcile these Changes against Change Requests identifying the Changes that occurred outside of the Change process. IT management can investigate these Changes and take steps and make policy Changes to prevent the occurrence of these Changes that occurred outside of the Change process in the future.

Controlling CIs: Configuration Management can help the IT management to know what assets are under IT control, who is responsible for these assets, and whether the actual inventory matches the official inventory.

Reduction in risk of audit non-compliance: Configuration Management maintains an inventory of all items of software within an IT infrastructure, so by using configuration audits (comparing actual CIs with authorized CIs), it is possible to identify software that is not authorized and has not been paid for.

Assisting disaster recovery: A CMDB repository can facilitate the restoration of an IT service in the event of a disaster, by identifying the required CIs to restore this service, including other important information about these CIs such as their location and inter-relationships with other CIs.

Cost reduction in Configuration Management process: Using automated processes to update the CMDB minimizes inconsistent data gathering, reduces update errors, and hence minimizes the cost of Configuration Management.

1.2 Architecture overview

In this section we explain the architecture of the overall CCMDB solution. The CCMDB solution is composed of multiple physical and logical components. The main components are the process runtime and the discovery environment.

1.2.1 CCMDB Data Layer and Process Layer

This topic includes the two major layers of the CCMDB solution: the *Data Layer* and the *Process Layer*. Both of those layers are themselves composed of multiple logical and physical components that are utilized in the overall solution layout.

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

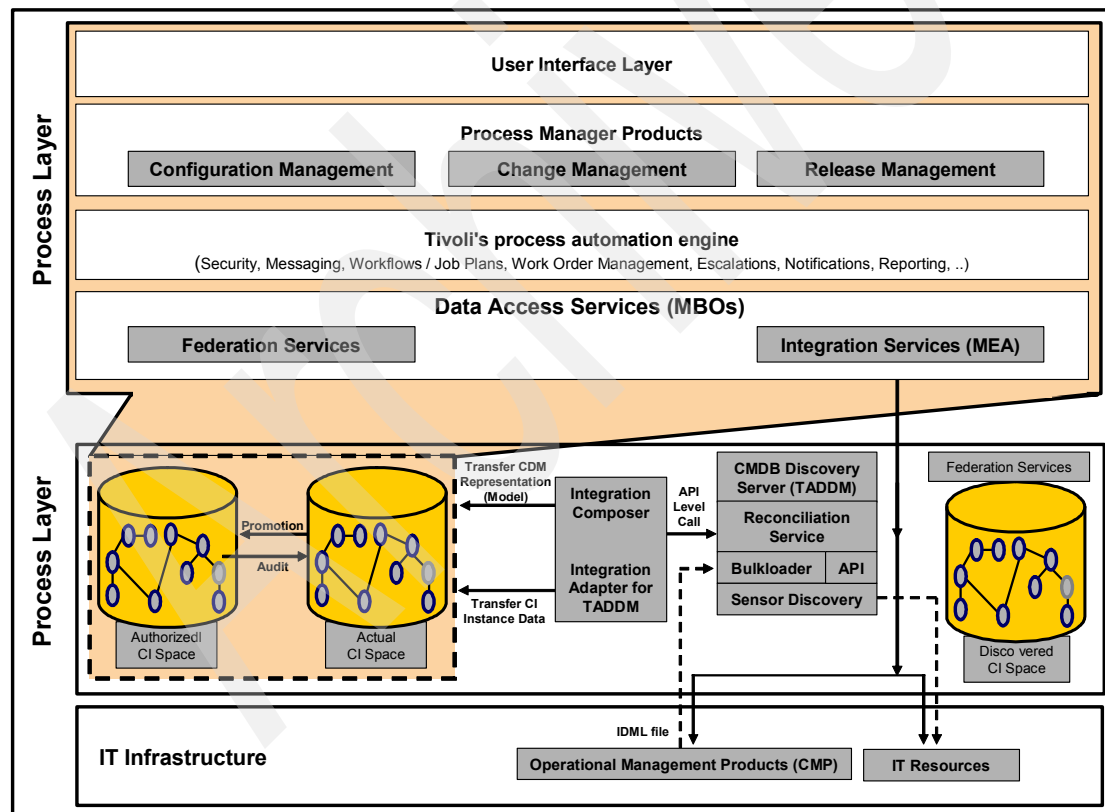


Figure 1-1 Logical architecture overview for CCMDB

The Data Layer, which is the base of the IBM Service Management architecture, 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 composed of various physical databases that exchange data with respect to configuration items. The Data Layer is physically composed 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 such as Change and Configuration Management.

The Process Layer consists of a foundation that provides runtime services for Process Manager Products (PMPs) that we refer to as *Tivoli's process automation engine*. 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 PMPs that utilize the process runtime environment for specific implementations of Service Management processes.

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

- ▶ CCMDB Discovery Server, also known as the Tivoli Application Dependency and Discovery Manager (TADDM)
- ▶ Tivoli's process automation engine
- ▶ The PMPs provided with the CCMDB solution:
 - The Configuration Manager PMP
 - The Change Management PMP
 - The Common Configuration Item PMP
 - The Common PMP, as a foundation layer or service for all other PMPs
- ▶ The Integration Composer and the Integration Adapter for TADDM

1.2.2 User interface layer

There are two major user interfaces provided by the overall CCMDB solution: 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 PMPs. 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 various 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 various IBM products, including many IBM Tivoli solutions. If you want to work with CCMDB applications from the ISC console, you can do so.

1.2.3 CCMDB discovery server: TADDM

The TADDM is the discovery server in the CCMDB environment. It provides discovery aggregation and reconciliation functionality, as well as a topology view into the discovered data space, also referred to as the discovered CI space.

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

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

<http://www.redbooks.ibm.com/abstracts/sg247519.html>.

Common Data Model and the discovered data: CCMDB is designed to be used with a discovery tool, such as TADDM, which discovers configuration items (CIs), and the relationships between them. Without the CIs and their relationships, the capabilities of the solution will be limited, because the relationships are used to determine the potential impact of the Change.

CIs can also be acquired from other discovery tools, but will have to be associated with relationships in order to be useful when determining outage impacts. This requirement also applies to CIs imported through other sources.

When CIs are entered into CCMDB, they are designed to be entered within the framework of a Common Data Model (CDM). This is a way of structuring the data to provide useful information as opposed to a vast number of data points, which might or might not be useful.

The problem with discovery tools is that while they can discover lots of data, this data might be irrelevant. The purpose of the CDM is to provide a structure built around a common need; for example, focusing on the major services that an IT department supports, such as a financials system, ERP, payments, or processing system. If these types of services are impacted, then this is critical to the operation of that organization. This is where successful Change Management can add the most value to an organization. Therefore the CDM can be structured around these critical services; the very top level of this CDM hierarchy can just name these services, and then the CIs that are associated to this top level service only through their relationships can be linked to the hierarchy. When Changes are made to the CIs, the impact on these critical services can be understood, and appropriate actions can be undertaken.

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 must 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 to extend reporting views of the discovered CI space with data federated from external data sources. Be aware that data federated from within the TADDM system is not transferred by the Integration Composer to the Process Layer of the CCMDB. 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.

The TADDM Server is separate from Tivoli's process automation engine, which provides the basis for the Service Management processes such as 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 Tivoli's process automation engine, it is best to install TADDM on a separate machine. 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.

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 mapping between the discovered CI space and the actual CI space.

Adapters: There is a TADDM adapter available for previous versions to transfer data into the CCMDB database. Note that this previous adapter transfers data into Asset representations inside the asset management database. The Integration Adapter for TADDM in the CCMDB solution populates CI structures and instance data rather than Asset Data. The TADDM integration adapter within the CCMDB 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.

1.2.4 Tivoli's process automation engine

Tivoli's process automation engine in the CCMDB solution is the component that provides the platform upon which PMPs such as Change Management or Release Management run and provide their specific functions. It is the foundation layer of the IBM Service Management Process Layer with technical functions such as these:

- ▶ 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 utilize the common process runtime 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 such as job plans, workflows, activities, and activity groups in order to automate processes and guide various personas through a process.
- ▶ A way to send a notification to a person or group. This can either be an email or a message in the Start Center of specific user roles.

- ▶ A reporting engine based on Eclipse Business Intelligence and Reporting Tools (BIRT) technology. The reporting engine is by default installed on every application server in the IBM WebSphere® Cell of application servers if you have a multi-server environment, but keep in mind that the reporting runtime is running in its own Java Virtual Machine. Reports are administered from the CCMDB Web User Interface. Report definitions are stored in the Process Layer database. Report security is based on the security model provided by Tivoli's process automation engine layer.
- ▶ Integrated tooling to customize data, application layouts (user interfaces), or process definitions. Tivoli's process automation engine includes integrated tooling to add, update, or modify new objects to the Process Layer database. From a CCMDB 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 Tivoli's process automation engine to create new database objects that are federated from various data sources. In addition, there is an Application Designer application that allows you to set users of the system to have views into the system other than those 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 1-1 on page 6, 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 such as 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 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, Tivoli's process automation engine (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.

In summary, Tivoli's process automation engine is the core runtime process environment that provides all the facilities that PMPs such as Change and Configuration Management utilization.

1.2.5 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 CCMDB, besides these two PMPs delivered with the CCMDB. The Common PMP does not provide specific functionality in terms of reflecting a Service Management process, but rather provides a common functionality that is used by all other implementations of PMP products.

IBM provides PMPs 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.

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

Common PMP

The 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 Change and Configuration Management PMPs build on top of the applications and functions that it provides.

If compared to Tivoli's process automation engine, the Common PMP provides application functionality rather than technology services. The most prevalent application that all other PMPs utilize from the Common PMP is the process request application. A process request forms the general interaction model into and between PMPs. 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*, allowing a controlled process to be initiated to update the CCMDB database and thus make sure that 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 another 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.

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) such as Tivoli Provisioning Manager or Tivoli Configuration Manager in order to initiate an action from within a step in a process into the OMP. 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.

Configuration Management PMP

The Configuration Process Manager Product is the part of the CCMDB 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 might 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 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.

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).

Defining lifecycles 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 requires a Change Request (RFC).

The Configuration Management PMP provides default lifecycles according to ITIL, but you can provide your own lifecycle definitions within the CI Lifecycle application.

The Configuration Management PMP provides two process implementations in CCMDB, *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.

The foregoing definitions define steps in order to move users with the appropriate role through formal process steps for the audit and update CI 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 ought to look like (authorized) versus what it really 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.

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 various types of Configuration Management users.

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 various 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).

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

The Change Management PMP provides various 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.

There are various 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. Various 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 can occur, for example, when you define a Change, where you can 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. After all the consequences are documented, the subsequent steps of the process will use this information. For example, authorizations 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 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, suppose 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.

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* such as Change Manager, Change Administrator, Change Owner, Change Analyst, Change Approver, Change Implementer, or Change Requester. In addition, predefined Start Centers for the various roles are provided; these Start Centers utilize 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.

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. Here are three examples of plans provided by the Change Management PMP for end-to-end process flows:

- ▶ *jStandard Change Job Plan* to represent a standard Change
- ▶ *Pre-Approved Job Plan* to represent a pre-approved Change
- ▶ *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 various 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, Authorization (or 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 can find specific tasks related to those phases.

1.2.6 Integration Composer and Integration Adapter for TADDM

Integration Composer (ITIC), formerly known as Fusion, is a component or tool that is used 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 environment, we focus on loading CI data rather than asset data into the system. Keep in mind that CI data can be linked to asset data through appropriate relationships in the Process Layer database.

Inside the CCMDB 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 models 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 (such as 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. This is the real data that you can see on the TADDM console. This is your Linux server *ABC* or the Business service *ITSO Application*.

The 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 another system from TADDM and the process environment of the TADDM. It just needs appropriately defined connections to the TADDM environment and the system hosting the Process Layer database of the TADDM.

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 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 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. Various adapter files are provided for exchanging the data model or CI type information as well as the instance data.

1.3 What is new with CCMDB V7.2 and V7.2.1

IBM has introduced important new features with CCMDB versions 7.2 and 7.2.1. Figure 1-3 lists the new features that are available with Version 7.2 and V7.2.1. It also shows in which version these features were introduced (shown with “X”).

To learn more about the new features available with CCMDB V 7.2.1, see the “What is new in this release” pages in the CCMDB InfoCenter:

For CCMDB V7.2:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.ccmdb.doc/overview/c_ccmdb_whatsnew.html

For CCMDB V7.2.1:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/index.jsp?topic=/com.ibm.ccmdb.doc_721/overview/c_ccmdb_whatsnew.html

Tip: The CCMDB 7.2.1 InfoCenter has excellent recorded videos about the new features of CCMDB 7.2.1, which you can access at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.ccmdb.doc_721/overview/c_link_demos.html

Table 1-1 CCMDB V7.2 and V7.2.1 new features

New features	Introduced in CCMDB 7.2	Introduced in CCMDB 7.2.1
Configuration item topology view	X	
CI reconciliation history view and user action enhancements	X	
CI promotion process improvements and Update promotion	X	
Workplan Map widget	X	
End-to-end Change Management process enhancements	X	
Automated impact analysis and Impact analysis topology view	X	
Progress map	X	
Restricted Change schedules and blackout periods	X	
Conflict detection	X	
CI baselines		X
CI-asset linking		X
Simplified Change process		X
Graphical scheduling GUI		X
Importing discovered CIs		X
Integrated Release Management		X
Multi-Customer/Service Provider support		X
Integration with other products		X
Context Menu Service and Data Integration Service		X

1.3.1 New features

In the following topics we provide more information about these features.

Topology CI View

Topology CI view improves service performance (reduce outages) and reduce costs (remove audit fines) by identifying the CIs that are critical to your business and understanding how they are related to each other.

On CCMDB you do not need to launch over to IBM Tivoli Application Dependency Discovery Manager (TADDM) Tool to get the topology view information, now you are able to access the Business View directly on CCMDB GUI. Using this feature the tool will allow us to get the CIs and components details and its relationship by information level enabling to get more information to help for taking decisions based on correct information.

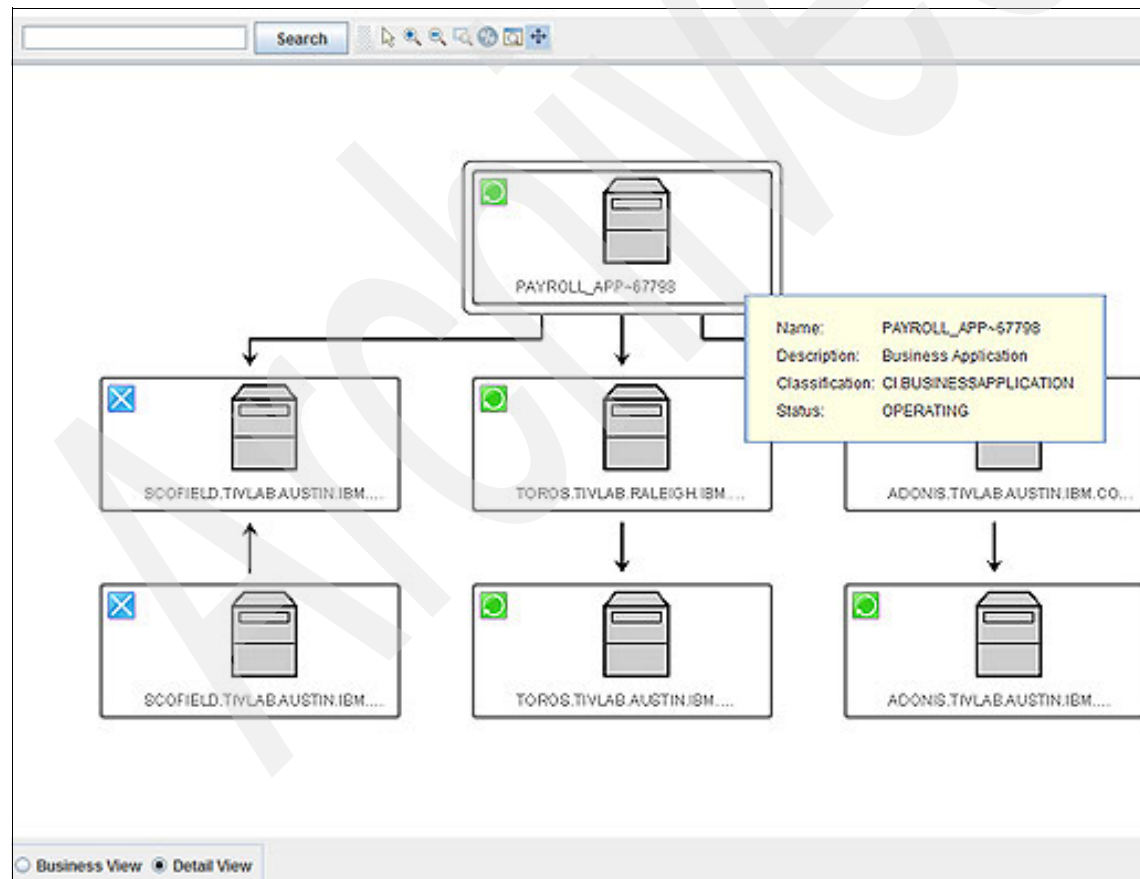


Figure 1-2 CCMDB Topology View

A new graphic topology view was added to easily view and identify important CIs. The features include:

- ▶ Availability in CI application, actual CI application, and Change Application for viewing Impacted CIs.
- ▶ Easy configuration to see exactly what is important for your business.
- ▶ Icons that make it easy identify CIs, what state they are in, and can be configured if needed.

CI reconciliation history view and user action enhancements

CI reconciliation history view and user action enhancements are available with CCMDB V7.2. These enhancements help you identify the causes of discrepancies in CI reconciliation results, and choosing and taking the appropriate actions.

CI reconciliation history view and user action enhancements allow you to determine exactly what remediation is needed from an audit variance. Using these enhancements you can:

- ▶ Create a Change or Service Request to handle the remediation of an audit variance
- ▶ View approved Changes for a CI when viewing an audit variance to help determine if there was an approved Change that caused the variance
- ▶ View CI attribute history while viewing an audit variance
- ▶ View the last audit results for the same CI

Figure 1-3 shows you how to remediate an audit variance by updating an authorized CI to reflect the actual value.

Authorized CI Information		Actual CI Information	
CI Object	CI	Actual CI Object	ACTCI
CI Attribute	COMPUTERSYSTEM_MEMORYSIZE	Actual CI Attribute	COMPUTERSYSTEM_MEMORYSIZE
CI Value	3,810,555,102.0000000000	Actual CI Value	4,290,772,992.0000000000
CI Unit of Measure		Actual CI Unit of Measure	
Top Level Site		Actual Configuration Item Number	NECHES1-1065
Configuration Item Number	NECHES1-1065	Actual Configuration Item Name	NECHES1
Configuration Item Name	NECHES1		
Update CI with Actual CI Value			

Figure 1-3 Authorized CI Information

See 3.2.9, “Auditing” on page 133 for more information about auditing.

CI promotion process improvements and Update promotion

CI Promotion Process improves time to value by making it quicker and easier to get the CIs needed for your business.

CCMDB allows authorized CIs to be created from actual CIs and then linked to the corresponding actual CI.

Update promotion was added in V7.2 to allow customers to promote CIs that have been previously promoted. This includes:

- ▶ Ability to synchronize existing authorized CIs with promotion
- ▶ Ability to synchronize attribute values and create related CIs that are newly discovered
- ▶ Expand the promotion scope and “re-promote” to create additional related CIs
- ▶ Add an attribute from a classification and all CI instances are affected
- ▶ Requires an RFC to update any CI that is in a protected state

Workplan Map widget

Workplan Map is a visual representation of all activities and tasks.

It exists on a new tab in a Change to provide a snapshot or visual indicator of all tasks. It includes the status for individual tasks, tasks in a hierarchy, and shows task dependencies. See Figure 1-4 on page 24.

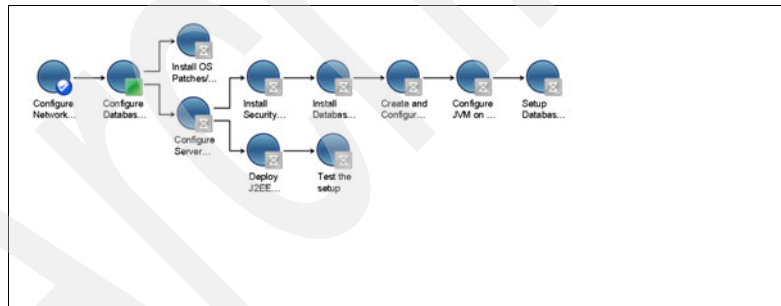


Figure 1-4 Workplan Map

End-to-end Change Management process enhancements

With CCMDB V7.2, a new workflow-based Change process is delivered out of the box. This new process provides enhanced job plans and approval enhancements.

Automated impact analysis and impact analysis topology view

Automated Impact Analysis reduces risk caused by a Change by easily and accurately determining the impacts ahead of time.

This feature was introduced in CCMDB V7.2. Using this feature, you can automate the identification of possible service outages. See Figure 1-5.



Figure 1-5 Calculate Impacts

Two techniques are provided to determine impacts:

- ▶ Calculated:
 - Engine executes a set of (configurable) rules against the CMDB
 - Follows relationships to find all impacts
- ▶ Historical Analysis:
 - Data mining of previously identified impacts
 - Re-use work done previously

See Figure 1-6 for an illustration of these techniques.

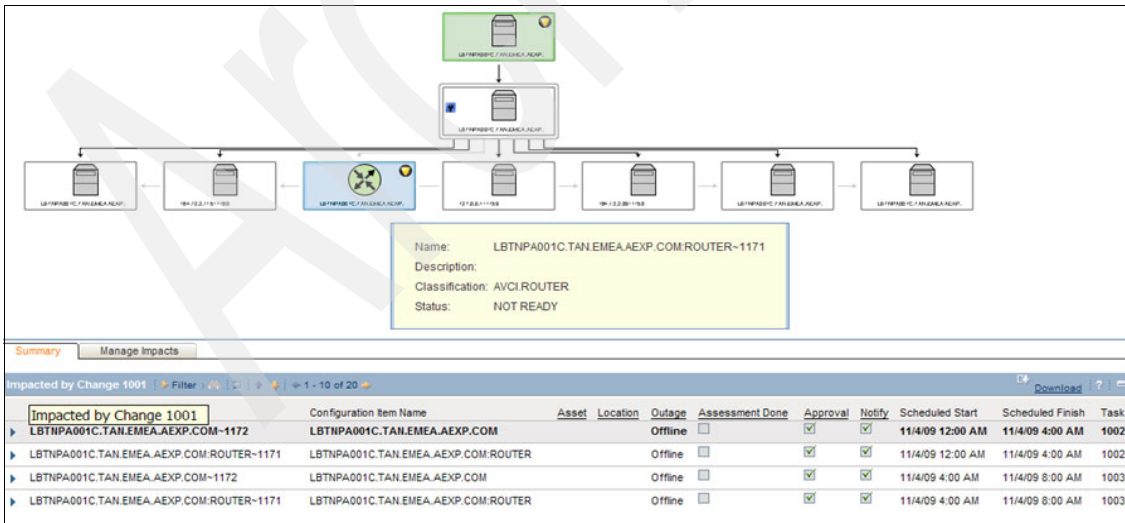


Figure 1-6 Automated Impact Analysis

Progress map

CCMDB V7.2 introduces a progress map that provides a visual representation of the states of a Change record. This map indicates the actual status of the Change Record and the flow expected to follow. The progress map steps can be defined by the System Properties variable defined by the name *pmchg.progressmap.seq.*

Figure 1-7 shows the progress map. See 2.2.2, “Workflows” on page 41 for more information.

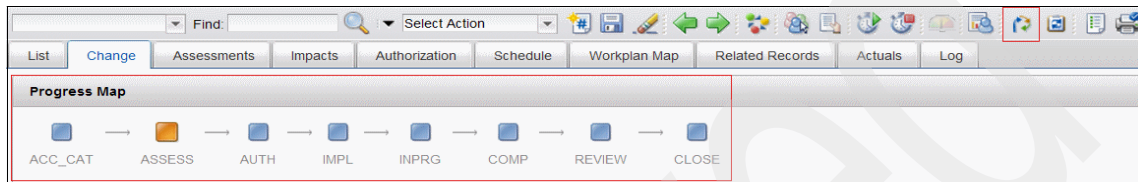


Figure 1-7 Route and progress map for a Change

Restricted Change schedules and blackout periods

Restricted Change schedules minimize the severity of impacts and disruption to services. Changes cannot be scheduled during blackout periods, as shown in Figure 1-8.

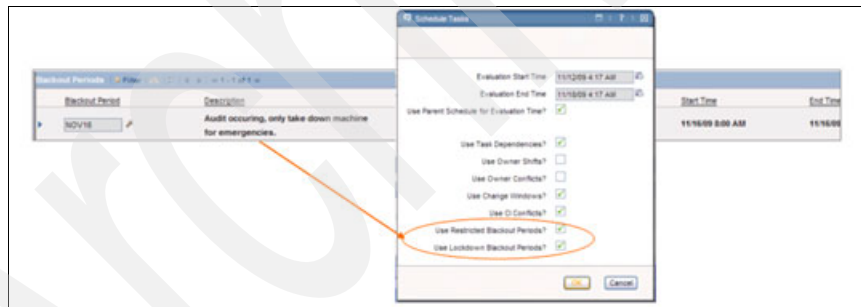


Figure 1-8 Restrict Change Scheduling

This feature was introduced in CCMDB V7.2. Blackout periods identify critical business periods when outages can be expensive. Automated Change scheduling will avoid blackout periods. Configuration item application shows blackout periods for the CI.

The blackout period has a type to indicate what is allowed:

- ▶ Restricted
- ▶ Lockdown

Conflict detection

Conflict detection was introduced in CCMDB V7.2. Conflict detection can automatically detect conflicts among scheduled tasks for a Change. You can examine and resolve detected conflicts.

CI baselines

A CI baseline contains the attributes and relationships for a CI or set of CIs at a point in time. It is typically created at a time when there is a known good configuration. It can later be compared against the current values of the corresponding actual CIs to see how the configuration has changed since the time the baseline was activated.

Figure 1-9 shows the CI baseline application.

The screenshot displays the 'CI Baseline' application interface. At the top, there are tabs for 'List' and 'Baseline'. The 'Baseline' tab is active, showing a form for creating or editing a baseline. The form includes fields for 'Name' (RBBaseline1), 'Version' (2), 'Active?' (checkbox), 'Active Date' (empty), 'Create Date' (6/10/10 23:14:10), 'Created By' (MAXADMIN), 'Change Date' (6/10/10 23:15:16), and 'Changed By' (MAXADMIN). There is also an 'Attachments' link. Below the form, a message states: 'A baseline includes member configuration items. You can add or delete member configuration items.' Below this message is a table of 'Member CIs'. The table has columns for 'Member CI', 'Configuration Item Name', 'Description', and 'Classification'. There are two rows of data. The first row shows 'ANISE.TIVLAB.AUSTIN.IBM.COM~11725' linked to 'ANISE.TIVLAB.AUSTIN.IBM.COM' with classification 'CI-512-2.WINDOWSOS'. The second row shows 'DROID.TIVLAB.AUSTIN.IBM.COM~12887' linked to 'DROID.TIVLAB.AUSTIN.IBM.COM' with classification 'CI-512-2.WINDOWSCOMPUTERSYSTEM'. At the bottom of the table, there is a 'Select CIs' dropdown and a 'New Row' button.

Member CI	Configuration Item Name	Description	Classification
ANISE.TIVLAB.AUSTIN.IBM.COM~11725	ANISE.TIVLAB.AUSTIN.IBM.COM		CI-512-2.WINDOWSOS
DROID.TIVLAB.AUSTIN.IBM.COM~12887	DROID.TIVLAB.AUSTIN.IBM.COM		CI-512-2.WINDOWSCOMPUTERSYSTEM

Figure 1-9 CI Baseline application

CI-asset linking

CI-asset linking was introduced in CCMDB V7.2.1. If you have installed an asset management product such as IBM Maximo for Asset Management, along with CCMDB, you can create linkages between configuration items and assets that represent the same objects. For a more information about Asset and CI integration, see "Chapter 17, Asset and CI integration" in the Redbooks publication, *Tivoli Integration Scenarios*, SG24-7878,

Simplified Change process

The simplified Change process reduces costs by making the Change process easier and quicker, as well as reducing risks and impacts by planning and communicating the Change before it actually occurs.

With CCMDB V7.2.1 you can now use a new workflow (PMCHGITLV3 workflow) based Change process provided out of the box:

- ▶ Invocation of automated impact analysis and scheduling
- ▶ Approval enhancements
- ▶ Multi-level Change authorization
- ▶ Support for multiple approvers at each level (for example, multiple CABs)
- ▶ Ease of adding additional adhoc approvers as needed
- ▶ Enhanced job plans to capture impact assessment workflow assignments

The simplified Change workflow process is shown in Figure 1-10. For more information about the simplified Change workflow process, see Chapter 4, “Customizing Change in Tivoli Change and Configuration Management Database” on page 301.

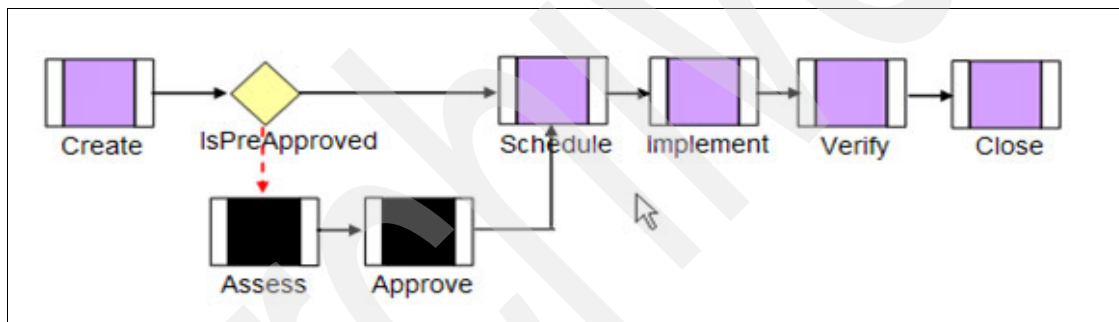


Figure 1-10 Simplified Change workflow process

Graphical scheduling GUI

Task scheduling for a Change occurs in the Scheduler application, which provides graphical tools that help you visually create a schedule that does not conflict with resource constraints, such as Change Windows and blackout periods. Figure 1-11 shows the Scheduler application - Gantt View. For more information, see “Scheduler Application” on page 201.

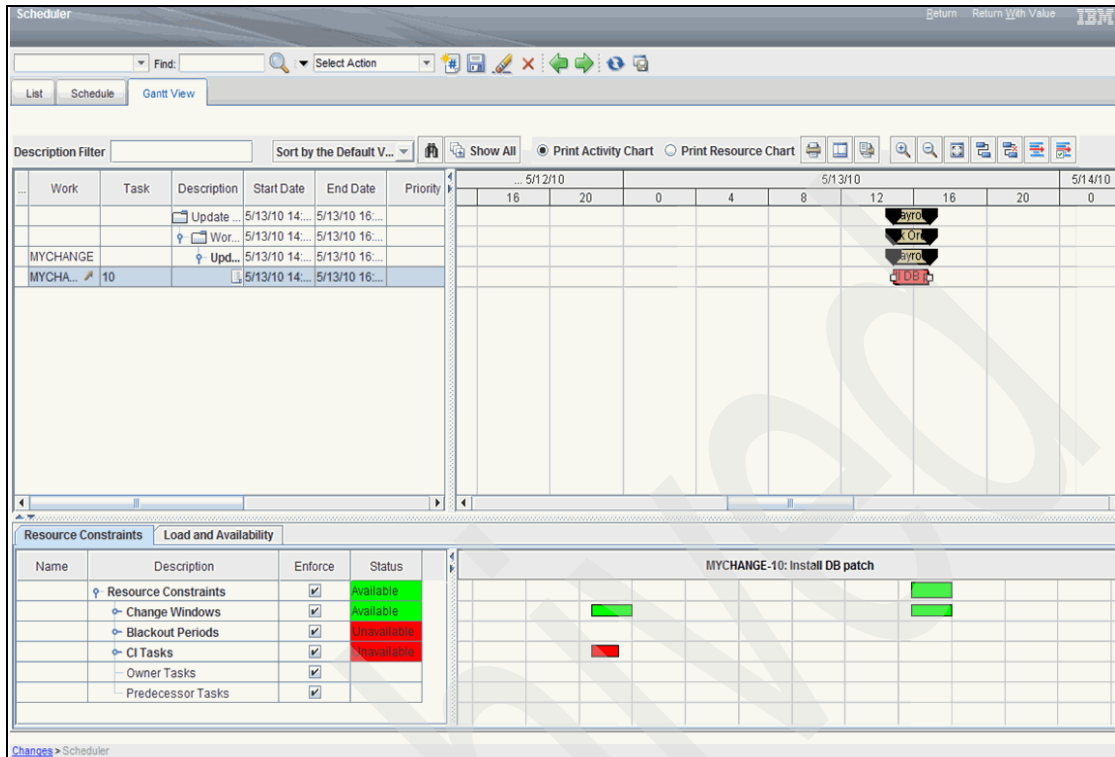


Figure 1-11 Scheduler Gantt View

Importing discovered CIs

Importing discovered CIs improves time to value by making it quicker and easier to get the CIs needed for your business. CCMDDB has the necessary performance improvements, automated configuration of ITIC TADDM CI adapters, and additional filter capabilities for importing discovered CIs from TADDM:

- ▶ Better control over the data that gets imported by specifying which CI types to include and exclude, as well as better performance due to less data
- ▶ Increased flexibility and control by specifying depth by CI type

Deployer's Workbench

CCMDDB V7.2.1 has introduced a new and improved CI design tool, Deployer's Workbench. It allows administrators to easily design the authorized CI space that their authorized CIs will exist in by choosing which CI types and relationships are critical to their business.

The following features are included:

- ▶ Best Practices authorized CI spaces, which can be used as is and can be imported into a CCMDB server with two clicks
- ▶ Drag and drop from Best Practices CI spaces, Common Data Model, or actual CI spaces to create your own authorized CI spaces easily
- ▶ Ease of importing and exporting CI spaces from one environment to another
- ▶ Documentation to guide you through every step

Figure 1-12 shows the Deployer's Workbench. See Appendix A, "A primer on Deployer's Workbench" on page 541 for more information about the Deployer's Workbench.

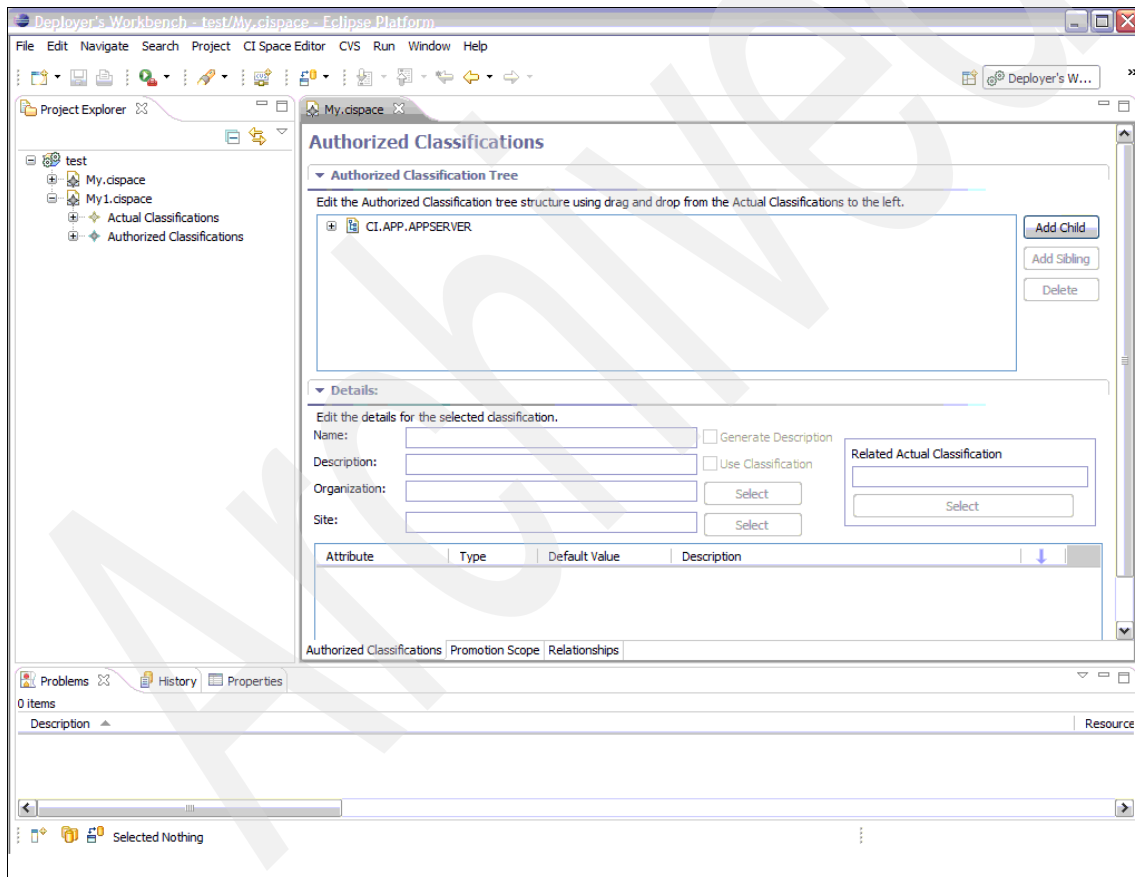


Figure 1-12 Deployer's Workbench

Integrated release management

Release management component is shipped with IBM Tivoli Release Process Manager Version 7.2.1 and is a separately charged feature. It can be used to manage simple and complex releases across the data center. See 3.4, “Release Process Manager” on page 253 for more information.

Multi-Customer/Service Provider support

Multi-Customer enablement or Service Provider support allows management of Change and Configuration for multiple customers in single instance deployment. With this functionality, one or more customers can be associated with a CI or an actual CI. These customer associations can be used to restrict who can see the CI data. There are implications for promotion, such as customer-copying behavior; and other implications for baselines, such as hiding members that you are not allowed to see. There are also implications for creating a new version, creating another baseline from an existing one, and running comparisons.

This functionality provides the following benefits:

- ▶ Improved efficiency of service delivery with automatic notification and automatic assignments
- ▶ Reduced cost by leveraging a single deployed instance to manage multiple customers for Change and Configuration Process Management
- ▶ New applications to manage multiple customers, as shown in Figure 1-13.

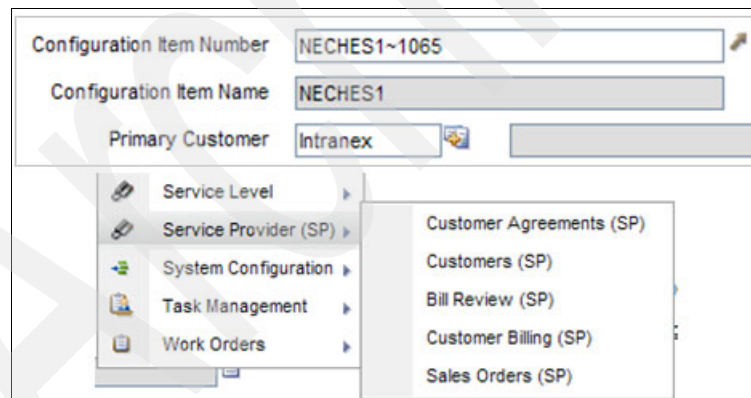


Figure 1-13 Multi-Customer/Service Provider support panels

Integration with other products

Starting with CCMDB V7.2.1, you can use the Change Management process to control updates to the IBM WebSphere Service Registry and Repository (WSRR), or automatically create assets in IBM Rational® Asset Manager from existing configuration items (CIs) in CCMDB.

For a full discussion of the benefits and implementation steps of IBM Rational Asset Manager and CCMDB integration, see “*Chapter 18 - IBM Rational Asset Manager and IBM Tivoli Change and Configuration Management Database integration*” in the Redbooks publication, *Tivoli Integration Scenarios*, SG24-7878.

For more information about IBM WebSphere Service Registry and Repository integration, see the InfoCenter at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.cmdb.doc_721/integrating/t_integrating_wsrr.html

Context Menu Service and Data Integration Service

CCMDB V7.2.1 enables a new Launch In Context (LIC) technology that addresses certain requirements that customers face today with the existing technology. The current LIC technology is still available in CCMDB V7.2.1 along with the new one. See Chapter 11, “Launch In Context” on page 507 for more information about this new feature.

1.3.2 Additional resources

Use this book in conjunction with the following references:

The InfoCenter Internet web page, which provides a more general overview of the CCMDB 7.2.1 product and information related to planning and installation of the product:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/index.jsp>

IBM Tivoli Change and Configuration Management Database (CCMDB) Wiki, where you can find and contribute information and scenarios about the product:

<https://www.ibm.com/developerworks/wikis/display/tivoliccmbd/Home>



Part 2

Process Layer

In this part of the book, we discuss the CCMDB Process Layer components.

Process flow technology

A CMDB implementation, or as ITIL V3 states, a Configuration Management System (CMS), has the main objective to support IT services management processes, which ensure that all persons and data necessary to execute a Change are aligned with the business objective.

In the terminology, CMS has been widely applied to explore many systems to support the IT lifecycle process in the configuration and Change process.

Process flow technology has gained improvements in the Information Technology Infrastructure Library (ITIL) toward ITIL V3. It maps the process flow of Change and Configuration Management at a better level with the ITIL process, which we cover here briefly before introducing the features of the process flow technology.

In this chapter, we describe the process flow technology components of the IBM CCMDDB solution. We highlight how the solution addresses the flexibility and adaptability requirements to cover client needs.

We explain the technologies involved in “Technology components” on page 39, and give an example in “An end-to-end example” on page 55. We also cover workflow, the main orchestration engine to automate the Change Management process, and job plans, which are used to define data templates and minor flow implementation.

Scope: The main purpose of this chapter is to give you an overview of the underlying components of process flow technology and explain how these components work together in an end-to-end scenario. Details about process managers, including process manager enhancements in CCMDB V7.2 and CCMDB V7.2.1 can be found in Chapter 3, “Process Managers” on page 83.

We cover the following topics:

- ▶ 2.1, “Change Management process” on page 37
- ▶ 2.2, “Technology components” on page 39
- ▶ 2.3, “An end-to-end example” on page 55
- ▶ 2.4, “Summary” on page 81

2.1 Change Management process

Service Transition principles, a discipline in ITIL V3, describes the Change Management process, whose objective is to bring business benefits and reduce errors towards the costs and efforts involved in doing Changes in IT services.

In order for a Change Management process to be managed successfully, it is necessary to reduce the risk exposure and the severity of any impact while being successful at the first attempt.

To achieve the objective of implementing a Change, ITIL best practice entails a considered approach to assess the risk and the business continuity, change impact, resource requirements, change authorization, and especially the realizable business benefit.

Certain changes are more frequently applied, and the Change Management board has established common procedures to drive this type of change. However, sometimes unusual changes must be applied and controlled, involving many aspects and persons responsible by the business impact.

A change is commonly formalized by a Request For Change (RFC) document, which includes details of the proposed Change. RFCs are submitted to a series of process activities to ensure that they bring the expected business benefits and that the persons involved in the Change are aligned with the process. The main activities outlined by the ITIL process in a Change process are as follows:

- ▶ **Accept:** Accept or reject the Change proposal, and categorize it in the IT and Business environment.
- ▶ **Assess:** Establish appropriate level of change authority and evaluate the relevant areas of interest and the business impacted.
- ▶ **Schedule:** Plan the updates and the resources involved.
- ▶ **Authorize:** Authorize or reject a Change, and communicate the decision to all stakeholders.
- ▶ **Implement:** Coordinate change implementation involving the resources and stakeholders needed.
- ▶ **Review closure:** After all actions are completed, determine if the Change has achieved its objectives (IT and business), closing it.

A set of best practice guidelines for IT service management processes has been documented in the ITIL library of documents. IBM has extended this best practice documentation based on experience and documented the results in a process reference model called the Process Reference Model for IT (PRM-IT).

The PRM-IT content is made publicly available through the IBM Tivoli Unified Process (ITUP) tool, including the Changes to the ITIL V3 model.

Figure 2-1 shows an example of the ITUP representation of the high level Change Management process flow. It represents the technology independent major steps of a best practice Change Management process.

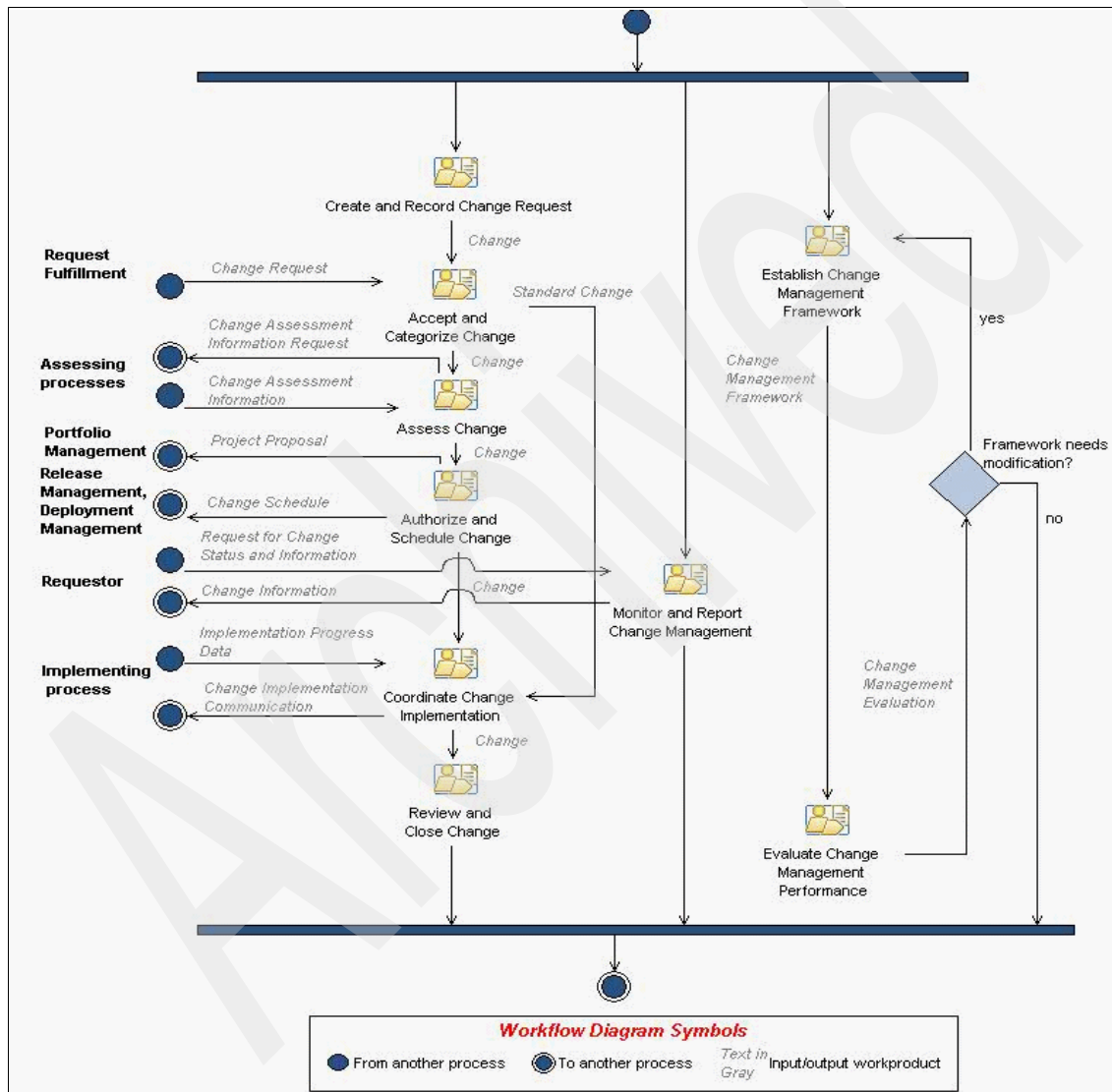


Figure 2-1 ITUP Change Management activities

ITUP describes major activities and their work breakdown structure, the input and output of each step, the roles responsible for each of the steps, and from a Tivoli perspective, the tools required to support the process from an operational perspective.

Best practice definitions such as ITIL or PRM-IT are independent descriptions of a specific type or classification of a process.

The main purpose of any CMDB implementation is to support IT service management processes. Processes require data in order to accomplish their work. Change and Configuration Management process templates are delivered as part of the CCMDB V7.2.1 package. In addition, the CCMDB is considered the fundamental building block for further ITSM process implementations. In the same way that business applications are geared towards a higher degree of automation, processes in the IT operations environment are required to increase the level of automation in order to decrease operational costs.

The CCMDB V7.2.1 has improved the process flow technology to implement these tasks. Workflows and job plans are the main process flow articulators to manage a Change. It is important to state that you might need to adjust these processes based on your business perspective.

2.2 Technology components

The IBM CCMDB solution is a combination of a Data Layer and a Process Layer. CCMDB V7.2.1 uses both the job plans and workflows to articulate the system to take benefit of the tool to build the process flow technology. Job plans are data templates to apply in work plans and work orders, used to build Change Requests and articulate them. Workflows are the main feature of the tool to provide flow control and implement process at a high level.

CCMDB provides workflows flexible enough to adapt to real life, while job plans are an important tool to describe data models to be applied in changes, its activities, and tasks.

In order to make process flows executable, a workflow technology is required in order to sequence personas in various roles through the activities and tasks of the process and deliver the right information at the right time. The final goal is to increase the operational efficiency.

Creating and submitting an RFC is a prerequisite for generating and working on the Change Record itself. The IBM solution is very much aligned to industry best practices. Its workflow technology provided by the CCMDB solution is capable of taking the theoretical process models into its software runtime in order to make it executable.

The IBM CCMDB solution delivers default process templates for Change and Configuration Management. Nevertheless, theory usually differs from implementations and real life requirements. While best practice definitions and reference implementations of IT service management processes can only provide a high level of abstraction, daily work requirements differ in each IT environment and depend very much on the specification or classification of the process request.

A Change Management process differs depending on its nature. Various people will be involved, various steps have to be taken for assessment, for authorization, or change implementation, depending on if a Change Request is, for example, for deploying a patch to an operating system or deploying a complex application into a composite, multi-tier application infrastructure. Very often, key roles of a process are department dependent, for example, the role of a Change Manager is represented by various people depending on the classification of a Change Request.

The workflow technology must be flexible enough to adapt to various requests and variations of a process in order to align to the daily work requirements. Easy modifications of process definition need to be handled without any programming skill for the administrative staff of the solution. The solution requires defining various process templates as permutations of generic flow definitions while using them based on the classification of a process request, for example, an RFC.

While the activities of a process usually are consistent from use case to use case, the tasks differ depending on the kind of request. Nevertheless, the workflow technology must be flexible enough to also handle various requirements with respect to the high level activities of the process flow. There is even the requirement and necessity to sometimes change the process flow while the process instance is already in progress. Activities and tasks definitions are required to be reusable for various process permutations in order to satisfy real live requirements.

The process request to Change Management or any other ITSM process has to specify a classification that describes the nature of the request. In the case of Change Management, the process request is also known as a Request for Change. Depending on the classification of the request, the activities and tasks (steps inside the activity level) that are necessary to fulfill the request are chosen by selecting and applying a predefined process template to the request. A classification, for example, defines an urgency, a priority, a detailed description of the request nature and, optionally, if known, the configuration items that are targeted in the request.

A template has predefined the sequence of activities and tasks necessary to handle the request and is considered to model the end-to-end model of the process flow. There can be various activities templates, also referred to as job plan templates, depending on the daily work requirement to fulfill the Change Request. A job plan template for deploying an operating system patch varies from a process template for password change or deploying a complex J2EE application.

2.2.1 Components overview

A combination of administrator facing applications and technologies in the CCMDB solution are used in order to define and apply a process model in order to achieve a high degree of automation for ITSM process implementations such as Change or Configuration Management.

In this section, we explain the concepts of a workflow, an action or action group, job plan and a work order, and we touch on Process Requests as related to the overall process flow.

We also explain each of the technologies and major roles in the overall implementation, while in 2.3, “An end-to-end example” on page 55, we walk through a concrete example provided in the CCMDB default implementation.

2.2.2 Workflows

Workflow facility is the orchestration tool used in the CCMDB to connect all the steps necessary to manage the Change process. It describes the process and its activities in an end-to-end flow control. It is the main tool to design all the process and activities in the Change Management process.

The workflow tool provides the following benefits to the administrators and users:

- Interactive automation with a user: If you need to provide interaction with the person in charge of the respective process step, for example, to make a decision between various choices to be taken, an authorization decision, or the forwarding to another application, a workflow is required. The Workflow Designer Application provides the capability to define a wizard-like interaction with the user to guide the user through a set of activities and decision points.

- ▶ **Verification of conditions:** Very often, you are required to analyze data before a decision can be taken on how to route the record to the next step in the overall process flow. Workflows allow you to define an evaluation of a record using a *Condition Node* in the workflow designer application. The evaluation indicates a true or false decision and can then redirect the record based on that evaluation.
- ▶ **Obtain authorizations:** Workflow designation nodes are the preferred way to obtain a positive or negative authorization from a person in charge of the decision. Depending on the decision, the workflow can route the record either to one or the other next step in the process flow.
- ▶ **Wait for another process to finish:** If there is a need to suspend the process execution until another process working on the same record completes its work, workflows provide the capability to define *Wait Nodes* in the workflow designer application for this purpose. An example is the interaction between Change and Release Management. In case a Change Record is transferred to release management for implementation, the Change Management progress is set on hold until the release process signals completion.
- ▶ **Divide process into subprocess:** Doing this provides for reusability. When a process has a lot of actions and designations to be executed, the workflow process enables you to divide the whole process into subprocesses to gain a better understating of the whole process flow.

Practically, a workflow designates the activities and tasks to the right persons in the appropriated time, considering the variables involved in the decisions related to the process.

Each node in the workflow implements a few of the benefits explained before. These nodes are connected, defining the process. They are modeled using a graphical designer called the Workflow Designer, located in the path, **Go To → System Configuration → Platform Configuration → Workflow Design**.

Each process in the Change Management discipline has its origin in a process request. A process request is submitted and classified by a requester. The Process Request application is responsible for creating the process requests. A process request is always necessary as the starting point for a process flow instantiation.

Figure 2-2 illustrates the master workflow for all kinds of process request submissions, regardless of the process request type. Here you can find the submit master workflow called ISMSUBMIT. The master workflow to handle the process request is defined by a global property in the System Properties application, named *pmp.submit.workflow*.

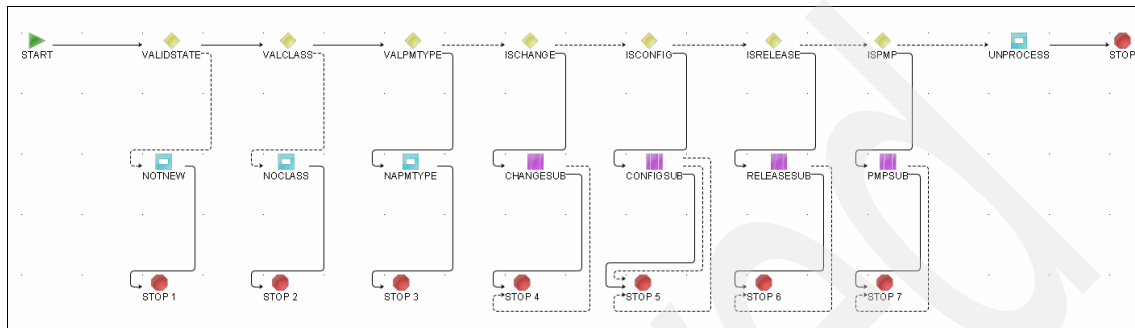


Figure 2-2 Process Request master workflow

Every workflow must have a START and a STOP node, and a connection line between them. After an object is routed to its workflow, the START node is executed followed by its action in the connecting arrow. While the workflow does not finish, the route action is available to the object application in the tool bar icon. After the workflow stops, the process flow is finished.

After a Process Request is routed to ISMSUBMIT workflow, it checks the type of process request involved in the request using these nodes: ISCHANGE, ISCONFIG, ISRELEASE or ISPMF. If certain nodes give a positive result, a subprocess handles the specific request.

When a Change is selected to be created by a process request, it creates a WOCHANGE object. After that, a series of workflows are defined in the CCMDB V7.2.1 to handle each step of the Change Management process.

Figure 2-3 represents an example of the tool bar used in the Change Application and its buttons to route and interact with workflows designed to handle each change activity: Accept, Assess, Authorize, Schedule, Implementation, and Review/Close.

One important indicator of process flow progress is the Progress Map frame, which indicates the actual status of the object and the flow expected to follow. The progress map steps can be defined by the System Properties variable defined by the name *pmchg.progressmap.seq*.

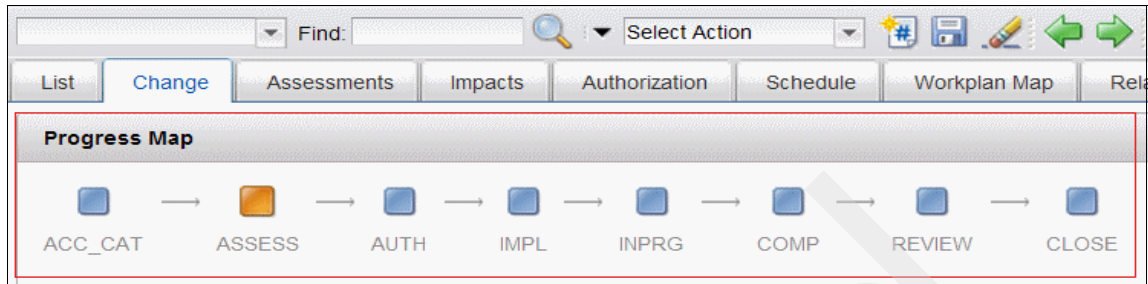


Figure 2-3 Route and progress map for a Change

The route button in the tool bar  interacts with workflow to execute the next step. When there is more than one workflow associated with the object, it opens a workflow selection box to designate the process to the right workflow (Figure 2-4).

To avoid this dialog, there is a new button called *ITIL V3 Change* that routes you directly to the workflow responsible for the ITIL V3 model.

The 'Start Workflow' dialog box contains a yellow message box at the top stating: 'There is more than one workflow process available for this application. Please select the one you want from the Process drop-down and press OK.' Below this is a 'Process' dropdown menu with 'PMCHGITLV3: C' selected, and a 'Memo' text field. At the bottom is a table titled 'Previously Started Workflow Processes' with columns: Process, Activation #, Description, and Active Instance. The table is currently empty, showing the message '...No rows to display...'. Navigation icons (back, forward, search, etc.) are located above the table. 'OK' and 'Cancel' buttons are at the bottom right.

Figure 2-4 Select workflow window to be started by the designated application and object

CCMDB V7.2.1 provides workflows that maps the whole lifecycle of a Change proposed by the ITIL V3. It is important to observe that the workflows proposed by this version of CCMDB are an abstraction of the Change Management process and they can be adapted to the environment. The workflows proposed by the tool to orchestrate the Change Management process and its functions are explored with more details in 2.3, “An end-to-end example” on page 55,

2.2.3 Action and action groups

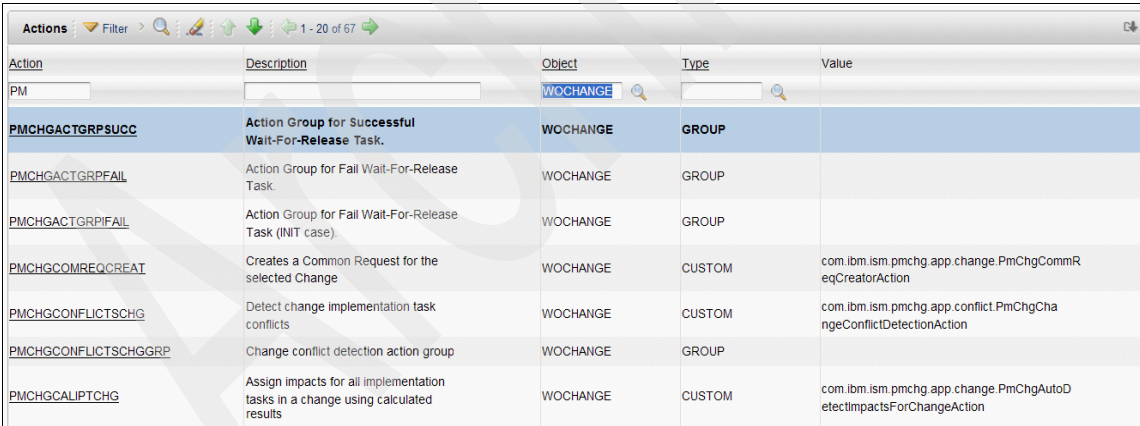
Workflows can call actions or action groups for automating part of its execution steps. Actions and actions groups are useful in workflow nodes. When a task, a designation, or even a decision is done in a workflow, an action can be executed to help to automate the process. Actions are usually a codified sequence of short logic. You use actions from within workflow nodes, in the flow represented by the arrows connecting the nodes logic.

An action can change the status of an object, set data values, run any custom action by using a custom Java class, execute a program on the application server, or initiate an application action, such as initiating a workflow.

Process Manager Products expose most of their capabilities by providing Java custom classes that can be used from within an action.

An action group is a list of actions bundled together in a sequential list of member definitions.

Actions and action groups are defined in the Action application, located at **Go To → System Configuration → Platform Configuration → Actions**. If you search in the Action application for the string “PM”, you see a list of all actions and action groups provided by the various process manager products (Figure 2-5).



Action	Description	Object	Type	Value
PM		WOCHANGE		
PMCHGACTGRPSUCC	Action Group for Successful Wait-For-Release Task.	WOCHANGE	GROUP	
PMCHGACTGRPFail	Action Group for Fail Wait-For-Release Task.	WOCHANGE	GROUP	
PMCHGACTGRPIFail	Action Group for Fail Wait-For-Release Task (INIT case).	WOCHANGE	GROUP	
PMCHGCOMREQCREAT	Creates a Common Request for the selected Change	WOCHANGE	CUSTOM	com.ibm.ism.pmchg.app.change.PmChgCommReqCreatorAction
PMCHGCONFLICTSCHG	Detect change implementation task conflicts	WOCHANGE	CUSTOM	com.ibm.ism.pmchg.app.conflict.PmChgChangeConflictDetectionAction
PMCHGCONFLICTSCHGGRP	Change conflict detection action group	WOCHANGE	GROUP	
PMCHGCAIPTCHG	Assign impacts for all implementation tasks in a change using calculated results	WOCHANGE	CUSTOM	com.ibm.ism.pmchg.app.change.PmChgAutoDetectImpactsForChangeAction

Figure 2-5 Listing of all actions provided by Process Manager Products

Figure 2-6 represents an action group, showing that two SETVALUE and one CHANGESTATUS actions are used inside this action group.

List

Action

Action:

Value:

Click to s

Object:

Parameter/Attribute:

Type:

Memo:

Classification:

Accessible From:

Class Description:

Members

Filter

1 - 2 of 2

Action	Description	Object	Type	Value
PMCHGFCOFF	Turn flow control off for the Change	WOCHANGE	SETVALUE	0
PMCHGCAN	Cancel the Change if the approval is rejected	WOCHANGE	CHANGESTATUS	CAN

Figure 2-6 Action Group executes two actions: set value and after changing a status

As the previous version of the CCMDB proposed, actions and actions groups can be used in job plans and tasks to automate specific process flows and increment the flow control functionality in the tasks perspective. This functionality is explored in 2.2.5, “Job plans” on page 48.

2.2.4 Work plans and work orders

A process request is submitted for authorization after being routed by the workflow. When the process request is approved, a work order object is created. Based on the process manager type that is classified in the process request, the work order is of a specific type. In our example, it becomes a *Change work order* or simply a Change. You do not find the Change Record before the related process request has been accepted.

A Change work order can now be assigned a detailed work breakdown structure by assigning a job plan template to the Change work order. A job plan defines the activities and tasks that need to be taken to successfully perform the Change Request. A job plan can either be applied manually to the Change Object in the Change Application or can be automatically applied to the Change by the Change workflow process. In both cases, the classification of the process request is a key factor to determine the right job plan process template to be applied.

After a job plan has been applied to the work order, a work plan object is now generated and the process flow can be set into progress. After all steps are completed, the final step is to close the originating process request.

When applying a job plan template to a work order, all activities and tasks defined in the job plan are copied over to the work order plan as children of the work order plan. If you need to modify the work order plan, you can modify the work order plan without requiring a modification to the original job plan process template.

The process request uses a description, a priority, a specification of the process manager type (for example, Change), an intended completion date, as well as a more granular classification of the specific type (such as a hardware, software, or storage change). The granular classification scheme can be adopted according to your needs. If required and known, the configuration items targeted for this Change can be specified in the process request.

Figure 2-7 shows an example of a process request to Change Management, including its classifications. A configuration item (CI) has been specified as well. Remember that a process request has to be submitted and approved before a work order object gets generated.

The screenshot displays the 'Process Request' application interface. At the top, there is a search bar with 'Find:' and a 'Select Action' dropdown. Below this are tabs for 'List', 'Process Request' (selected), 'Related Records', and 'Log'. The main header area shows 'Process Request: 1006', 'Owner: MAXADMIN', 'Owner Group:', 'Process State: ACCEPTED', and 'Status: INPROG'. A 'User Information' section is visible below the header. The 'Process Request Details' section contains the following fields:

- Description: PATCH Tivoli Monitoring
- Process Manager Type: Change
- * Site: AUSTIN
- Requested Completion: 5/31/10 4:30 AM
- Asset: (empty field with a double arrow icon)
- Location: (empty field with a double arrow icon)
- Configuration Item: 10004 (highlighted with an orange border) with a double arrow icon pointing to 'Miami Linux WAS server'
- Configuration Item Name: MIALINUXWAS01
- Response Plan: (empty field with a double arrow icon)
- Impact: 2
- Urgency: 3
- Priority: 3

Figure 2-7 Process Request application

All PMPs provide a type of work order. Change, Configuration, or Release Management are examples of PMPs that provide specific work order objects that get generated when accepting the respective process request.

2.2.5 Job plans

Job plans are used to represent data definition to work plans to execute a specific series of activities and tasks. A job plan is a detailed description of the work to be performed after a process request gets a work order object and a job plan gets assigned to the work order object.

A job plan is a reusable template of a work item description. After a job plan gets assigned to a work order object, it becomes a work order plan. The key difference between a job plan template and a work order plan is that the work

order plan represents a process in progress, while a job plan template is, first of all, a description of a work breakdown structure.

Another difference between the job plan or work order plan and the workflow technology is that you can change a work order plan even at runtime in case you detect an incident that requires you to do so. You cannot change an instance of a workflow while it is in progress. You have to think about all possible exceptions and conditions while designing the workflow. This is not true for a work order plan, which you can change it at runtime.

Attention: The workflow facility is the main engine to host end-to-end process workflows. The overall workflow engine uses the workflow application as well as the job plan to define and run automated service management processes.

Job plans are used to define templates for the activities and tasks of a specific process flow, such as implementing a Change. Remember that the job plan does not include a description of the steps that are part of the process request phase, such as submitting and accepting an RFC.

In order to represent a specific process flow, job plans use a nested structure. A top level job plan includes the major activities of a process. These activities depend on the process type and are usually aligned with best practice definitions such as ITIL or PRM-IT, as reflected in ITUP. Nevertheless, they can be adopted to the requirements of the organization. The top level job plan is of type *Process*.

Each activity represented in the top level job plan is defined as a nested job plan itself. The IBM CCMDB solution delivers three default examples for Change Management out of the box, a top level job plan for a standard Change, one for representing a pre-approved Change, and a more specific example for representing a Change to a more complex J2EE application.

We expect the J2EE Application Job Plan to be a more detailed, production oriented guideline, whereas the standard and Pre-Approved Job Plans are abstracted guidelines oriented along the best practice definitions. Job plan definitions need to be defined in the level of detail required for your production environment. Think of job plans as a technology-based representation of your project plan documentation. All steps are defined inside the system and are handled by a workflow engine.

Examples of job plans

Figure 2-8 shows a list of available job plans. Drilling into the example of the Standard Change Job Plan, you see the activity shown in Figure 2-9 defined.

List	Job Plan	Work Assets	Specifications	Change Template	Change Assessments	Change Authorization
Advanced Search : Save Query : Bookmarks :						
Job Plans : Filter > 1 - 8 of 8						
Job Plan	Description	Duration	Supervisor	Status	Template Type	Organization
<u>PMCHG</u>						
<u>PMCHGNORML</u>	Normal Change JobPlan	4:00		ACTIVE	PROCESS	
<u>PMCHGSTAND</u>	Standard Change JobPlan	4:00		ACTIVE	PROCESS	
<u>PMCHGJ2EE</u>	J2EE Application JobPlan	44:00		ACTIVE	PROCESS	
<u>PMCHGDATAB</u>	Implement Database Change	6:00		ACTIVE	PROCESS	
<u>PMCHGMIDWR</u>	Implement the Middleware Change	25:00		ACTIVE	PROCESS	
<u>PMCHGSERV</u>	Implement the Server Build Change	59:00		ACTIVE	PROCESS	
<u>PMCHGSTANA</u>	Standard Automated Change JobPlan	4:00		ACTIVE	PROCESS	
<u>PMCHGEMERG</u>	Emergency Change Jobplan	4:00		ACTIVE	PROCESS	
☐ Select Records						

Figure 2-8 Default job plans for Change Management

Figure 2-9 represents a job plan with information such as the name of the job plan, the template type, default object used for this job plan tasks, and the duration of the process. The *Process Flow Sequence* field specifies the sequence followed by the object status for this specific job plan and workflow. The values described in this field are the status expected for the object controlled.

Tasks in a job plan are the series of *Activities* in a process. In Figure 2-9, there is only one activity for the job plan, the *Implementation of the Change*.

Job Plans

Find: [] Select Action []

List Job Plan Work Assets Specifications

Job Plan [CHG-P-F1] Standard Change JobPlan Organization [] Site []

Details

Status: ACTIVE

Template Type: Process

Duration: 0:00

Classification: []

Class Description: []

Launch Entry Name: []

Default WO Class: []

WO Priority: []

Interruptible?: ☐

Flow Controlled?: ☒

Suspend Flow Control?: ☐

Flow Action: []

Flow Action Assist?: ☐

Job Plan Tasks Filter [] 1 - 4 of 5

Sequence	Task	Description	Nested Job Plan
1	10	Assessment	CHG-F1
2	20	Approval	CHG-F2
3	30	Schedule	CHG-F3
4	40	Implement the Change	CHG-F4

Labor Materials Services Tools

Planned Labor Filter []

Task	Craft	Skill Level	Labor	Quantity	Hours
...No rows to display...					

Figure 2-9 jStandard Change Job Plan activity definitions

Figure 2-10 shows a list of job plan tasks. Each task has a sequence number to be followed. The flow controlled check box guarantees that each task's predecessors need to be completed before going to the next task level.

The screenshot displays the 'Job Plan Tasks' interface. At the top, there is a toolbar with icons for filter, search, and navigation. Below this is a table listing tasks with columns for Sequence, Task, Description, Nested Job Plan, and Duration.

Sequence	Task	Description	Nested Job Plan	Duration
10		Setup and configure Network Infrastructure ch		4:00
20		Setup and configure Database and Application		4:00
30		Install OS Patches/Upgrades		4:00
40		Setup and configure Database Server Instance		4:00

Below the table is a 'Details' panel for the selected task (Task 20: Setup and configure Database and Application). The panel contains various fields for configuration:

- Organization: [Field]
- Site: [Field]
- Task: 20 Setup and configure Database and Application
- Sequence: [Field]
- Nested Job Plan: [Field]
- Classification: [Field]
- Class Description: [Field]
- Launch Entry Name: [Field]
- Implementation Task? ☒
- Duration: 4:00
- Meter: [Field]
- Owner: [Field]
- Owner Group: PMCHGIMP
- Flow Controlled? ☒
- Suspend Flow Control? ☐
- Flow Action: [Field]
- Flow Action Assist? ☐
- Assisted Workflow: [Field]
- Predecessors: 10

Figure 2-10 Sequential job plan tasks to setup a J2EE server.

A sequence of steps is defined enabling dependencies between tasks, which can define one or multiple predecessor tasks that have to be finalized before it starts executing the successive ones.

Following the Change data model enhancement, which conciliates more information regarding the activities in the ITIL process, job plans provide new template information to populate specific data to the Change Management process. Three new tabs support the new functions in the CCMDB tool:

- ▶ Change Template tab: Provides information to populate a Change with pre-defined values regarding reason for Change, verification, and back out plan.
- ▶ Change Assessment tab: Provides a template to populate technical and business assessments impacts for that specific Change.
- ▶ Change Authorization tab: Provides information about the authorization levels that the Change needs to meet before being implemented.

Figure 2-11 shows the Assessment tab for a job plan that implements database Changes. An example of this new information in the job plans is explored in more detail in 2.3, “An end-to-end example” on page 55.

Technical Assessments			
Filter > [Icons] 1 - 7 of 7 [Icons] Download ? [Icon]			
Description	Assessment Type	Owner	Owner Group
Assess Network Impact	Network		PMCHGANA
Assess Capacity Impact	Capacity		PMCHGANA
Assess Storage Impact	Storage		PMCHGANA
Assess Security Impact	Security		PMCHGANA
Assess Server Impact	Server		PMCHGANA
Assess Application impact	Application		PMCHGANA
Assess OS Impact	OS		PMCHGANA
New Row			
Business Assessments			
Filter > [Icons] 1 - 3 of 3 [Icons] Download ? [Icon]			
Description	Assessment Type	Owner	Owner Group
Assess Operational Impact	Operational		PMCHGANA
Assess Financial Impact	Financial		PMCHGANA
Assess SOX Impact	SOX		PMCHGANA

Figure 2-11 Job plan assessments tab

Comparison: Job plans are used to provide data information to an activity or a task to be executed, while workflows control the orchestration engine of the process and its activities.

Nested job plans

A job plan models hierarchal structures using nested structures. Tasks of a nested job plan hierarchy can run in parallel or sequentially, depending on the predecessor definitions. This capability allows you to define a parent child hierarchy as well as a way to define sibling relationships between the various execution steps of the process definition. The flow control of a process while in progress always propagates status Changes of the lower tasks up in the hierarchy. So, for example, if a task inside an activity is completed, it declares its completion status in order to start the execution of the next task in the defined sequence.

Figure 2-12 illustrates an example of a dependency tree of nested job plans and a dependent sequence of task definitions.

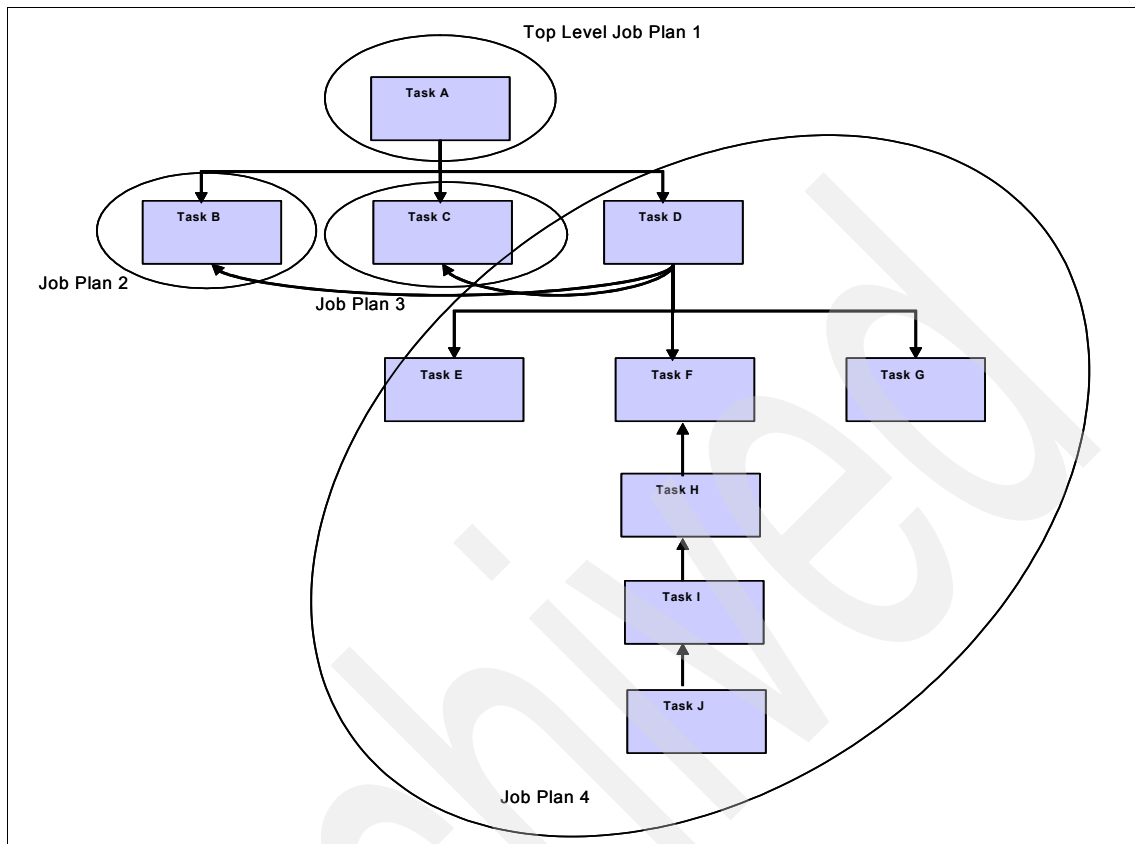


Figure 2-12 Nested job plan hierarchy and task dependencies

Figure 2-12 shows four job plans, the top level job plan 1 and its nested job plans B, C, and D. Technically, the nested job plans are defined as tasks inside the top level job plan 1.

The nested job plans 2 and 3 start in parallel, while the execution of job plan 4 depends on 2 and 3 to finish. This illustrates the sibling concept.

Tasks E, F, and G of job plan 4 all start in parallel, while tasks J, I, and H all depend on the completion of their respective predecessors.

To summarize, job plans are used to model specific process flow, regardless if the flow is an abstract, type independent flow or a concrete example reflecting your daily real life process flow requirements. Job plans are templates that get assigned manually or automatically to a work order that has been generated through a process request and result in a work order plan. Job plans define a hierarchy of tasks, including a way to define a dependency on task completion of its predecessors. Tasks can run in parallel or sequentially.

There are various types of tasks. You can define manual, automated, or interactive task types. Depending on the task type, you link the task definition to an action, action group, workflow, or integration module.

In CCMDB 7.2.1 there are changes in how end-to-end process flows are modeled. Whereas in the previous versions, parts of the *process* were designed through job plans activities and tasks, relying upon tasks and flow actions to control the end-to-end process flows, the CCMDB V7.2.1 proposes the workflow technology as the primary tool to be used to model end-to-end process flows. In this case, job plans are an auxiliary data model to create data and tasks definitions to implement a process.

The workflow technology and job plans can still be used as they have been used in previous versions; nevertheless, the default process templates for Change and Configuration Management are modeled according to what we describe in this chapter.

The main reason for this update in this release resides in the fact that workflow is easier to model and proposes a centralized solution, while using job plans creates innumerable work plans, activities, and tasks templates that are sometimes hard to maintain.

Complexity: In CCMDB V7.2.1, job plans are expected to keep a low level of complexity, while workflows articulate the end-to-end process flow in the system.

2.3 An end-to-end example

Now that we have explained the various technology components involved in the overall process flow technology, including their interaction, we present concrete examples of default Change and Configuration Management process flow definitions provided by the CCMDB V7.2.1 solution and explain where the interactions of the technologies are applied. Where sections in this chapter explain the flow process, the technology components are presented at the same time.

This section uses the Change and Configuration Management process proposed by the ITIL V3 to exemplify how the process flow technology works. However, this section does not focus on the Change Management process, but on the process flow components. For more information about the Change Management process, see Chapter 3, “Process Managers” on page 83.

2.3.1 Request for a Change

The start point to do a Change is to request formally for an RFC. This can be achieved by the Process Request application, located at **Go To** → **Change** → **Process Request**.

When creating a process request for a Change, the user defines it as a Change type, and provides the main information, such as classification, configuration items, asset, locations, and the target configuration items. Figure 2-13 represents a Change Process Request for an IBM DB2® Database server.

The screenshot shows a web form titled "Process Request Details". It contains several input fields and dropdown menus. The "Description" field is set to "Change the DB2 Database Server". The "Process Manager Type" is set to "Change". The "Site" is set to "AUSTIN". The "Requested Completion" field is empty. The "Asset" field is empty. The "Location" field is empty. The "Configuration Item" is set to "10001". The "Configuration Item Name" is set to "MIALINUX01". The "Response Plan" field is empty. The "Impact" field is set to "1". The "Urgency" field is set to "1". The "Priority" field is set to "1". A "Details" box on the right contains the text "Change the DB2 Database server in Configuration Item machine.".

Figure 2-13 Change Request process to change a DB2 Database server

After the Request for Change is created, it is possible to *submit* the RFC to the next step. After the submit action is requested, ISMSUBMIT workflow (Table 2-1 on page 57) is responsible to generate a Change work order based on the process submitted in the Process Request application. Remembering, the master workflow to handle the process request is defined by a global property in the System Properties application, named *pmp.submit.workflow*.

2.3.2 Change Request process

After a Change Object is created by the process request application, it is possible to submit the Change to its workflow process. The activities and the related proposed workflows used in this chapter to handle a Change process are described in the Table 2-1.

Table 2-1 Workflow process modeled in CCMDB

Process	Workflow Name	Description
Change Management	PMCHGITLV3	Workflows that drive the whole process.
Accept	PMCHGACCV3	Subprocess to accept and categorize
Assess	PMCHGASSV3	Subprocess to assess
Schedule	PMCHGSCHV3	Subprocess to schedule the activities
Authorize	PMCHGAUTH	Subprocess to authorize
Implement	PMCHGIMPV3	Subprocess to coordinate
Review/Close	PMCHGREVV3	Subprocess to review and close a Change

Observe that these workflows are examples to an end-to-end process execution. They can be adapted to the environment needs, or even be changed by new ones to attend in a properly way the client needs.

A Change Object is the focus of the presented workflows, and this object is represented by the Figure 2-14 (WOCHANGE), which was created by the previous RFC. It is possible to view the actual phase of the Change in the *Process Map* frame. Actually, the Change described by the figure has not been submitted to the Change workflow process.

ListChangeAssessmentsImpactsAuthorizationScheduleWorkplan MapRelated RecordsActualsLog

Progress Map

ACC_CAT

ASSESS

SCHED

AUTH

IMPL

INPRG

COMP

REVIEW

CLOSE

Change: 1177

Owner:

Owner Group:

Status: WAPPR

At

Change Details

Summary: Change the DB2 Database Server

Change Type: Normal

Reason for Change:

Effect Of Not Implementing:

Verification Plan:

Remediation Plan:

Risk:

Failure Probability: 2

Impact: 1

Urgency: 1

Priority: 1

Change Category:

Figure 2-14 Change example to replace DB2 Database server.

The Change Management workflow is the main workflow definition to manage the whole Change process. This workflow is used to control the Change Object (WOCHANGE) in its whole lifecycle.

Figure 2-15 represents the Change Management workflow process provided by the CCMDB tool. This workflow is divided into the following activities, mapped as subprocess: *Accept, Assess, Schedule, Authorize, Implement and Review*. Each subprocess must be finished with success before going to the next one. After all subprocesses finish, the Change is finished. If a certain subprocess is not completed with success, the workflow maps all the negative paths to the correct process phase to handle the return properly.

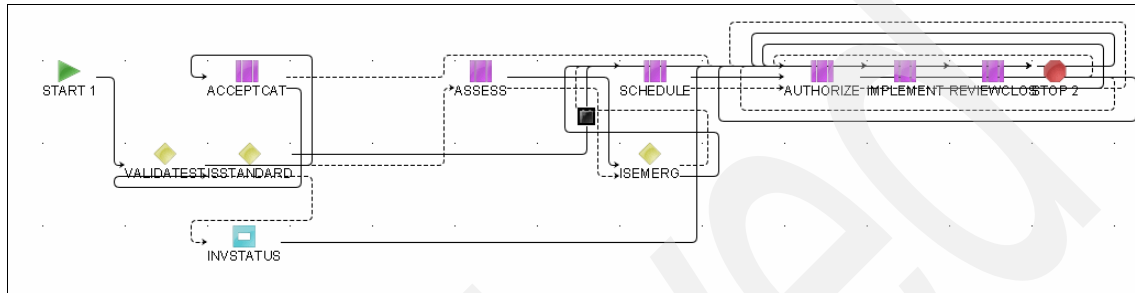


Figure 2-15 Change Management master workflow process

Figure 2-16 describes a subprocess node in the workflow.

Figure 2-16 Accept and Categorize subprocess node

Accept and categorize

Accept and categorize is the main activity that classifies the type of the Change, considering its risk to the IT and business environment, leveraging the impact, urgency and priority information.

The workflow responsible for that phase is represented by Figure 2-17. After the workflow starts, it changes the status of the Change Object to *Accept and Categorize* by applying the first action provided by the connection line between the START and VALIDATE nodes. The first node is a subprocess to filter, validate, and accept a Change. The second subprocess classifies the Change, giving the risk number. Observe that a subprocess can have a positive and a negative output. If the CLASSIFY subprocess completes successfully, then the workflow proceeds to the STOP node, otherwise, it goes back for special treatment.

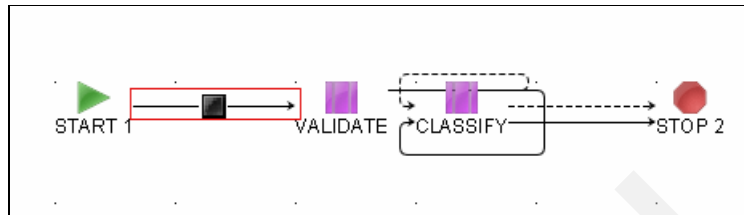


Figure 2-17 Accept and Characterize workflow

Figure 2-18 show the action to change the status of the current Change involved in the workflow process.

Figure 2-18 Action to change a status to Accept and Categorize

The subprocess classify automatically calculates the risk of this Change based on the information previously provided in the Change tab and also in the RFC. Figure 2-19 shows the dialog box that displays after entering in the accept and categorize phase.

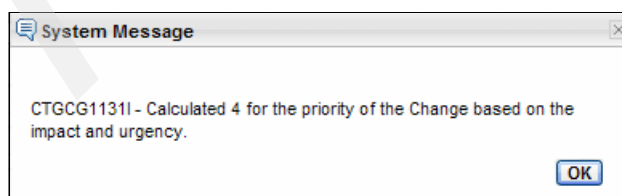


Figure 2-19 Automatic risky calculation provided by the workflow

This workflow automatically applies, considering the type of the Change selected; one of the following template job plans is provided by the CCMDDB: Normal Change job plan (PMCHGNORML) for normal Changes, Standard Change Job Plan (PMCHGSTAND) for standard Changes, and Emergency Change Job Plan (PMCHGEMERG) for Emergency Changes. In this example, it was automatically selected as the emergency job plan.

Figure 2-20 represents the Change in the initial phase of the Change Management process: accept and categorize. Observe that the current status is mapped in the Progress Map frame.

The screenshot shows the IBM Tivoli Change and Configuration Management Database (CCMDDB) interface. At the top, there is a navigation bar with tabs: List, Change, Assessments, Impacts, Authorization, Schedule, Workplan Map, Related Records, Actuals, and Log. The 'Change' tab is active. Below the navigation bar is the 'Progress Map' section, which shows a workflow diagram with steps: ACC_CAT, ASSESS, AUTH, IMPL, INPRG, COMP, REVIEW, and CLOSE. The 'ACC_CAT' step is highlighted. Below the progress map, there are input fields for 'Change: 1177', 'Owner: MAXADMIN', 'Owner Group:', and 'Status: ACC_CAT'. The 'Change Details' section is below, containing a 'Summary' field with the text 'Change the DB2 Database Server'. To the right of the summary is a 'Risk' field with the value '2'. Below the summary are several other fields: 'Change Type: Emergency', 'Reason for Change:', 'Effect Of Not Implementing:', 'Verification Plan:', and 'Remediation Plan:'. To the right of these fields are 'Failure Probability: 1', 'Impact: 1', 'Urgency: 1', 'Priority: 1', and 'Change Category:'. Each of these fields has a magnifying glass icon next to it.

Figure 2-20 Accept activity to change a DB2 Database Server

The job plan applied brings pre-defined information about the Changes to be executed. In the CCMDDB V7.2.1 version, job plans have specialized tabs to automate Change process steps. Figure 2-21 shows the Change Template tab for an emergency job plan. This job plan tab brings information about the reason for Change and Verification procedures, describes a back out plan, and specifies if the job plan has a workflow that is fully automated. When this option is checked, the workflow can skip any manual tasks to fully automate the Change tasks.

The screenshot shows the 'Job Plans' application interface. At the top, there's a navigation bar with 'Go To' and links for Reports, Start Center, Profile, Sign Out, and Help. Below this is a toolbar with various icons. The main area has tabs for List, Job Plan, Work Assets, Specifications, Change Template, Change Assessments, and Change Authorization. The 'Change Template' tab is active, showing a form for 'Emergency Change Jobplan'. The 'Job Plan' field is set to 'PMCHGEMERG' and the 'Change Type' is 'Emergency'. The 'Details' section at the bottom has fields for 'Reason for Change' (Emergency change), 'Verification', 'Back Out Plan', and a 'Fully Automated?' checkbox which is checked.

Figure 2-21 Emergency job plan example

Assess

Assess is the phase in which you can measure or evaluate all the impacts that a Change has in the IT environment. CCMDB V7.2.1 provides two types of assessments: Business and IT. Business assessments focus on describing the impact of implementing (or not implementing) a Change to the company business, while IT assessment evaluates the impact of that Change in the IT infrastructure.

In this case example, the Emergency Jobplan does not provide any assessment checklist for a Change classified as emergency. However, when applying a normal Change, the Job Plan Application provides a template for classifying the impacts for that Change. Figure 2-22 describes the assessments for a Change when the Normal Job Plan (PMCHGNORML) is applied.

The screenshot shows two tables for assessments. The top table is 'Technical Assessments' and the bottom is 'Business Assessments'. Both tables have columns for 'Assessment Type', 'Impact', and 'Results'. The 'Technical Assessments' table lists OS, Application, Server, Security, Storage, Capacity, and Network. The 'Business Assessments' table lists SOX and Financial.

Assessment Type	Impact	Results
OS		
Application		
Server		
Security		
Storage		
Capacity		
Network		

Assessment Type	Impact	Results
SOX		
Financial		

Figure 2-22 Assessments measurements for a Normal Change provided by the Normal Job Plan template

When the Normal Job Plan template is applied to a Change, all the assessment evaluations are copied to the Change work plan. To proceed to the next step of

the workflow, it is necessary to evaluate the risks for the assessments presented, because the workflow uses these values to calculate the risk and assessment risks to the environment.

Figure 2-23 shows the assessment workflow for ITIL V3 provided by default in the CCMDB V7.2.1 tool.

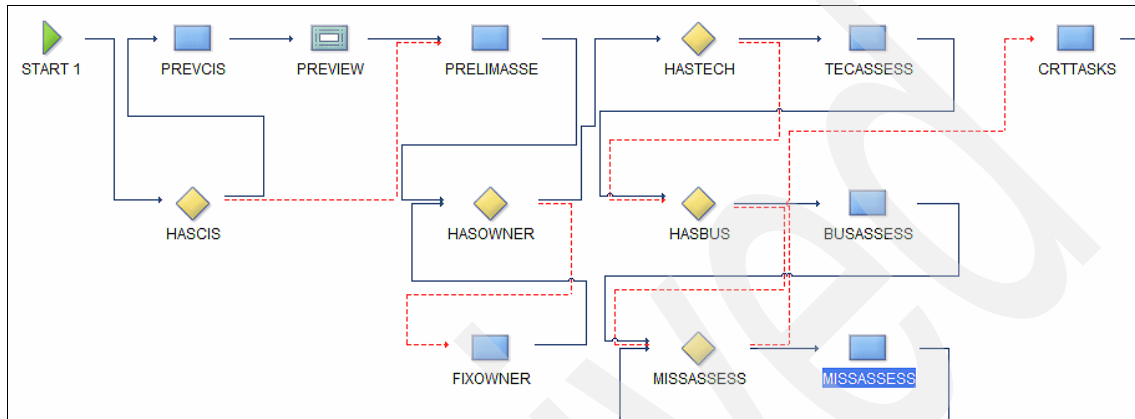


Figure 2-23 Assessment workflow

The workflow starts by showing a preview (PREVIEW node) for the Change owner about all impacted CIs related to the Change. The PREVIEW *interactive action* executes a specific action provided by the Change Application. This action shows the impacted CIs calculated by the relationships provided in the target CI and all CIs related.

Figure 2-24 describes the action *Preview Impacts* for the Change, which is showed automatically by the assessment workflow. This analysis can also be viewed by the action *Preview Impacts*, located at **Select Action** → **Impact Analysis** → **Preview Impacts**.

Preview Impacts of the Change					
Calculated Historical Filter 1 - 2 of 2 Download					
List of CIs impacted by Change.					
Impacted CI	Impacted CI Name	Business Impact	Impacted Description	Target CI Causing Impact	Target CI Name
10006	WEBSPPHERE ND		WebSphere app on Miami Linux WAS server	10004	MIALINUXWAS01
DB2 INSTANCE	MIAMI DB2		Software Image, DB2, 9.5	10003	MIALINUXDB2

Figure 2-24 Impacted CIs view

Another important point in the workflow consists of the designations rules for the technical and business assessments owners, provided by the TECASSESS and BUSASSESS nodes, which create designations for each technical and business assessment owners to calculate their impacts.

Figure 2-25 shows the task node for the technical assessment owners to classify the impacts of their assessments for the Change. A designation can be created to a specific user or to a role: a user or a set of users responsible to execute the designated task. The roles and users defined to do a technical and business assessments can be defined in the job plan, as previously demonstrated in Figure 2-11 on page 53.

The workflow remains stopped in the designation point until the assignments are accepted or rejected by the users. The workflow administrator has the option to proceed when any assignment is accepted or all assignments are accepted.

Task Node Properties

Title: Time Limit:

Application: Display One? ☒

WF Task Type:

Assignments Filter > 1 - 2 of 2 Download ?

Role ID	Description	Send E-mail	Task Description	Time Limit
▶ PMCHGTEC2 >>	Owners of the pending technical assessments	☐	Provide Technical Assessment Impact Score o	0:00
▶ PMCHGTECA >>	Owners of the pending technical assessments	☐	Provide Technical Assessment Impact Score o	0:00

Notifications Filter > Download ?

 Send To

...No rows to display...

Perform Accept Action

☐ When any assignment is accepted
☒ When All assignments are accepted.

Figure 2-25 Task node to create designations for technical assessment owners

Each designation is created to the users described by the roles. Each designated user can visualize the assignment in the user's Start Center application, specifically in the Inbox/Assignments section (Figure 2-26).

Inbox / Assignments				
Next Assignment Due: 09/06/10 00:24 Refresh				
Description	APP	DESCRIPTION	DUE DATE	PRIORITY Route
Some Change assessments are missing impacts for Change 1005, please complete them.	CHANGE	Some Change assessments are missing impacts for Change 1005, please complete them.	28/05/10 07:35	
Some Change assessments are missing impacts for Change 1130, please complete them.	CHANGE	Some Change assessments are missing impacts for Change 1130, please complete them.	08/06/10 11:26	
Determine whether Change 1177 is within our policy	CHANGE	Determine whether Change 1177 is within our policy	09/06/10 00:24	
1 - 3 of 3				

Figure 2-26 Designations for impacts assessments

Schedule activities and tasks

Schedule is the phase in which activities and tasks are mapped to skilled professionals at a specific date and time to execute the Change. For example, the scheduled workflow phase can be observed in the normal Change (Normal Job Plan), while an Emergency Change does not provide it by default.

Figure 2-27 represents the Normal Job Plan, which defines a process template type for the Change work order class.

Job Plan: PMCHGNORML **Normal Change JobPlan**

Organization:

Details

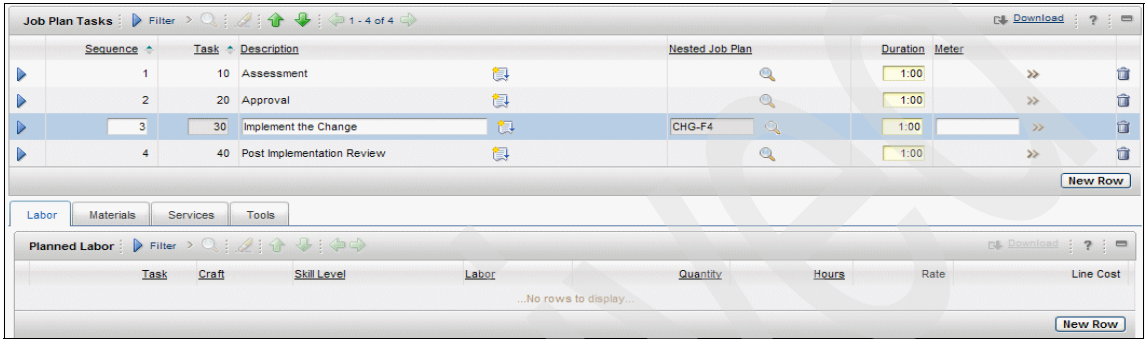
Status: ACTIVE	Default WO Class: CHANGE
* Template Type: Process	WO Priority:
* Duration: 4:00	Interruptible? ☐
Classification:	Flow Controlled? ☒
Class Description:	Suspend Flow Control? ☐
Launch Entry Name:	Flow Action:
Include Tasks in Schedule? ☒	Flow Action Assist? ☐
Start Constraint Offset:	
Finish Constraint Offset:	
Process Flow Sequence:	

Figure 2-27 Normal Change job plan: process template for Change work order class

Each job plan template has a header section that applies to the work order object, while each task has its own detailed definition. The task definition is mapped for the child work order object created. Remember that a child work order object is created after a job plan template has been applied to the work order object.

The activities and tasks are inherited from the job plan template, and have information such as duration, ICs impacted, professionals necessary, and the data of the expected activity to start. When all of this information is loaded to the Change, it is expected to designate all the resources necessary to execute the Change.

Figure 2-28 exemplifies the job plan tasks provided by the Normal Job plan. The tasks described are assessment, authorization, implementation, and post implementation review.



The screenshot displays the 'Job Plan Tasks' window. It features a table with columns: Sequence, Task, Description, Nested Job Plan, Duration, and Meter. The table contains four rows of tasks. Below the table is a 'Planned Labor' section with tabs for Labor, Materials, Services, and Tools. The Labor tab is active, showing a table with columns: Task, Craft, Skill Level, Labor, Quantity, Hours, Rate, and Line Cost. The Labor table is currently empty, displaying the message '...No rows to display...'. The interface includes various icons for filtering, saving, and downloading data.

Sequence	Task	Description	Nested Job Plan	Duration	Meter
1	10	Assessment		1:00	>>
2	20	Approval		1:00	>>
3	30	Implement the Change	CHG-F4	1:00	>>
4	40	Post Implementation Review		1:00	>>

Planned Labor

Task	Craft	Skill Level	Labor	Quantity	Hours	Rate	Line Cost
...No rows to display...							

Figure 2-28 Normal Change job plan tasks

In the lower section of the job plan template window are additional tabs such as Labor, Material, Services, and Tools. Though these capabilities stem from the Enterprise Asset Management world in order to specify what people, services, or tools are needed to perform the Change, you can use them in the IT world as well.

It is important to keep in mind that these tasks are work order objects (activities and tasks) to register information related to each phase of the workflow process. The objective of these tasks is not to replace the workflow orchestration mechanism, but only to register data related to the phase of these tasks when necessary.

After these tasks are associated to a Change by applying a job plan, each task creates a work order object, which type reflects the level of association defined in the job plan. Commonly, job plans applied directly to the work orders objects are the type of Process ones (Figure 2-27 on page 64), which generates activities objects (WOACTIVITY) for each task described. The normal Change job plan tasks (Figure 2-28) are mapped into activities when applied in a Change Object.

Figure 2-29 represents the tasks created for a normal Change using the Normal Change job plan. Each job plan task was turned into an activity object (WOACTIVITY).

The screenshot shows the 'Schedule' tab for a change plan. The 'Progress Map' section includes fields for Target Start, Target Finish, Estimated Duration (4:00), Start No Earlier Than, Finish No Later Than, Scheduled Start, Scheduled Finish, Time Remaining, Actual Start, Actual Finish, and Scheduler Project. The 'Tasks for Change 1214' section displays a table of tasks:

Sequence	Task	Summary	Estimated Duration	Status	Owner	Owner Group
1	10	Assessment	1:00	WAPPR		
	20	Approval	1:00	WAPPR		
4	40	Post Implementation Review	1:00	WAPPR		

Figure 2-29 Schedule plan for a normal Change

The nested CHG-F4 is an activity job plan type, and it generates objects of the type specified in the default WO class. Figure 2-30 shows the nested job plan description for CHG-F4. When a job plan has a task followed by a nested job plan, this means that this task is mapped to a work order object followed by the tasks described by the nested job plan.

The screenshot shows the 'Job Plan' tab for the nested CHG-F4 plan. The 'Details' section includes the following fields:

- Job Plan: CHG-F4
- Implement the Change
- Organization:
- Status: ACTIVE
- Template Type: Activity
- Duration: 0:00
- Classification:
- Class Description:
- Launch Entry Name:
- Include Tasks in Schedule? ☒
- Start Constraint Offset:
- Finish Constraint Offset:
- Process Flow Sequence:
- Default WO Class: ACTIVITY
- WO Priority:
- Interruptible? ☐
- Flow Controlled? ☒
- Suspend Flow Control? ☐
- Flow Action:
- Flow Action Assist? ☐

Figure 2-30 Nested CHG-F4 plan

For each task or activity generated, there is a set of options to automate the process flow. One of the most important flags in the header section is the Flow Controlled? check box. Make sure the check box is checked; otherwise the work order plan does not automatically start the children and tasks of the work order object. Each task has a Flow Controlled? check box as well. Make sure it is checked.

By default, the check box is checked as needed. The flow control specification determines if the task participates in the enforced sequencing.

Because the top level job plan template is defining the activities of the process flow, the task definition refers to a nested job plan. You can see that the Implementing Change phase is referring to the nested job plan CHG-F4.

We explain the other most relevant fields of the task definition section while we walk through concrete examples in the rest of this chapter.

Authorize (Approval)

The *Authorization* phase is the Change activity that decides the “go/no go” for the Change. In the authorization phase, certain activities information has already been registered and authorized by the respective owners. These activities are classification, categorization, assessments impact, and schedule implementation activities.

After these activities are defined, it is in the authorization phase that a specific group of users are designated to authorize the Change to be finally implemented.

ITIL suggests various boards to authorize a Change depending on the impact and urgency categorized on it. For example, a standard Change (pre-authorized) does not require any authorization to proceed, while normal Changes require the Change Advisory Board (CAB) authorization. For Emergency Changes, the Emergency Change Advisory Board (ECAB) enters an action to handle the situation.

In CCMDB the situation is similar. The group responsible to authorize a Change before it proceeds can be defined in the Change Object. A job plan can define a template to apply the groups necessary to approve a Change, followed by the level necessary to a specific group to be scheduled for the authorization phase. Figure 2-31 shows the groups responsible to approve a normal Change, followed by its level definition.

List	Job Plan	Work Assets	Specifications	Change Template	Change Assessments	Change Authorization
Job Plan: PMCHGNORML		Normal Change JobPlan				Organization:
Change Type: Normal						
Other Approvers Filter > [Search] [Add] [Remove] [Refresh] 1 - 3 of 3						
Description		Approver		Approver Group		
CAB				PMCHGCAB		
IT Management Board				PMCHGTM		
Business Executive Board				PMCHGBUS		

Figure 2-31 Normal Change job plan authorization groups

When a job plan is applied to a Change, the authorization information is copied to the Approval tab in the work order Change Object. Figure 2-32 shows the information copied from the Normal Change job plan to the generated Change. In the schedule phase, it is possible to request specific approvers for the Change.

Approvers for Change 1214 Filter > [Search] [Add] [Remove] [Refresh] 1 - 3 of 3					
Description		Approver		Approver Group	
CAB				PMCHGCAB	
IT Management Board				PMCHGTM	
Business Executive Board				PMCHGBUS	
Schedule Approvers for Change 1214 Filter > [Search] [Add] [Remove] [Refresh] 1 - 1 of 1					
Description		Approver		Approver Group	
Authorization target				PMCHGCAB	
				Type	
				Blackout	
Approval Logs Filter > [Search] [Add] [Remove] [Refresh]					
Summary		Decision		Person	
				Transaction Date	
...No rows to display...					

Figure 2-32 Approvers designated for a normal Change

CCMDB provides, by default, workflows for the authorization process. Figure 2-33 shows a piece of the Change authorization process workflow (named PMCHGAUTH) defined in the CCMDB.

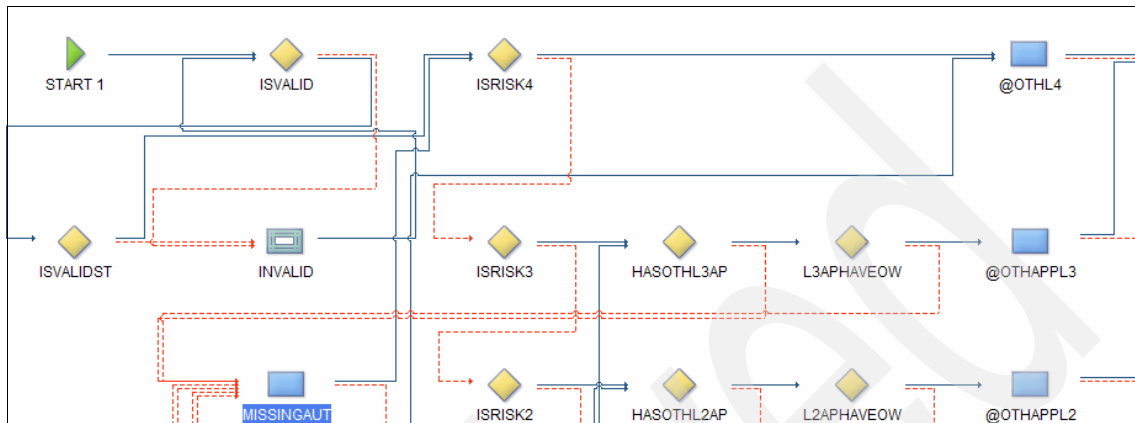


Figure 2-33 Workflow for authorization process

The workflow PMCHGAUTH starts checking for a valid (ISVALID) owner. If it is valid, it checks the risk level of the Change. Considering the approvers exemplified by Figure 2-32 on page 68, if a Change has the following risks:

- ▶ If low risk (ISRISK4): It designates to the Change owner.
- ▶ If medium risk (ISRISK3): It designates to the approvers level 3; CAB group.
- ▶ If high risk (ISRISK2): It designates to the approvers level 2; IT Management board.
- ▶ If critical (ISRISK1): It designates to the approvers level 1; Executive Business board.

It is important to observe that this workflow designates the Change directly to the level of authorization targeted, not passing by lower levels. If the client wants to designate sequentially from the lowest level to the targeted level, there is another workflow that simulates this situation, named PMCHGAP.

Figure 2-34 exemplifies this workflow. Observe that in this workflow, after the designation is authorized for one level, it goes sequentially to higher levels, until it reaches the targeted designation level.

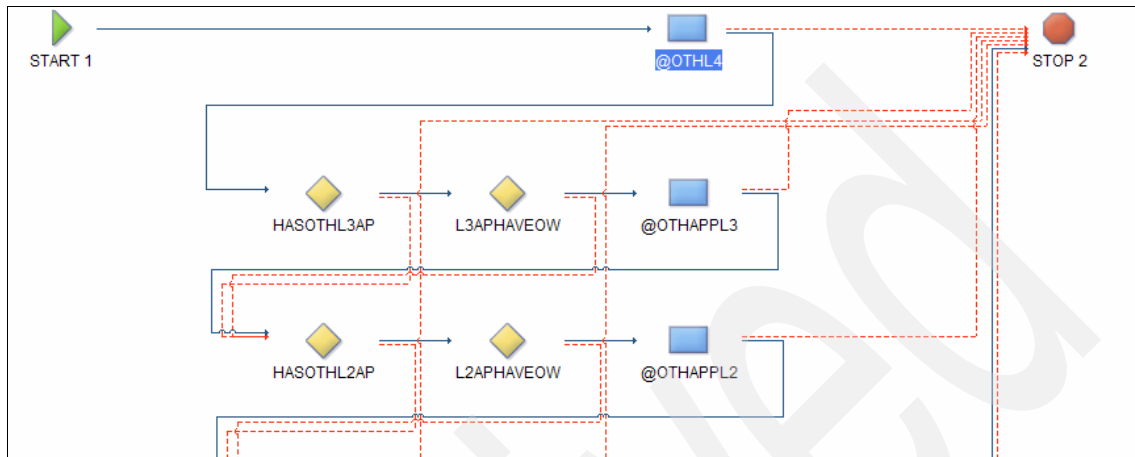


Figure 2-34 Workflow for hierarchical authorization process

Implementation

After the Change has been already scheduled and authorized, it is time to implement it. The activities and tasks defined in the Change Record are the steps and procedures necessary to implement the Change.

The tasks and activities information can be found in the Work Plan map tab of the Change Object. Each activity and task are mapped into work order objects which are designated to skilled resources in the proposed time.

Figure 2-28 on page 65 describes a list of tasks to be executed during the implementation. Figure 2-35 here shows the Work Plan tab that graphically represents these activities to be implemented. There is a navigation tree, which describes the nodes and the predecessors for these tasks.

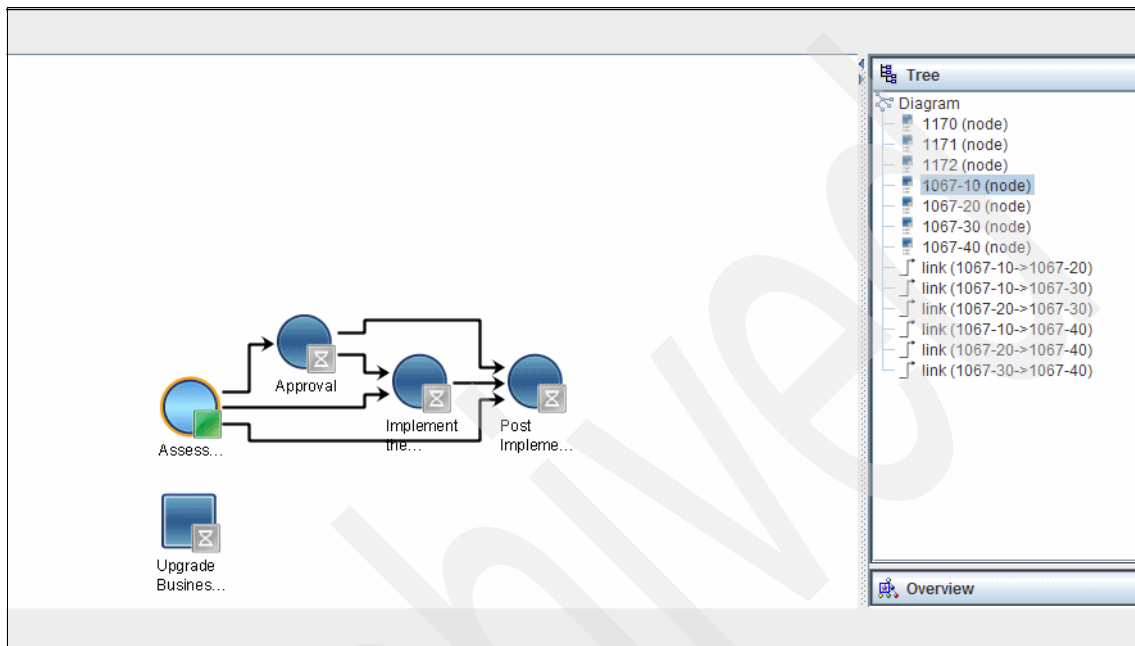


Figure 2-35 Work plan tab for normal Change

These links are built from the tasks predecessors information. From the work plan, it is possible to navigate to the record that maps the activity and its tasks. Figure 2-36 represents the Activity and Tasks application showing the activity work order record generated by the normal Change job for the Implementation task (Figure 2-28 on page 65).

List	Activities and Tasks	Resources	Related Records	Log
CHANGE 1067 Attachments				
Task: 30 Implement the Change		Activity Type:		
Location: >>		Priority: 2		
Asset: >>		Reported By: ADRIAN		
Configuration Item: >>		SLA Applied?		
Configuration Item Name:		Under Flow Control? ☒		
Target Description:		Suspend Flow Control?		
Classification: >>		Flow Action: PLUSPACTAPI >>		
Class Description:		Flow Action Assist?		
Launch Entry Name: >>		Implementation Task? ☒		
Parent Process: 1067 >>		Status: WAPPR		
		Status Date: 6/1/10 9:38 PM		
		Service Group: >>		
		Service: >>		
		Owner: PMCFGADM		
		Owner Group:		
		Assisted Workflow: >>		

Figure 2-36 Activity work order record generated for the Change

In Figure 2-36, the task is defined as an implementation task by the implementation flag. When a task is defined as an implementation task, it defines a task that creates release work orders to implement Changes in the IT environment. Releases are detailed procedures to be executed in the IT environments.

Figure 2-37 shows more information about the Implement the Change task. It shows the schedule information and its predecessors relationship which needs to be completed before executing this step. This activity has two predefined predecessors, which are tasks 10 and 20. By default, these predecessors tasks need to be completed before starting the current task. If one of these predecessors tasks is an activity, all the tasks inside the activity have to be completed before the first task inside the Implementing activity can start.

Scheduling Information	
Target Start:	
Target Finish:	6/10/10 1:15 PM
Scheduled Start:	
Scheduled Finish:	
Actual Start:	
Actual Finish:	
* Estimated Duration:	1:00
Predecessors:	10,20 >>
Start No Earlier Than:	
Finish No Later Than:	

Figure 2-37 Activity schedule and predecessors sequence

When defining tasks predecessors, the task can define the kind of relationship between the current work record and its predecessors. This gives more flexibility to run tasks in parallel. Figure 2-38 demonstrates the tasks predecessors relationships provided by the application.

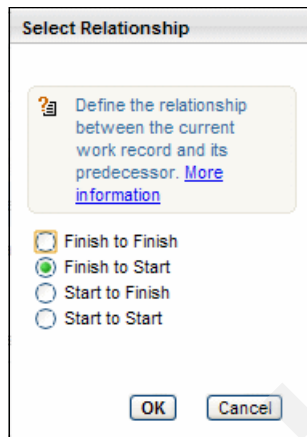


Figure 2-38 Predecessors relationship

There are four kinds of relationships:

- ▶ “The predecessor task needs to finish” to “the following task also finishes”
- ▶ “The predecessor task needs to finish” to “the following task starts (default)”
- ▶ “The predecessor task needs to start” to “the following task finishes”
- ▶ “The predecessor task needs to start” to “the following task also starts”

In an activity and a task, you can take benefit from predecessors definitions and automate completely the process by specifying actions and assisted workflows to be executed when the task is set to in progress. Remember that we are being orchestrated by the implementation workflow, however, other workflows and actions can be launched to control small activities and tasks. For example, in Figure 2-36 on page 72, the action that applies a response plan (called PLUSPACTAPPRESPL) is specified in the Flow Action field. This action is executed when this task is set into progress.

If the Flow Action Assist? check box is checked, the action will not run automatically when the task is set into progress. An example might be if you are required to start the action manually by pushing an additional button that you have defined on the application user interface. By default, the Flow Action Assist? check box is unchecked.

If the Suspend Flow Control? check box is checked, the automation of the task is suspended until it gets resumed again.

Next, we explain how to define a task that calls out to a workflow definition. The main reason to call out to a workflow is to guide the user within the process flow through interactive automation, for example, by requiring a decision or routing the user to another application. Another reason to use the workflow technology is that if interactions with other processes are required. workflows are usually triggered if a particular user activity needs to run in the foreground.

If a task needs to call out to a workflow, the workflow name needs to be specified in the Assisted Workflow field of the task definition. In Figure 2-39, the activity implementation task starts another process by assisted workflow, which starts a release to distribute software (PMRELSWDST).

The screenshot shows the 'Implement the Change' task definition form. The 'Task' field is set to '30' and 'Implement the Change'. The 'Parent Process' is '1067'. The 'Assisted Workflow' field is set to 'PMRELSWDST', with a 'Start Assisted Workflow' button next to it. Other fields include 'Location', 'Asset', 'Configuration Item', 'Target Description', 'Classification', 'Class Description', 'Launch Entry Name', 'Activity Type', 'Priority' (2), 'Reported By' (ADRIAN), 'SLA Applied?', 'Under Flow Control?' (checked), 'Suspend Flow Control?', 'Flow Action', 'Flow Action Assist?', and 'Implementation Task?' (checked). The 'Status' is 'WAPPR' and the 'Status Date' is '6/1/10 9:38 PM'. The 'Owner' is 'PMCFGADM'.

Figure 2-39 Activity with an assisted workflow

If we look at the graphical view of the workflow definition in the Workflow Designer application, it shows that the workflow has an interaction node definition with the name NEWSWDST (Figure 2-40). An interaction node is used to route a user to another application in the system.

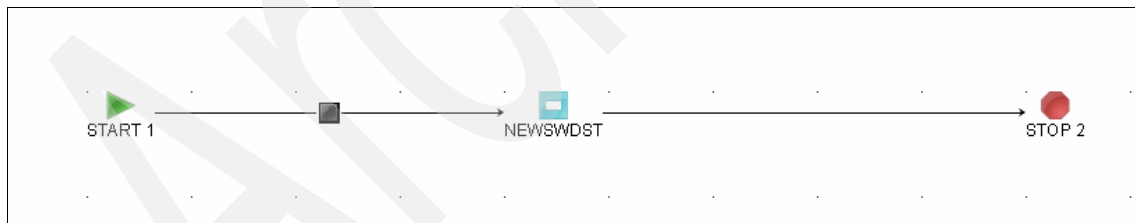
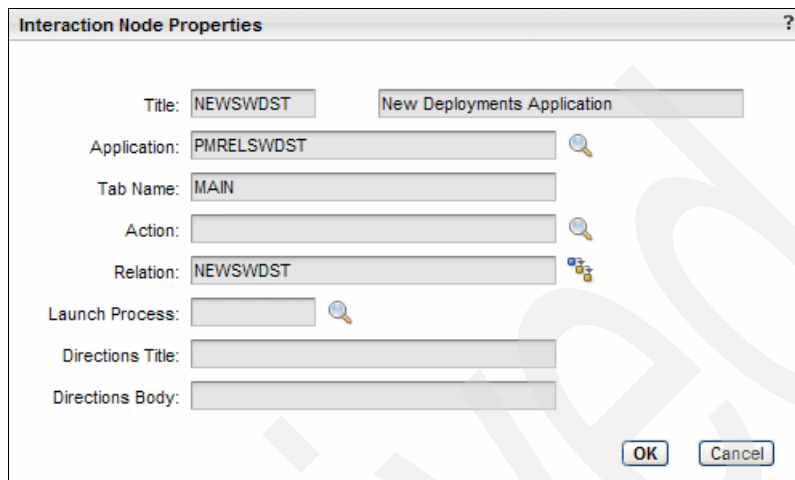


Figure 2-40 Launch to Deployments Application from task

The properties of the interaction node (right-click the icon) show the definition shown in Figure 2-41. Observe that this workflow directs the user to the release and distribution application (PMRELSWDST) from an activity.



The dialog box titled "Interaction Node Properties" contains the following fields:

- Title: NEWSWDST (with a secondary field "New Deployments Application")
- Application: PMRELSWDST
- Tab Name: MAIN
- Action: (empty)
- Relation: NEWSWDST
- Launch Process: (empty)
- Directions Title: (empty)
- Directions Body: (empty)

At the bottom right are "OK" and "Cancel" buttons.

Figure 2-41 Workflow Interaction Node Properties

The property definition shows that the user is guided to the New Deployments application when this task of the workflow is initiated. It is important to observe the Relation field, which defines the relationship to bring the object to the current application.

Actions: A workflow can use an action or action groups inside its execution model. It is preferable to call an action from a workflow rather than calling the task directly from the task definition of the jjob plan if, for example, we require a conditional check before a decision can be made as to which action to run.

Using the activities and tasks structure, it is possible to implement the Change, supported by the workflow automation process.

After all activities and tasks for a Change are completed, it is time to finalize the Change implementation phase. To better illustrate the implementation phase, Figure 2-42 shows the workflow process that coordinates the Change implementation. The WAITCOMP node waits for the Change Record to be completed before proceeding to the next steps.

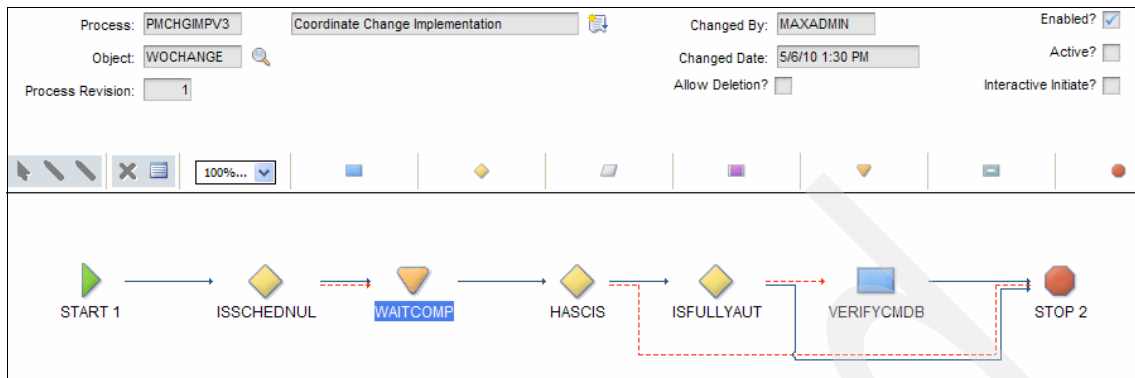


Figure 2-42 Coordinate Change implementation workflow

If the flow control is enabled for the tasks in the Change, it is completed only when the last task is completed. After the Change status is changed to complete, the implementation workflow can call the wait node to proceed to the next step. Figure 2-43 shows the wait node properties, which waits until the WOCHANGE current object is in the completed or closed status.

Wait Node Properties

Title: WAITCOMP Wait for the Change Implementation to complete

Wait List: Filter > 1 - 2 of 2

Event
maximo.wochange.statuschange.close
maximo.wochange.statuschange.comp

Notifications: Filter > ...No rows to display...

Figure 2-43 Wait for the Change Implementation to complete

The last step in the implementation phase is to check if the CMDB was updated with the Changes done in the IT environment. This task is handled by the Configuration discipline. After the configuration activities are completed, it is time to close the Change.

Close and review

Close and review is the final step of the Change Management process. This workflow creates assignments to verify if the Change was successfully implemented to the stakeholders.

If the Change was not completed with success, it is closed with the status, failed. In this case, the Change owner can create a follow up process to correct the gaps necessary to complete the Change objectives.

Figure 2-44 shows the workflow responsible to close and review the Change. The VERIFY designation verifies with the stakeholders if the Change is implemented as desired. If the stakeholders answer yes, the workflow finishes, and the master workflow for the Change process finishes as well.

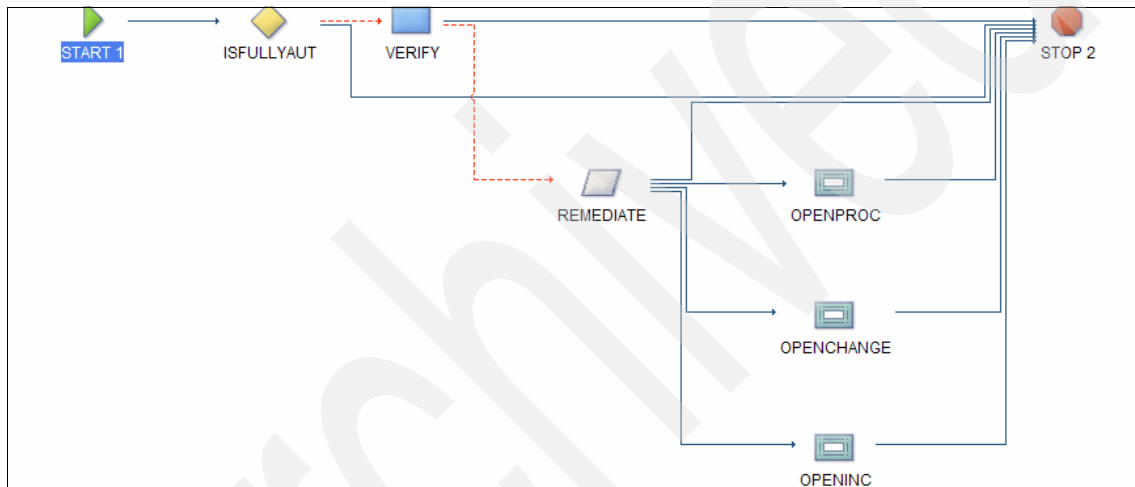


Figure 2-44 Close and review workflow process

If VERIFY is answered negatively, the workflow requires a manual input mode represented by the REMEDIATE node.

A manual interaction is a node that requires manual user interaction to choose the next step to be executed. In this case, the manual interaction gives three options: open a process (OPENPROC), open another Change (OPENCHANGE) or open an incident (OPENINC).

Figure 2-45 shows the input mode node and its possible actions to be taken. When this node is executed by the workflow, it will pop up a dialog box with the options described in the Sequence defined.

Title: Remediate the Change

Display One? ☐

Sequence	Action	Instruction	To Node	Positive
1	>>	Open Follow Up Incident	OPENINC	☒
2	>>	Open Process Request	OPENPROC	☒
3	>>	Open Follow Up Change Workorder	OPENCHANGE	☒
4	>>	Do Nothing	STOP 2	☒

Notifications

Communication Template

Send To

...No rows to display...

New Row

Figure 2-45 Manual input mode properties

After closing the Change, the Change Management process finishes, as we can observe in Figure 2-15 on page 58.

2.3.3 Configuration process

The configuration process is the mechanism to update the Changes done in the IT infrastructure to the CMDB database, and is executed after the task for the Change is being completed. The final example is another one explaining how to define an assisted workflow as part of the task definition to interact two processes.

This time our intention is to highlight the interaction between the Change and Configuration Management process flows. In Figure 2-46, the CI Data Update task of the Implement the Change job plan template calls out to an assisted workflow called PMCHGCRAWF. This informs the Configuration Manager of the Changes that have been performed by Change Management.

The Configuration Manager is in charge of updating the data in the CCMDDB in a controlled way. The assisted workflow is interactively guiding the user of Change Management through the process of generating a process request to Configuration Management.

Sequence	Task	Description	Nested Job Plan	Duration	Meter
10	Implements the change			0:00	>>
20	Verify Implementation			0:00	>>
30	CI Data Update			0:00	>>
40	Document Results			0:00	>>

Details

Organization:
Site:
Task: CI Data Update
Sequence:
Nested Job Plan:
Classification:
Class Description:
Launch Entry Name:
Implementation Task? ☐

Duration:
Meter:
Owner:
Owner Group:
Flow Controlled? ☒
Suspend Flow Control? ☐
Flow Action:
Flow Action Assist? ☐
Assisted Workflow:
Predecessors:

Attachments

New Row

Figure 2-46 Change Management generates a Process Request to Configuration Management

The definition of the PMCHGCRWF workflow in the workflow designer application is shown in Figure 2-47.

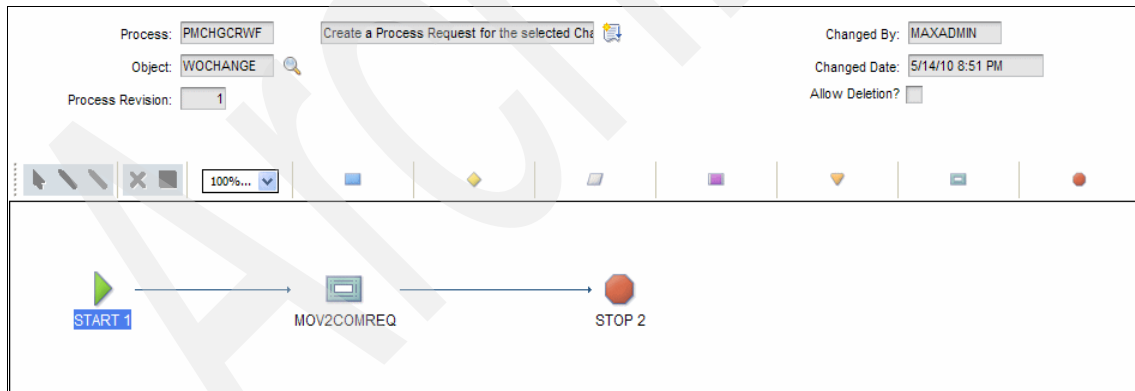


Figure 2-47 Definition to generate Process Request Configuration Management

The workflow is using an interaction node called MOV2COMREQ that interactively guides the user to the process request application in order to let the user fill out the details of the process request definition into Configuration Management.

This is an example of an interaction between processes of various types. Remember that a process interaction always requires a process request before a work order object can be generated.

Another good example for a predecessor definition is shown in Figure 2-48.

Sequence	Task	Description	Nested Job Plan	Duration	Meter
50	Update Change Progress to Implemented			1:00	

Details

Organization: Site: Task: 50 Update Change Progress to Implemented

Nested Job Plan: Classification: PMREL \ PMRELTsk \ PMRELCPR Class Description: Change Progress task Launch Entry Name: Implementation Task? ☐

Duration: 1:00 Meter: Owner: Owner Group: Flow Controlled? ☒ Suspend Flow Control? ☐ Flow Action: PMCHGPREVOL Flow Action Assist? ☒ Assisted Workflow: Predecessors: 10,20,30,40

[Attachments](#) [New Row](#)

Figure 2-48 Predecessor definitions in task definition

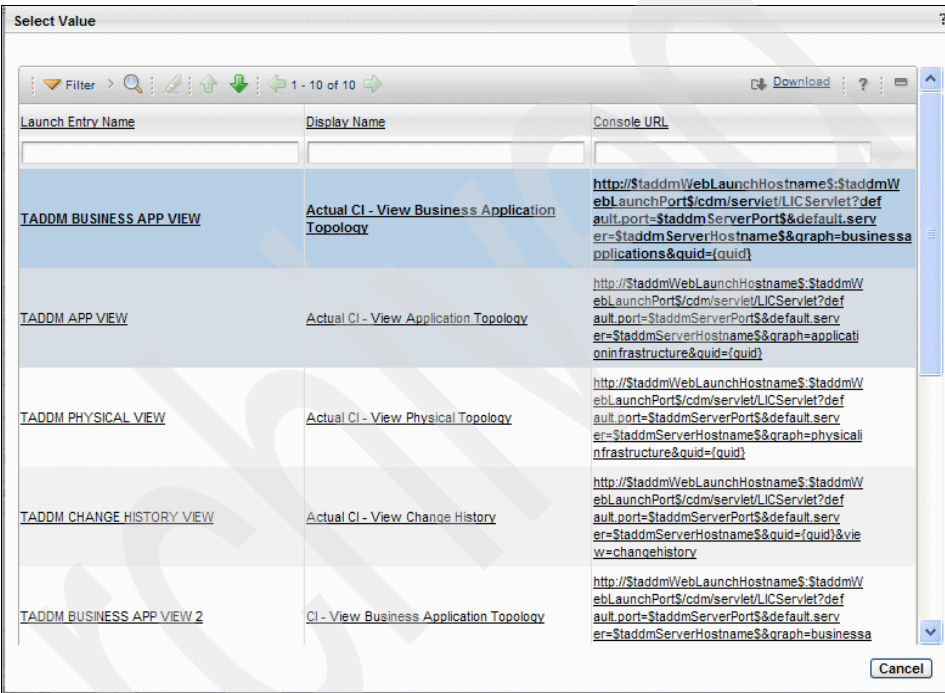
The Update Change Progress to implemented task of the Implement the Change job plan template is the final task defined in the overall sequence. The predecessor field reveals that all other tasks have to be completed before the task can be started. It cannot start in parallel to any other task of this job plan.

So far we have explained how a task can define various types of automation inside the job plan template. We have shown how to call out to an action, a workflow, as well as a workflow that guides the user interactively through the process of generating a process request.

Sometimes it is necessary to have the support of an external system within the overall process flow. The external system, for example, holds valuable information that is not synchronized into the CCMDB, such as current status information about a configuration item. We refer to external systems usually as *Operational Management Products (OMPs)*.

If, for example, during a Change review you want to verify if your Changes took effect, you can Launch in Context to an operational management product such as IBM Tivoli Monitoring to check if the status of your CI has been changed as intended. To do so, in the task definition, you need to define a value in the Launch Entry Name field. This requires an entry of a Launch in Context definition that is defined in the Launch in Context application.

Figure 2-49 reveals the values of definitions that have been defined in the Launch in Context application.



Launch Entry Name	Display Name	Console URL
TADDM BUSINESS APP VIEW	Actual CI - View Business Application Topology	http://\$staddmWebLaunchHostname:\$staddmWebLaunchPort/\$cdm/servlet/LICServlet?default.port=\$staddmServerPort&default.server=\$staddmServerHostname&graph=businessapplications&quid={quid}
TADDM APP VIEW	Actual CI - View Application Topology	http://\$staddmWebLaunchHostname:\$staddmWebLaunchPort/\$cdm/servlet/LICServlet?default.port=\$staddmServerPort&default.server=\$staddmServerHostname&graph=applicationinfrastructure&quid={quid}
TADDM PHYSICAL VIEW	Actual CI - View Physical Topology	http://\$staddmWebLaunchHostname:\$staddmWebLaunchPort/\$cdm/servlet/LICServlet?default.port=\$staddmServerPort&default.server=\$staddmServerHostname&graph=physicalinfrastructure&quid={quid}
TADDM CHANGE HISTORY VIEW	Actual CI - View Change History	http://\$staddmWebLaunchHostname:\$staddmWebLaunchPort/\$cdm/servlet/LICServlet?default.port=\$staddmServerPort&default.server=\$staddmServerHostname&quid={quid}&view=changehistory
TADDM BUSINESS APP VIEW 2	CI - View Business Application Topology	http://\$staddmWebLaunchHostname:\$staddmWebLaunchPort/\$cdm/servlet/LICServlet?default.port=\$staddmServerPort&default.server=\$staddmServerHostname&graph=businessa

Figure 2-49 Task Launch in Context Definition

2.4 Summary

In this chapter, we explain the major capabilities of workflow and a job plan template task definition. We explain how to define manual and automated tasks, tasks to launch in context to the automation engine, how to interact with other process types, and define the sequence of tasks inside a process by defining predecessor relationships between the tasks.

Process Managers

In this chapter we describe the basics of the Configuration Management, Change Management, and Release Management Process Managers (PMs) and how they can be used and integrated with other Process Managers.

We cover the following topics:

- ▶ 3.1, “Overview of Process Managers” on page 84
- ▶ 3.2, “Configuration Management Process Manager” on page 94
- ▶ 3.3, “Change Management Process Manager” on page 167
- ▶ 3.4, “Release Process Manager” on page 253

3.1 Overview of Process Managers

Process Managers (PMs) are out of the box workflows for Tivoli's process automation engine that are based on Service Management best practices. PMs extend the CCMDB to provide integration, automation, and implementation of Service Management process workflows. The flexible workflows are completely customizable and provide a good starting point for implementing these best practice processes.

PMs enable the creation of executable process flows, provide a user interface to allow users to perform process procedures, gather information from various sources, interact with external tools, utilize and update information in the CCMDB, and provide information to monitoring, analysis, and reporting. In addition, PMs provide capabilities to track execution metrics and provide dashboards and reports that allow IT organizations to identify bottlenecks and improve organizational productivity.

PMs utilize industry best practices such as IT Infrastructure Library® (ITIL), Control Objectives for Information and Related Technology (COBIT), and enhanced Telecom Operations Map (eTOM). PMs enable quick implementation of many IT service management processes, because they provide predefined workflows based on best practices, so it is not necessary to build a process from scratch.

Process: A process is a sequence of interrelated activities that receive an input, add value to it, and produce an output that achieves a specific objective. It is guided by policies and needs to be controlled through key performance indicators.

PMs are essential components of the IBM Service Management architecture. Figure 3-1 shows how PMs relate to the IBM Service Management architecture.

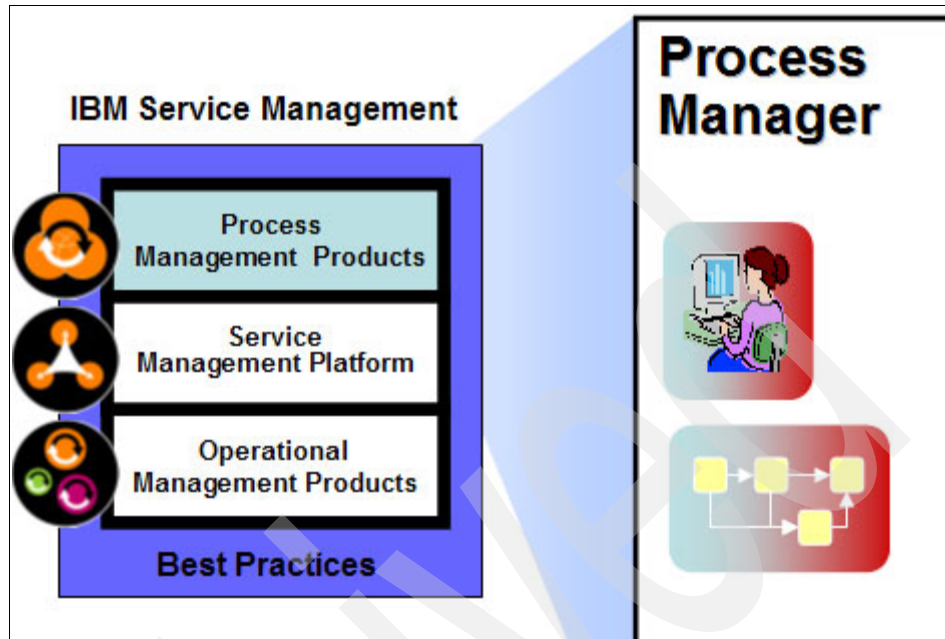


Figure 3-1 Architectural context of IBM Service Management

3.1.1 Process Manager role

Organizations that have decided to implement best practices processes require the necessary tools to help them automate process workflows. Before a tool such as the Tivoli CCMDDB was created, it was necessary to hire developers to build workflow-based applications from scratch. This required organizations to go through a full development cycle to model, simulate, develop, deploy, and execute these processes. This was an expensive proposition, requiring advanced development tools to build workflow-based applications.

Today there is a better option, using a prebuilt process management product that provides an implementation of a particular process of interest, along with significant process flow management and automation capabilities. Such products need to provide IT operations managers with the ability to reconfigure process flows as needed directly through the PM graphical user interface (GUI) without going through an expensive development cycle. Reconfigured processes need to remain consistent with corporate guidelines to ensure compliance with corporate and legal requirements. After the PM is installed and configured, it is ready for use by the IT operations staff to perform process tasks in their daily operations. The PM is responsible for routing tasks to the right user and keeping track of the progress of the tasks assigned to various users participating in the process.

Process execution metrics are often gathered for analysis and reporting, both by the PM itself and by external tools. Analysis of these metrics is used to understand process Key Performance Indicators (KPIs) and can be used to re-engineer processes to improve organizational efficiency (Continual Service Improvement).

Figure 3-2 provides an overview of the Process Manager role.

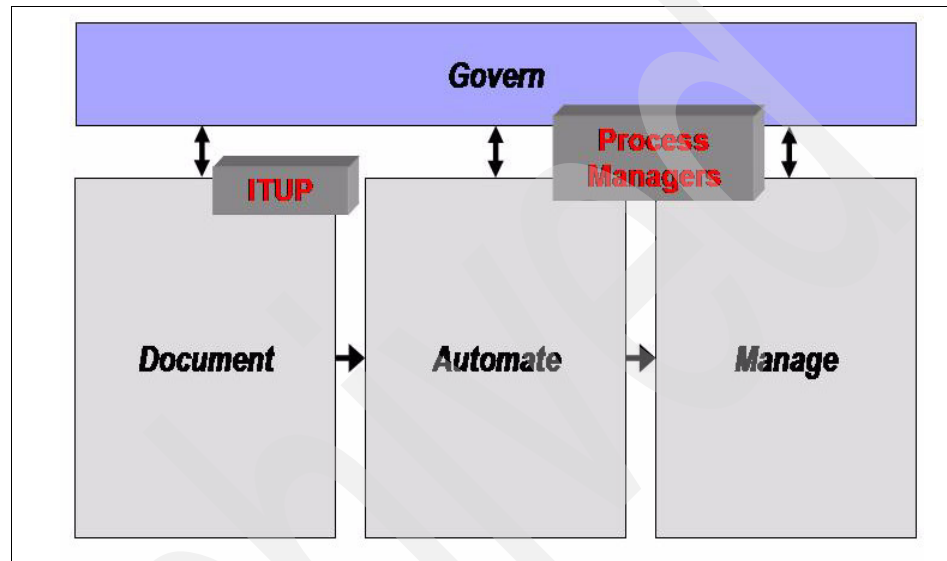


Figure 3-2 Process Manager role

3.1.2 How Process Managers work

Process Managers (PMs) enable the creation of flexible process flows and provide a user interface to allow users to perform the tasks and complete the process steps. They enable aggregation of information from various sources (including external tools) through tool integration modules.

As described in 2.2.5, “Job plans” on page 48, a job plan is a template, with a detailed description of work to be performed on an asset, configuration item, or location.

Job plans can be applied to various types of Changes:

- ▶ J2EE Changes
- ▶ Standard Changes
- ▶ Emergency Changes
- ▶ Change with Release

You can use the Job Plans Application to create, view, modify, or delete job plan records. A job plan typically includes procedural descriptions and lists of estimated labor, items and materials, services, and tools to be used on the job.

A process request is the initial step for the Process Manager. Process requests are submitted through the Process Request Application. This application is responsible for administering the process requests that were submitted. Within the application, a process request is submitted, accepted, rejected, or closed.

After a process request is accepted, a job plan can be applied to a work order by the Process Manager.

Figure 3-3 illustrates the process request flow and how it interacts with the Process Manager.

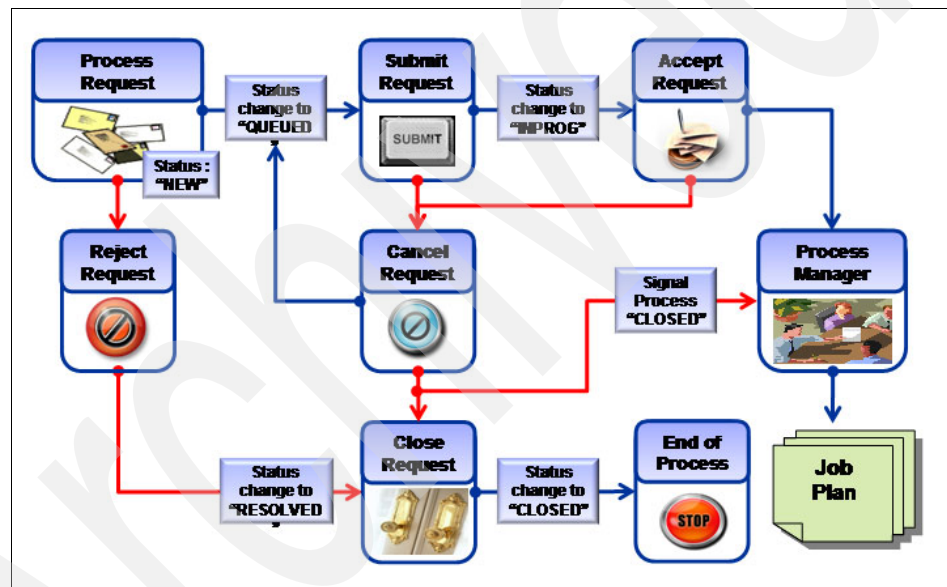


Figure 3-3 Process request and Process Manager

3.1.3 Job plan

This section explains how to work with job plans.

Certain information in CCMDB is specific to Enterprise Asset Management, so just the important and appropriate information about Change and Configuration Process are highlighted here.

Creating a job plan

To access the job plans application, click the application link in Start Center, or select **Go To → Planning → Job Plans**. Within the Job Plan Application, click the **New Job Plan** button. Figure 3-4 shows the job plans form.

Defining a job plan consists of the following steps:

1. Defining the tasks by breaking the job down into steps in the job plan tasks table window
2. Defining the labor craft/skills and hours on the Labor sub-tab
3. Defining the materials needed on the Materials sub-tab
4. Defining the services needed on the Services sub-tab
5. Defining the tools needed on the Tools sub-tab

These steps are applied according to the purpose of each job plan. For example, certain job plans might not require materials.

A required field is the template type. It defines the type of template to be applied to the job plan. It can be any of the following choices:

- ▶ **Process:** Use this choice to classify a job plan as a process. It is the top level of the hierarchy.
- ▶ **Activity:** Use this choice to classify a job plan as an activity. It is the lowest level of the hierarchy.
- ▶ **Maintenance:** Use this choice to classify a job plan as maintenance. It is mostly used for enterprise asset management situations.
- ▶ **Configuration PMP:** Use this choice to classify a job plan as a job plan for configuration PMP purposes.

Job Plans

Bulletins: (0) Go To Reports Start Center Profile Sign Out Help

Find: Select Action

List Job Plan Work Assets Specifications Change Template Change Assessments Change Authorization

Job Plan: PMCHGNORM Normal Change JobPlan Organization: Site: Attachments

Details

Status: ACTIVE Default WO Class: CHANGE

Template Type: Process WO Priority:

Duration: 4:00 Interruptible?

Classification: Flow Controlled? ☒

Class Description: Suspend Flow Control?

Launch Entry Name: Flow Action:

Include Tasks in Schedule? ☒ Flow Action Assist?

Start Constraint Offset:

Finish Constraint Offset:

Process Flow Sequence:

Responsibility

Supervisor: Work Group:

Crew: Owner:

Lead: Owner Group:

Job Plan Tasks: Filter 1 - 4 of 4

Sequence	Task	Description	Nested Job Plan	Duration	Meter
1	10	Assessment		1:00	
2	20	Approval		1:00	
3	30	Implement the Change		1:00	
4	40	Post Implementation Review		1:00	

New Row

Labor Materials Services Tools

Planned Labor: Filter

Task	Craft	Skill Level	Labor	Quantity	Hours	Rate	Line Cost
...No rows to display...							

New Row

Figure 3-4 Job plans form

Defining job plan tasks

Each job plan can be broken down into a series of steps that must be performed to complete the job; these steps are called job plan tasks. The Tasks table window on the job plans page contains a list of numbered tasks that have been defined for a job plan, along with a description of the work to be done for that step, and the estimated time for its completion. These tasks can also use nested job plans.

It is possible to assign the tasks number to any estimated labor, materials, services, and tools that are associated with the task. This is helpful to track and report information by task.

Each job plan task can include the following information:

- ▶ **Sequence:** Used to identify the order in which tasks must be performed. Tasks can have the same sequence number if they must or can be performed simultaneously. After you apply a job plan to a work order, CCMDDB copies the sequence numbers to the work order's tasks.
- ▶ **Task ID:** Unique identifier of the task. The default is for CCMDDB to increment task numbers by 10, for example, 10, 20, 30, and so on. This gives you the flexibility to add new tasks between existing ones.
- ▶ **Description:** Description of the work to be done for the task.
- ▶ **Duration:** Estimated number of hours to perform the task.
- ▶ **Meter:** The meter associated with the measurement point of an asset, for example, a pressure gauge.

Figure 3-5 shows the jjob plan tasks window.

The screenshot displays the 'Job Plan Tasks' window. At the top, there is a toolbar with a 'Filter' button, a search icon, and a status bar indicating '3 - 6 of 6'. Below this is a table with columns: Sequence, Task, Description, Nested Job Plan, Duration, and Meter. The table contains four rows of tasks. The third row is selected, showing 'Implement the Change' with a duration of 1:00. Below the table is a 'Details' panel with various input fields for task configuration, including Organization, Site, Task (set to 30), Sequence (set to 3), Nested Job Plan, Classification, Class Description, Launch Entry Name, Implementation Task (checked), Duration (set to 1:00), Meter, Owner, Owner Group, Flow Controlled (checked), Suspend Flow Control, Flow Action, Flow Action Assist, Assisted Workflow, and Predecessors. A 'New Row' button is located at the bottom right of the details panel.

Sequence	Task	Description	Nested Job Plan	Duration	Meter
1	10	Assessment		1:00	
2	20	Approval		1:00	
3	30	Implement the Change		1:00	
4	40	Post Implementation Review		1:00	

Details

Organization:

Site:

Task: 30 Implement the Change

Sequence: 3

Nested Job Plan:

Classification:

Class Description:

Launch Entry Name:

Implementation Task? ☒

Duration: 1:00

Meter:

Owner:

Owner Group:

Flow Controlled? ☒

Suspend Flow Control? ☐

Flow Action:

Flow Action Assist? ☐

Assisted Workflow:

Predecessors:

Attachments

New Row

Figure 3-5 Job plan tasks

A key point at this point is to define the predecessor task. When a predecessor task is defined for one task, it means that the task will only be performed when the predecessor task is completed.

To define a predecessor task, click the arrow near the Predecessor field and then choose the predecessor tasks.

Figure 3-6 illustrates the predecessor tasks.

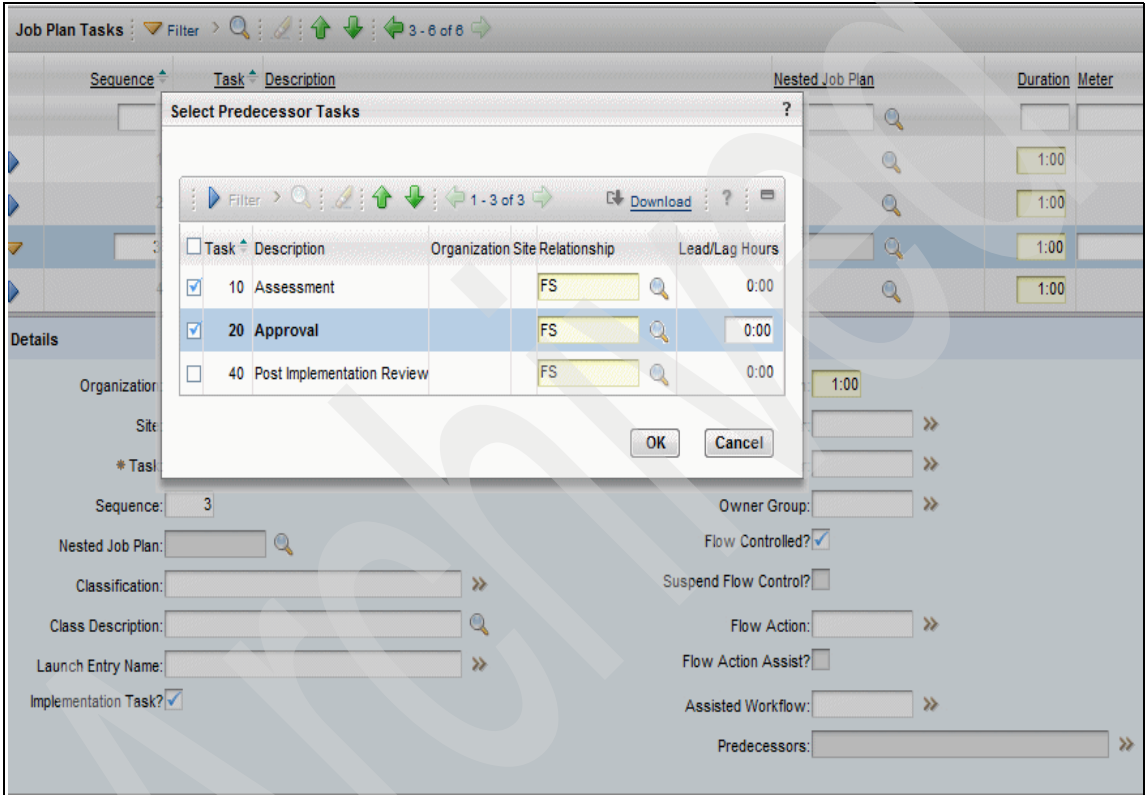


Figure 3-6 Selecting Predecessor Tasks

After defining all nested job plans, they must be associated to the main job plan; Figure 3-7 illustrates a job plan with nested job plans.

Job Plans | Bulletins: (0) | Go To | Reports | Start Center | Profile | Sign Out | Help

Find: [] Select Action []

List | **Job Plan** | Work Assets | Specifications | Change Template | Change Assessments | Change Authorization

Job Plan: **PMCHGNORM** | Normal Change JobPlan | Organization: [] | Site: [] | Attachments []

Details

Status: **ACTIVE**

Default WO Class: **CHANGE**

Template Type: **Process**

Duration: **4.00**

Classification: []

Class Description: []

Launch Entry Name: []

Include Tasks in Schedule? ☒

Start Constraint Offset: []

Finish Constraint Offset: []

Process Flow Sequence: []

WO Priority: []

Interruptible? ☐

Flow Controlled? ☒

Suspend Flow Control? ☐

Flow Action: []

Flow Action Assist? ☐

Responsibility

Supervisor: [] Work Group: []

Crew: [] Owner: []

Lead: [] Owner Group: []

Figure 3-7 Job plan with nested job plans

To associate a nested job plan, select the job plan task to view its properties, then click the arrow near the field Nested Job Plan.

All job plans available will be shown; select the appropriate one. Figure 3-8 illustrates associating a nested job plan.

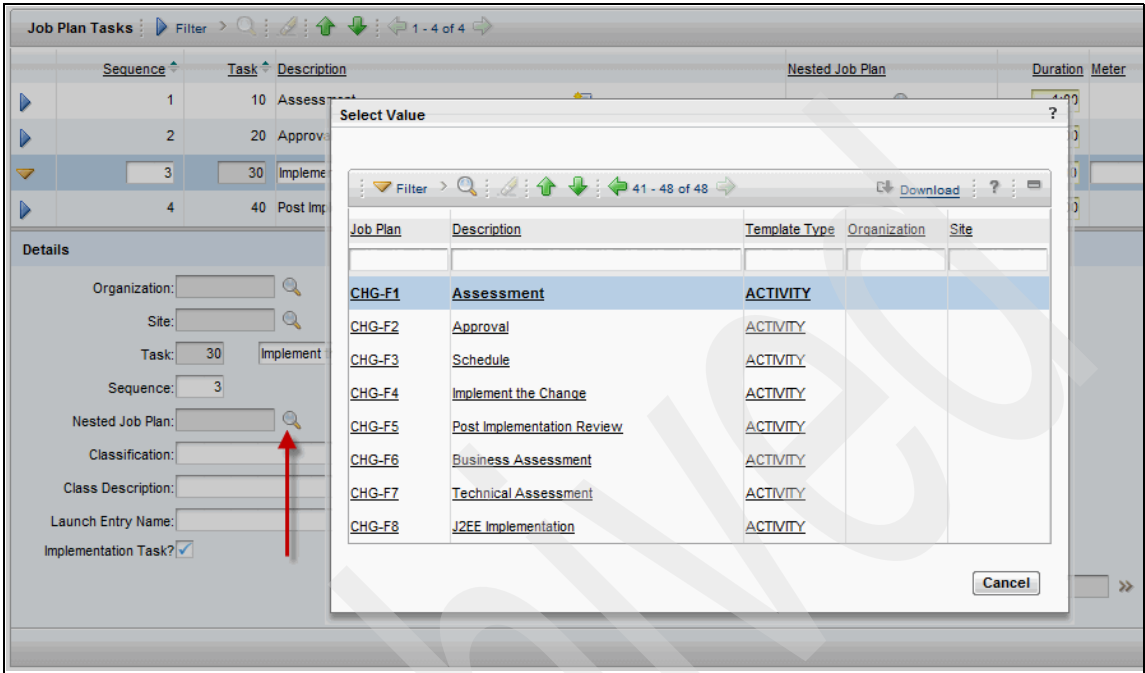


Figure 3-8 Associating a nested job plan

Attention: To have an action automatically performed after a task is completed, you have to define the action in the Flow Action field and *not* check the Flow Action Assist field.

Deleting a job plan

If a job plan record is not listed on any other CCMDB record (for example, listed on an RFC), it can be deleted by selecting **Delete job plan** from the Select Action menu. CCMDB will display an error message if the job plan cannot be deleted.

A job plan can only be deleted by someone who has security authorization to view all of the job plans, that is, access to all Organizations and Sites listed on the job plan.

Duplicating a job plan

The Duplicate Job Plan action must be used to create a copy of an existing job plan, for example, if you want to create a similar job plan for two differing sites. After a job plan is duplicated, it can be modified as needed.

When duplicating a job plan, it duplicates only the portions of the job plan that security access permits to be duplicated, based on security authorizations.

Job plan status

A job plan record can have one of the following statuses:

- ▶ **DRAFT:** Default status for new records. A job plan is still being created and has not yet been approved for use on work orders. Job plan records with a status of DRAFT cannot be associated with PMs or work orders.
- ▶ **ACTIVE:** Job plans that have been approved for use on work orders. A job plan record must be active to be associated with PMs and work orders.
- ▶ **INACTIVE:** Job plans that are no longer required, for example, one that has been replaced by another job plan. Inactive job plan records do not appear in select value lists.

As a best practice, job plan records that are no longer needed must be deactivated by having their status changed to INACTIVE rather than deleted.

3.2 Configuration Management Process Manager

Configuration Management is the process responsible for provide a logical model of IT Infrastructure components and its relationships. It also supports all of the ITSM processes by providing accurate and up-to-date information about the configuration items.

A configuration item (CI) is any component of an information technology infrastructure that is under the control of Configuration Management. CIs can be individually managed, and they are usually treated as self contained units for the purposes of identification and change control. CIs can be a service, hardware, software, support staff, or documentation. They are uniquely identified by names and other attributes.

Relationship: A relationship consists of the physical and logical connections between the configuration items, such as is connected to, parent-child, component-subcomponent, runs-on, and so on.

Configuration Management ensures that selected components of a complete service, system, or product (the configuration) are identified, baselined, and maintained and that Changes to them are controlled. These objectives are achieved using the following subprocesses:

- ▶ Identify configuration items: Defines scope, naming conventions, attribute values, policies, roles, responsibilities, templates, and standards.
- ▶ Control configuration items: Ensures that all additions, updates, and deletions have the appropriate controlling documentation.
- ▶ Report configuration status: Ensures that all configuration data and documentation is recorded as each asset or CI progresses through its lifecycle.
- ▶ Verify and audit configuration items: Verify that the CMDB accurately reflects the environment and established standards by performing periodic audits followed by remediation process to rectify any discrepancies.

In the IBM Service Management Solution, the process “Identify configuration items” is implemented through TADDM; control configuration items through the Update CI Process in the Configuration Process Application; report configuration status through CI LifeCycle Application; and verify and audit configuration items through the Reconciliation CI Process in the Configuration Processes Application.

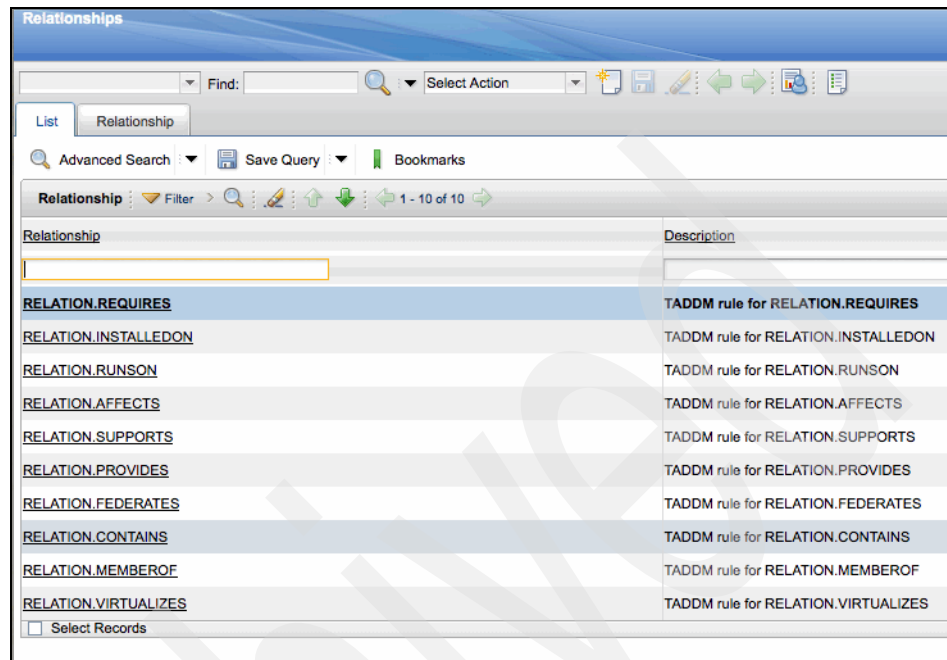
3.2.1 Relationships

Relationships are a representation of the association that exists between configuration items. This information is useful for all other IT Service Management processes, for example, during a Change assessment, problem determination, or incident categorization.

CCMDB comes with certain relationships defined, and it is possible create new relationship rules through the Relationship Application.

Each Relationship Type (for example, runsOn) defines the valid source and target CI Types (for example, Operating System runsOn Computer System).

Figure 3-9 illustrates the default relationship list.



The screenshot shows a web-based application titled 'Relationships'. It features a search bar with 'Find:' and a 'Select Action' dropdown. Below the search bar are tabs for 'List' and 'Relationship'. The 'List' tab is active, showing a table with two columns: 'Relationship' and 'Description'. The table contains ten rows, each representing a different relationship type and its corresponding TADD rule. The first row is highlighted in blue. At the bottom of the table, there is a checkbox labeled 'Select Records'.

Relationship	Description
<u>RELATION.REQUIRES</u>	TADD rule for RELATION.REQUIRES
<u>RELATION.INSTALLEDON</u>	TADD rule for RELATION.INSTALLEDON
<u>RELATION.RUNSON</u>	TADD rule for RELATION.RUNSON
<u>RELATION.AFFECTS</u>	TADD rule for RELATION.AFFECTS
<u>RELATION.SUPPORTS</u>	TADD rule for RELATION.SUPPORTS
<u>RELATION.PROVIDES</u>	TADD rule for RELATION.PROVIDES
<u>RELATION.FEDERATES</u>	TADD rule for RELATION.FEDERATES
<u>RELATION.CONTAINS</u>	TADD rule for RELATION.CONTAINS
<u>RELATION.MEMBEROF</u>	TADD rule for RELATION.MEMBEROF
<u>RELATION.VIRTUALIZES</u>	TADD rule for RELATION.VIRTUALIZES

☐ Select Records

Figure 3-9 Relationship list

3.2.2 Configuration Management roles

In this section we discuss the roles defined for Configuration Management.

Overview

If you enable the installation process to configure your directory server, it will create the following roles as security groups. Each role will have its own Start Center and application access:

- ▶ Configuration manager: Primarily responsible for the definition and the quality of the Configuration Management process.
- ▶ Configuration administrator: Manages the configuration process management applications in CCMDB, including administering the Configuration Management security groups and their access to applications.
- ▶ Configuration librarian: Custodian of configuration item information.
- ▶ Configuration auditor: Responsible for running configuration audits.

Start Centers by roles

Each role in CCMDB can have a unique Start Center according to their activities and information needs.

To customize a Start Center, select **Start Center** → **Change Content/Layout** and then select the content that you want to see in the right and left column.

Figure 3-10 shows the Start Center layout and configuration window.

Layout and Configuration

Layout: **Narrow-Wide** Description: **Process Management Re**

Left Column Filter > 1 - 3 of 3

Portlet	Display Name
Quick Insert	Quick Insert
Favorite Applications	Favorite Applications
Result Set	ITUP Documents

Right Column Filter > 1 - 5 of 5

Portlet	Display Name
Inbox / Assignments	Inbox / Assignments
Bulletin Board	Bulletin Board
Result Set	My Active Process Requests
Result Set	My Process Requests Rejected in the past 7 day
Result Set	My Draft Process Requests

Figure 3-10 Customize Start Center

The Configuration Management Start Centers are illustrated in the following figures.

Figure 3-11 shows the configuration Manager Start Center.

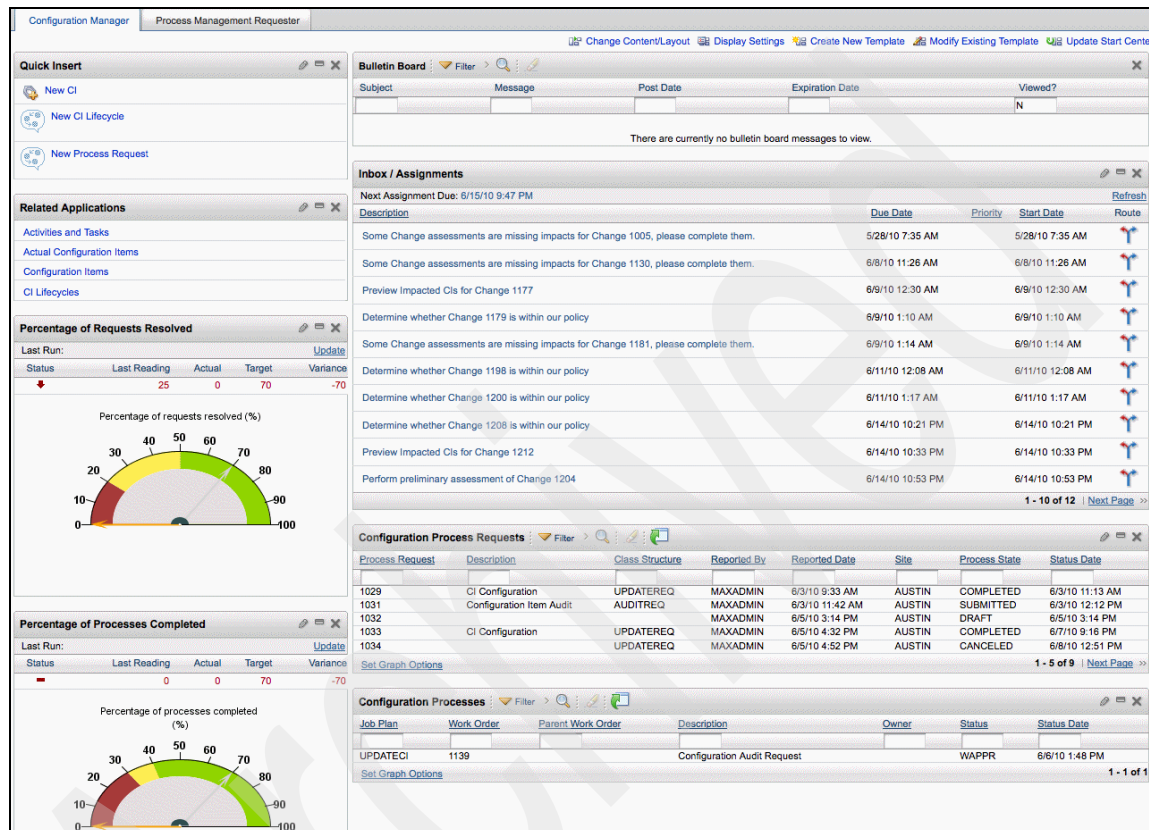


Figure 3-11 Configuration Manager Start Center

Figure 3-12 illustrates a Start Center for the Configuration Administrator role.

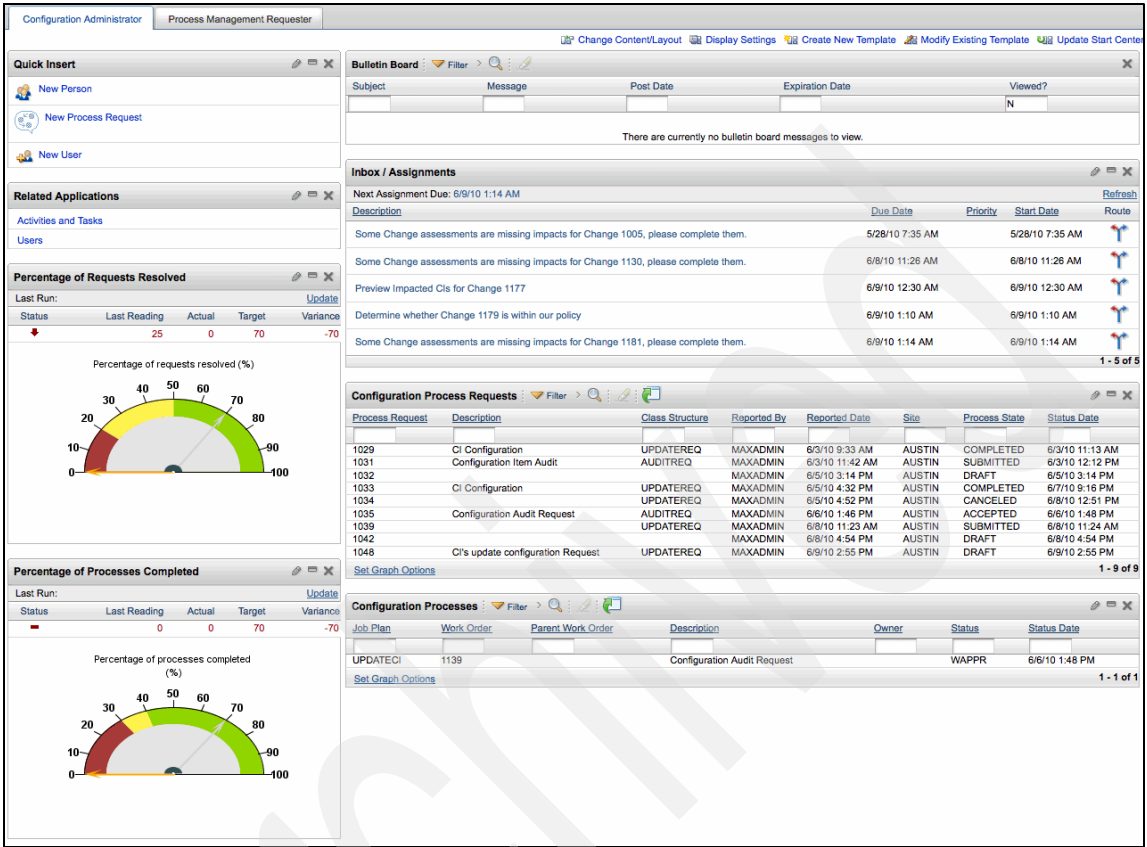


Figure 3-12 Configuration Administrator Start Center

Figure 3-13 illustrates a Start Center for the Configuration Auditor role.

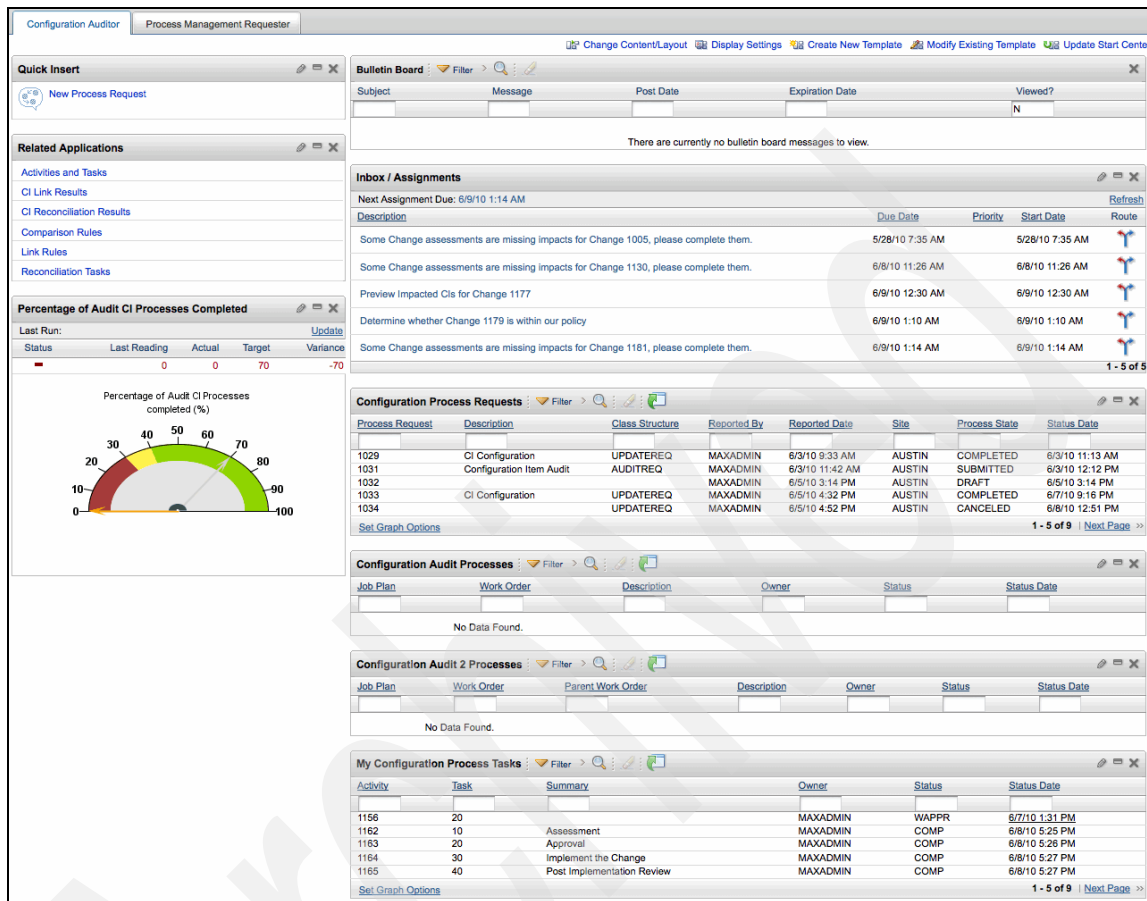


Figure 3-13 Start Center for Configuration Auditor

Figure 3-14 illustrates a Start Center for the Configuration Librarian role.

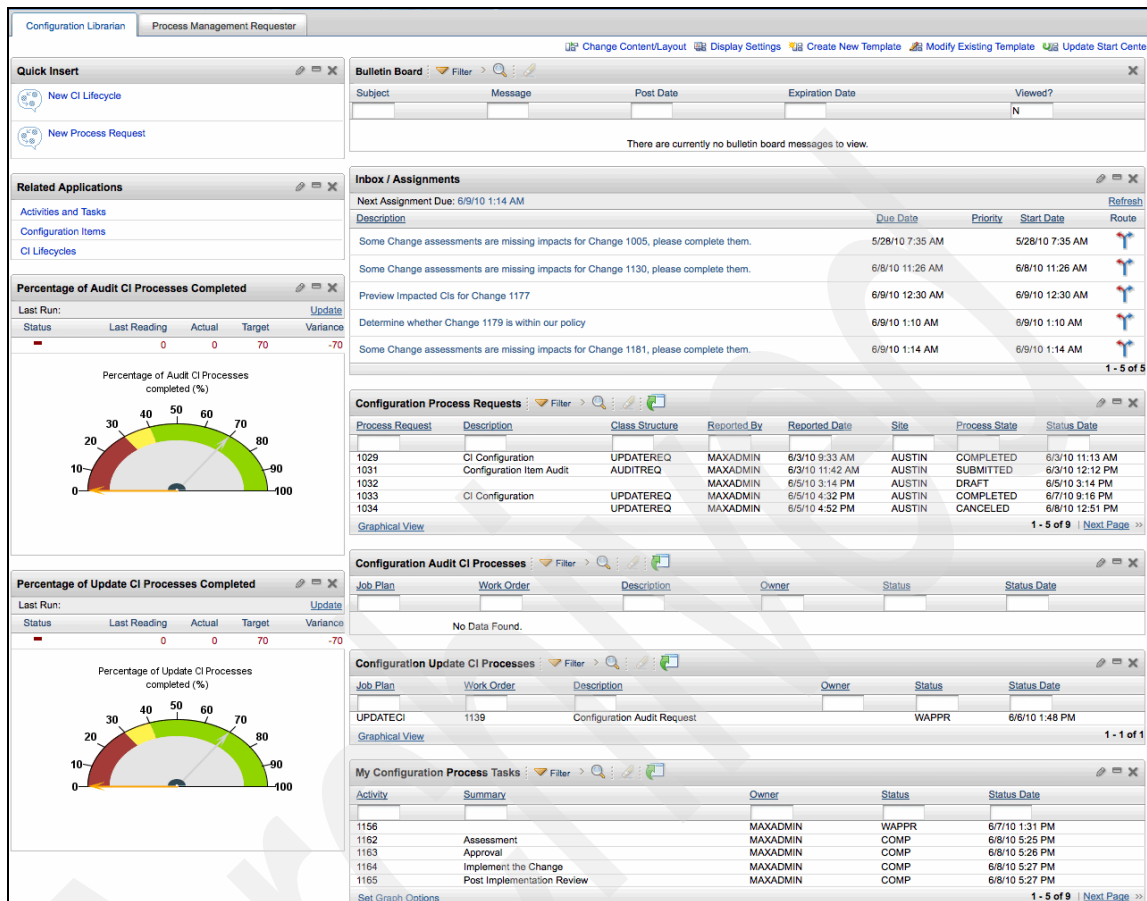


Figure 3-14 Start Center for Configuration Librarian

3.2.3 Configuration item lifecycle

A lifecycle consists of the various stages in the life of a configuration item (CI). You can customize lifecycles, but we show a few of the more common lifecycles in Figure 3-15.

The *lifecycle* defines a set of states and the permitted transitions that occur between these states. You must designate one state for each lifecycle as the default state. When a new CI is created, it automatically takes the default state of the lifecycle that has been applied to that type of CI. If there is no lifecycle explicitly associated with that CI type, the new CI takes the default state from the system default lifecycle.

States can be protected or unprotected. A *protected state* requires a higher level Change process, such as a Request for Change (RFC) in order to move CIs to or from it.

Use a *protected* state if changes on the CI might have a significant effect on the environment, in order to have more control over the movement of CIs to and from this state.

The *production* state is an example of a commonly used protected state. *Transitions* are the process of changing from one state to another state. They correspond to a movement of a CI from one lifecycle status to the next lifecycle status. If a transition occurs to or from a protected state, an RFC must be initiated and approved before that transition can take place.

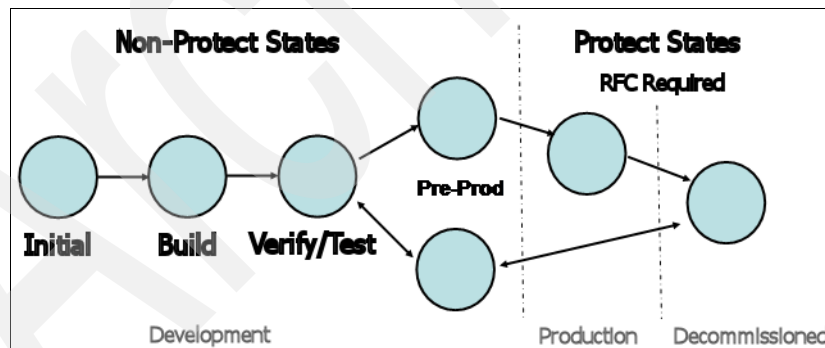


Figure 3-15 CI lifecycle

Certain terms are specific to the IBM Tivoli Change and Configuration Management Database (CCMDB) Lifecycle application. We use these terms throughout this section:

- ▶ *Lifecycle*: The various stages in the life of a configuration item (CI). The *lifecycle* defines a set of states and the permitted transitions between them.
- ▶ *Default lifecycle*: The lifecycle that is applied to a CI type that has not had a lifecycle applied to it. You can designate any defined lifecycle as the default lifecycle.
- ▶ *Status*: The name of a required field in many types of records that shows the current stage in the lifecycle of the associated CI.
- ▶ *Transition*: A Change in state that corresponds to a movement of a CI from one lifecycle status to the next lifecycle status.
- ▶ *Default state*: The state within a lifecycle to which a new CI is assigned.
- ▶ *Protected state*: A lifecycle state requiring a Request for Change (RFC) in order to move a CI into or out of that state.
- ▶ *Internal state*: One of the names defined for lifecycle states while they are processed by CCMDB logic.

3.2.4 CI lifecycle management

This section describes how to perform the following actions within a CI Lifecycle:

- ▶ Creating a new lifecycle
- ▶ Adding a lifecycle state
- ▶ Defining lifecycle transitions
- ▶ Applying the lifecycle to CI types
- ▶ Modifying an existing lifecycle
- ▶ Deleting a lifecycle state

Creating a new lifecycle

All configuration items (CIs) require an associated lifecycle to govern the states in which the CI can enter and the rules by which those states can transition. This topic describes how to create a lifecycle, assign a state or states to it, and designate the transition from that state to another state.

Perform the following set of actions to add a new lifecycle:

1. Select **Go To** → **IT infrastructure** → **CI Lifecycle** (Figure 3-16).

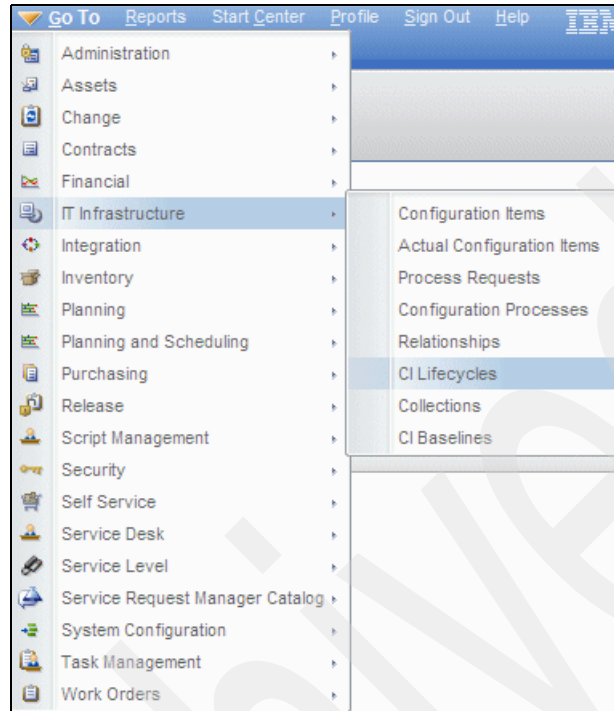


Figure 3-16 CI Lifecycle menu

2. You can choose to either click **New CI Lifecycle** from the Select Action menu or you can click the **New CI Lifecycle** icon to the right of the Select Action drop-down menu (Figure 3-17). Selecting **New CI Lifecycle** opens the Lifecycle tab. An ID designation for the new lifecycle is automatically generated and cannot be altered.

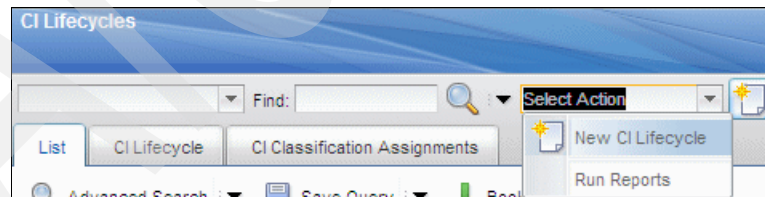


Figure 3-17 Add New CI Lifecycle

3. In the Lifecycle Name field, type the name of the new lifecycle that you want to create. If you want this lifecycle to function as the default lifecycle, select the **Default** (Figure 3-18). Optionally, type a description of the lifecycle in the Description field.

Find: Select Action

List CI Lifecycle CI Classification Assignments

ID: Is Default? ☐

* Lifecycle Name: Description:

Figure 3-18 New CI Lifecycle

Adding a lifecycle state

The second step in creating a lifecycle is adding the state of that lifecycle.

Ensure that you have finished the Add a New Lifecycle task by performing the following steps:

1. Click **New Row**. The Lifecycle State Details table is displayed underneath the States table (Figure 3-19).
2. Select a state from the State drop-down menu.
3. Select the **Is protected?** if you want to create a protected state. You cannot delete a protected state without having a Request for Change (RFC) associated with the lifecycle.
4. Select **Is Default?** if you want this state to be the default.
5. Optional: Enter a description in the Description field for the state.
6. Repeat this task until you have defined all the desired states.

States Filter 1 - 1 of 1

State	Description	Is Protected	Is Default
DEVELOPMENT	Development	☐	☒

Lifecycle State Details

* State: Description:

Check External KM? ☐

Is Default? ☒

Figure 3-19 New CI State

Defining lifecycle transitions

When you define a lifecycle state, you must indicate the other states to which a CI can move from the current state. The third step in creating a lifecycle is the processing of defining the transitions to each lifecycle state (Figure 3-20).

1. Click the state where you want to set a transition.
2. Click **Set Transition** in Transitions from (Name) State, where (Name) is the name of the state you clicked. This opens a dialog box that allows you to select the state to where the CI can move.

The screenshot displays the CCM interface for defining lifecycle transitions. At the top, a table lists lifecycle states:

State	Description	Is Protected	Is Default
DEVELOPMENT	Development	☐	☒
VALIDATE	Validate	☐	☐

Below the table is a 'Lifecycle State Details' section for the selected 'VALIDATE' state. It includes a dropdown for 'State' (set to 'Validate'), a text field for 'Description' (containing 'Validate'), and checkboxes for 'Check External KM?' and 'Is Default?'. Below this is a 'Transitions from VALIDATE State' section, which is currently empty, showing '...No rows to display...'. At the bottom of this section, a button labeled 'Set Transitions' is highlighted with a red rectangular box.

Figure 3-20 Define lifecycle transitions

Applying the lifecycle to CI types

The fourth step in creating a lifecycle involves applying the lifecycle to one or more classifications or CI types.

This function permits you to apply a lifecycle to one or more CI types. When the system subsequently identifies new CIs, they are automatically assigned to the lifecycle that you have applied to their CI type.

Attention: Many of the classifications listed are unsuitable for CIs, so make your selections carefully.

Complete the Setting the lifecycle transition task before you apply the lifecycle to CI classifications.

Here are examples of lifecycle classifications:

- ▶ Software/operating system
- ▶ User issue
- ▶ User issue/hardware/printer

To apply the lifecycle to CI types, click the **CI Classification Assignments** tab, then click **New Row**. If you know the name of the classification you want to associate with your lifecycle, type it into the Classification field. If you do not know the name, click the **Detail Menu** icon next to the Classification field and select the appropriate classification from the list by clicking the blue box beside the listing (Figure 3-21).

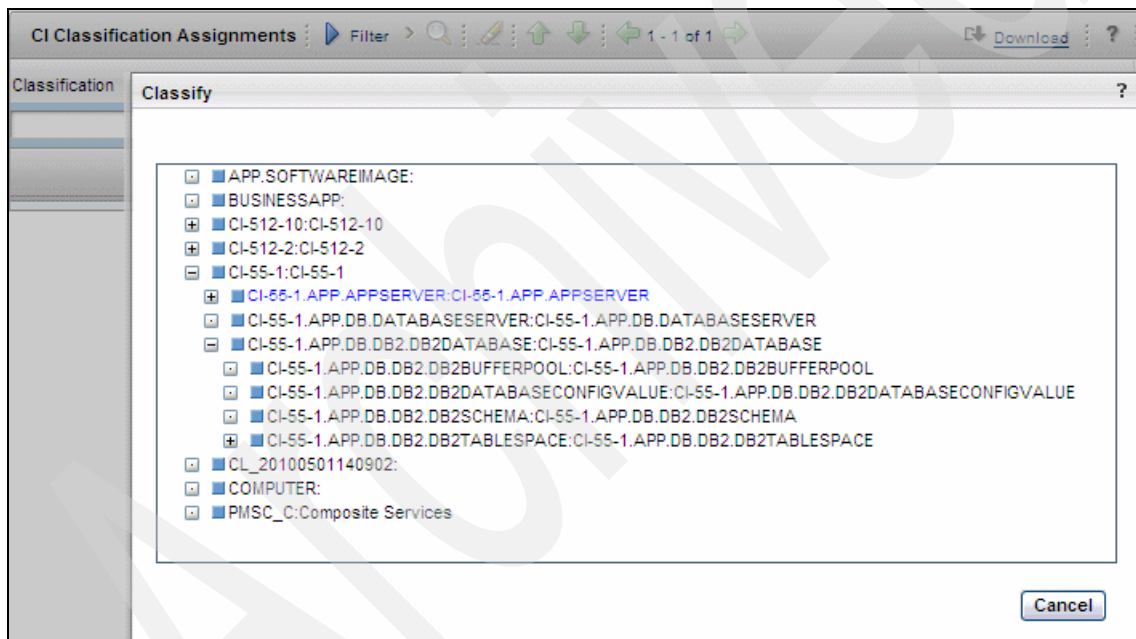


Figure 3-21 Classify lifecycle

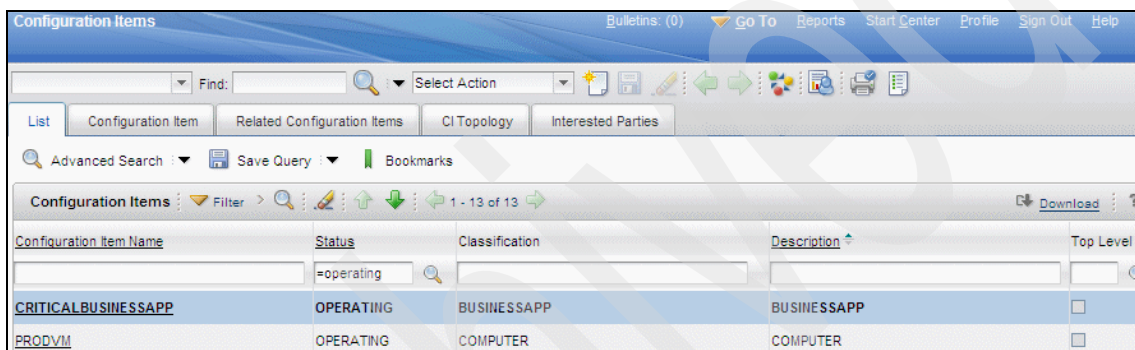
Modifying an existing lifecycle

It might become necessary to modify certain of the existing attributes of a lifecycle after they have already been applied. To add a new state name to the existing states of a lifecycle or to change existing transitions between states, choose the lifecycle you want to modify, make the Changes, and save it.

Deleting a state from a lifecycle

If the state from a lifecycle is not being used by any CIs, it can be deleted. When you try to delete a state that is being used, an error message will pop up, telling you that you cannot delete the state. Before deleting the state, move each CI to another state.

To find CIs in a certain state, open the configuration items application, enter the state in the Status field, and press Enter (Figure 3-22). If you have states with the same name in more than one lifecycle, particular CIs in the list might be in another lifecycle that has a state with the same name. In that case, check the classification of each CI in the list to determine whether it uses the lifecycle from which you plan to delete the state.



The screenshot shows the 'Configuration Items' application window. The 'Status' filter is set to 'operating'. The table below shows the results of the filter.

Configuration Item Name	Status	Classification	Description	Top Level
CRITICALBUSINESSAPP	OPERATING	BUSINESSAPP	BUSINESSAPP	☐
PRODVM	OPERATING	COMPUTER	COMPUTER	☐

Figure 3-22 Filtering CIs by status

To delete a state from a lifecycle, follow these steps:

1. Open the lifecycle in the CI Lifecycles application.
2. If the state that you plan to delete is the default state, choose another state to be the default state and check the **Is Default?** box on the line of the new default state. You cannot delete the default state.
3. Click the trash icon on the line of the state you want to delete (Figure 3-23). The name of the state will be struck through in the list.
4. Click **Save** to save the updated lifecycle.



The screenshot shows the 'States' application window. The table below shows the states in the lifecycle. The 'VALIDATE' state is highlighted, and a trash icon is visible in the 'Is Default' column.

State	Description	Is Protected	Is Default
VALIDATE	Validate	☐	☐

Figure 3-23 Deleting state from lifecycle

Deleting a lifecycle

If there are no CIs in classifications to which that lifecycle has been applied, the CI Lifecycle can be deleted. When you try to delete a Lifecycle that is being used, an error message will pop up, telling you that you cannot delete the Lifecycle. Before deleting the Lifecycle, move each CI to another one.

To delete a Lifecycle, follow these steps:

1. Open the CI Lifecycles application.
2. If the lifecycle that you plan to delete is the default lifecycle, choose another lifecycle to be the default lifecycle and check the **is Default?** box on the line of the new default lifecycle. You cannot delete the default lifecycle.
3. Click the name of the lifecycle you want to delete in the Lifecycle Name column.
4. Select **Delete this Lifecycle** from the Select Action drop-down menu.
5. A dialog box will pop up. Click **Yes** if you want to delete the lifecycle.

3.2.5 Discovering configuration items

The configuration items in the environment are discovered by sensors or operational management products. Discovery is the process of identifying the configuration items (CIs) that exist in your IT infrastructure, including their attributes and how they are related to other CIs. Information about discovered CIs is stored in the Tivoli Application Dependency Discovery Manager (TADDM) database. TADDM receives information about CIs in two ways:

- ▶ Through a TADDM sensor, a tool that traverses the IT infrastructure to discover specific types of CIs
- ▶ From an operational management program (OMP), such as Tivoli Provisioning Manager, whose data TADDM imports through its discovery library

The TADDM database stores CI data using the common data model, which is a standard model for describing configuration items, including their attributes and relationships.

For more details about the common data model, TADDM and its role in the discovery process, see the *Certification Study Guide: IBM Tivoli Application Dependency Discovery Manager V7.1*, SG24-7764.

3.2.6 Authorized configuration items

The discovery process collects a large amount of data about configuration items because it scans the infrastructure. The configuration items have many attributes and relationships, and you probably will not want to control all configuration items and all attributes, so it is possible to refine the data to include only the information that aggregates value and has to be managed.

Each configuration item in the CCMDB is either an actual CI or an authorized CI. Figure 3-1 shows the characteristics of each type.

Table 3-1 Actual CIs versus authorized CIs

Actual CIs	Authorized CIs
Represents an item in the environment.	It is a logical representation of a corresponding Actual CI.
Actual CIs are imported from TADDM into CCMDB and cannot be modified in CCMDB.	It is usually created from an Actual CI by a process called promotion, but can also be created manually in the configuration items application.
Cannot be chosen as targets CIs of a Change Request or other process.	Can be chosen as targets of a Change Request or other process.

Audit reports can be run to check on the differences between actual and authorized versions of configuration items, and then take corrective actions as needed.

Authorized configuration items are usually created through the Promotion process. It is also possible to create them manually in the configuration items application.

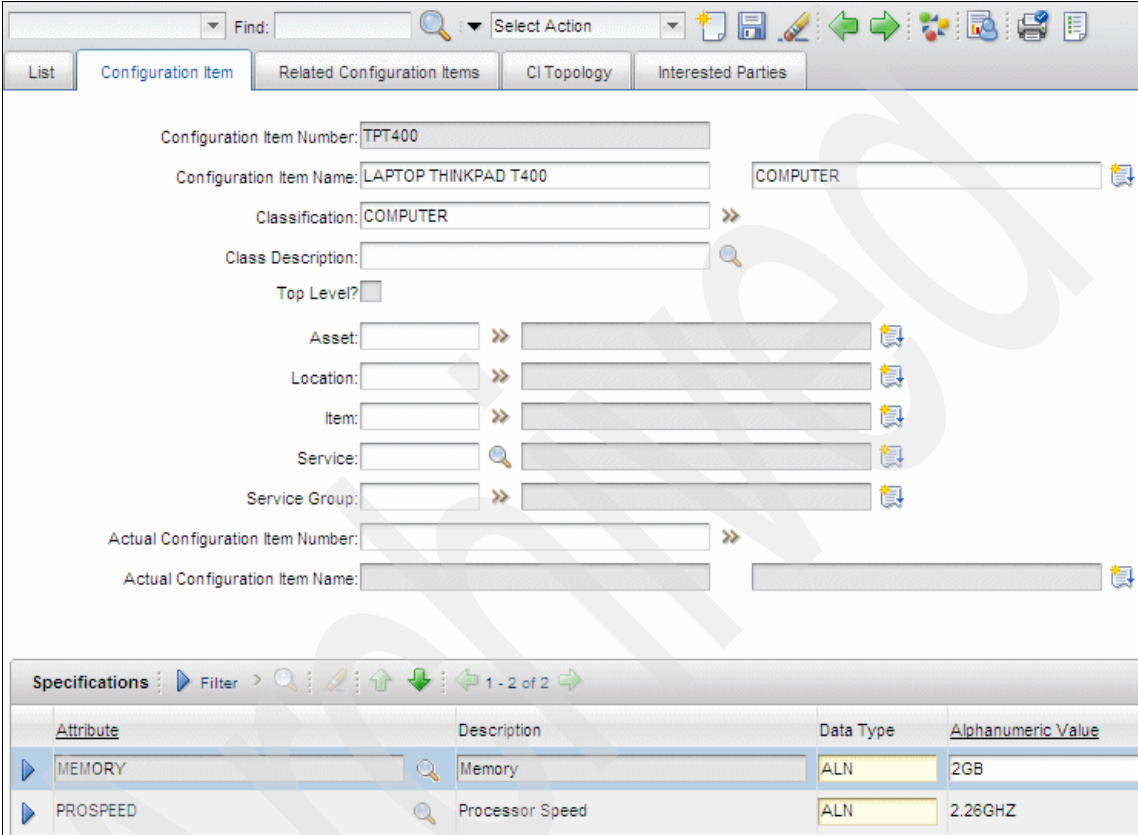
A manually created authorized CI can be linked to an actual CI later, or it can exist on its own with no link to an actual CI.

This section discusses how to manually create authorized CIs.

Creating an authorized configuration item

To create an authorized configuration item, follow these steps:

1. Select **Go To Menu** → **IT Infrastructure** → **Configuration Items**, and then click the **New CI** button (Figure 3-24).



Configuration Item Number: TPT400

Configuration Item Name: LAPTOP THINKPAD T400

Classification: COMPUTER

Class Description:

Top Level? ☐

Asset:

Location:

Item:

Service:

Service Group:

Actual Configuration Item Number:

Actual Configuration Item Name:

Specifications

Attribute	Description	Data Type	Alphanumeric Value
MEMORY	Memory	ALN	2GB
PROSPEED	Processor Speed	ALN	2.26GHZ

Figure 3-24 Create New CI

The information required to create a new CI is:

- **Configuration item number:** Unique ID for the CI.

The optional information to create a new CI is as follows:

- **Configuration item name:** Name for the CI.
- **Actual CI number:** To base the configuration item on an actual CI, enter its value in the actual CI field. If the Classification field is empty when you enter the actual CI value, the system uses the classification and attributes from the actual CI.

- Enter the following fields as needed:
 - **CI owner:** Person responsible for that CI, which is useful when approving a Change because you will know the best person to consult.
 - **Change Window:** Change Window that must be applied to the new CI.
 - **Asset:** Related asset to associate the CI.
 - In the **Classification** field, associate the CI with a classification. Using the Detail Menu, select **Classify** to select a classification, or select **Go To Classifications** to open the Classifications application. The Specification table window displays the attributes associated with the classification.
 - Using other fields, associate the CI with an asset, location, item, service, or service group. Certain associations preclude making other associations.

For example, if you associate the CI with an asset, the Location, Item, and Service fields become read-only and you cannot associate the CI with them.

Similarly, the CI Location, Site, Organization, Item Set, Calendar, and Shift fields can be affected by other choices.
 - If you want the CI to be owned by someone, enter a value in the **CI Owner** field.
- 2. In the same tab, define the attributes for the new CI. It can only be defined if a Classification is associated with it.
- 3. To add the attributes, click the **New Row** button in the Specifications section. In the attribute field, enter a value or select it from the opened list by clicking **Select Value** (Figure 3-25).
- 4. Fill the following fields as needed:
 - Data Type
 - Unit of Measure
 - Section
 - Alphanumeric Value
 - Numeric Value
 - Table Value
 - Inherited from
 - Apply Down Hierarchy?

Attribute	Description	Data Type	Alphanumeric Value
MEMORY	Memory	ALN	2GB
PROSPEED	Processor Speed	ALN	2.26GHZ

Details

Attribute:
Description:
Alphanumeric Value:

Data Type:
Numeric Value:

Unit of Measure:
Table Value:

Section:
Inherited from:

Apply Down Hierarchy? ☐

Figure 3-25 New CI Attributes

At the bottom of the Configuration Item tab, a table lists the blackout periods to which this CI has been assigned. A blackout period is a time, designated by administrators, during which Changes require extra scrutiny. Changes during a blackout period might require extra authorizations, or might be prohibited altogether. You can use Select Action menu items to add a CI to a blackout period or to a Change Window.

Defining relationships

To define the relationships (Figure 3-26), follow these steps:

1. In the Related CIs tab, define relationships that the New CI has with other CIs.

Source Configuration Item	Classification	Relation	Target Configuration Item
WIRELESS MOUSE	COMPUTER	RELATION.AFFECTS	THINKPAD T400

Details

* Source Configuration Item Number:

Source Configuration Item Name:
Classification:

* Relation:
TADDM rule for RELATION.AFFECTS:

* Target Configuration Item Number:

Target Configuration Item Name:
Classification:

Containment? ☐

Figure 3-26 Add a Relationship

2. To define a relationship, click the **New Row** button, then select the source CI and the relation or the target CI and the relation.

When selecting the source CI, the target CI will be automatically filled with the name of the CI that is being created, otherwise when selecting the target CI, the source CI will be automatically filled with the name of the CI that is being created.

If both configuration items are associated with classifications, the relationship must be valid. If the relationship is not valid, an error box will pop up.

3. Use the **Containment** check box to determine if there is a containment relationship between the source and target configuration items. The Containment check box is read only.

Updating an authorized configuration item

To update an authorized configuration item, follow these steps:

1. Select a configuration item from the list, edit it, define the modifications, and save it.

Whether you can edit a field sometimes depends on the value in another field. When the classification is changed, the attributes reflect the new classification.

2. If the configuration item is in a protected state in its lifecycle, provide an approved Change record in order to change the value of an attribute or the status of the CI:
 - a. The Change must be in Approved or In Progress state, and it must identify the CI as a target.
 - b. Enter the Change number or select it from a list when you modify the CI.

Deleting an authorized configuration item

A configuration item that has been used on any other record cannot be deleted. For example, a CI that is part of a collection and is displayed on an incident, problem, or Change record.

If the CI that you want to delete is in a protected state in its lifecycle, you must supply the name of an approved Change in order to delete the CI or to change its status.

Instead of deleting a configuration item, its status must be changed to decommissioned. Decommissioning a CI makes the CI inaccessible in all other applications.

To delete a configuration item, follow these steps:

1. View the CI that you want to delete in the configuration items application (Figure 3-27).
2. Check the status column to see whether the CI is in a protected state. If it is in a protected state, you must have a valid Change (it must be in Approved or Progress state, and it must identify the CI as a target), in order to delete it or to change its status. You can enter the Change number or select it from a list.
3. To verify that there are no work items associated with the CI, click **Select Action** → **View Work Details** to see a list of work items. A work item can be a service request, a process request, a problem, an incident, an activity or task, or a workorder. You can also view the CI in its Topology sub-tab; if any work items are associated with the CI, a number is displayed in the icon representing the CI.
4. If there are any work items associated with the CI, you can either remove the CI from the work item or delete the work item.
5. When there are no work items associated with the CI, select **Delete** from the Select Action drop-down menu.

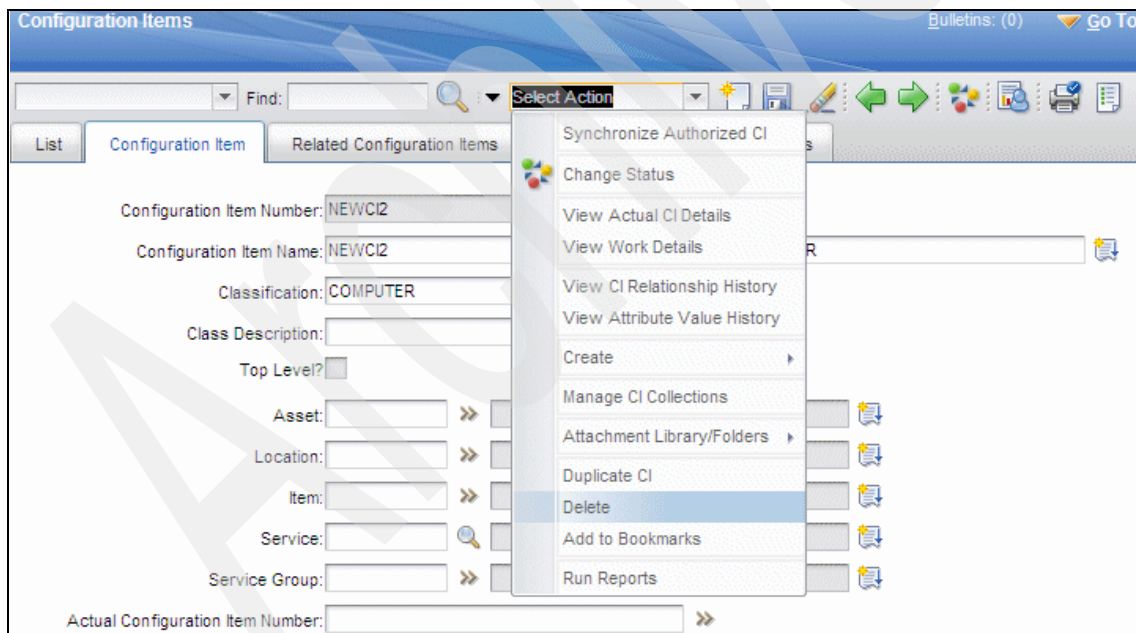


Figure 3-27 Delete authorized configuration item

Duplicating an authorized configuration item

To duplicate a configuration item, follow these steps:

1. Select a configuration item from the list and then select **Select Action** → **Duplicate CI**.
2. All fields of the CI record will be duplicated, except the Configuration Item field and information in the Related CIs tab.

Tip: Duplicating a CI and modifying it as needed is a quick way to create similar configuration items (CI) without having to enter the same information again.

Managing a CI Collection

Use the Manage CI Collection dialog (Figure 3-28), to associate collections with a configuration item (CI), to delete a collection, or activate or deactivate an association.

Associating a CI to a Collection helps to better organize the CIs. For example, you might have a collection called **Command Center CIs** and associate all its CIs to this collection, and another called **Data Center CIs** and associate only CIs related to the data center.

Follow these steps:

1. Select the CI to be associated and select **Select Action** → **Manage CI Collection**. The Manage CI Collections dialog box opens and displays any collections already associated with the CI.
2. Perform any of the following actions:
 - Associate a Collection with CI:
Click **New Row** to add one collection or click **Select Collections** and select one or multiple collections.
 - Delete an association of a Collection with CI:
Select the row and click the recycle bin icon.
 - Activate or deactivate an association:
Click the **Active** check box in the row that has the collection.

Configuration Item Name:
THINKPAD T400
COMPUTER,Laptops, 4GB, 2,26MHZ

Configuration Item Number:
TPT400

Filter
1 - 2 of 2
Download
?

Collection	Description	Active
COLL1 >>		☒
COLL-WIN >>		☒

Select Collections
New Row

OK
Cancel

Figure 3-28 Manage CI Collection

3.2.7 Controlling and updating the CI process

The Control CIs process ensures that changes to authorized CIs are made with the appropriate review, authorization process, and control documents. The Control CI Request can be issued as a service request by the PMP process (Change or Release PMP) or by an authorized role (for example, Change Manager, Configuration Librarian, or Other IT users).

Multiple CI changes can be requested in one control CI request. The CI attribute changes in CCMDB is done through the Update CI process that is responsible for performing all updates in the CCMDB.

Figure 3-29 provides an overview of the CI processes.

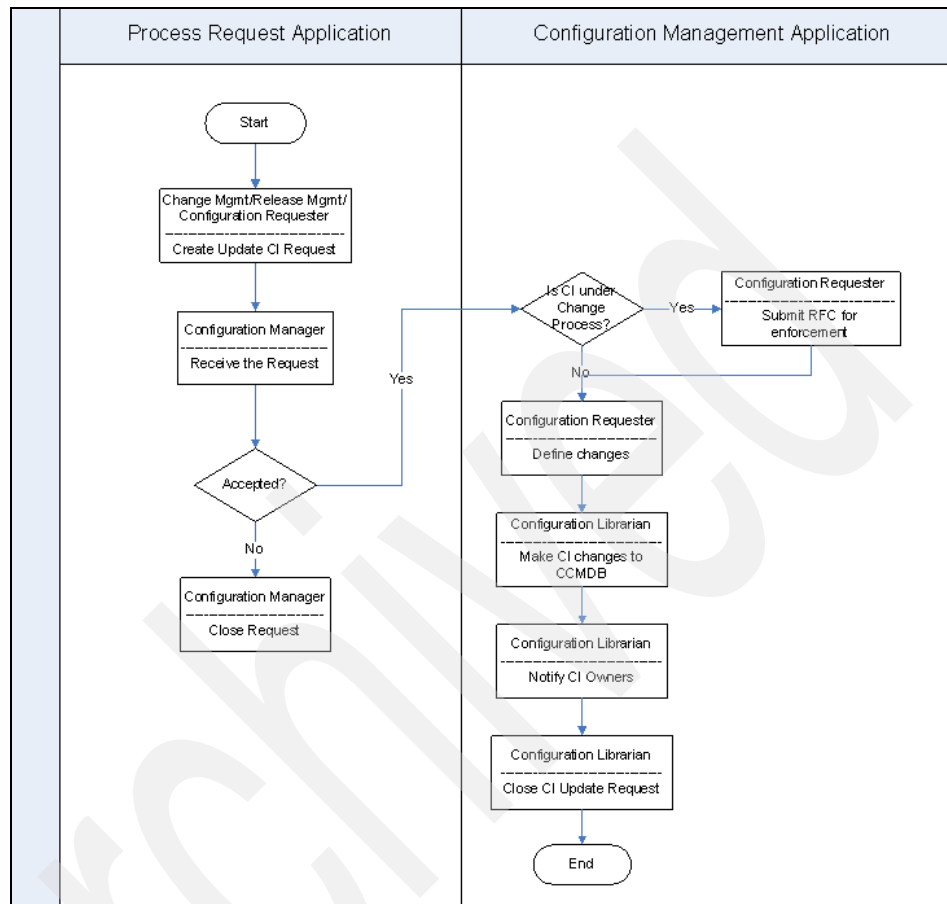


Figure 3-29 Control process flow

Creating an Update CI Request

Responsible role: Configuration Requester.

An Update CI Request can be created from the Configuration Application or can be issued from other PMPs as Change Management.

To create an Update CI Request from Configuration Application, perform the following steps:

1. Select **Go To → IT Infrastructure → Process Requests**. Click **New Process Request**.
2. Provide the following information to characterize the process request as an Update CI Request:
 - **Process Manager Type**: Choose the option Configuration in the list.
 - **Configuration Item**: Choose a CI you want to update.
 - **Classification**: PMCFGUR.
 - **Class Description**: CIs Configuration Control / Update Request.
3. In the Classification Attributes table, fill in two very important fields: CHGRECD and CHANGENUM. They are used together to determine if all the updates for an RFC are recorded.
 - a. Open the CHGRECD attribute, and In the Table Value field, type or select YES.
 - b. For the CHANGENUM attribute, select the number of the Change that resulted in modifications to the CIs.

When you specify the YES table value for the CHGRECD attribute, you are informing the configuration manager who receives the request that the modifications were specified in the Move/Modify/Swap dialog box. The configuration manager can then go to the specified Change number in the Changes application and examine the modifications; in addition, the configuration manager will be aware that the CI was automatically updated as specified when the Change status moved to COMPLETE.

4. After providing all the required information, click **Submit**.

Figure 3-30 shows an example of this configuration.

The screenshot shows the 'Process Request Details' form. It contains several fields for configuring a request:

- Description:** CI Configuration
- Process Manager Type:** Configuration
- Site:** AUSTIN
- Requested Completion:** (empty)
- Asset:** (empty)
- Location:** (empty)
- Configuration Item:** TPT400 (selected from a dropdown showing 'COMPUTER, Laptops, 4GB, 2.26MHZ')
- Configuration Item Name:** THINKPAD T400
- Response Plan:** (empty)
- Priority:** 3

Below these fields is the 'Request Classification Details' section:

- Classification:** PMCFGUR (selected from a dropdown)
- Class Description:** CIs Configuration Control / Update Request

At the bottom is the 'Classification Attributes' table:

Attribute	Description
CHANGENUM	Change Number
CHGRECD	All Updates Recorded

A 'New Row' button is located at the bottom left of the table.

Figure 3-30 Create a Process Request to update a CI

Accepting an Update CI Request

Responsible role: Configuration Librarian / Configuration Manager.

The Configuration Librarian or Configuration Manager receives the request, verifies the information provided, and decides to accept it or not. To accept an Update CI Request, select the record and click **Accept**.

Updates: When an Update CI Request is provided by the Modify option in Change PMP, the updates are made automatically using RFC from Change PMP and do not need to be approved again.

Figure 3-31 illustrates an Update CI Request to be approved.

The screenshot shows the bottom toolbar of the application. It includes a search bar, a 'Select Action' dropdown, and a series of icons. The 'Accept' icon, which is a green checkmark inside a red square, is highlighted with a red box.

Figure 3-31 Accept CI Process Request

Creating a Configuration Process Record

Responsible role: CCMDB.

After the Update Process Request is accepted, it automatically becomes a Configuration Process Record. Check the Configuration Process number on Related Records tab (Figure 3-32).

ListProcess RequestRelated RecordsLog

Process Request:1033CI ConfigurationSite:AUSTINProcess State:ACCEPTEDStatus:INPROG

Related TicketsFilterDownload

Related Work OrdersFilterDownload

Work Order	Description	Class	Status	Relationship
1132	CI Configuration	PMCFGWO	WAPPR	FOLLOWUP

Select Work OrderNew Row

Figure 3-32 Configuration Process Related Record

Applying a job plan to Configuration Process Record

Responsible role: CCMDB.

A job plan is automatically applied to the new configuration process record. As with the Change Record, the tasks created can be modified in each configuration process record without affecting the job plan itself.

To create new tasks, click the **New Row** button.

To delete a task, click the **Trash** button in the row to be deleted. Figure 3-33 shows the application of a job plan to a Configuration Process Record.

The screenshot displays the 'Plans' tab for Configuration Process 1132. The interface includes a header with tabs: List, Configuration Process, Plans, Related Records, Actuals, Log, Specifications, and Workplan Map. Below the tabs, there are input fields for Configuration Process (1132), CI Configuration, Supervisor, Status (WAPPR), Parent WO, Lead, Site (AUSTIN), Job Plan (UPDATECI), and Work Group. A section titled 'Audit CIs Details' is followed by a 'Children of Configuration Process 1132' section with a 'Filter' button and a 'Download' button. Below this is a 'Tasks for Configuration Process 1132' section with a 'Filter' button, a 'Download' button, and a table of tasks. The table has columns: Sequence, Task, Summary, Estimated Duration, Status, Owner, and Owner Group. The tasks are listed in a table with 7 columns: Sequence, Task, Summary, Estimated Duration, Status, Owner, and Owner Group. The tasks are: 1. Review Update CI request and work (0:00, WAPPR), 2. Determine requested CI changes (0:00, WAPPR), 3. Make CI attribute changes if it is requ (0:00, WAPPR), 4. Make CI relationship changes if it is (0:00, WAPPR), and 5. Send email notification to CI Owner (0:00, WAPPR). At the bottom right, there are buttons for 'Generate Task Schedules' and 'New'.

Sequence	Task	Summary	Estimated Duration	Status	Owner	Owner Group
1	10	Review Update CI request and work	0:00	WAPPR		
2	20	Determine requested CI changes	0:00	WAPPR		
3	30	Make CI attribute changes if it is requ	0:00	WAPPR		
4	40	Make CI relationship changes if it is	0:00	WAPPR		
5	50	Send email notification to CI Owner	0:00	WAPPR		

Figure 3-33 Configuration Process job plan

Approving a Configuration Process Record

Responsible role: Configuration Librarian / Configuration Manager.

The Configuration Librarian or Configuration Manager receives the Configuration Process Record and approves it or not.

To approve the record, the Status of the record has to be changed to Approved, as shown in Example 3-34.

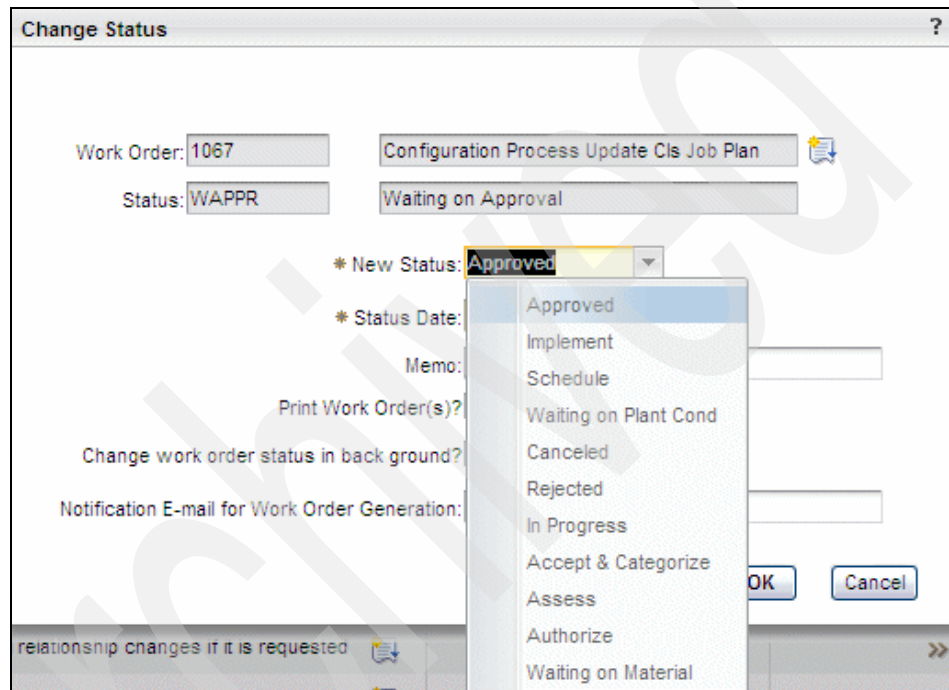


Figure 3-34 Approve Configuration Process Record

Initiating tasks

Responsible role: Configuration Librarian.

To start working on the Configuration Process Record, it has to be changed to In Progress.

When the status is changed to In Progress, the first task automatically receives the same status.

When a task has its status changed to Complete, the next task status automatically gets the status In Progress, as shown in Figure 3-35.

Configuration Process: 1132 CI Configuration Supervisor: Status: INPRG

Parent WO: Lead: Site: AUSTIN

Job Plan: UPDATECI Work Group:

Audit CIs Details

Children of Configuration Process 1132 Filter Download

Tasks for Configuration Process 1132 Filter Download

Sequence	Task	Summary	Estimated Duration	Status	Owner	Owner Group
1	10	Review Update CI request and work	0:00	COMP		
2	20	Determine requested CI changes	0:00	INPRG		
3	30	Make CI attribute changes if it is requ	0:00	WAPPR		
4	40	Make CI relationship changes if it is	0:00	WAPPR		
5	50	Send email notification to CI Owner	0:00	WAPPR		

Generate Task Schedules

Figure 3-35 Initiate Tasks

Executing job plan tasks

Responsible role: Configuration Librarian.

The following tasks are part of the default job plan. It can be modified as needed in each Configuration Process Request.

After each task is implemented, its status has to be changed to Complete.

Task 10

The first task, Review Update CI request and work plan, must be executed by the Configuration Librarian and its objective is to review the work plan.

Task 20

The second task, Determine requested CI Changes, is to determine what CI Changes must be made by the request. It can be determined:

- By reviewing the related RFC
- By checking the current authorized CIs and actual CIs
- By creating an audit CI request

Task 30

The third task, Make CI Attribute Changes if requested, performs the Changes determined in task 20:

1. Select **Selection Action** → **Move/Swap/Modify** to make the CI Changes (Figure 3-36).
2. Click the **Modify** tab and then open the Configuration Items tab.
3. Select the CI and its Attribute you want to modify and click **Modify Attribute**.
4. After inserting a new value, click **Save as Plan**. Clicking **Save as Plan** causes the CI to be updated as specified when the configuration process is completed. Save as Plan is the preferred choice if you want to synchronize the Change and Configuration Management processes.
5. If you do not want to synchronize, click **Execute now**.

The screenshot displays the 'Modify' tab in a configuration management tool. It is divided into three main sections:

- CIs:** A table with columns 'Configuration Item Number', 'Configuration Item Name', and 'Description'. It contains one entry: 'TPT400' with the name 'THINKPAD T400' and description 'COMPUTER,Laptops, 4GB, 2,26MHZ'. A 'New Row' button is at the bottom right.
- CI Specifications:** A table with columns 'Attribute', 'Description', 'Section', 'Data Type', 'Alphanumeric Value', 'Numeric Value', and 'Unit of Measure'. It contains one entry: 'MEMORY' with description 'Memory', section 'ALN', and value '4GB'. A 'Modify Attribute' button is at the bottom right.
- Planned Modifications:** A table with columns 'Configuration Item', 'Configuration Item Name', 'Attribute', 'Data Type', 'Old Value', and 'New Value'. It contains one entry: 'TPT400' with name 'THINKPAD T400', attribute 'MEMORY', data type 'ALN', old value '4GB', and new value '8GB'. A search icon is next to the new value field.

At the bottom of the interface are three buttons: 'Execute Now', 'Save As Plan', and 'Cancel'.

Figure 3-36 Make Changes in CI Attribute

Task 40

The fourth task, Make CI Relationship Changes if requested, performs Changes in CI relationships.

The Changes in CI relationships must be done in the configuration items application.

Select **Go to** → **IT Infrastructure** → **Configuration Items**. Then select the CI, go to the Related CIs tab, and make the Changes.

“Defining relationships” on page 113 shows the Related CIs tab.

Task 50

The fifth and last task, Send an email notification to CI Owner, is to ensure that the CI owner is notified about Changes in the CI.

To send an email, select **Select Action** → **Create** → **Communication**.

Figure 3-37 shows a communication form.

Create Communication

To create communication, select a template or directly enter the message and details.

Template: >>

To:

cc:

bcc:

* Send From:

Reply To:

Subject:

Message:

Attachments | Filter >

Document	Description
...No rows to display...	

Figure 3-37 Send an email notification

Closing the Configuration Process Record

Responsible role: Configuration Librarian.

After the last task is implemented, the Configuration Process Record status is automatically changed to Complete.

The Configuration Librarian has to verify if every task was implemented correctly and change the Configuration Process Record status to Close or Failed.

When the Configuration Process Record has the status changed to Closed, the related Update CI Request is changed to closed also, as shown in Figure 3-38.

The screenshot displays a software interface for managing Configuration Process Records. At the top, there are tabs: List, Configuration Process, Plans, Related Records (selected), Actuals, Log, Specifications, and Workplan Map. Below the tabs, there are input fields for 'Configuration Process: 1132', 'CI Configuration', 'Site: AUSTIN', and 'Status: CLOSE' (the latter is highlighted with a red box). Below these fields are sections for 'Related Configuration Process' and 'Related Tickets', each with a 'Filter' button and a 'Download' button. The 'Related Tickets' section contains a table with the following data:

Related Record Key	Description	Class	Status	Relationship
1033	CI Configuration	SR	CLOSED	ORIGINATOR

The 'Status' column for the record with key 1033 is highlighted with a red box. At the bottom right of the table, there are buttons for 'Select Ticket' and 'New Row'.

Figure 3-38 Closed Configuration Process Record

3.2.8 Verifying and auditing the CI process

Auditing of authorized CIs is a periodic process designed to maintain the accuracy of the authorized CIs. The Verify and Audit process compares the authorized CIs with actual CIs and the following variances:

- ▶ Reporting attribute and relationship variances
- ▶ Reporting on actual CIs for which no authorized CI was found
- ▶ Reporting on authorized CIs for which no actual CI was found

The process of auditing CIs and resolving identified variances is also called auditing.

Terminology: Starting with CCMDB V7.2.1, all CCMDB terminology has been aligned with ITIL V3 terms. In this book we are using Audit for all titles involving CI comparisons and no longer using the Maximo term “reconciliation” for this. We only use “reconciliation” when referring to the actual application names in CCMDB. For example, we call the CCMDB task that is used for audit the “reconciliation task”, but we call the actual process the “audit process.”

Creating, submitting, and accepting a CI audit request

This section provides a step-by-step explanation of how to request and run the audit process in CCMDB.

Submitting an audit request

Responsible role: Configuration Manager or Configuration Auditor.

To create an audit request, the first step is to create a Process Request. The Configuration Manager or Configuration Auditor submits an audit request using the Process Request application:

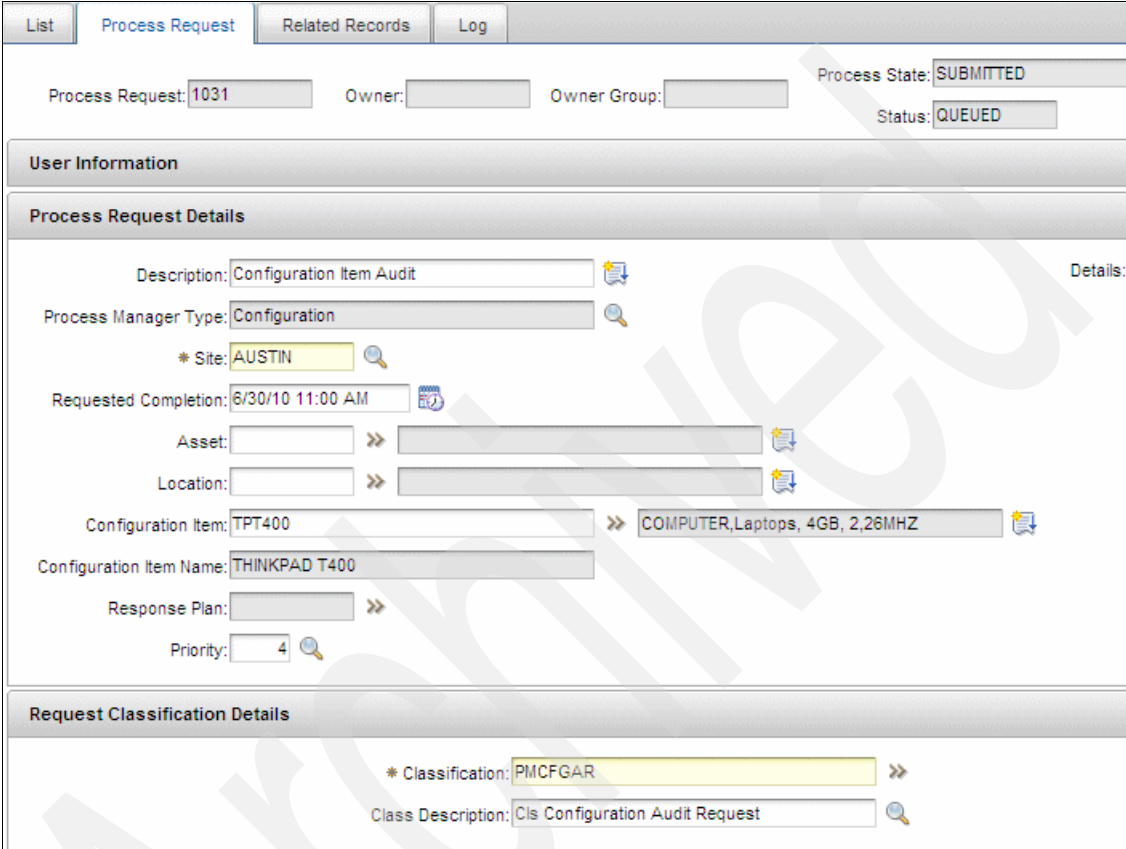
1. Select **Go To → IT Infrastructure → Process Request**.
2. Within the Process Requests Application, click **New Process Request**.

An automatic ID will be assigned for the new request. Provide the following information:

- **Description:** A title for the request.
- **Priority:** A priority suggestion.
- **Details:** A detailed explanation.
- **Process Manager Type:** The type of the request, in this case, Configuration.
- **Site:** The site where the configuration audit request will be applied.
- **Requested Completion:** The target date.
- **Configuration Item:** Select the CI that will be audited.
- **Classification:** The request classification, in this case, PMCFGAR.
- **Class Description:** The description of classification. It will be fulfilled automatically according to the classification chosen.
- **Target CIs:** Use Target CIs if you need to audit more than one CI. You can specify either assets or CIs associated with the process request.

3. Click **Submit** to submit the request.

Figure 3-39 shows a sample Process Request window for the configuration audit request.



The screenshot displays the 'Process Request' window for a configuration audit. The interface includes a top navigation bar with tabs: 'List', 'Process Request' (selected), 'Related Records', and 'Log'. Below the tabs, there are input fields for 'Process Request' (value: 1031), 'Owner', 'Owner Group', 'Process State' (value: SUBMITTED), and 'Status' (value: QUEUED). The main content area is divided into three sections: 'User Information', 'Process Request Details', and 'Request Classification Details'. The 'Process Request Details' section contains several fields: 'Description' (Configuration Item Audit), 'Process Manager Type' (Configuration), '* Site' (AUSTIN), 'Requested Completion' (6/30/10 11:00 AM), 'Asset', 'Location', 'Configuration Item' (TPT400), 'Configuration Item Name' (THINKPAD T400), 'Response Plan', and 'Priority' (4). The 'Request Classification Details' section includes '* Classification' (PMCFGAR) and 'Class Description' (CIs Configuration Audit Request). The window also features a large, faint watermark reading 'PMCFGAR' diagonally across the center.

Figure 3-39 Configuration Audit Process Request

Accepting or rejecting the Process Request

Responsible role: Configuration Manager or Configuration Auditor.

The Configuration Manager or Configuration Auditor receives the request for configuration audit in the queue displayed in their Start Center (default Template-PMCFGAUD).

The approver (Configuration Manager or Configuration Auditor) opens the request and reviews it, sees that it meets the basic requirements, accepts the request, and assigns an owner.

Figure 3-40 shows a sample of the Configuration Process Requests window.

Process Request	Description	Class Structure	Reported By	Reported Date	Site	Process State	Status Date
1031	Configuration Item Audit	AUDITREQ	MAXADMIN	6/3/10 11:42 AM	AUSTIN	SUBMITTED	6/3/10 12:12 PM

Figure 3-40 Configuration Process Requests Queue

Assigning an owner for the Process Request

Responsible: Configuration Manager or Configuration Auditor.

The process request record is displayed for the Configuration Manager (or the Configuration Auditor) in their Start Center. If the owner was not assigned in the Process Request Application, the Configuration Manager has to assign one.

The Configuration Manager can assign another person to be the owner of the configuration audit request or can take ownership for the request. Figure 3-41 shows how to assign an owner.

Process Request: 1033 Owner: Owner Group:

Figure 3-41 Assigning owner for configuration audit request

Reviewing job plan (and tasks for the job plan)

Responsible role: Configuration Audit Owner.

The Configuration Audit Owner opens the record in the Configuration Processes application. In the configuration audit example, a job plan was automatically associated when the process request classification for audit was selected.

The Configuration Audit Owner can review the list of tasks associated with the job plan or select a new one. This populates the configuration process with a set of activities and tasks and now becomes a work order.

Select **IT Infrastructure** → **Configuration Processes** and select the **Configuration Process**. Go to **Plans** tab and review tasks or select a new job plan.

Figure 3-42 shows a sample window of the configuration audit job plan.

Sequence	Task	Summary	Estimated Duration	Status	Owner	Owner Group
1	10	Review Update CI request and work	0:00	WAPPR		
2	20	Determine requested CI changes	0:00	WAPPR		
3	30	Make CI attribute changes if it is re	0:00	WAPPR		
4	40	Make CI relationship changes if it	0:00	WAPPR		
5	50	Send email notification to CI Owner	0:00	WAPPR		

Figure 3-42 Configuration Audit job plan

The Configuration Audit Owner can customize the set of activities and tasks to be used to complete the audit process or change the job plan. If another job plan is needed, follow these steps:

1. Click the **Select Action** → **Remove Work Plan**.
2. In the job plan field, click the **Details** → **Select Value** and select a new job plan.

Note in Figure 3-42 that the work order record number was used as a Configuration Process. When the Process Request is accepted, CCMDDB generates another work order record number and associates it with the Process Request. For our next steps, the work order record number will be used.

Another way to navigate through the windows to reach the right work order number is to select **IT Infrastructure** → **Process Requests**. Select the process request and go to the Related Records tab. In the work order field, click the **Details** menu and select **Go to Configuration Processes**.

Figure 3-43 shows how to navigate to Configuration Processes from the Process Request application.

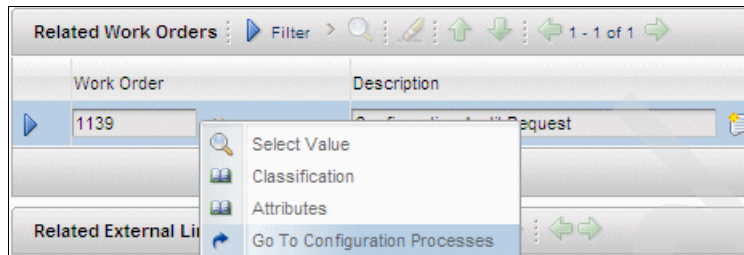


Figure 3-43 Navigating to Configuration Processes from the Process Request application

When a job plan is applied to a configuration process, its activities are inherited by the configuration process.

Audit request is accepted

Responsible role: Configuration Audit Owner.

The Configuration Audit Owner changes the status of the Configuration Process record to **In Progress** to initiate the first activity in the job plan.

Click the **Change Status** button and select the **In Progress** status.

Figure 3-44 shows how to change the status of the configuration process record.

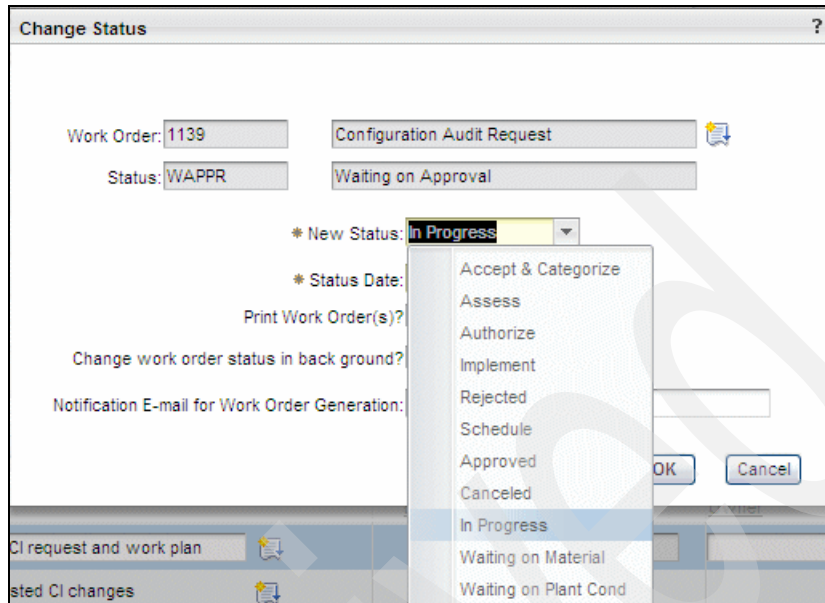


Figure 3-44 Changing the Configuration Process record status to In Progress

3.2.9 Auditing

The Reconciliation module application allows the comparison of actual and authorized configuration items, investigate discrepancies, and take actions on the results. CCMDB maintains both versions of CI in two applications: the configuration items application and the actual configuration items application.

Configuration items application

In the configuration items application, it is possible to create and maintain data about configuration items that conform to rules and relationships specified. This is data that is recorded about what was acquired and installed. It represents the authorized inventory, how things must be, and what has been planned. These configuration items are called authorized configuration items.

Actual configuration items application

In the actual configuration items application, information about data is collected directly from components actually installed in an enterprise. To gather this data, discovery tools, such as TADD, scan computers, network devices, and other information technology components deployed in the enterprise and record information about the hardware and software installed on those components. An integration tool, such as Integration Composer, imports the collected data into CCMDB.

Reconciliation module applications

Reconciliation module applications are designed to work with both CCMDB and TAMIT. For this reason, a few fields in the applications display both Asset and Deployed Asset options as well as CI and actual CI options. Certain applications in the module are specific to the software that you purchase, and the license key determines which applications your system displays.

The six applications included on Reconciliation module are: Reconciliation Tasks, Task Filters, Link Rules, Comparison Rules, CI Link Results, and CI Reconciliation Results. They can be accessed by **Go to → Administration → Reconciliation** as shown in Figure 3-45.

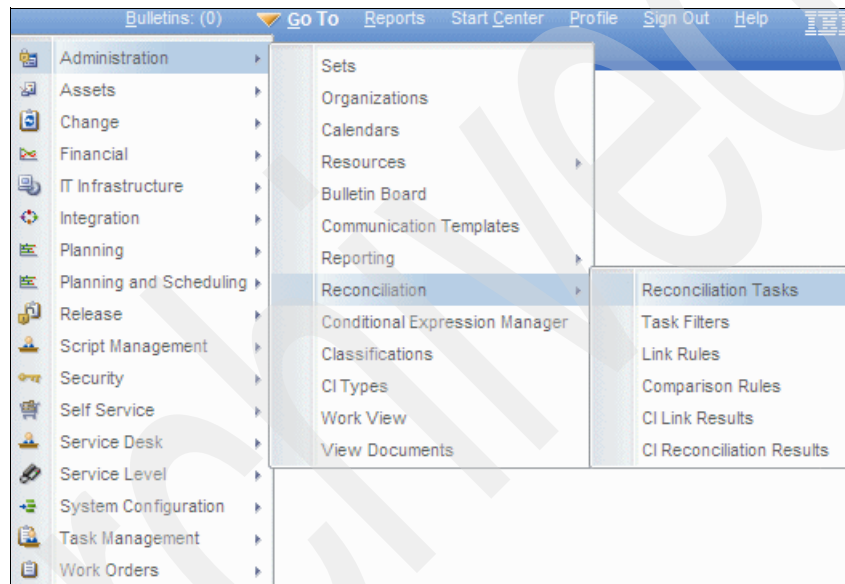


Figure 3-45 Reconciliation Menu

Auditing overview

The Reconciliation module applications permit evaluating data about information technology devices and networks. Reconciliation module applications can be used to perform two functions:

- ▶ Define auditing tasks that allows comparison of information from one data set with information in another data set.
- ▶ View and manage the results of Audits.

Reconciliation module applications can be used to configure a background process that reconciles objects from one data set (Data Set 1) with objects in another data set (Data Set 2). For CCMDB, typically Data Set 1 contains information about authorized CI objects, that is, data that you maintain in the configuration items application. Data Set 2 typically contains information maintained in the actual configuration items database.

The auditing process identifies successful matches as well as discrepancies and variances between the two sets of data. The results of an audit can be used to determine whether the objects actually deployed comply with corporate plans and whether the Changes over the lifecycle of an object are in compliance with corporate policies. Discrepancies might be caused by a variety of factors, including:

- ▶ Incorrect data entry
- ▶ Reconfigured equipment
- ▶ Retired equipment
- ▶ Theft
- ▶ Unauthorized use of hardware and software in the enterprise
- ▶ Unauthorized Changes or Changes implemented without correctly following all defined steps for the process (no updates in configuration items database, in this case)

To define the parameters for an audit, create a reconciliation task that combines the elements required for an audit into a specific task (see “Setting up reconciliation tasks” on page 154). A reconciliation task consists of three possible components: a task filter (optional), one or more link rules (required), and one or more comparison rules (optional). Use the Reconciliation module applications to create these components. After creating a reconciliation task, use the Cron Task Setup application to create a schedule for running the audit.

Auditing options

For CCMDB, there are three basic types of auditing options you can perform:

- ▶ **Attributes equality:**

Compares an attribute, attributes of a child or parent object in Data Set 1 with a specific attribute, attributes of a child or parent object in Data Set 2.

For example, authorized CI records for computer systems can be evaluated at a specific site to determine whether the RAM on the computers in authorized CIs matches the RAM actually installed on computers in actual CIs.

- ▶ **Matches found:**

Specifies the ratio of object instances in Data Set 1 to object instances in Data Set 2 to look for in the comparison.

For example, authorized CI records for computers can be compared at a specific site with actual CI records to determine whether a specific software application is actually installed as expected on computers that are actually deployed. In other words, the authorized CI records indicate that the software is installed on certain computers. Determine if the records in actual CI data include an instance of that software on the corresponding computers.

► Full CI comparison:

Compares the relationships of authorized CIs and the attributes associated with the authorized CIs with the relationships and attributes associated with the corresponding actual CIs.

You can run a full CI comparison and use the results to determine whether the relationships in authorized CI data between a computer system, operating system, and software component matches the relationships in actual CI data. For example, when you reconcile authorized CIs against actual CIs, determine if there is a successful match to indicate that for the computer system Mail Server 1, IBM Lotus® IBM Domino® Server is installed on Microsoft® Windows XP and that Windows XP runs on Mail Server 1.

Setting up an audit

CCMDB reconciles Data Set 1 and Data Set 2 by performing a rule-based compare operation defined in a reconciliation task. Use the Reconciliation module applications to define a reconciliation task and then use the Cron Task Setup application in the System Configuration module to create a cron task that schedules the reconciliation task to run. After the reconciliation task runs, authorized users can view results of the audit in the CI Link Results and CI Reconciliation Results applications.

Use the following steps to set up and execute an audit:

1. Set up a task filter. A task filter is optional.
2. Define one or more link rules.
3. Define one or more comparison rules. Comparison rules are optional.
4. Set up a reconciliation task.
5. Create a cron task to schedule the audit.
6. View results of the audit.
7. If appropriate, resolve discrepancies and document how you resolve them.

Setting up a task filter

A task filter record specifies a subset of either Data Set 1 or Data Set 2 that will be evaluated when a reconciliation task is executed. For example, it can be defined as a filter that selects CIs only at the Boston site; in this case the reconciliation task evaluates top-level authorized CIs only at the Boston site, but evaluates all actual CIs.

A task filter is an optional component of a reconciliation task that can be used to limit the scope of a reconciliation task. Use the Task Filters application to set up task filters.

After a task filter is created, use the Reconciliation Tasks application to associate the filter with a specific reconciliation task, and the system applies the task filter each time the reconciliation task runs. If a task filter is not defined for a reconciliation task, the system compares all top-level authorized CI objects with all top-level actual CI objects.

The Task Filters application can be used to perform the following actions:

- ▶ Create a new task filter.
- ▶ Delete a task filter.
- ▶ Duplicate a task filter.
- ▶ Modify an existing task filter.

To open the Task Filters application, select **Go To** → **Administration** → **Reconciliation** → **Task Filters**.

Task filter components

A task filter includes the following components:

- ▶ **Filter name:** A unique name (specified in the Filter field) that identifies the task filter.
- ▶ **Description (optional):** A brief description of the task filter.
- ▶ **Filter type:** Type of task filter. The type selected determines which set of objects the filter applies to. For CCMDB configuration items, either CIs or actual CIs can be selected.
- ▶ **Filter clause(s):** In the Task Filter Clauses table window, at least one clause that specifies an attribute and a value for the task filter must be defined. Multiple attribute clauses can be created for a task filter.

Task filter example

Figure 3-46 shows a Task Filters application window for creating a task filter.

The screenshot shows the 'Task Filters' application window. At the top, there's a menu bar with 'Bulletins: (0)', 'Go To', 'Reports', 'Start Center', 'Profile', 'Sign Out', and 'Help'. Below the menu bar is a search bar with 'Find:' and a 'Select Action' dropdown. The main area has two tabs: 'List' and 'Task Filter'. The 'Task Filter' tab is active, showing a filter named 'Redbook Task Filter' with a description 'Task Filter used by CCMDB Redbook' and a filter type of 'CI'. Below this, the 'Task Filter Clauses' section shows a single clause with the attribute 'LOCATION' and the value 'R001'. A 'Details' section at the bottom shows the attribute 'LOCATION' and value 'R001'. A 'New Row' button is at the bottom right.

Figure 3-46 Creating a Task Filter

If multiple clauses that specify various attributes are created, the system processes the clauses using a logical AND between the clauses. For example, if a task filter is set up for actual CIs based on the Location and Site attributes, the system selects only actual CIs that meet with both criteria.

If multiple clauses for the same attribute are created, the system processes the clauses using a logical OR between clauses. For example, a task filter is created for authorized CIs with two filter clauses for Site, one for Boston and one for New York; the system selects records that have either Boston or New York as a site.

To specify an attribute, select it from a predefined value list. The values in the list depend on the value selected for the Filter Type field. Table 3-2 shows the values available for each filter type.

Table 3-2 Filter type values

Filter type	Attribute
CI	CI Number Class Structure Collection Item Location Organization ID Service Service Group Site ID Status Work Order
Actual CI	Class Structure GUID
Asset (This filter type does not apply to CIs.)	Asset Asset Class Structure Collection Custodian GL Account Organization Site Status Usage Work Order
Deployed Asset (This filter type does not apply to CIs.)	Asset Class Organization Site System Role

You can change the filter clauses, but you cannot change the task filter name or filter type. In this case, you need to create a new record with another name and delete the existing one. When you save a modified task filter, the application evaluates the filter to determine whether the clauses are valid expressions; if a clause is not valid, the application displays an error message and does not save the task filter.

You can **Delete Task Filter** from the Select Action menu on the Task Filter tab, but you cannot delete a task filter if it is associated with a task defined in the Reconciliation Tasks application.

You can **Duplicate Task Filter** from the Select Action menu on the Task Filter tab.

Setting up link rules

A link rule is a required component of a reconciliation task. Link rules establish the basis for auditing by identifying which top-level object in Data Set 1 to link with a top-level object in Data Set 2. Link rules are often based on unique identifiers. The attribute most commonly used to link CIs with actual CIs is the actual CI number (ACTCINUM).

After a link rule is created, use the Reconciliation Tasks application to associate the link rule with a specific reconciliation task. When the system executes the reconciliation task, it evaluates each link rule on the task and attempts to match the object and attribute defined in the rule for Data Set 1 with the object and attribute defined in the rule for Data Set 2.

The system evaluates link rules in a reconciliation task in a cascading sequence, based on the sequence numbers, until it finds a match or until it reaches the end of the cascading rule list. If the system finds a match, it displays the link result in the Link Results application. If the system does not find a match or finds multiple matches, it displays a link rule failure result in the CI Reconciliation Results application.

Use the Link Rules application to perform the following actions:

- ▶ Create a new link rule.
- ▶ Delete a link rule.
- ▶ Duplicate a link rule.
- ▶ Modify an existing link rule.

Link rule components

A link rule consists of the following elements:

- ▶ **Link name:** A unique name (specified in the Link field) that identifies the link rule.
- ▶ **Description (optional):** A brief description of the link rule.
- ▶ **Data set specifications for Data Set 1 and Data Set 2** that indicate what data to reconcile.
- ▶ **Link clauses:** In the Link Clauses table window, at least one clause that defines a relation (or link) between a top-level object in Data Set 1 and a top-level object in Data Set 2 must be created. Each link clause identifies an object and attribute in Data Set 1 to link to a specific attribute in Data Set 2 when the system executes a reconciliation task.

Clauses: The Link Clauses table window displays selected fields for each clause. To view all fields for a clause, select a row and click **View Details**.

Table 3-3 describes the elements of a link clause.

Table 3-3 Link clause elements

Field	Function	Rules and requirements
Sequence	Number that specifies the order in which to process the clause.	▶ Mandatory. ▶ Use a unique number for each clause. ▶ Use a number greater than 0. ▶ The default is increments of ten in ascending order.
Open Parenthesis (...)	Marks the beginning of a set of clauses grouped together so that the system can perform operations on them in a specific order.	Optional. However, for each open parenthesis, use a corresponding close parenthesis.
Data Set 1 Object	Specifies the target object in Data Set 1.	▶ Mandatory. ▶ Selected from a value list that includes the following values: – For assets: • ASSET (Asset). • ASSETSPEC (Asset Specification). – For configuration items (CIs): • CI (configuration item). • CISPEC (CI Specification).
Data Set 1 Class Structure	When selecting a specification as the Data Set 1 object, this field identifies a specific class structure for auditing.	▶ Mandatory if ASSETSPEC or CISPEC is selected for the object. ▶ Selected from a value list. Values in the list are class structure identifiers for the top-level objects.

Field	Function	Rules and requirements
Data Set 1 Class Structure Description	Displays a description of the selected class structure.	Read-only field.
Data Set 1 Classification	Displays the classification for the selected class structure.	Read-only field.
Data Set 1 Attribute	Identifies the specific attribute of the object or class structure to link. For assets and deployed assets, the attribute is typically a serial number or asset tag. For CIs and actual CIs, it is typically ACTCINUM, the actual CI number.	▶ Mandatory. ▶ Selected from a value list. Values in the list depend on the value selected in the Data Set 1 Object field and, if applicable, the Data Set 1 Class Structure field.
Data Set 1 Attribute Title	Displays the title of the Data Set 1 object attribute.	Read-only field.
Operator	Identifies the type of link between Data Set 1 and Data Set 2.	The equals (=) operator is read-only; it cannot be changed.
Data Set 2 Object	Identifies the target object in Data Set 2.	Selected from a value list that includes the following values: ▶ For assets: DEPLOYEDASSET. ▶ For configuration items (CIs) : ACTCI (actual CI). ACTCISPEC (actual CI Specification).
Data Set 2 Class Structure	When selecting a specification as the Data Set 2 object, this field identifies a specific class structure for auditing.	▶ Mandatory if ACTCISPEC is selected. ▶ Selected from a value list. Values in the list are class structure identifiers for the top-level objects.

Field	Function	Rules and requirements
Data Set 2 Class Structure Description	Displays a description of the selected class structure.	Read-only field.
Data Set 2 Classification	Displays the classification for the selected class structure.	Read-only field.
Data Set 2 Attribute	Identifies the specific attribute in Data Set 2 to link. For assets and deployed assets, the attribute is typically a serial number or asset tag. For CIs and actual CIs, it is typically ACTCINUM, the actual CI number.	▶ Mandatory. ▶ Selected from a value list. Values in the list depend on the value selected in the Data Set 2 Object field and, if applicable, the Data Set 2 Class Structure field.
Data Set 2 Attribute Title	Displays the title of the Data Set 2 attribute selected.	Read-only field.
Close Parenthesis ...)	Marks the end of a set of clauses grouped together so that the system can perform operations on them in a specific order.	Optional. However, for each open parenthesis, use a corresponding close parenthesis.
Sequence Operator	When more than one link clause exists, this operator prescribes how the current clause relates to the next clause in the sequence.	▶ Required if a link rule consists of more than one clause. ▶ Must be empty for the last row in the sequence (that is, the row with the highest sequence number). ▶ Selected from a value list that includes the following values: – AND – OR

Creating link rules

It is possible to create link rule records from the List tab or from the Link Rule tab in the Link Rules application.

Before saving a link rule, the following requirements must be satisfied:

- ▶ Assign a unique link rule name.
- ▶ Define data set specifications for Data Set 1 and Data Set 2 that indicate what data to reconcile.
- ▶ Create at least one link rule clause.
- ▶ Clauses must be valid expressions. When saving a link rule, the application uses the following rules to determine whether clauses are valid expressions. If the application determines that a clause is not valid, it displays an error message and does not save the link rule.
 - Each open parenthesis must have a corresponding close parenthesis.
 - The number in the Sequence field must be unique.

Attention: When entering sequence numbers in random order, the application sorts the clauses and displays them in ascending numerical order when saving the record.

- All rows except the row with the highest sequence number must have a value specified in the Sequence Operator field.
- The row with the highest sequence number must not have a sequence operator. (After the application sorts the clauses, this is the last row in the table window.)

You can change the Link Rule, but you cannot change neither the name of the link rule in the Link field and data sets selected in the Data Set 1 and Data Set 2 fields. When you save a modified link clause, the application evaluates the link rule to determine whether the clauses are valid expressions; if a clause is not valid, the application displays an error message and does not save the link rule.

You can **Delete Link Rule** from the Select Action menu on the Link Rule tab, but you cannot delete a link rule if it is associated with a task defined in the Reconciliation Tasks application.

You can **Duplicate Link Rule** from the Select Action menu on the Link Rule tab.

Figure 3-47 shows a sample of a Link Rules application. To open the Link Rulers application, select **Go To** → **Administration** → **Reconciliation** → **Link Rules**.

The screenshot displays the 'Link Rules' application interface. At the top, there are two input fields: 'Link: Redbook Link Rule' and 'Link Rule used by CCMDB Redbook'. Below this is a section titled 'Type of Reconciliation'. Underneath, there are two data set selection fields: 'Data Set 1: CI' and 'Data Set 2: ACTUAL CI', with a note 'will be linked to' and 'based on the link clause provided.' between them. A 'Link Clauses' section follows, featuring a toolbar with icons for filter, search, add, remove, and sort, along with a 'Download' button. Below the toolbar is a table with columns: 'Sequence (...)', 'Data Set 1 Attribute', 'Operator', 'Data Set 2 Attribute', and 'Sequence Operator'. The first row contains the value '10' in the 'Sequence' column, 'CINUM' in the 'Data Set 1 Attribute' column, an equals sign '=' in the 'Operator' column, 'ACTCINUM' in the 'Data Set 2 Attribute' column, and an empty field in the 'Sequence Operator' column. A 'New Row' button is located at the bottom right of the table.

Sequence (...)	Data Set 1 Attribute	Operator	Data Set 2 Attribute	Sequence Operator
10	CINUM	=	ACTCINUM	

Figure 3-47 Creating link rules

Setting up comparison rules

A comparison rule is an optional component of a reconciliation task. It defines how to compare objects or attributes of a child or parent object in one data set with a child or parent object in another data set when the system executes a reconciliation task. For example, you can set up a comparison rule to compare software on computers in authorized CIs with software on computers in actual CIs. A task can include more than one comparison rule.

To create a comparison rule, define Data Set 1 and Data Set 2. Then define the comparison rule. It is possible to define a filter for Data Set 1 or Data Set 2 to limit your comparison to a subset of either data set.

For CCMDB, there are three basic types of comparison rules:

- ▶ **Attributes equality:** Compares an attribute or attributes of a child or parent object in Data Set 1 with a specific attribute or attributes of a child or parent object in Data Set 2.
- ▶ **Matches found:** Specifies the ratio of object instances in Data Set 1 to object instances in Data Set 2 to look for in the comparison.
- ▶ **Full CI comparison:** Compares the relationships of authorized CIs and the attributes associated with the authorized CIs with the relationships and attributes associated with the corresponding actual CIs. Full CI comparison rules are a special type of comparison rule.

After a comparison rule is created, use the Reconciliation Tasks application to associate the rule with a specific reconciliation task, and the system includes the comparison rule each time it executes the reconciliation task. When the system runs a reconciliation task, it processes link rules first. The link rule specifies the top-level object and attribute in one data set to match with a specific attribute of a top-level object in another data set. When the system processes a reconciliation task, it processes comparison rules in the task only if the link rule successfully links an object in Data Set 1 with an object in Data Set 2.

When defining a reconciliation task in the Reconciliation Tasks application, it is possible to specify how to report comparison results by selecting from the following options:

- ▶ All results, both successful and failed matches
- ▶ Instances where the object from Data Set 1 failed to reconcile against the object from Data Set 2
- ▶ Instances where the object from Data Set 1 successfully matched the object from Data Set 2

When the system executes a reconciliation task, it lists the results of comparison rule evaluations in the CI Reconciliation Results application.

In summary, use the Comparison Rules application to perform the following actions:

- ▶ Create a new comparison rule.
- ▶ Delete a comparison rule.
- ▶ Duplicate a comparison rule.

Figure 3-48 shows a sample of a Comparison Rules application. To open the Comparison Rules application, select **Go To → Administration → Reconciliation → Comparison Rules**.

Comparison rules identify object(s) or attributes(s) of child or parent Data Sets which are compared when a reconciliation task is executed. Specify at least one Matches Found or one Attributes Equality, but not both. Use the Data Set 1 Filter and Data Set 2 Filter to limit or focus the comparison.

Comparison:

Type of Reconciliation

Data Set 1: will be compared to Data Set 2: based on the comparison clause provided.

Full CI Comparison

Full CI Comparison? ☐

Data Set 1 Filter Data Set 2 Filter Matches Found **Attributes Equality**

Attributes Equality Clauses Filter 1 - 1 of 1

Sequence	Data Set 1 Attribute	Operator	Data Set 2 Attribute
10	CLASSSTRUCTUREID	=	CLASSSTRUCTUREID

Figure 3-48 Defining Comparison Rules

Comparison rules components

You can use the following components to create comparison rules:

- ▶ **Comparison name (required):** A unique name (specified in the Comparison field) that identifies the comparison rule.
- ▶ **Description (optional):** A brief description of the comparison rule.
- ▶ **Data set specifications for Data Set 1 and Data Set 2** that indicate what data to reconcile (required).
- ▶ **Full CI comparison specification (optional):** Selecting the Full CI Comparison check box creates a full configuration item (CI) comparison rule that lets you compare CI relationships. Selecting this check box disables all sub-tabs in the application. This feature applies only to customers who install CCMDB.
- ▶ **Data Set 1 filter clause(s):** A Data Set 1 filter is optional; however, if a matches found clauses is included, a Data Set 1 filter or Data Set 2, or both, must be defined.

- ▶ Data Set 2 filter clause(s): A Data Set 2 filter is optional; however, if a matches found clauses is included, a Data Set 1 filter or Data Set 2, or both, must be defined.
- ▶ One of the following definitions:
 - Matches found: To define the ratio of object instances in Data Set 1 to object instances in Data Set 2 that one wants to look for in the comparison.
 - Attributes equality: To define how to compare the specific attribute or attributes of a child or parent from Data Set 1 with specific attribute or attributes of a child or parent in Data Set 2.

Data Set 1 and Data Set 2 filter clauses

On the Data Set 1 Filter and Data Set 2 sub-tabs are defined filter clauses that specify a subset of Data Set 1 objects to reconcile against Data Set 2 objects when using a comparison rule and vice-versa. Each clause identifies an object or attribute in Data Set 1 or Data Set 2 to evaluate when the system processes a comparison rule. Data Set 1 is used as an example in the following text to make our explanation easier. Points to consider in Data Set 1 must also be considered in Data Set 2.

When working with comparison rules, it is important to understand that all the filtering and comparisons work on sets of objects and to be aware of the way expressions operate with sets in auditing comparison. To designate which output objects to select, a Data Set 1 and Data Set 2 filter clauses define one of the following conditions:

- ▶ Select an object from Data Set 1 if the selected attribute matches a specified value based on the operator selected. Using the operator specified in the clause, the system evaluates each top-level object in Data Set 1 and all its children and selects any objects that matches the value specified.
- ▶ Select an object from Data Set 1 if the selected attribute of a class specification matches a specified value. Using the operator specified in the clause, the system evaluates each top-level object in Data Set 1 and all its children and selects any objects that belong to the class specified in the clause. Any object that has another class is skipped. Then the filter uses the operator to evaluate the attribute value and selects all objects that match the value specified.
- ▶ Select an object from Data Set 1 if the specified classification exists. The system evaluates each top-level object in Data Set 1 and all its children and selects all instances that have the specified class.

Table 3-4 describes the elements of a Data Set 1 or Data Set 2 filter clause.

Table 3-4 Elements of a Data Set 1 or Data Set 2 filter clause

Field	Function	Rules and Requirements
Sequence	Number that specifies the order in which to process the clause.	▶ Mandatory. ▶ Use a unique number for each clause. ▶ Use a number greater than 0.
Open Parenthesis (...	Marks the beginning of an expression. Parenthesis marks group expressions to control the order of operations when using multiple clauses joined by a logical operator (AND or OR).	Optional. However, for each open parenthesis, use a corresponding close parenthesis.
Data Set 1 / Data Set 2 Object	Specifies the target object in Data Set 1 or Data Set 2.	▶ Mandatory for both Data Set 1 and Data Set 2. ▶ Selected from a value list that includes the following values: – If Data Set 1 is assets: • ASSET (Asset). • ASSETSPEC (Asset Specification). • ITEM (Item). • ITEMSPEC (Item Specification). – If Data Set 1 is configuration items (CIs): • CI (configuration item). • CISPEC (CI Specification).

Field	Function	Rules and Requirements
Data Set 1 / Data Set 2 Class Structure	When selecting a specification as the Data Set 1 object, this field identifies a specific class structure for auditing.	► Mandatory if a specification for the object is selected. ► Selected from a value list. Values in the list are class structure identifiers for the top-level objects.
Data Set 1 / Data Set 2 Class Structure Description	Displays a description of the selected class structure.	Read-only field.
Data Set 1 / Data Set 2 Classification	Displays the classification for the selected class structure.	Read-only field.
Data Set 1 / Data Set 2 Attribute	Identifies the specific attribute of the object or class structure to use for the Data Set 1 / Data Set 2 filter.	► Optional. ► Selected from a value list. The object selected determines which values the system displays in the value list.
Data Set 1 / Data Set 2 Attribute Title	Displays the title of the attribute selected.	Read-only field.
Operator	Identifies the operator for the attribute specification.	Mandatory if an attribute is selected. Otherwise, the field is read-only.
Value	Specifies a value for the attribute selected.	► If an attribute is not selected, the field is read-only. ► If an attribute is selected, the field is mandatory unless NOTEMPTY or NOTNULL is selected as an operator. If NOTEMPTY or NOTNULL is selected, the field is read-only.

Field	Function	Rules and Requirements
Close Parenthesis ...)	Marks the end of an expression. Parenthesis marks group expressions to control the order of operations when you use multiple clauses joined by a logical operator (AND or OR).	Optional. However, for each close parenthesis, use a corresponding open parenthesis.
Sequence Operator	When more than one clause exists, this operator prescribes how the current clause relates to the next clause in the sequence.	▶ Required if the filter consists of more than one clause. ▶ Do not enter a value for the last row in the sequence (that is, the row with the highest sequence number). ▶ Selected from a value list that includes the following values: – AND – OR

Table 3-5 describes the operators that can be used to define the ratio between Data Set 1 and Data Set2 object instances.

Table 3-5 Operators to define data set ratios

Operator	Description
At least 1 to at least 1	At least one Data Set 1 object exists, but you can have more than one, and at least one Data Set 2 object exists, but you can have more than one.
At least 1 to exactly 1	At least one Data Set 1 object exists, but you can have more than one, and only one Data Set 2 object exists.
Exactly 1 to at least 1	Only one Data Set 1 object exists, and at least one Data Set 2 object exists, but you can have more than one.
Exactly 1 to exactly 1	Only one Data Set 1 object exists, and only one Data Set 2 object exists.
Exactly N to exactly N	N Data Set 1 objects exist, and N Data Set 2 objects exist, where N is the same number for each.

Wildcard use

You can use a “wildcard” character or characters with letters or numbers to indicate that you want to find records that begin with, end with, or contain those letters/numbers. The Reconciliation module applications use standard wildcard rules. The wildcards available are asterisk (*), percent sign (%), underscore (_) and question mark (?). Table 3-6 shows examples.

Table 3-6 Wildcard examples

Enter...	To find...
123* or 123%	Records that start with 123, such as 123, 12345, 123ABC, etc.
*123 or %123	Records that end in 123, such as 123, 5123, or PUMP123.
123 or %123%	Records that contain 123, such as 123, 1234, PUMP123, or XX12300Valve.
Elec* or Elec%	Records that contain words that start with “Elec,” such as electric, electromagnetic, or electrode.
123? or 123_	Any four-character records that start with 123, such as 1234, 1230, 123g, etc.
_18 or ?18	Any three-character records that end with 18, such as 418 or J18.

If no wildcard is specified, it is compared as %<value>%. To search for the *, %, _ or ? as characters without using them as wildcard, duplicate them.

Logical AND/OR

A Data Set 1 / Data Set 2 filter can be a logical expression made of multiple clauses using the logical operators AND and OR. Clauses are evaluated from left to right and two values are considered at a time (binary operations). You can use parentheses to alter the order of operations. By default, AND has a higher priority than OR; consequently, if no parentheses are present, operands joined by AND are processed first.

Example:

```
ClauseA OR ClauseB AND ClauseC AND ClauseD is evaluate as:
ClauseA OR ((ClauseB AND ClauseC) AND ClauseD)
ClauseA OR (Result AND ClauseD)
ClauseA OR Result
```

Matches found clauses

A matches found definition is a component of a comparison rule defined in the Comparison Rules application. You use the Matches Found sub-tab on the Comparison Rule tab in the application to create matches found clauses. Each clause specifies the acceptable ratio of Data Set 1 object instances to Data Set 2 object instances in the comparison after the system applies the data set filter or filters. Each clause identifies an object in Data Set 1 and an object in Data Set 2 to evaluate; the clause also includes an operator that specifies the number of instances of the Data Set 1 object and the number of instances of the Data Set 2 object.

The elements of the Matches Found tab have a similar function of the elements from Data Set 1 and Data Set 2 filter. See Table 3-4 on page 149.

Attributes equality clauses

An attributes equality definition is a component of a comparison rule defined in the Comparison Rules application. You use the Attributes Equality sub-tab on the Comparison Rule tab in the application to create attributes equality clauses that define how to compare the specific attribute or attributes of a child or parent object in Data Set 1 with specific attribute or attributes of a child or parent object in Data Set 2 when the system processes a comparison rule.

The elements of the Attributes Equality tab have a similar function of the elements from Data Set 1 and Data Set 2 filter. See Table 3-4 on page 149.

Creating comparison rules

A comparison rule can be created in Comparison Rules application:

Go To → Administration → Reconciliation → Comparison Rules.

The required fields must be filled before saving the comparison rule:

- ▶ Define a unique comparison rule name.
- ▶ Define data set specifications for Data Set 1 and Data Set 2 that indicate what data to reconcile.
- ▶ Define a comparison based on one of the following components:
 - matches found definition.
 - attributes equality definition.
 - full CI comparison specification.
- ▶ If you create a matches found clause, define a Data Set 1 filter or a Data Set 2 filter or both.
- ▶ Create clauses that are valid expressions. When you save a comparison rule, the application uses the following rules to determine whether clauses are valid expressions.

If the application determines a clause is not valid, it displays an error message and does not save the comparison rule:

- Each open parenthesis must have a corresponding close parenthesis.
- The number in the Sequence field must be unique.

Order: You can enter sequence numbers in random order. When you save a record, the application sorts the clauses and displays them in ascending numerical order.

- ▶ All rows except the row with the highest sequence number must have a value specified in the **Sequence Operator** field.
- ▶ The row with the highest sequence number must not have a sequence operator. After the application sorts the clauses, this is the last row in the table window.

Duplicating comparison rules

Comparison rules can be duplicated using the **Duplicate Comparison Rule** option from the Select Action menu.

Deleting comparison rules

It is possible to select **Delete Comparison Rule** from the Select Action menu on the Comparison Rule tab when it is not associated with a reconciliation task defined in the Reconciliation Tasks application. When a comparison rule is deleted, the application deletes all Data Set 1 filter, Data Set 2 filter, matches found, and attributes equality clauses associated with the rule. If you attempt to delete a comparison rule associated with a reconciliation task, the application displays an error message.

Setting up reconciliation tasks

Before executing an auditing to compare two data sets, a reconciliation task must be set up. A reconciliation task record combines a task filter (optional), one or more link rules (required), and one or more comparison rules (optional) into a specific job task that the system executes based on the schedule created in the Cron Task Setup application.

It is possible to use the Reconciliation Tasks application to perform the following actions:

- ▶ Create a new reconciliation task.
- ▶ Delete a reconciliation task.
- ▶ Duplicate a reconciliation task.
- ▶ Modify an existing reconciliation task.

Figure 3-49 shows a sample of the Reconciliation Tasks application. To open the Reconciliation Tasks application, select **Go To** → **Administration** → **Reconciliation** → **Reconciliation Tasks**.

List **Reconciliation Task**

A reconciliation task combines a task filter (optional), one or more link rules, and one or more comparison rules (optional) into a task that task filter specifies a subset of objects to reconcile. The value in the Comparison Results field specifies how the system reports results reconciliations, or successful reconciliations.

* Reconciliation Task: Filter Type:

Task Filter: Last Completion Date:

Comparison Results: Is Case Sensitive? ☐

Type of Reconciliation

Data Set 1: will be reconciled to Data Set 2:

Link Rules Filter > 1 - 1 of 1 Download ?

Sequence	Link	Description	Data Set 1	Data Set 2
10	Redbook Link Rule	Link Rule used by CCMDDB Redbook	CI	ACTUAL CI

Select Link Rule New Row

Figure 3-49 Reconciliation Tasks

Reconciliation task components

Use the following components to create reconciliation tasks:

- ▶ Reconciliation task name: A unique name (specified in the Reconciliation Task field) that identifies the reconciliation task.
- ▶ Description: A brief description of the audit.
- ▶ Task filter (optional): A task filter for the reconciliation task. When selecting a task filter, the application displays the type for the selected filter in the Filter Type field. Task filters apply only to link rules, not comparison rules.
- ▶ Filter type: Type of task filter associated with the reconciliation task.
- ▶ Case sensitivity specification: The Is Case Sensitive? check box specifies whether or not the reconciliation task is case sensitive. Selecting the check box makes all elements of the reconciliation task case sensitive, including the task filter and any link rules and comparison rules associated with the task.

- ▶ **Comparison results specification:** The Comparison Results field specifies what kind of result records to add when a comparison rule is included in the reconciliation task. This field is not active unless you define a comparison rule.
- ▶ **Data set specifications for Data Set 1 and Data Set 2** that indicate what data to reconcile.
- ▶ **Link Rule(s):** In the Link Rules table window, you specify one or more link rules for the reconciliation task. A reconciliation task must include one or more link rules. If you add multiple link rules, you must assign a sequence number to each rule to specify the order in which to process the rule. When the system executes the reconciliation task, it evaluates each link rule and attempts to match the object and attribute in Data Set 1 with the object and attribute in Data Set 2. The Link Rules table window on the Reconciliation Task tab displays the following information about the link rules used in the reconciliation task:
 - **Sequence:** Unique sequence number to specify the order in which to process the link rule when multiple link rules exist.
 - **Link:** Unique name to identify the link rule.
 - **Description:** Link rule description.
 - **Data Set 1:** Data Set 1 specified in the link rule.
 - **Data Set 2:** Data Set 2 specified in the link rule.

Select the **Link Rule** button in the Link Rules table window to select one or more link rules for a reconciliation task. Selecting this button opens a dialog box that lists link rules that you have created.

- ▶ **Comparison Rule(s) (optional):** Use the **Select Comparison Rule** button in the Comparison Rules table window to select one or more comparison rules for a reconciliation task. If you add a comparison rule, you must select a value in the Comparison Results field to specify what kind of comparison results to display in the CI Reconciliation Results application. The system applies comparison rules only after it establishes a successful link between an object from Data Set 1 and an object from Data Set 2. The Comparison Rules table window on the Reconciliation Task tab displays the following information about the comparison rules used in the reconciliation task:
 - **Comparison:** Unique name to identify the comparison rule.
 - **Description:** Comparison rule description.
 - **Data Set 1:** Data Set 1 specified in the comparison rule.
 - **Data Set 2:** Data Set 2 specified in the comparison rule.

Use the Reconciliation Tasks application to create reconciliation tasks that can be scheduled for execution using the Cron Task application. Create reconciliation task records from the List tab or from the Reconciliation Task tab in the application.

Scheduling tasks

The Cron Task Setup application is used for scheduling a reconciliation task. A reconciliation task record combines the components required for an audit into a specific job task that the system runs based on the schedule that is set up in the Cron Task Setup application.

The same name specified in the Reconciliation Task field must be used in the cron task as a value of RECONTASKNAME parameter.

The system allows scheduling cron tasks for multiple reconciliation tasks. Reconciliation task records that are associated with a cron task cannot be deleted. When the system executes a reconciliation task, it lists results in the CI Link Results application and in the CI Reconciliation Results application.

The cron task must point to the following class:

`psdi.app.recontask.engine.ReconCronTask`

Figure 3-50 on page 157 shows a sample of a Cron Task Setup window for a reconciliation task. To open the Cron Task Application, select **Go To** → **System Configuration** → **Platform Configuration** → **Cron Task Setup**.

The screenshot displays the 'Cron Task Setup' application window. At the top, there are tabs for 'List' and 'Cron Task'. Below the tabs, the 'Cron Task' section shows the following fields:

- Cron Task:** ReconciliationCi
- Reconciliation Cron Task:** Reconciliation Cron Task
- Class:** psdi.app.recontask.engine.ReconCronTask (highlighted with a red box)
- Access Level:** FULL

Below these fields is a table titled 'Cron Task Instances'. The table has columns: 'Cron Task Instance Name', 'Schedule', 'Run as User', 'Active', 'Keep History', and 'Max Number of History Records'. The first row shows:

Cron Task Instance Name	Schedule	Run as User	Active	Keep History	Max Number of History Records
ReconciliationCi	1h,*0,*****	MAXADMIN	☒ (highlighted with a red box)	☐	0

Below the table are buttons for 'Duplicate' and 'New Row'. Underneath the table is a section for 'Cron Task Parameters'. It has tabs for 'Parameters' and 'History'. The 'Parameters' tab is active, showing a table with columns: 'Parameter', 'Value', and 'Description'. The first row shows:

Parameter	Value	Description
RECONTASKNAME	RBtask (highlighted with a red box)	The name of the reconciliation task that will be

Figure 3-50 Setting up a cron task with reconciliation task name

Note that the Class field contains the class file for the audit process and the Active field is checked. The Value field for the parameter RECONTASKNAME in the Cron Task Parameters table window contains the reconciliation task name entered in the Reconciliation Task field in the Reconciliation Tasks application.

Scheduling cron tasks

Multiple cron tasks: The system allows the creation of cron tasks for multiple reconciliation tasks. If various cron tasks are set up to process overlapping sets of data, the results are unpredictable. Be sure not to set up multiple cron tasks with overlapping schedules.

Make sure that Integration Composer imports data before reconciliation cron tasks are processed. Also, it is best that reconciliation cron tasks that import from Integration Composer and reconciliation cron tasks do not occur simultaneously.

Viewing the results of the audit

After the system executes a reconciliation task, it is possible to view the results of the audit in either the CI Link Results or CI Reconciliation Results application.

CI link results

In the CI Link Results application, you can view the instances where the actual CI number on an authorized CI matched an actual CI number on an actual CI. When the system processes a reconciliation task, link rules sometimes successfully identify a link between objects, and sometimes they do not. The CI Link Results application lists the successful links.

Figure 3-51 shows an example of CI Link Results.

The screenshot displays the 'CI Link Result' application window. At the top, there are tabs for 'List' and 'CI Link Result', with 'CI Link Result' being the active tab. Below the tabs, the 'Rule Name' is set to 'RBLinkRule' and the 'Link Date' is '6/10/10 15:58:35'. The main area is divided into two panels: 'Authorized CI Information' on the left and 'Actual Information' on the right. In the 'Authorized CI Information' panel, the 'CI' field contains 'IBUWIN02.TIVLAB.AUSTIN-5900' and the 'Configuration Item Name' field contains 'IBUWIN02.TIVLAB.AUSTIN'. In the 'Actual Information' panel, the 'Actual Configuration Item Number' field contains 'IBUWIN02.TIV' and the 'Actual Configuration Item Name' field contains 'IBUWIN02.TIV'. The 'Description' field is currently empty.

Figure 3-51 CI Link Results

CI reconciliation results

In the CI Reconciliation Results application, you can view two types of audit results:

- ▶ **Link Rule Failures:** A link failure occurs when the system processes a link rule and does not find a successful one-to-one link between the object in Data Set 1 and the object in Data Set 2. Link failures occur when the audit finds no links or finds multiple links.
- ▶ **Comparison Rule Results:** The system produces comparison rule results when it processes a comparison rule. The specific kind of comparison rule data depends on a parameter set in the Reconciliation Tasks application, which lets you select one of the following options for comparison results when you set up a reconciliation task:
 - All results, both successful and failed matches.
 - Instances where the object from Data Set 1 failed to reconcile against the object from Data Set 2.
 - Instances where the object from Data Set 1 successfully matched the object from Data Set 2.

To view only the link rule failure or comparison rule results, use the advanced search features to display only link results or comparison results.

In the CI Reconciliation Results application, you can also perform these actions:

- ▶ Update the authorized CI with the information from actual CI using the button **Update CI with Actual CI Value**.
- ▶ Mark a record as **Resolved** if the discrepancy have been evaluated and resolved.
- ▶ Use the Comments field to enter information about how the issue was resolved or an explanation about why the discrepancy exists.
- ▶ If a review of the auditing result indicates that an action or a response is needed, you can create a Service Requests, Process Request, Incidents, Problems, work orders, Changes or Releases requesting it. It can be created through **Select Action → Create** menu.

Figure 3-52 shows one CI Reconciliation Result.

List CI Reconciliation Result

Reconciliation Task: MemComplianceAustin >> Top Level CI Number: IBUWIN02.TIVLAB.AUSTIN.IBM.COM-5900

Result Type: COMPARISON Top Level CI Name: IBUWIN02.TIVLAB.AUSTIN.IBM.COM

Rule Name: RBComparisonRule >> Modified Date: 6/10/10 16:01:21

Created Date: 6/10/10 15:55:43

Message: No CI attributes found for comparison

Resolved? ☐

Comments:

Reconciliation Result Details

Authorized CI Information	Actual CI Information
CI Object: CISPEC	Actual CI Object: ACTCISPEC
CI Attribute: COMPUTERSYSTEM_MEM	Actual CI Attribute: COMPUTERS
CI Value:	Actual CI Value: 1072693248
CI Unit of Measure:	Actual CI Unit of Measure:
Top Level Site:	Actual Configuration Item Number: IBUWIN02.TN
Configuration Item Number: IBUWIN02.TIVLAB.AUSTIN-5900 >>	Actual Configuration Item Name: IBUWIN02.TN
Configuration Item Name: IBUWIN02.TIVLAB.AUSTIN	

Update CI with Actual CI Value

Figure 3-52 Reviewing the CI Reconciliation result

The comparison of CI to actual CI relationships and attributes always takes the data from the authorized CI and compares this against the actual CI data. Because only a portion of the actual CI data is promoted to authorized CIs, only discrepancies with the authorized set are processed and written as results to the CI Reconciliation Results application. No results are generated for actual CI relationships or attributes that exist outside of the linked authorized CI relationships and attributes.

The results that the CI Reconciliation Results application displays for comparison rule evaluations depend on the parameter you select in the Comparison Results field when you set up a reconciliation task in the Reconciliation Tasks application, where you can choose either all results, successful comparison results, or failed comparison results. The following table describes the result messages CCMDDB displays for comparison rule evaluations in the Message field.

Table 3-7 Comparison Rule Result messages

Message Key	Message	Comment
C1 ^a	No CIs were retrieved by the comparison filter.	The comparison rule has a Data Set 1 filter, and the system did not retrieve any objects from Data Set 1 when applying this filter.
C2 ^a	No actual CIs were retrieved by the comparison filter.	The comparison rule has a Data Set 2 filter, and the system did not retrieve any objects from Data Set 2 when applying this filter.
C3	Matches found comparison has succeeded.	The number of Data Set 1 objects and the number of Data Set 2 objects (after the system applied any applicable filter or filters) matched the ratio specified in the rule. If you specify CI for Data Set 1 Object: ► The system counts the top-level object and all children of that object. or ► If you specify a Data Set 1 filter, the system counts the objects selected by the filter. If you specify a CI specification, for Date Set 1 Object: ► The system counts objects with the class identifier specified. or ► If you specify a Data Set 1 filter, the system counts all objects in Data Set 1 with the specified class identifier selected by the filter. The system always lists the Data Set 2 object.
C4 ^b	Attribute equality comparison has succeeded.	The attributes for the Data Set 1 object and the Data Set 2 object (and, if applicable, their measurement units) match exactly.
C5	Matches found comparison has failed.	The number of Data Set 1 objects and the number of Data Set 2 objects (after the system applied any applicable filters) did not match the ratio specified in the rule.
C6 ^b	Attribute equality comparison has failed.	The attributes for the Data Set 1 and the Data Set 2 objects (or, if applicable, their measurement units) did not match exactly.

Message Key	Message	Comment
C7	No CI object for attribute equality comparison.	This result message occurs only when a specification is selected for the Data Set 1 Object field. The message indicates that the system did not find any object in Data Set 1 with the class specified.
C8	No actual CIs object for attribute equality comparison.	The system did not find any of the Data Set 2 objects specified by the comparison rule.
C9	No CI and actual CI object for attribute equality comparison.	The system did not find any Data Set 1 object or Data Set 2 object specified in the comparison rule.

- ^a If a comparison rule has both a Data Set 1 filter and a Data Set 2 filter and neither filter returns any records, the system might display both C1 and C2 messages for the same objects.
- ^b The number of attribute equality comparisons processed by an attributes equality comparison rule is equal to the number of the objects in the leading set (The leading set is determined by the type of ask filter in the reconciliation task). For every comparison, the result is either a success or failure. The system writes these results to the CI Reconciliation Results application based on the parameter that you set for Comparison Results in the Reconciliation Tasks application. In that application, you specify whether you want all results, only successful results, or only failed comparison results.

Messages C3 and C4 indicate that the system successfully matched the Data Set 1 object and Data Set 2 object when the comparison rule was processed. All other messages indicate that the system did not successfully match a Data Set 1 object to a Data Set 2 object.

3.2.10 Interaction with other processes

Because Configuration Management Process is responsible for maintaining information related to CIs, it interacts with all other ITSM processes, as in the following examples:

- ▶ Incident Management uses CI information to understand what CIs are involved in an incident.
- ▶ Problem Management uses CI information to help track down the root cause of a problem.
- ▶ Change Management uses CI information to understand the ramifications of a proposed Change to analyze its impacts.

- ▶ Release Management updates the CMDB with information about deployed Releases.
- ▶ Service Level Management maintains service level agreements in the CMDB.
- ▶ Availability Management uses CI information to identify pockets of unavailability.
- ▶ Capacity Management uses CI information in capacity analyses.
- ▶ IT Service Continuity Management uses CI information to determine what resources need to be restored in the event of a major outage.

In the CCMDB context, the Configuration PMP works closely with the Change PMP.

3.2.11 CI baselines

A CI baseline contains the attributes and relationships for a CI or set of CIs at a point in time. It is typically created at a time when there is a known good configuration. It can later be compared against the current values of the corresponding actual CIs to see how the configuration has changed since the time the baseline was activated.

CI Baselines application can be used by a Change Owner, Change Manager, Problem Manager or Configuration Auditor in the following ways:

- ▶ Saving the configuration of a set of CIs before a Change Request is implemented to allow comparing that configuration to the actual configuration after the Change. This allows checking to see if there were unauthorized Changes, and to ensuring that the expected Changes were made.
- ▶ Verifying that the action of rolling back to a saved baseline configuration is successful. You might do this either because of a failed Change or because of another incident or problem.
- ▶ For problem determination, comparing the baseline configuration with the actual configuration to see what Changes might be causing the problem.
- ▶ For audit purposes, when you want to compare the actual configuration to a saved baseline configuration instead of the current authorized configuration.

Certain actions such back out a Change, rebuild a baseline, or rollback to a baseline need to be performed manually.

Creating a CI baseline

The CI Baseline application is used to create a new CI baseline. To create a baseline, select **Go To** → **IT Infrastructure** → **CI Baselines** → **New CI Baseline**.

The basic fields on this application are as follows:

- ▶ **Name:** A unique name for the CI baseline.
- ▶ **Version:** The CI baseline version. Many versions of the same CI baseline can be created by clicking the **Create Version** icon or selecting **Action** → **Create Version**. The version is used when an activated baseline needs to be modified and it keeps a history of the baselines.
- ▶ **Active:** An active baseline is used for comparisons and cannot be changed. After the baseline is activated, you must create a new version to be able to change its configuration.
- ▶ **Active Date:** Date/time when the baseline was activated.
- ▶ **Member CIs:** The authorized CIs that are members of the baseline. When the baseline is activated, the member CI attributes and relationships at the baseline activation time are used for the comparison action.

Figure 3-53 shows an Inactive baseline with two members added.

The screenshot displays the 'CI Baseline' application interface. At the top, there are tabs for 'List' and 'Baseline'. The 'Baseline' tab is active, showing a form for editing a baseline. The form includes fields for 'Name' (RBBaseline1), 'Version' (2), 'Active?' (unchecked), 'Active Date' (empty), 'Create Date' (6/10/10 23:14:10), 'Created By' (MAXADMIN), 'Change Date' (6/10/10 23:15:16), and 'Changed By' (MAXADMIN). Below the form, there is a message: 'A baseline includes member configuration items. You can add or delete member configuration items.' Below this message is a table titled 'Member CIs' with columns for 'Member CI', 'Configuration Item Name', 'Description', and 'Classification'. The table contains two rows of data. Below the table, there are buttons for 'Select CIs' and 'New Row'.

Member CI	Configuration Item Name	Description	Classification
ANISE.TIVLAB.AUSTIN.IBM.COM~11725	ANISE.TIVLAB.AUSTIN.IBM.COM		CI-512-2.WINDOWSOS
DROID.TIVLAB.AUSTIN.IBM.COM~12887	DROID.TIVLAB.AUSTIN.IBM.COM		CI-512-2.WINDOWSCOMPUTERSYSTEM

Figure 3-53 CI Baseline application

There are many options to list and add members to a baseline:

- ▶ **New Row:** Use this option to add a known CI or search for it based on the Classification or Attribute.
- ▶ **Select CIs from Available CIs:** Lists the CIs using filters, such as location, classification and status.
- ▶ **Select CIs from Collection:** Lists all the CIs that are members from the chosen Collection.
- ▶ **Select CIs from Baseline members:** Lists the CIs that are already members from another baseline. It is possible to list all of them or filter by baseline name and version.
- ▶ **Select CIs from Related to Member CI:** Lists the CIs that have any direct relation with the member CI selected.
- ▶ **Select CIs from Children of Member CI:** Lists the children of the Member CI selected. Children CIs are CIs that are related to the selected member CI because they were promoted from actual CIs based on a promotion scope. The selected Member CI must be a top level authorized CI whose classification is the top-level authorized classification in a promotion scope.

Creating a new version of the CI baseline: In addition to creating a CI baseline, you can also create a new version of a CI baseline. When you create a new version of a baseline, it copies all of the existing member CIs to the new version and the version number is incremented. Until the new version is activated, it is editable. That means you can add and remove member CIs. You might want to create a new version of a baseline if a configuration has changed and you want to preserve the new configuration.

Activating a CI baseline

After all desired member CIs are added in the baseline, it needs to be activated. Figure 3-54 shows the **Activate the Baseline** icon.

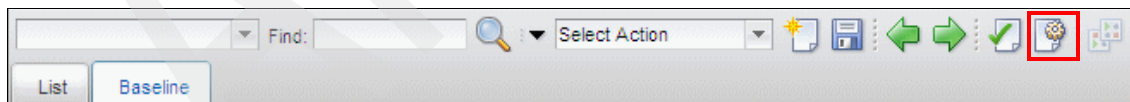


Figure 3-54 Activating a CI baseline

Tip: It is not necessary to validate the baseline first. This is done automatically for you when you click **Activate the Baseline**.

After the activation, the **Compare with Actual CI** button is displayed. Clicking it starts running the comparison, which will compare each member CI to the actual CI it is currently linked to. It compares the attributes and relationships that existed for the member CI at the time the baseline was activated to the current values for the actual CI. When the comparison process is done, it will pop up the results, which are organized in four tabs:

- ▶ **Member CIs With Differences:** This tab shows a list of the CIs with a difference between the baseline and the actual CI. It helps to identify any actual CIs that have been changed, added, or removed since the baseline was activated.
- ▶ **Member CIs Not Compared:** This tab shows the member CIs in the baseline that have no corresponding actual CI.

Comparisons: For the following reasons, a member might not be compared:

- ▶ It has no corresponding actual CI specified.
- ▶ The corresponding actual CI does not exist.
- ▶ The user is not authorized to see data for the customer associated with the corresponding actual CI.

- ▶ **Member CIs With No Differences:** This tab shows the member CIs that have not been changed since the baseline was created.
- ▶ **All Relationship Differences:** This tab shows only the CIs that have differences between their relationships.

Figure 3-55 shows the CI Baseline Results window, with attributes differences between the member CI and actual CI.

Member CI Name	Member CI Classification	Summary
DCMJ21	GENCSYS	NOACTCILINKED
WINXP123	OS	NOACTCILINKED

Figure 3-55 CI Baseline Comparison with attribute differences

3.3 Change Management Process Manager

This section discusses the Change Management Process Manager.

3.3.1 Change Management overview

Change Management is the process responsible for protecting the environment from unauthorized Changes, and reducing the risks associated with planned Changes. Through it, standardized methods and procedures are defined for efficient and prompt handling of all Changes in order to minimize or avoid the impact of Change-related incidents on service quality and consequently on the business.

Information Technology has become a critical success factor for business, meaning that interruptions in IT service and quality issues can cause significant impact to the business, including financial damages. Therefore, you need a rigorous control of Changes.

A wide variety of reasons can drive a Request for Change (RFC). The following items include the most common occurrences in the IT environment that require an RFC:

- ▶ Business requirements have changed or evolved.
- ▶ An incident or problem resolution modifies one or more configuration items.
- ▶ A new configuration item needs to be introduced to the IT infrastructure.
- ▶ An existing configuration item needs to be removed.
- ▶ An existing configuration item needs to be upgraded.
- ▶ New or changed legislation requires a corresponding Change in the IT infrastructure (a regulatory Change).
- ▶ A business unit has changed locations, requiring the wholesale relocation of office equipment and computing resources.
- ▶ Vendors or contractors have changed their products or services.

Objective: ITIL defines the objective of Change Management as to ensure that Change Requests are recorded and then evaluated, authorized, prioritized, planned, tested, implemented, documented, and reviewed in a controlled manner.

Change lifecycle flow

The Change Management process begins when a Change Requester submits a Request for Change (RFC) with minimum information. The person responsible for receiving the request analyzes it and accepts or rejects it. If the request is accepted, a Change Owner is assigned and then a Change Record is opened. The Change Owner is responsible for providing all information about the requested change; the level of detail will depend on the type of Change and the process defined.

CCMDB organizes a Change in two phases: First it is handled as a process request, and after being accepted, it becomes a Change Record.

Here we list typical information that a Change Record needs to contain:

- ▶ Description
- ▶ Configuration items impacted (can be changed during the assessment step)
- ▶ Reason for the Change
- ▶ Effects if the Change is not implemented
- ▶ Proposed date, time, and time frame
- ▶ Proposed category (for example, minor, significant, or major)
- ▶ Proposed priority
- ▶ Risk assessment and risk management plan
- ▶ Backout plan
- ▶ Backup information
- ▶ Activity plan
- ▶ Estimated resources
- ▶ Estimated costs and quality of service

The Change Owner receives the Change and completes the information required. Then, Change Analysts and Subject Matter Experts assess the Change Record. During the assessment, both the technical and business aspects must be analyzed:

- ▶ Change reason
- ▶ Impact on business
- ▶ Risks of Change implementation
- ▶ Resources needed
- ▶ Proposed date, time, and time frame
- ▶ Relationship with other Changes
- ▶ Backout plan
- ▶ Backup available and required

According to ITIL, Changes that are categorized as standard/pre-approved do not need to go through Change assessment and authorization for each Change. These types of Changes are approved once, and each time the Change is made, the authorization applies. A good example of this type of Change might be provisioning a laptop for a new employee.

Potential Changes: Preapproved or standard Changes are potential Changes that have already been analyzed and approved by the Change Manager or Change Advisory Board. Standard Changes are those that tend to reoccur, are well understood, and are relatively risk-free. Change Management maintains a list of pre-approved/standard Changes.

After the assessment is completed, the Change Owner schedules the activities, determines the CI attribute modifications, and sends the RFC to a Change Approver that examines the analysis results and determines whether to approve the RFC.

When the RFC is approved, the Change Manager schedules the RFC. The activity for scheduling a Change takes into account the Forward Schedule of Change, eliminating conflict between differing Changes, and assigning appropriate resources accordingly.

Approved Changes can be subsequently scheduled into target Releases, in line with the policy for determining Releases. The result is an updated Forward Schedule of Change (FSC), containing details of all approved Changes and their implementation dates, along with a Projected Service Outage (PSO), containing details of Changes to agreed Service Level Agreements and service availability.

The Change Implementer receives the RFC and implements as planned. Approved Changes are implemented primarily using Release Management, but other Changes are implemented using a task assignment by the Change Manager. This determination is made by Change Management policies and the appropriate Change model. Regardless of who implements the Change, Change Management monitors the deployment of the Change.

After the Change has been implemented, the Change Owner opens a process request to Configuration Manager to update CI attributes in CCMDB.

The Change Manager conducts a post implementation review after a predefined elapsed time. It ensures that the Change has had the desired effect and met its objectives, and that Users and Customers are satisfied with the results, or to identify any shortcomings. Finally, the RFC record is closed.

Figure 3-56 illustrates the Change Management process flow just described.

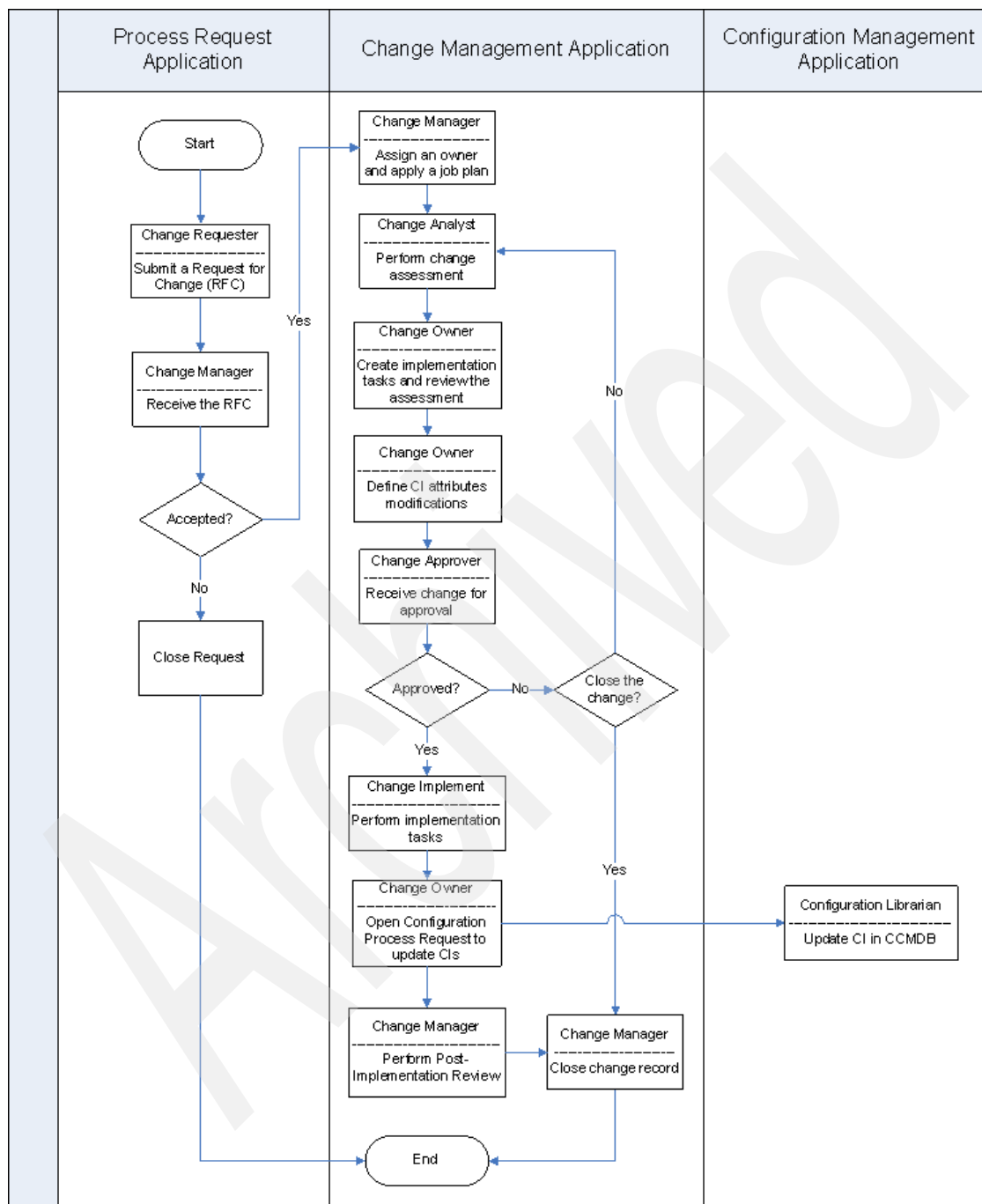


Figure 3-56 Change Management Process

Change Management roles

A role is an abstract definition of a set of responsibilities that encompass tasks to be performed and work products to be produced. All processes have roles and responsibilities, and they are typically realized by an individual, or a set of individuals, working together as a team. An individual can fulfill many types of roles. Roles are not individuals, nor are they necessarily equivalent to job titles; instead, they describe the responsibilities or accountabilities assigned to each assigned individual.

The roles in CCMDB are created as security groups and can be customized to meet process needs. Each role will have its own Start Center and application access. The roles are as follows:

- ▶ **Change Manager:**

The Change Manager is primarily responsible for the overall quality of the Change Management process. This role is the main coordinator within this process and is the focal point regarding Changes for both the customer and the IT organization. Therefore, all managers in the IT organization must support that person in their role.

- ▶ **Change Administrator:**

The Change Administrator supports the Change Manager by managing records, tracking action items, and providing process-related reports.

- ▶ **Change Owner:**

The Change Owner is responsible for an individual Change. The Change Owner follows the Change from beginning to end, bringing in analysts and specialists as needed to complete the project. The Change Owner is responsible for seeing that analysts and specialists bring the Change to a close.

- ▶ **Change Analyst:**

A Change Analyst is responsible for performing the assessment of the Change.

- ▶ **Change Approver:**

A Change Approver is responsible for assessing an RFC and assigning it to one of the authorization statuses. Change Approvers are typically representatives of groups directly involved in or impacted by the Change.

- ▶ **Change Implementer:**

The Change Implementer is responsible for implementing the Changes (including execution of backout procedures if available and needed).

► Change Requester:

The Change Requester proposes Changes to the IT infrastructure. The Requester is the person responsible for proposing and submitting an RFC. An RFC can also have an author, who creates the Change in Change Management for the Requester, but is not responsible for proposing or submitting the Change.

Start Center by role

Each role in CCMDB can have a particular Start Center according to their activities and information needs.

To customize a Start Center, select **Start Center** → **Change Content/Layout** and then select the content that you want to see in the right and left column.

Figure 3-57 illustrates the Change Content/Layout.

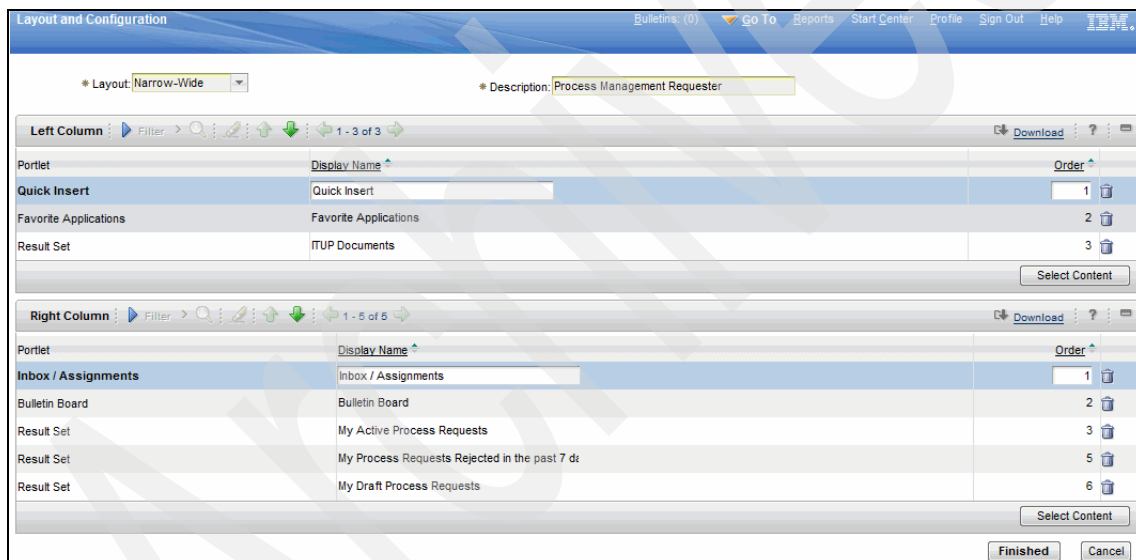


Figure 3-57 Customizing a Start Center

Change Management roles are shown in the following figures.

Figure 3-58 illustrates the Change Managers Start Center.

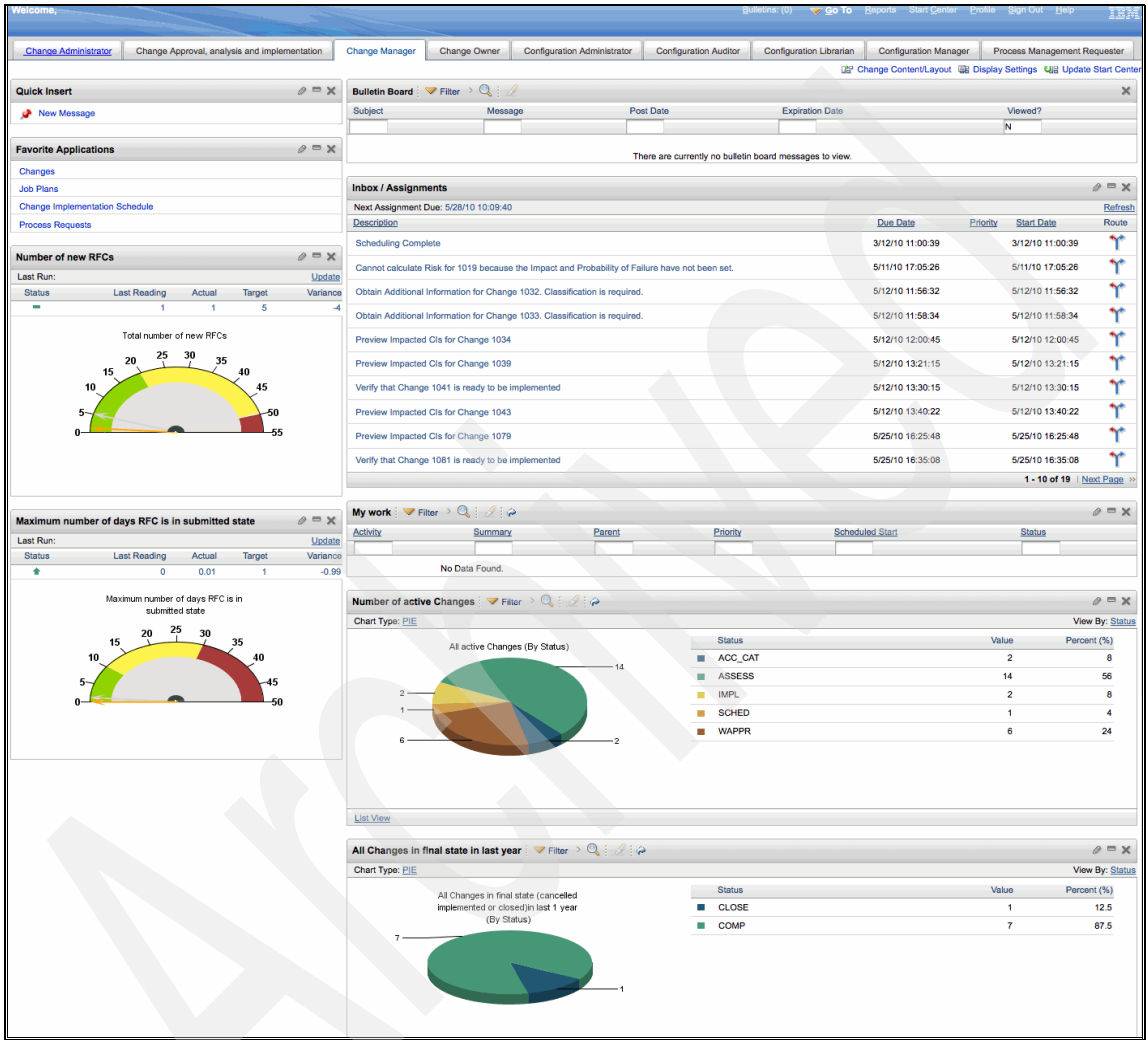


Figure 3-58 Change Manager's Start Center

Figure 3-59 illustrates the Change Administrator's Start Center.

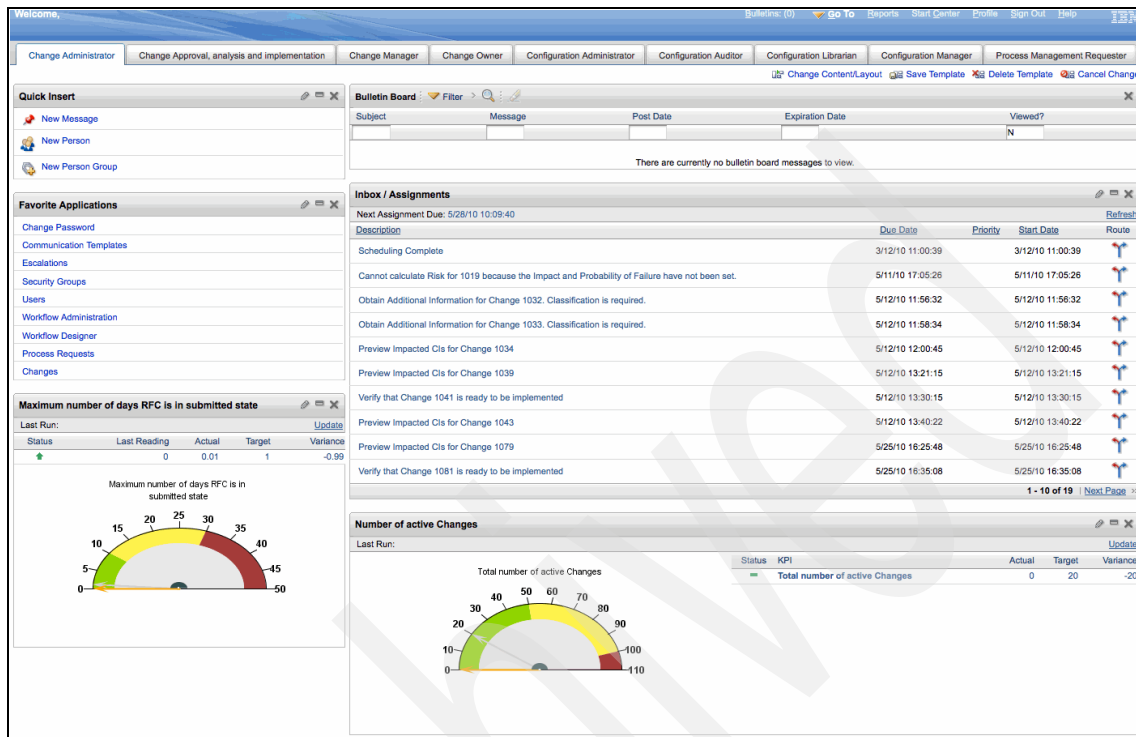


Figure 3-59 Change Administrator's Start Center

Figure 3-60 illustrates the Change Owner's Start Center.

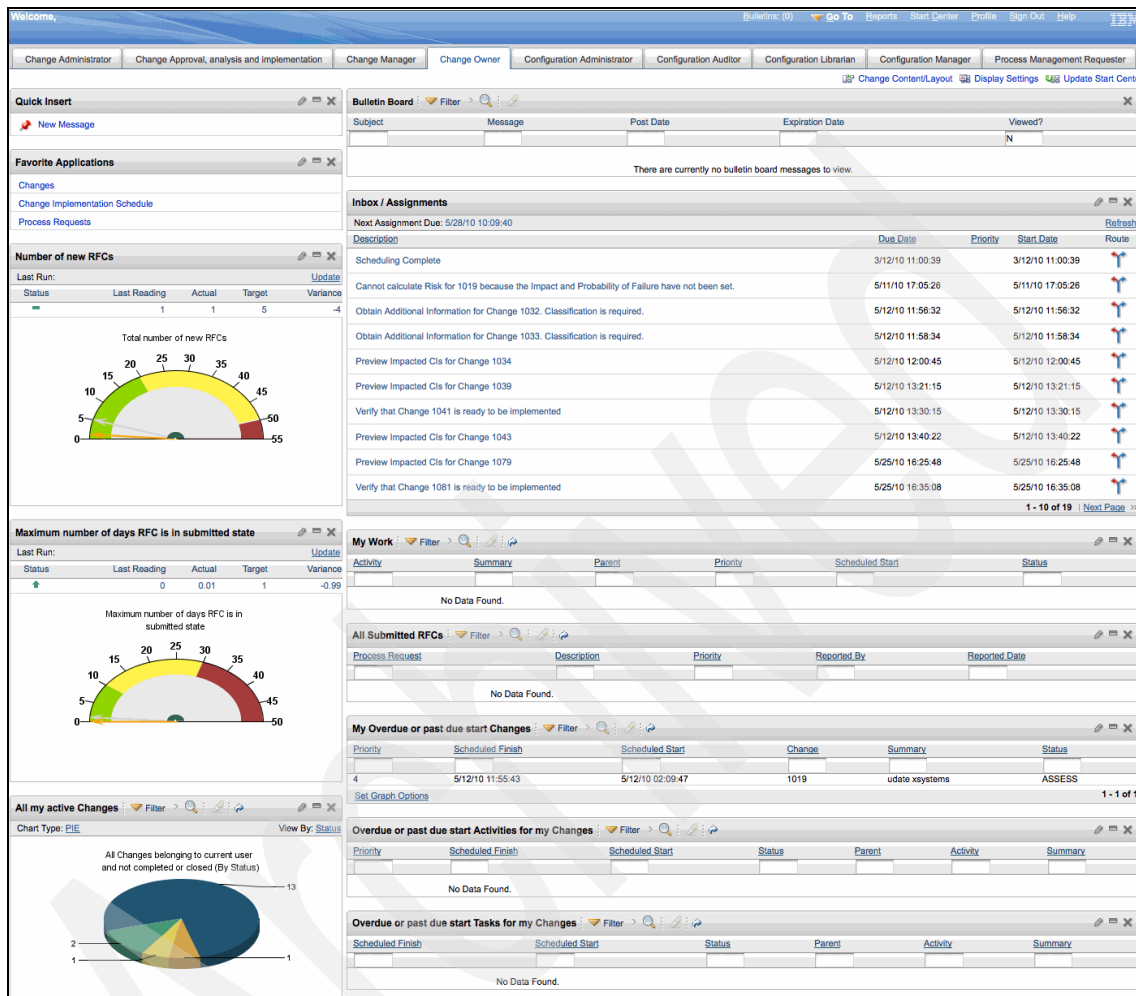


Figure 3-60 Change Owner's Start Center

Figure 3-61 illustrates the Change Analyst's Start Center.

Welcome, Bulletins (0) [Go To](#) [Reports](#) [Start Center](#) [Profile](#) [Sign Out](#) [Help](#)

[Change Administrator](#) [Change Approval, analysis and implementation](#) [Change Manager](#) [Change Owner](#) [Configuration Administrator](#) [Configuration Auditor](#) [Configuration Librarian](#) [Configuration Manager](#) [Process Management Requester](#)

[Change Content/Layout](#) [Display Settings](#) [Update Start Center](#)

Favorite Applications

[Activities and Tasks](#)

[Changes](#)

Bulletin Board [Filter](#) [Search](#)

Subject	Message	Post Date	Expiration Date	Viewed?
				N

There are currently no bulletin board messages to view.

Inbox / Assignments [Refresh](#)

Next Assignment Due: 5/28/10 10:09:40

Description	Due Date	Priority	Start Date	Route
Scheduling Complete	3/12/10 11:00:39		3/12/10 11:00:39	Route
Cannot calculate Risk for 1019 because the Impact and Probability of Failure have not been set.	5/11/10 17:05:26		5/11/10 17:05:26	Route
Obtain Additional Information for Change 1032. Classification is required.	5/12/10 11:56:32		5/12/10 11:56:32	Route
Obtain Additional Information for Change 1033. Classification is required.	5/12/10 11:58:34		5/12/10 11:58:34	Route
Preview Impacted CIs for Change 1034	5/12/10 12:00:45		5/12/10 12:00:45	Route
Preview Impacted CIs for Change 1039	5/12/10 13:21:15		5/12/10 13:21:15	Route
Verify that Change 1041 is ready to be implemented	5/12/10 13:30:15		5/12/10 13:30:15	Route
Preview Impacted CIs for Change 1043	5/12/10 13:40:22		5/12/10 13:40:22	Route
Preview Impacted CIs for Change 1079	5/25/10 16:25:48		5/25/10 16:25:48	Route
Verify that Change 1081 is ready to be implemented	5/25/10 16:35:08		5/25/10 16:35:08	Route

1 - 10 of 19 | [Next Page](#)

All Active Changes [Filter](#) [Search](#) [Refresh](#)

Owner	Priority	Change	Summary
		1001	Automation 1001
		1019	update xsystems
MAXADMIN	4	MYCHG2	Update Payroll DB server
		1023	Virtualization demo scenario change
		DEMOCHG	Scheduler Demo
		1032	usecase 2 -- move prodvm to a new vm host
MAXADMIN	1	1033	usecase 3 -- update local av on laptop
MAXADMIN	4	1034	usecase 4 -- urgent update of av
MAXADMIN	4	1039	usecase 2 -- move prodvm to a different vm
		1041	usecase 3 -- update antivirus on local laptop

[Set Group Options](#)

1 - 10 of 25 | [Next Page](#)

My Work [Filter](#) [Search](#) [Refresh](#)

Activity	Summary	Parent	Scheduled Start	Priority	Status

No Data Found.

My Change Policy Documents [Filter](#) [Search](#) [Refresh](#)

Document	Description	Changed Date	Revision
CADMS1	Change Policy Information for Change Administrator role		
CADMG1	General Change Management Policy Information		
CADMG2	Change Tool Policy Information		
CALYS1	Change Policy Information for Change Analyst role		
CALYG1	General Change Management Policy Information		
CALYG2	Change Tool Policy Information		
CAPRG1	Change Policy Information for Change Approver role		
CAPRG1	General Change Management Policy Information		
CAPRG2	Change Tool Policy Information		

Figure 3-61 Change Analyst's Start Center

3.3.2 Change process step-by-step within CCMDB

This section provides a step-by-step explanation of how the Change Management process works in CCMDB.

Functions that are more complex are further explained in detail later in this chapter.

Submitting an RFC

Responsible role: Change Requester

A Change Requester submits a Request for Change (RFC) using the Process Request application.

Select **Go To** → **Change** → **Process Request**.

Within the Process Request Application, click the button **New Process Request**.

An automatic ID will be assigned for the new request. The following information can be provided:

- ▶ Description: An RFC title
- ▶ Priority: A suggestion priority
- ▶ Details: A detailed explanation
- ▶ Process Manager Type: The type of the request, in this case, Change
- ▶ Site: The site to which the RFC will be applied
- ▶ Requested Completion: The target date
- ▶ Classification: The request classification
- ▶ Class Description: The description of classification, which will be fulfilled automatically according to the classification chosen

Optionally, it is possible to define attributes classifications and target CIs. Figure 3-62 illustrates a New Process Request form.

Process Requests | Bulletin: (0) | Go To | Reports | Start Center | Profile | Sign Out | Help

Find: [] | Select Action: []

Process Request: 1027 | Owner: [] | Owner Group: [] | Process State: DRAFT | Status: NEW | Attach: []

User Information

Requestor: MAXADMIN
 Name: MAXADMIN
 Phone: []
 E-mail: []
 Reported Date: 8/1/10 6:46 PM

Process Request Details

Description: J2EE Application Upgrade
 Details: Upgrade B2B Web Application
 Process Manager Type: Change
 Site: AUSTIN
 Requested Completion: 8/31/10 6:49 PM
 Asset: []
 Location: []
 Configuration Item: []
 Configuration Item Name: []
 Response Plan: []
 Impact: 3
 Urgency: 3
 Priority: 3

Request Classification Details

Classification: PMCHG \ PMCHGSFW \ BUSAPPL \ J2EEBSVA
 Class Description: J2EE Business Server Applications

Classification Attributes | Filter | 1 - 2 of 2 | Download

Attribute	Description	Value
J2EEVERS	J2EE Version	
PKGNAME	Name	

New Row

Target CIs | Filter | Download

Asset	Location	Configuration Item Number	Configuration Item Name	Target Description
...No rows to display...				

Select | Clear All | New Row

Figure 3-62 New Process Request

Accepting or rejecting the RFC

Responsible role: Change Manager.

The Change Manager receives the Request for Change in the queue displayed in their Start Center. The Change Owner opens the request and reviews it, sees that it meets the basic requirements, accepts the request, and assigns an owner (the Change Owner can also be assigned in Change Application). Accepting the request does not mean that the requested Change will be completed, merely that it will be further assessed. After an RFC is accepted, it becomes a Change Record.

Important: Remember that until this step, the RFC was been handled by the Process Request Application; after it is accepted, it becomes a Change Record and will be under the Change Application.

Figure 3-63 illustrates the Change Manager's RFC queue.

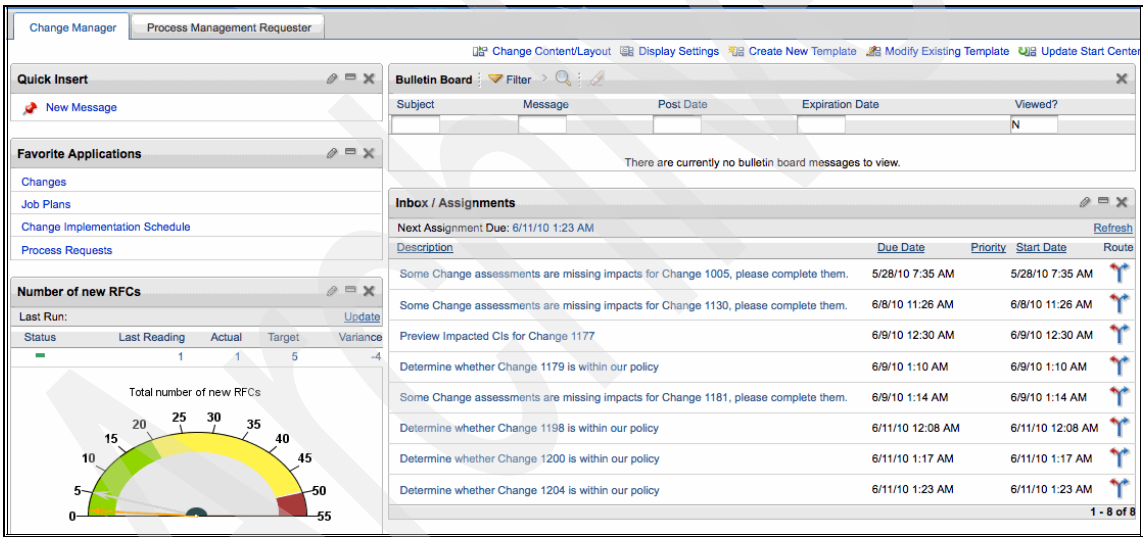


Figure 3-63 Change Managers Inbox (RFC Queue)

The Change Manager clicks the **Route** icon in order to view and approve or reject the RFC.

Figure 3-64 illustrates the Change Manager Accepting the RFC.

The screenshot displays the IBM Tivoli Change Manager web interface. At the top, there is a navigation bar with tabs: List, Change, Assessments, Impacts, Authorization, Schedule, Workplan Map, Related Records, Actuals, and Log. Below this is a 'Progress Map' section with a sequence of steps: ACC_CAT, IMPL, INPRG, COMP, REVIEW, and CLOSE. The main content area shows details for Change 1200, with fields for Owner (MAXADMIN), Owner Group, Status (ACC_CAT), and Attachments. A 'Change Details' section is partially visible on the left. A modal dialog box titled 'Complete Workflow Assignment' is open in the center. It contains a task description: 'Determine whether Change 1200 is within our po'. Under the 'Action' section, there are two radio buttons: 'I have reviewed this Change and accept that it is within our Change policy. I have set the Change Type and Change Owner if they were not automatically set.' (which is selected) and 'This Change is not within our policy so will be rejected.' Below the actions is a 'Memo' field. At the bottom of the dialog is a table for 'Earlier Memos' with columns for Memo, Person, and Transaction Date. The table currently shows '...No rows to display...'. At the bottom right of the dialog are buttons for OK, Reassign, and Cancel.

Figure 3-64 Change Manager accepting an RFC

Assigning a Change Owner

Responsible role: Change Manager.

The Change Record will be displayed to the Change Manager in their Start Center. If the Change Owner was not assigned in the Process Request Application, the Change Manager has to assign it.

This action can be done in the Change Application in two ways:

- ▶ The person who will be the Change Owner has to click the **Take Ownership** button. The Change will be assigned to that person's name. Figure 3-65 illustrates **Take Ownership**.

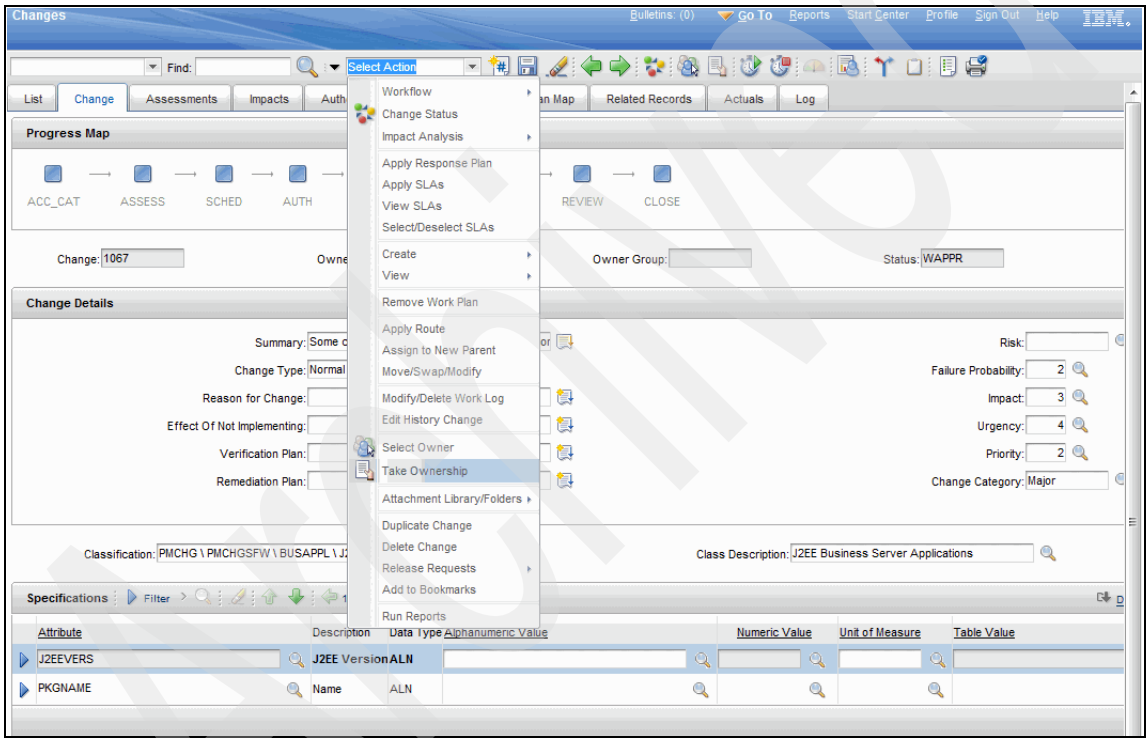


Figure 3-65 Take Ownership

- The Change Manager assigns the Change Owner.
The Change Manager clicks the **Select Owner** button and then selects the Change Owner. The Change will be assigned to the selected name.
Figure 3-66 illustrates the **Select Owner**.

The 'Select Owner' dialog box contains the following table of data:

Person Group	Person	Name	Shift	Open Work
PMCFGADM	PMCFGADM	PMCFGADM		0
PMCFGAUD	PMCFGADM	PMCFGADM		0
PMCFGLIB	PMCFGADM	PMCFGADM		0
PMCFGMGR	PMCFGADM	PMCFGADM		0
PMCHGANA	MAXADMIN			5
PMCHGBUS	MAXADMIN			5
PMCHGCAB	MAXADMIN			5
PMCHGIMP	MAXADMIN			5
PMCHGITM	MAXADMIN			5
PMCHGMA	MAXADMIN			5
REQUESTR	MAXADMIN			5
	DEFLT	DEFLT		0
	DEFLTREG	DEFLTREG		0
	MAXREG			0
	MXINTADM			0

Figure 3-66 Select Owner

Selecting an appropriate job plan

Responsible role: Change Owner.

The Change Owner opens the Change in the Changes application, selects an appropriate job plan from the list of available job plans, and assigns it to this Change. This populates the Change with a set of activities and tasks and now becomes a work order.

The procedure for this is as follows:

1. Select the Change Owner, go to Schedule tab, and select a job plan.

Figure 3-67 illustrates selecting a job plan.

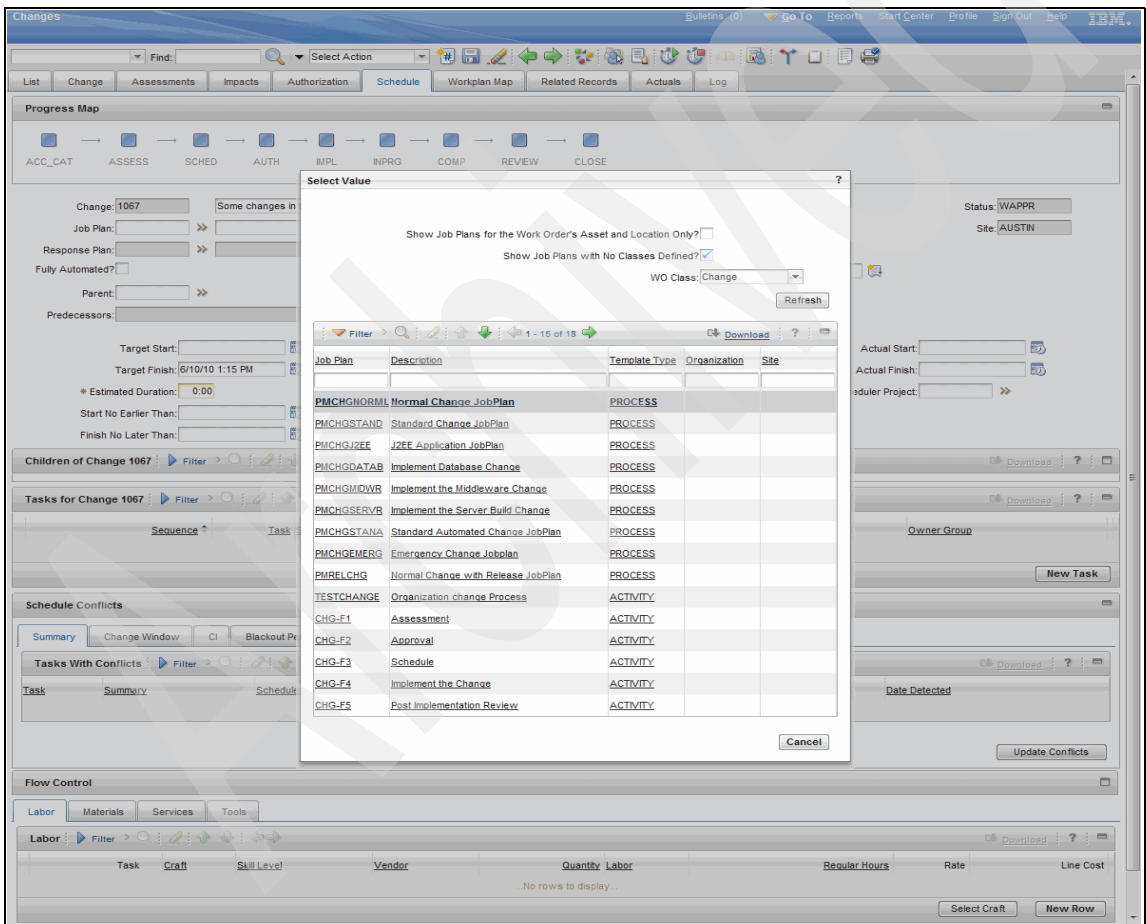


Figure 3-67 Job plan selection

Tip: We have explained how to manually apply a job plan to a Change Object. However, you can automatically apply it based on the classification of the Change Request (RFC). To apply the object automatically, select a job plan and define the field Default WO Class.

When a job plan is applied to a Change, its activities become children of the Change. The Change Owner can customize the set of activities and tasks to be used to complete the Change.

2. Click the **Actuals** menu tab:
 - a. To add a new Activity, click the **New Activity** button.
 - b. To add a new Task, click the **New Row** button.

Figure 3-68 illustrates the job plan applied.

The screenshot shows the IBM Tivoli CCM interface with the 'Actuals' tab selected. The 'Progress Map' section shows a sequence of steps: ACC_CAT, IMPL, INPRG, COMP, REVIEW, and CLOSE. The 'Change' field is set to 1067, and the 'Summary' is 'Some changes in the J2EE business application'. The 'Children of Change 1067' table lists the following activities:

Sequence	Work Order	Summary	Class	Asset	Location	Configuration Item	Configuration Item Name	Status
1098		Approval	ACTIVITY					WAPPR
1099		Schedule	ACTIVITY					WAPPR
1100		Implement the Change	ACTIVITY					WAPPR
1101		Post Implementation Review	ACTIVITY					WAPPR
1102			ACTIVITY					WAPPR

The 'Tasks for Change 1067' table is empty, showing 'No rows to display...'. The 'Labor' table is also empty, showing 'No rows to display...'. Buttons for 'Select Work Orders', 'New Activity', 'New Row', 'Select Labor', and 'Select Planned Labor' are visible at the bottom of the interface.

Figure 3-68 Job plan applied

Initiating the Activities

Responsible role: Change Owner.

The Change Owner changes the status of the Change to INPROGRESS to initiate the first activity in the job plan.

Click the **Change Status** button and select **In Progress**, as shown in Figure 3-69.

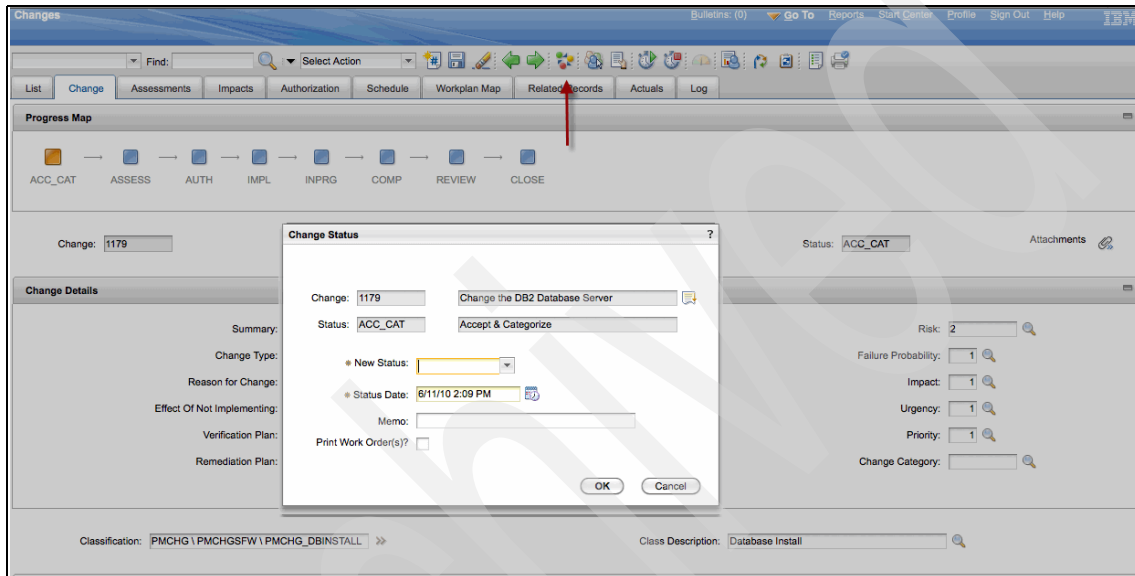


Figure 3-69 Change process status

Performing technical Change assessment

Responsible role: Change Analysts.

The Change will be displayed in the Start Center of each Change Analyst required to analyze the Change.

Change Analysts with appropriate technical expertise have to assess the technical impacts of the Change and use the Impact Assessment tab of the Changes application to record their assessments, which might include costs as well as notes about implementation tasks that will be required to carry out the Change. The technical assessment can be performed by multiple SMEs; in this case the tasks have to be routed to the appropriate experts.

To start an assessment within the selected Change, go to the **Assessments** tab.

Each Change Analyst has to click the **New Row** button to add an assessment analysis as demonstrated in Figure 3-70.

The screenshot displays the 'Changes' application interface. At the top, there's a navigation bar with tabs: List, Change, Assessments (selected), Impacts, Authorization, Schedule, Workplan Map, Related Records, Actuals, and Log. Below the tabs is a 'Progress Map' showing a sequence of steps: ACC_CAT, ASSESS, AUTH, SCHED, IMPL, INPRG, COMP, REVIEW, and CLOSE. The main section is divided into three parts: 'Change' details (Change: 1023, Summary: virtualization demo scenario change, Risk, Impact, Failure Probability), 'Details' (a large empty box), and 'Owner' information (Owner, Owner Group, Status: WAPPR). Below this is the 'Analysis Summary' section with fields for Maximum Assessed Impact, Assessment Summary, Estimated Total Work Effort (0.00), and Estimated Total Cost (0.00). The bottom section is 'Technical Assessments', which includes a table with columns: Assessment Type, Results, Implementation Notes, Cost, Effort, Impact, Owner, and Owner Group. The table lists five assessment types: Application, Capacity, Server, Storage, and Security. A 'New Row' button is located at the bottom right of the table.

Assessment Type	Results	Implementation Notes	Cost	Effort	Impact	Owner	Owner Group
Application							
Capacity							
Server							
Storage							
Security							

Figure 3-70 Technical Assessment

Each technical assessment row has an implementation note associated with it, which is to be used to record notes that will be transformed during implementation tasks.

To add an implementation note, select the assessment row that the note will be associated with and then click the **Long Description** button next to the Implementation Notes Field as illustrated in Figure 3-71.

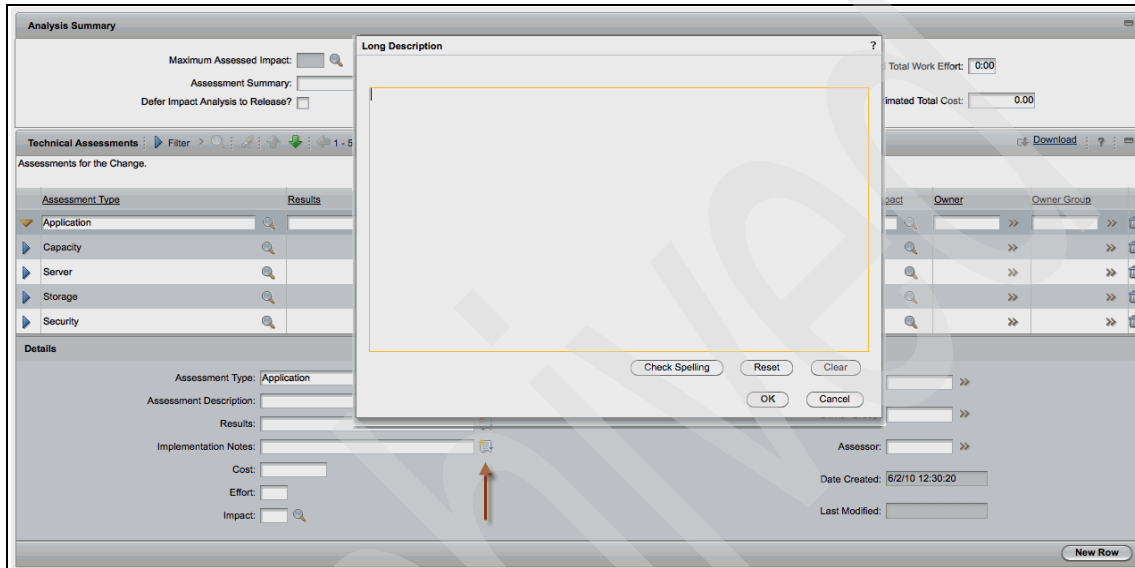


Figure 3-71 Technical assessment implementation notes

Routing tasks

Responsible role: Change Owner or Automatic Task in job plan.

The Change is routed to the next task by changing the status of the current task to Completed in the **Actuals** tab, or an automatic task can be defined in the job plan.

To change a task's status, click the **Change Status** icon on the far right of the task row, as illustrated in Figure 3-72.

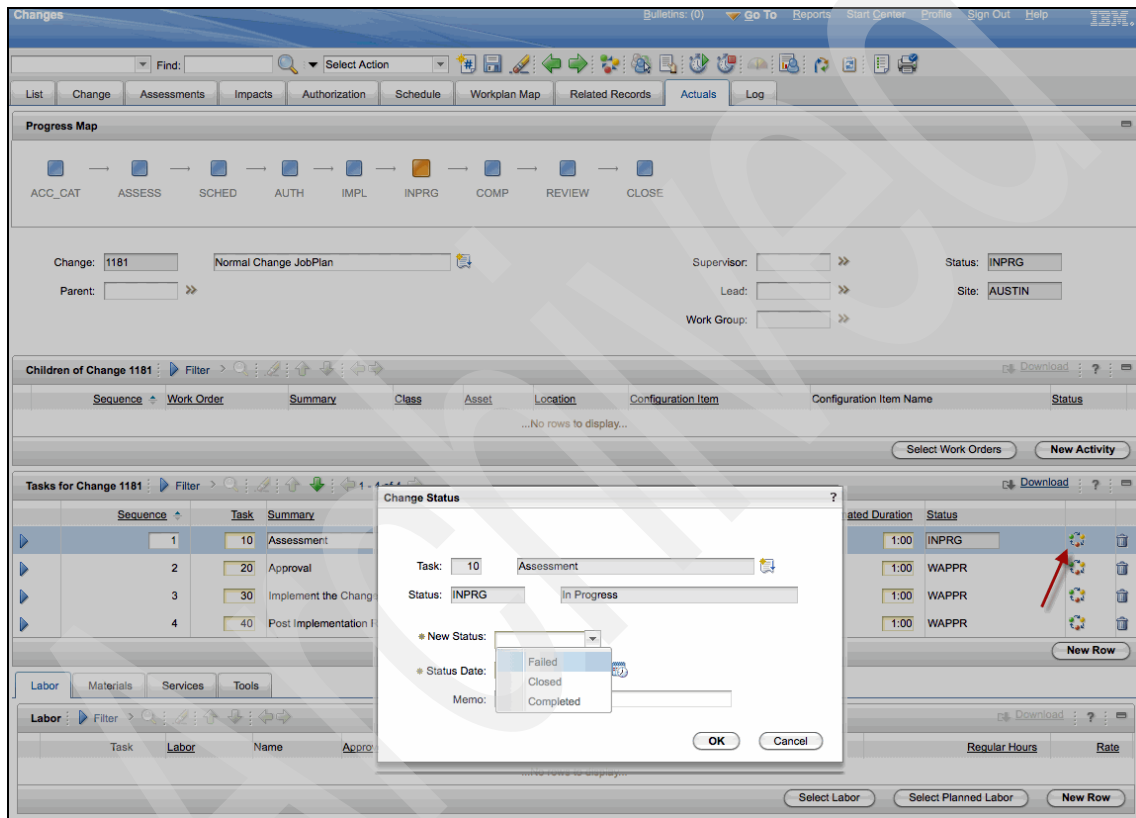


Figure 3-72 Change Task Status

Performing a business Change assessment

Responsible role: Business Experts.

Business Experts have to assess the business impacts of the Change and use the Impact Assessment tab of the Changes application to record their assessments.

To start an assessment, within the selected Change, go to the **Assessments** tab. Each Change Analyst has to click the **New Row** button to add a Business Assessment analysis as demonstrated in Figure 3-73.

The screenshot displays the 'Changes' application interface. At the top, there's a navigation bar with tabs: List, Change, Assessments (selected), Impacts, Authorization, Schedule, Workplan Map, Related Records, Actuals, and Log. Below the navigation bar is a 'Progress Map' section showing a sequence of steps: ACC_CAT, ASSESS, SCHED, AUTH, IMPL, INPRG, COMP, REVIEW, and CLOSE. The 'Change' section shows 'Change: MYCHG' and 'Summary: Update Payroll system'. Below this, there are fields for 'Risk', 'Impact' (set to 3), and 'Failure Probability' (set to 2). The 'Analysis Summary' section shows 'Maximum Assessed Impact: 1' and 'Estimated Total Work Effort: 0.00'. The 'Technical Assessments' section is currently empty, showing 'No rows to display...'. The 'Business Assessments' section contains two rows: 'SOX' and 'Financial'. The 'Financial' row has an 'Impact' of 2. A red arrow points to the 'New Row' button at the bottom right of the 'Business Assessments' table.

Assessment Type	Impact	Results	Cost	Effort	Owner	Owner Group
SOX	1					
Financial	2					

Figure 3-73 Business Assessment

Note that Business Assessment does not have implementation notes because it is a business analysis and will not have implementation tasks associated.

Authorization (Approval)

Responsible role: Change Approver.

The Change record is sent to an approver. The Changes that have to be approved are shown in the Start Center of the Change Approver role as shown in Figure 3-74.

The screenshot displays the 'Change Approval, analysis and implementation' interface for a 'Process Management Requester'. The left sidebar shows 'Favorite Applications' with links to 'Activities and Tasks' and 'Changes'. The main content area includes several sections:

- Bulletin Board:** A table with columns: Subject, Message, Post Date, Expiration Date, Viewed?. It shows 'There are currently no bulletin board messages to view.'
- Inbox / Assignments:** A table with columns: Description, Due Date, Priority, Start Date, Refresh, Route. It lists two assignments:

Description	Due Date	Priority	Start Date
Some Change assessments are missing impacts for Change 1005, please complete them.	5/28/10 7:35 AM		5/28/10 7:35 AM
Provide Business Assessment Impact Score on the Impact Analysis tab for Change 1130	6/6/10 2:57 PM		6/6/10 2:57 PM
- All Active Changes:** A table with columns: Owner, Priority, Change, Summary. It lists several changes:

Owner	Priority	Change	Summary
	3	1003	Patch for ITSO Framemaker App
TEST2	3	1004	Change request 1 for testing purpose in ITSO
MAXADMIN	3	1005	Patch System XYZ
		1012	AD_TEST_CHG
		1013	AD_TEST_CHG2
		1014	AD_TEST_CHG3
		AD_CHG3	Deploy WAS ND NODE
PMCFGADM	2	1056	Some requirement in test applications
	2	1067	Some changes in the JZEE business applications
	3	1107	Deploy DB2 Patch
- My Work:** A table with columns: Activity, Summary, Parent, Scheduled Start, Priority, Status. It shows one activity:

Activity	Summary	Parent	Scheduled Start	Priority	Status
1112	Setup and configure Database and Application Server Hardware	CHOW			INPRG
- My Change Policy Documents:** A table with columns: Document, Description, Changed Date, Revision. It lists three documents:

Document	Description	Changed Date	Revision
CAPRS1	Change Policy Information for Change Approver role		
CAPRG1	General Change Management Policy Information		
CAPRG2	Change Tool Policy Information		

Figure 3-74 Change Approver Start Center

A Change can have multiple Change Approvers according to its categorization.

The concept of a Change Advisory Board (CAB) defined by ITIL is applied in this step where personnel responsible for approving the Changes receive them, analyze the assessment information, and decide whether to approve them or not.

The Change Approver uses the assessment data to approve the Change proceeding with the next steps, or requests further analysis if the assessments show a significant risk.

An option to request further analysis information sends a communication to the responsible Change Analyst. A file or web page can be attached in the communication and all communications sent are logged in the Log tab.

To send a communication, select **Select Action** → **Create** → **Communication** as shown in Figure 3-75.

The screenshot shows a software interface for creating a communication. At the top, there is a 'Select Action' dropdown menu. Below it, a 'Progress Map' shows a sequence of steps: ACC_CAT, ASSESS, SCHED, AUTH, IMPL, INPRG, COMP, REVIEW, and CLOSE. The 'Change' field is set to '1005', 'Owner' is 'MAXADMIN', 'Owner Group' is empty, and 'Status' is 'APPR'. The 'Change Details' section is titled 'Create Communication'. It includes a 'Template' dropdown, 'To', 'cc', 'bcc', 'Subject', and 'Message' fields. There are also 'Send From' and 'Reply To' fields. At the bottom, there is an 'Attachments' section with a 'Filter' dropdown and a 'Description' field. The 'Attachments' section shows 'Document' and 'Description' with a message '...No rows to display...'. There are buttons for 'Attach File', 'Attach Web Page', 'Send', and 'Cancel'.

Figure 3-75 Send Communication

Another way to request more information is by adding a new task for the Change Record and assigning it to the responsible person.

To approve or reject an RFC, click the **Change Status** icon as shown in Figure 3-76.

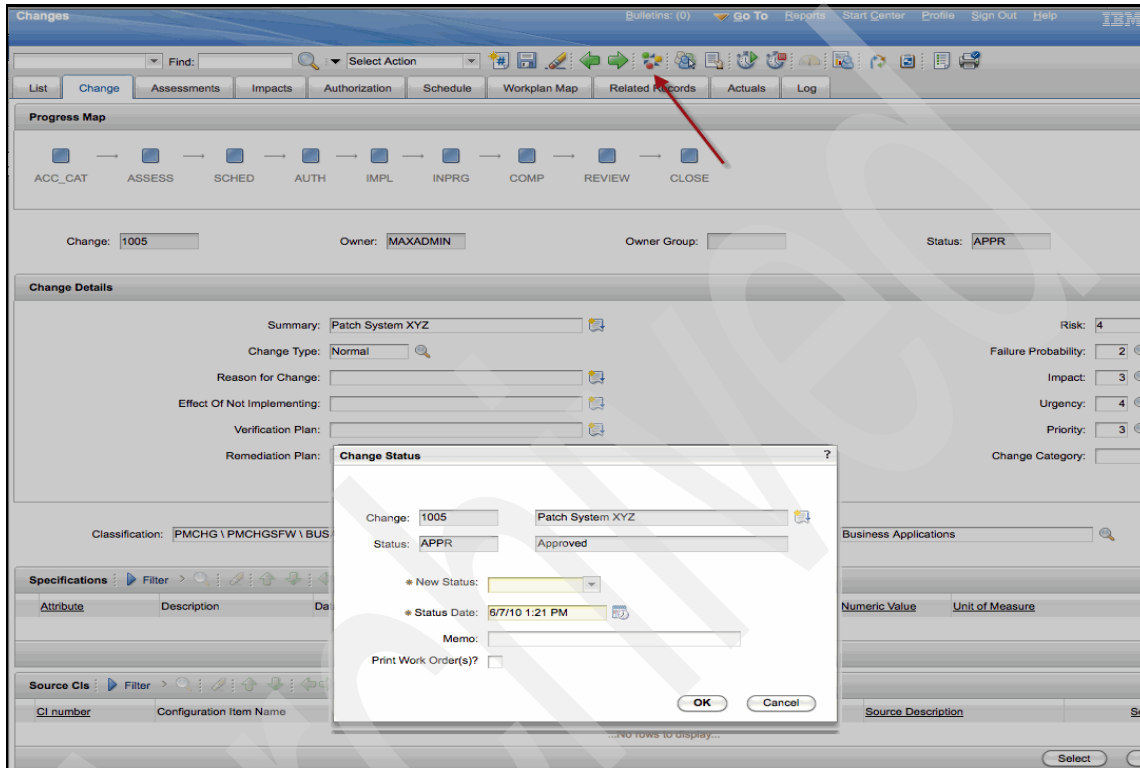


Figure 3-76 Approving a Change

Creating implementation tasks

Responsible role: Change Owner

After the assessments and authorizations are carried out, the Change Owner reviews the implementation notes created during the assessment phase and translates them into specific implementation tasks. For each implementation task, the Change Owner chooses target CIs from the list of CIs identified in the original request, then identifies impacted CIs (those that are affected even though they are not direct targets of the Change).

To add implementation tasks, click **New Task** in the **Tasks for Change** box in the on the **Schedule** tab. as shown in Figure 3-77.

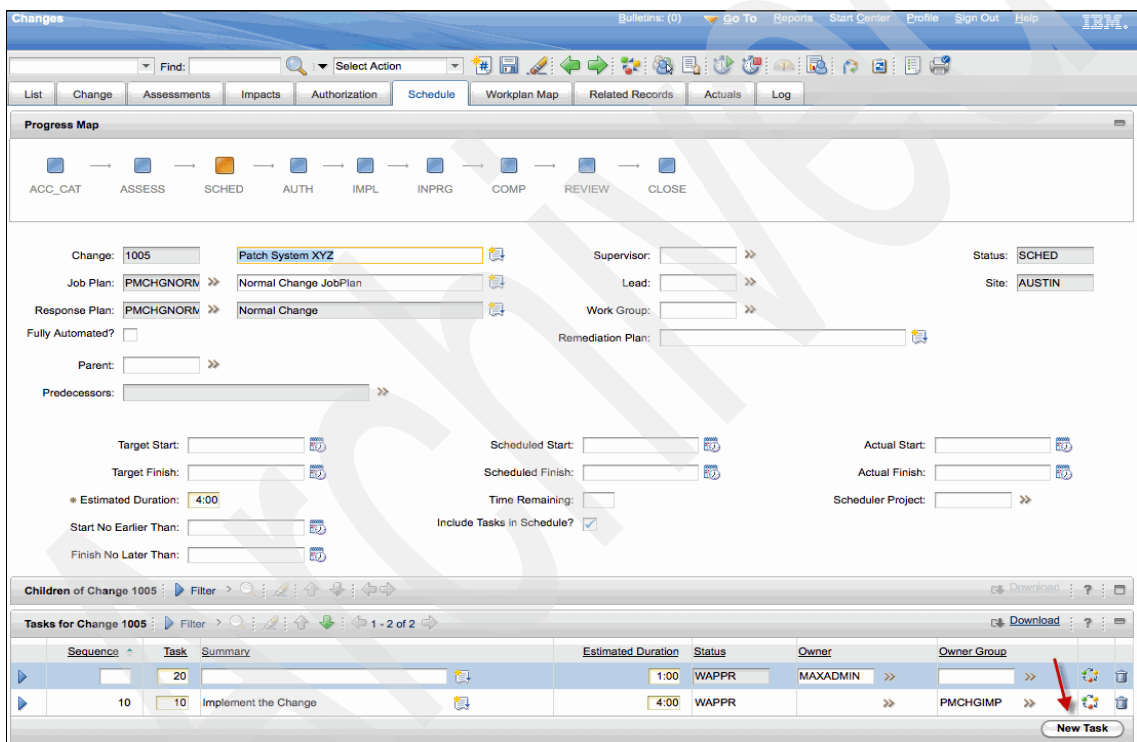


Figure 3-77 Adding Tasks to Changes

When scheduling implementation tasks, the Change Owner consults the Change Window application to determine whether certain affected CIs have defined Change Windows, which specify the times when the CI can be taken out of service in order to make Changes.

The Change Owner also looks at the Change Implementation Schedule application to see whether implementation tasks for other Changes are scheduled for the CIs. Based on this information, and the required sequence of implementation tasks, he creates a schedule for the implementation tasks and then completes the planning phase of the Change.

Defining CI attributes modifications

Responsible role: Change Owner.

The Change Owner has to define what attributes must be modified after the Change implementation.

To perform this action, within the selected Change, choose the option **Move/Swap/Modify** in the Select Action menu, define the attributes modifications, and then click **Save As Plan** as shown in Figure 3-78.

Move/Swap/Modify

Move/Swap | **Modify** | Users and Custodians

Asset | Location | **Configuration Item**

CIs | Filter > [Icons] | 1 - 1 of 1 | Download | ? | [Icon]

Configuration Item Number	Configuration Item Name	Description
10006	WEBSHERE ND	WebSphere app on Miami Linux WAS server

New Row

CI Specifications | Filter > [Icons] | 1 - 1 of 1 | Download | ? | [Icon]

Attribute	Description	Section	Data Type	Alphanumeric Value	Numeric Value	Unit of Measure
☐ SOFTWAREIMAGE_IMAGE_NAME	SOFTWARE IMAGE NAME	ALN				

Modify Attribute

Planned Modifications | Filter > [Icons] | Download | ? | [Icon]

Configuration Item	Configuration Item Name	Attribute	Data Type	Old Value	New Value
...No rows to display...					

Execute Now | Save As Plan | Cancel

Figure 3-78 Defining CI attributes modifications

Saving as Plan means that the modifications will be done when the Change is complete.

Creating a schedule for implementation tasks

Responsible role: Change Owner

After creating the implementation tasks, the Change Owner has to organize the sequence of tasks and assign an owner. This information and other data must be configured in the Activities and Tasks application.

To open the **Activities and Tasks** Application, select **Select Go To → Change → Activities and Tasks**, as shown in Figure 3-79 on page 195.

The screenshot displays the 'Activities and Tasks' application interface. The main form includes fields for Task (10), Location, Asset, Configuration Item, Target Description, Classification (PMREL \ PMRELTSK \ SWDIST), Class Description (Software Distribution), Launch Entry Name, and Parent Process (1005). It also features checkboxes for Under Flow Control, Suspend Flow Control, Flow Action, Flow Action Assist, and Implementation Task. A 'Start Assisted Workflow' button is located at the bottom right of the main form. Below the main form is a 'Scheduling Information' section with fields for Target Start, Target Finish, Scheduled Start, Scheduled Finish, Actual Start, Actual Finish, Estimated Duration (4.00), Predecessors, Start No Earlier Than, and Finish No Later Than. To the right of the scheduling section is a 'Multiple Assets, Locations and CIs' table with columns for Asset, Location, Configuration Item Number, and Target Description. The table shows two rows: one with Configuration Item Number 10003 and another with 10002. Below this table is an 'Attributes' section with a table showing attributes like SWDISTID (Software Deployment ID) and SWPKG (Software Package). The bottom section is 'Parent's Attributes' with a table showing attributes like SWDISTID and SWPKG.

Figure 3-79 Activities and Tasks application

Task owners: In our examples, the task owners are always *maxadmin*, but in practice, these can be unique persons.

There are two ways to access the Activities and Tasks application.

The first way is from the GO TO menu as shown in Figure 3-80.

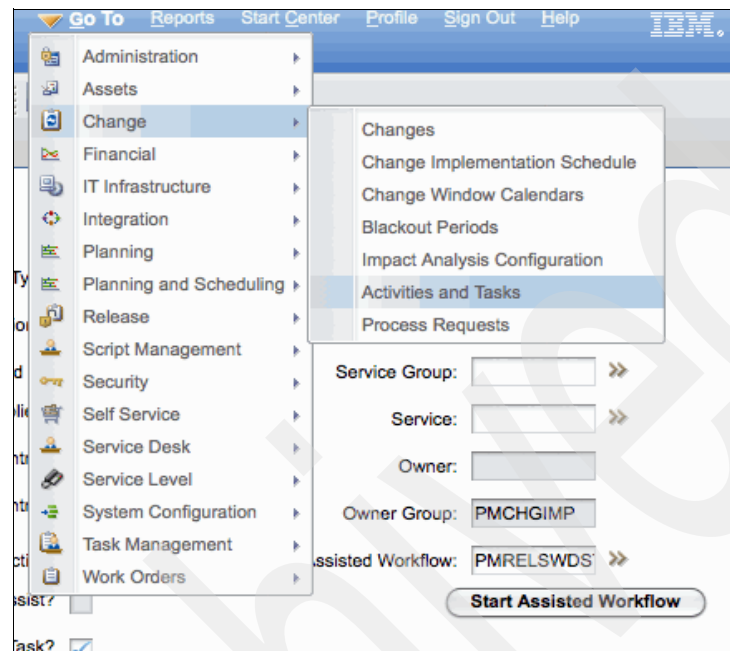


Figure 3-80 Activities and Tasks - option 1

The second way is from within the Change in the **Actuals** tab. Choose the activity, click the arrow near the Record field, and then click **Go To** → **Activities and Tasks** as shown in Figure 3-81.

Changes Bulletins: (0) [Go To](#)

Find: Select Action
[List](#) [Change](#) [Assessments](#) [Impacts](#) [Authorization](#) [Schedule](#) [Workplan Map](#) [Related Records](#) [Actuals](#) [Log](#)

Progress Map

ACC_CAT → ASSESS → SCHED → AUTH → IMPL → INPRG → COMP → REVIEW → CLOSE

Change: Supervisor: >>
 Parent: >> Lead: >>
Work Group: >>

Children of Change 1160 [Filter](#) 1 - 2 of 2

Sequence	Work Order	Summary	Class	Asset	Location	Configuration Item	
1	1173	Upgrade WAS	ACTIVITY			10006	W
2	1174	Go To Activities and Tasks	ACTIVITY			10006	W

Tasks for Change 1160 [Filter](#) 1 - 4 of 4

Sequence	Task	Summary	
1	10	Assessment	
2	20	Approval	
3	30	Implement the Change	
4	40	Post Implementation Review	

Figure 3-81 Activities and Tasks - option 2 - from Actuals Tab

Implementing the tasks

Responsible role: Change Owner.

After the Change Owner has defined and approved the implementation tasks, the Change Implementer or Implementers must consult their Start Center Activities and Tasks Application to verify what tasks they have to perform.

Change Implementer(s) have to perform the task and update the task status, status date, and a memo, as necessary.

Here are a few suggestions for the task status:

- ▶ Failed
- ▶ Approved
- ▶ Canceled
- ▶ Closed
- ▶ Completed
- ▶ In Progress
- ▶ Waiting on Material
- ▶ Waiting on Plant Cond

To update a task status, go to the Activities and Tasks application and select **Select Action** → **Change Status** as shown in Figure 3-82.

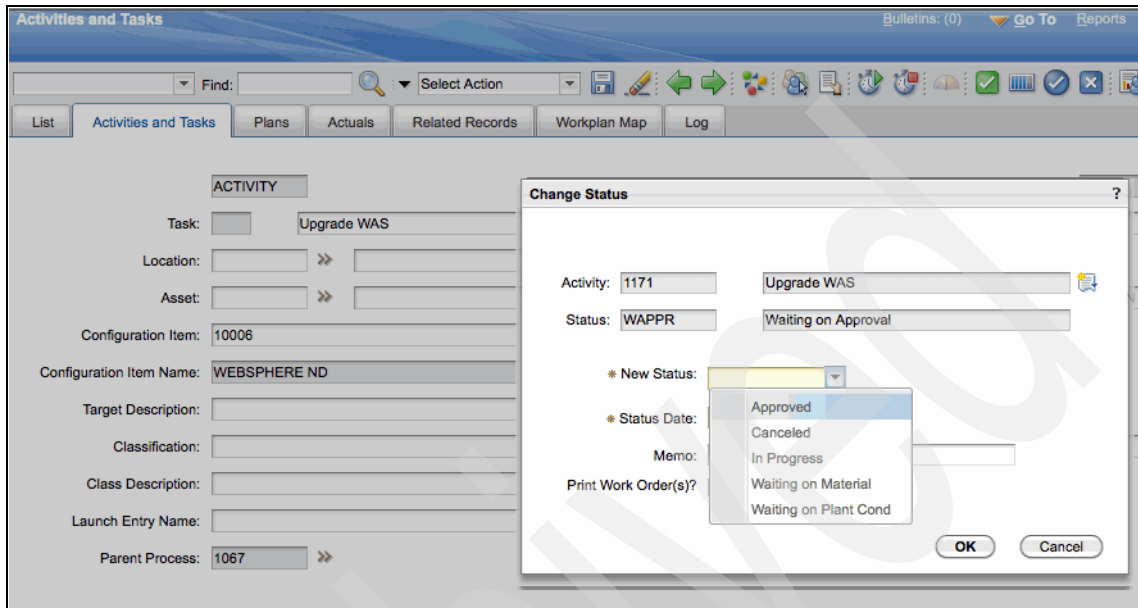


Figure 3-82 Update task status

Tasks: The status options displayed depend on the current status of the task. If the task's current status is Waiting for Approval, then only Approved, Canceled, In Progress, Waiting on Material, or Waiting on Plant Cond are valid options. If the task's current status is In Progress, then the status options can be Failed, Closed, or Completed.

The Change Implementer or Implementers must also update the details of each task implemented in the Log tab as shown in Figure 3-83.

The screenshot shows the IBM Activities and Tasks interface. At the top, there's a toolbar with various icons and a 'Return' button. Below the toolbar, there's a navigation bar with tabs: List, Activities and Tasks, Plans, Actuals, Related Records, Workplan Map, and Log. The 'Log' tab is selected. In the center, there's a form for a work log entry. It includes fields for 'Activity' (1173), 'Upgrade WAS', 'Status' (WAPPR), and 'Site' (AUSTIN). Below this, there's a 'Work Log' section with a table of logs. The table has columns: Record, Class, Created By, Date, Type, Summary, and Viewable. A single record is shown: Record 1173, Class ACTIVITY, Created By MAXADMIN, Date 6/8/10 4:22 PM, Type CLIENTNOTE, Summary Task completed successfully, and Viewable checkbox. Below the table, there's a 'Details' section with fields for Record (1173), Class (ACTIVITY), Created By (MAXADMIN), Date (6/8/10 4:22 PM), Type (CLIENTNOTE), and Viewable? checkbox. The Summary field shows 'Task completed successfully.' and the Details field shows 'No errors.'

Figure 3-83 Work log task implemented

Submitting a Configuration Process Request

Responsible role: Change Owner.

As discussed in “Defining CI attributes modifications” on page 194, a Configuration Process Request must be submitted to the Configuration Management Process.

This action ensures that all attribute Change modifications are appropriately logged.

To create a configuration process request, select **Select Action** → **Create Process Request** as shown in Figure 3-84.

The screenshot displays the 'Process Requests' interface in the IBM Tivoli CCM database. At the top, there's a navigation bar with 'Process Requests' and a search bar. Below this, a toolbar contains various icons for actions like 'Select Action', 'Create', 'Update', 'Delete', etc. The main form is divided into several sections:

- Process Request Header:** Shows 'Process Request: 1042', 'Owner: MAXADMIN', 'Owner Group: ', 'Process State: DRAFT', and 'Status: NEW'.
- User Information:** Includes fields for 'Requestor: MAXADMIN', 'Name: MAXADMIN', 'Phone: ', 'E-mail: ', and 'Reported Date: 6/8/10 4:54 PM'.
- Process Request Details:** Contains fields for 'Description: ', 'Process Manager Type: Configuration', 'Site: AUSTIN', 'Requested Completion: ', 'Asset: ', 'Location: ', 'Configuration Item: ', 'Configuration Item Name: ', 'Response Plan: ', and 'Priority: '.
- Request Classification Details:** Includes 'Classification: ' and 'Class Description: '.
- Classification Attributes Table:** A table with columns 'Attribute', 'Description', and 'Value'. It currently shows 'No rows to display...'. There is a 'New Row' button below it.
- Target Classification Table:** A table with columns 'Asset', 'Location', 'Configuration Item Number', 'Configuration Item Name', and 'Target Description'. It also shows 'No rows to display...'. There is a 'New Row' button below it.

Figure 3-84 Create a configuration process request

For a detailed explanation of this function, see 3.3.10, “Interaction with other processes” on page 246.

Scheduler Application

Responsible role: Change Owner.

The scheduler is the CCMDB application that allows you to pull in work orders necessary to complete a Change, and schedule the start and finish time for them. The types of work orders that can be scheduled are WOACTIVITY, WOCHANGE, and WORELEASE. The Scheduler can be used to determine a start and finish time for the Change and implementation tasks, taking into account the CI's Change Windows and blackout periods.

To schedule a Change, select the **Schedule** tab within a Change Record. From the *Scheduler Project Field*, select **Create a Project Schedule** from the pull down menu as shown in Figure 3-85.

The screenshot displays the 'Scheduler' tab within a Change Record in the CCMDB application. The interface includes a 'Progress Map' at the top with a sequence of steps: ACC_CAT, ASSESS, AUTH, SCHED (current), IMPL, INPRG, COMP, REVIEW, and CLOSE. Below this, the 'Change' field is set to 'MYCHANGE' with the description 'Update payroll system'. Other fields include 'Job Plan', 'Response Plan', 'Requires Scheduling?' (checked), 'Fully Automated?' (unchecked), 'Parent', 'Predecessors', 'Target Start/Finish', 'Estimated Duration' (0:00), 'Start No Earlier Than', 'Finish No Later Than', 'Sched Start/Finish', 'Time Remaining', 'Include Tasks in Schedule?' (checked), 'Supervisor', 'Lead', 'Work Group', 'Remediation Plan', 'Status' (WAPPR), and 'Site' (AUSTIN). A red box highlights the 'Scheduler Project' field, which has a dropdown menu open showing options: 'Select Value', 'Go To Scheduler', 'Clear', and 'Create a Project Schedule'. Below the form, there is a section for 'Children of Change MYCHANGE' and a table for 'Tasks for Change MYCHANGE'.

Sequence	Task	Summary	Estimated Duration	Status	Owner	Owner Group
10	Install DB patch		2:00	WAPPR		

Task Information: Task: 10, Install DB patch, Under Flow Control? [checked], Attachments

Figure 3-85 Creating a Project Schedule

Attention: If there is already a Scheduler Project listed in this field, the new Schedule will overwrite the existing one.

This will launch the Scheduler Application, as shown in Figure 3-86.

The screenshot displays the 'Scheduler' application window. At the top, there's a toolbar with icons for Find, Select Action, and other functions. Below the toolbar, there are tabs for 'List', 'Schedule', and 'Gantt View'. The 'Schedule' tab is active, showing configuration details for a schedule named 'CHANGE_9' with the description 'Update payroll system'. The configuration includes fields for Calendar, Shift, Organization, Start Date, and Public? (checked). On the right, there are fields for Last Saved Date (5/13/10 14:48:32), Last Committed Date, Created By (MAXADMIN), and Created Date (5/13/10 14:48:32). There are also checkboxes for 'Preserve Schedule Start Dates?' (checked) and 'Refresh When Opening Gantt View?' (unchecked). Below the configuration section, there are three tables: 'Work Queries', 'Additional Resources', and 'Resource Availability'. The 'Work Queries' table has one row with Data Source 'WOCHANGE', Query Name 'CHANGE_9', and Where Condition 'woum=MYCHANGE'. The 'Additional Resources' and 'Resource Availability' tables are currently empty, showing '...No rows to display...'. At the bottom left, there's a breadcrumb trail 'Changes > Scheduler'.

Data Source	Query Name	Description	Where Condition
WOCHANGE	CHANGE_9		woum=MYCHANGE

Data Source	Query Name	Description	Where Condition
...No rows to display...			

Figure 3-86 Scheduler Application

To view the Scheduler graphically as a Gantt chart, select the **Gantt View** tab as shown in Figure 3-87.

The Gantt View page is divided into four sections; the upper part of the window shows a Gantt Chart, and the lower sections show the Resource Constraints view. The Gantt view visually maps out the Change tasks against the schedule constraints to show the available Change Windows, blackout periods, and other tasks.

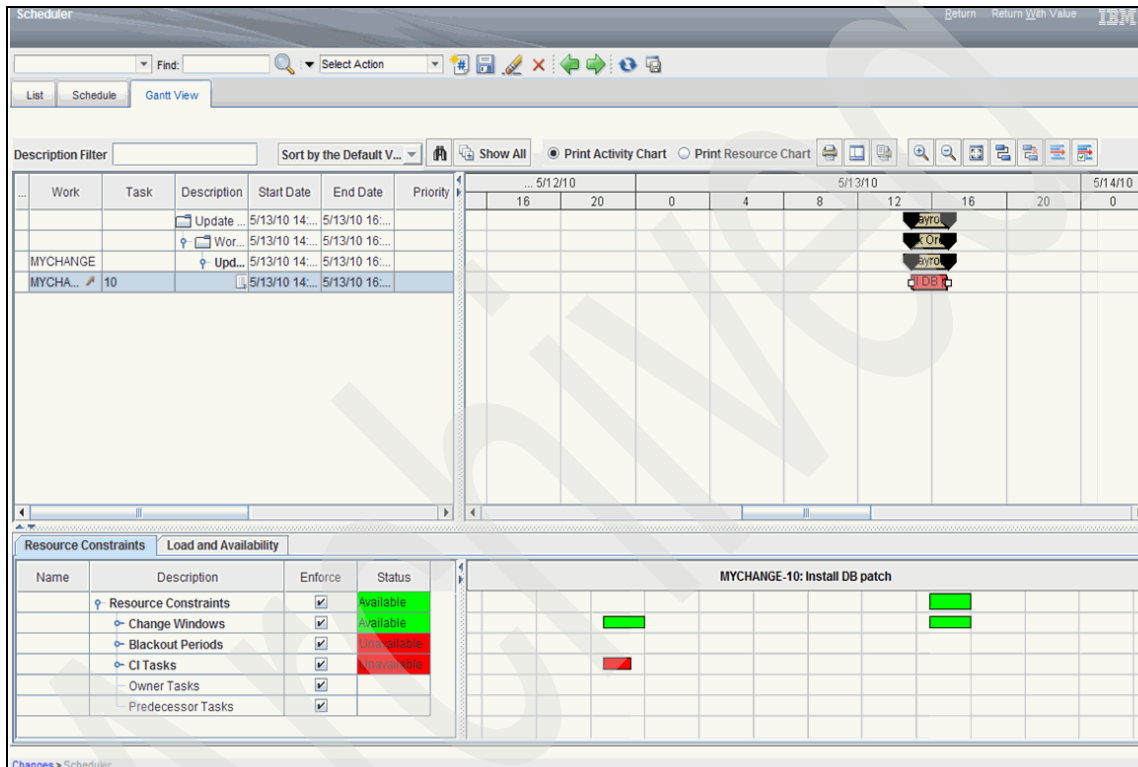


Figure 3-87 Scheduler Gantt View

To automatically schedule a Change, click the toolbar button **Perform Critical Path Method on all Rows** as shown in Figure 3-88. This will automatically find the next available time period for this Change.

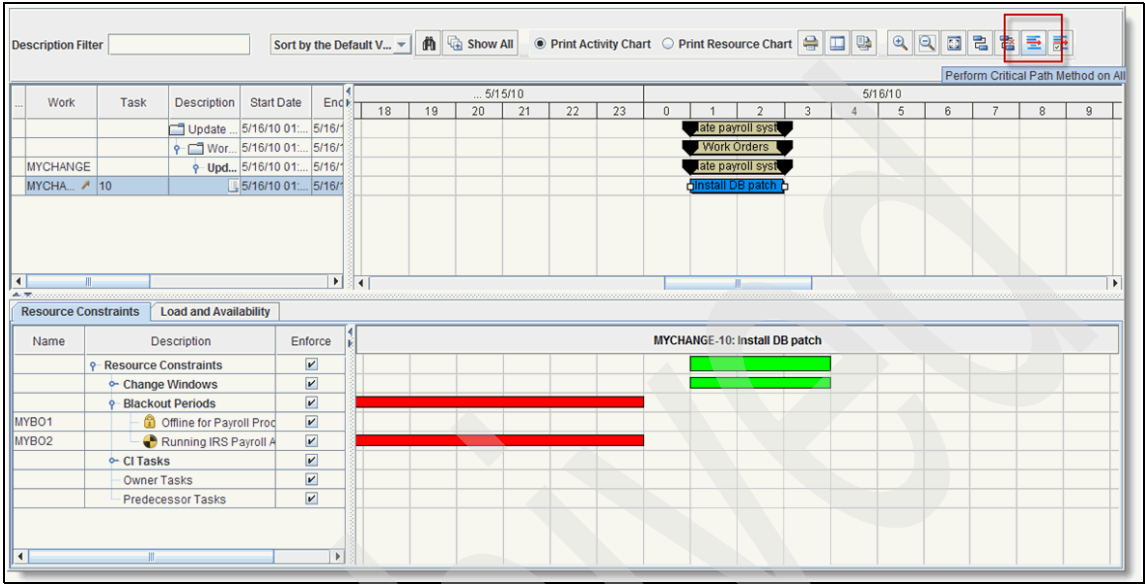


Figure 3-88 Scheduler Gantt View - Automatic Schedule creation.

After the Change has been scheduled, click the **Commit Changes** toolbar button to bind the schedule to the Change as shown in Figure 3-89, then select **Return with Value** to return to the Change.

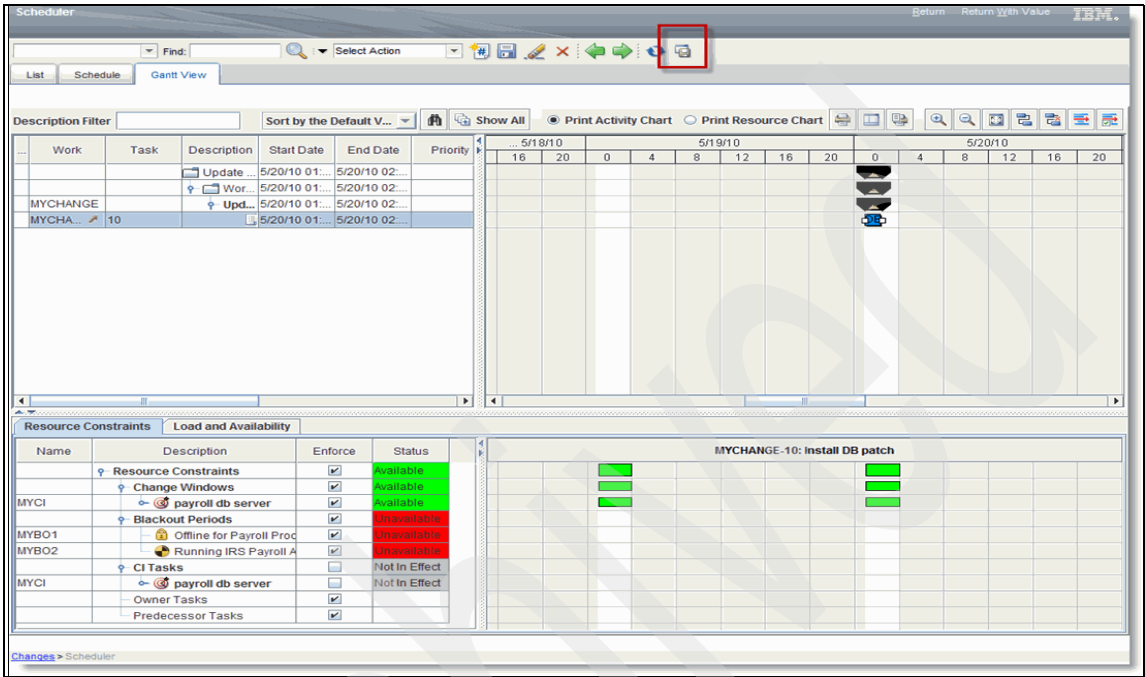


Figure 3-89 Scheduler Gantt View - Scheduler Commit Changes.

When returning to the Change application, now the schedule information is updated and shown in the Change as depicted in Figure 3-90.

Figure 3-90 Change Record - updated Change schedule

Post-implementation review

Responsible role: Change Manager.

A post-implementation review must be performed after all Changes are implemented. This activity will ensure that the Change was implemented according to the plan. The Change Manager or Change Owner has to verify whether the Change achieves its purpose by using the comments or feedback in each task performed. It is important that there be high quality comments or feedback from Change Implementers during the implementation.

Post Implementation Review also provides information to the team to compare the plan to actual data to improve its ability to predict costs and times, and review any other aspects of implementing the Change that are of interest.

After completing the review, the post implementation review must have its status updated to Complete.

To update a Change status, select the option **Change Status** from the Select Action menu and then select the appropriate status as shown in Figure 3-91.

The screenshot displays the Change Manager application interface. At the top, a navigation bar includes tabs for List, Change, Assessments, Impacts, Authorization, Schedule, Workplan Map, Related Records, Actuals, and Log. Below this, a 'Progress Map' section shows a sequence of steps: ACC_CAT, ASSESS, SCHED, AUTH, IMPL, INPRG, COMP, REVIEW, and CLOSE. The 'REVIEW' step is currently selected.

The main content area shows details for Change 1160, 'Online Banking Application Upgrade'. Fields include Change (1160), Parent (empty), Supervisor (empty), Lead (empty), Work Group (empty), Status (REVIEW), and Site (AUSTIN).

Below the details, there are two tables. The first table, 'Children of Change 1160', lists two work orders:

Sequence	Work Order	Summary	Class	Asset	Location	Configuration Item	Configuration Item Name	Status
1	1173	Upgrade WAS	ACTIVITY			10008	WEBSPPHERE ND	COMP
2	1174	Upgrade WAS Application	ACTIVITY			10008	WEBSPPHERE ND	COMP

The second table, 'Tasks for Change 1160', lists four tasks:

Sequence	Task	Summary	Estimated Duration	Status
1	10	Assessment	1:00	COMP
2	20	Approval	1:00	COMP
3	30	Implement the Change	1:00	COMP
4	40	Post Implementation Review	1:00	COMP

At the bottom, there is a 'Labor' section with a table for labor tracking. The table has columns for Task, Labor, Name, Approved, Start Date, Start Time, End Time, Regular Hours, and Rate. The message '...No rows to display...' is shown below the table.

Figure 3-91 Post implementation review

Closing an RFC

Responsible role: Change Manager.

The final step of the process is to close the RFC record. To update a Change status, select the option **Change Status** from Select Action menu and then select the appropriate status as shown in Figure 3-92.

Change: 1160 Online Banking Application Upgrade

Supervisor: Lead: Status: COMP Site: AUSTIN

Change Status

Change: 1160 Online Banking Application Upgrade

Status: COMP Completed

New Status: Closed

Status Date: Completed

Memo:

Print Work Order(s)?

OK Cancel

Children of Change 1160

Sequence	Work Order	Summary
1	1173	Upgrade WAS
2	1174	Upgrade WAS Application

Tasks for Change 1160

Sequence	Task	Summary
1	10	Assessment
2	20	Approval
3	30	Implement the Change
4	40	Post Implementation Review

Labor

Task	Labor	Name	Approved	Start Date	Start Time	End Time	Regular Hours	Rate
No rows to display...								

Figure 3-92 Closing an RFC

The Request For Change record in the Process Request application is updated to status CLOSED as shown in Figure 3-93.

The screenshot displays the 'Changes' application interface. At the top, there is a navigation bar with tabs: List, Change, Assessments, Impacts, Authorization, Schedule, Workplan Map, Related Records, Actuals, and Log. Below this is a 'Progress Map' section showing a workflow from ACC_CAT to ASSESS, SCHED, AUTH, IMPL, INPRG, COMP, REVIEW, and finally CLOSE. The main form area shows details for Change 1160, 'Online Banking Application Upgrade'. Fields include Change (1160), Parent (empty), Supervisor (empty), Lead (empty), Work Group (empty), Status (CLOSE), and Site (AUSTIN). Below this is a table for 'Children of Change 1160' with columns: Sequence, Work Order, Summary, Class, Asset, Location, Configuration Item, Configuration Item Name, and Status. The table contains two rows: 1. Sequence 1, Work Order 1173, Summary 'Upgrade WAS', Class 'ACTIVITY', Asset empty, Location empty, Configuration Item 10006, Configuration Item Name 'WEBSPHERE ND', Status 'COMP'. 2. Sequence 2, Work Order 1174, Summary 'Upgrade WAS Application', Class 'ACTIVITY', Asset empty, Location empty, Configuration Item 10006, Configuration Item Name 'WEBSPHERE ND', Status 'COMP'. Below the table are buttons for 'Select Work Orders' and 'New Activity'. The next section is 'Tasks for Change 1160' with columns: Sequence, Task, Summary, Estimated Duration, and Status. It contains four rows: 1. Sequence 1, Task 10, Summary 'Assessment', Estimated Duration 1.00, Status 'COMP'. 2. Sequence 2, Task 20, Summary 'Approval', Estimated Duration 1.00, Status 'COMP'. 3. Sequence 3, Task 30, Summary 'Implement the Change', Estimated Duration 1.00, Status 'COMP'. 4. Sequence 4, Task 40, Summary 'Post Implementation Review', Estimated Duration 1.00, Status 'COMP'. Below this is a 'Labor' section with tabs: Labor, Materials, Services, Tools. The 'Labor' tab is active, showing a table with columns: Task, Labor, Name, Approved, Start Date, Start Time, End Time, Regular Hours, and Rate. The table is empty with the message '...No rows to display...'. At the bottom are buttons for 'Select Labor', 'Select Planned Labor', and 'New Row'.

Figure 3-93 RFC Closed

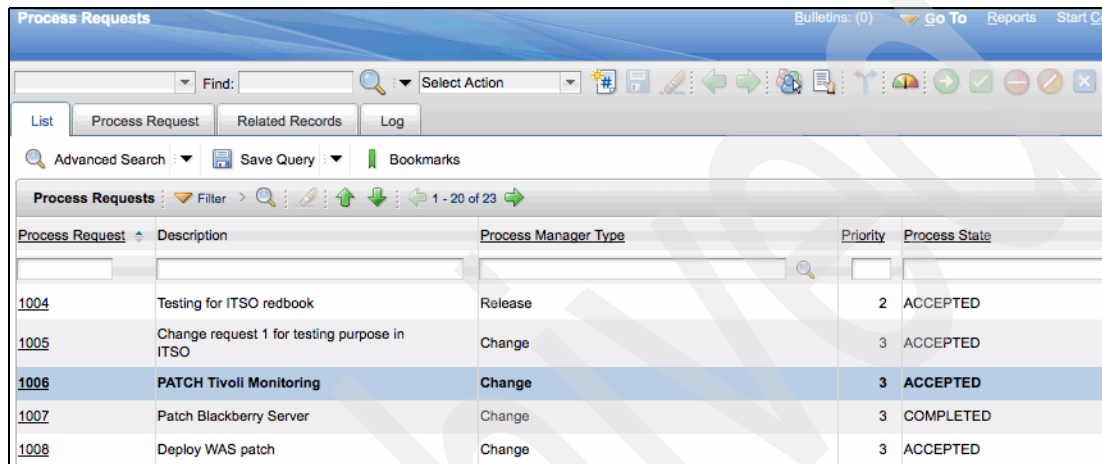
3.3.3 Functions applicable to Change Management

The following sections provide detailed information about the most important functions that can be used with an RFC and Change Records.

3.3.4 Accepting or rejecting a Request for Change

The role responsible for responding to Requests for Change can either accept or reject them. The Requests for Change can be viewed by selecting **Change** → **Process Requests**. It shows the process requests number, description, Process Manager type, priority, and process state.

Figure 3-94 illustrates a list of process requests for acceptance or rejection.



The screenshot shows the 'Process Requests' window with a table of requests. The table has columns for Process Request, Description, Process Manager Type, Priority, and Process State. The following table represents the data shown in the screenshot:

Process Request	Description	Process Manager Type	Priority	Process State
1004	Testing for ITSO redbook	Release	2	ACCEPTED
1005	Change request 1 for testing purpose in ITSO	Change	3	ACCEPTED
1006	PATCH Tivoli Monitoring	Change	3	ACCEPTED
1007	Patch Blackberry Server	Change	3	COMPLETED
1008	Deploy WAS patch	Change	3	ACCEPTED

Figure 3-94 Process Lists

The Change Owner or the role whose responsibilities include making the initial response to Requests for Change can also see a list of Changes that have been requested in their Start Center.

To start an action with a Process Request, it is necessary to choose one from the following list:

- ▶ Use Submit to submit a Request for Change. The status of the request is changed to Submitted.
- ▶ Use Accept to accept a Request for Change. The status of the request is changed to Accepted.
- ▶ Use Reject to reject a Request for Change. The status of the request is changed to Rejected.
- ▶ Use Cancel to cancel a Request for Change. The status of the request is changed to Canceled.
- ▶ Use Close to close a Request for Change. The status of the request is changed to Completed.

Figure 3-95 shows the list of actions that can be performed on a Process Request.

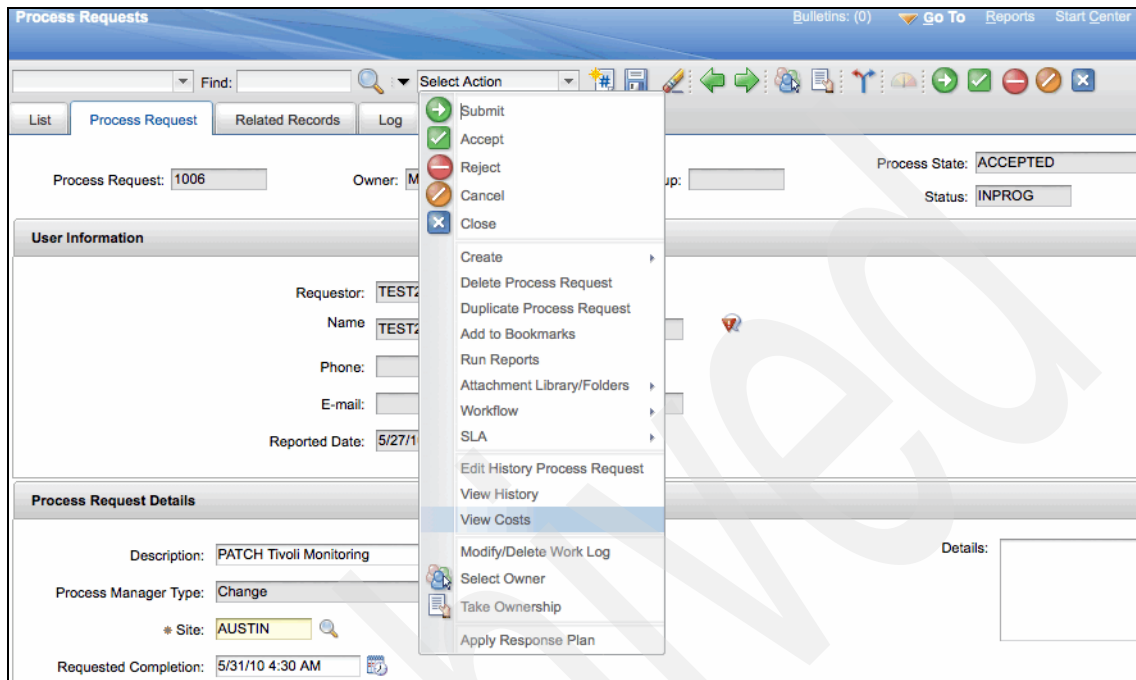


Figure 3-95 Process List of Actions

Organizations need to have rules that help decide how to respond to each request. As a suggestion, the following questions can be asked:

- ▶ View the details of the request. Are they complete? What details are mandatory?
- ▶ Evaluate the Change according to standard practice. Is there any reason why the request ought not to be accepted? Note that when a request is accepted, it is not a promise that the Change will be made. It is an agreement that the request meets the criteria for being evaluated further.
- ▶ If the request does not meet the criteria, reject it. With the Change selected in the list, or while viewing the details of the request, select **Reject** from the Select Action menu. The status of the request is set to Resolved and the resolution is set to Reject.
- ▶ If the request does meet the criteria, accept it. With the Change selected in the list, or while viewing the details of the request, select **Accept** from the Select Action menu. The status of the request is set to Resolved and the resolution is set to Accept.

When a request is accepted or rejected, it no longer displays on lists of requests that require a response. If it was accepted, it displays on lists of Changes that are in progress. Depending on each organizations's process, it might act as follows:

- ▶ Choose a job plan to be used to assess and implement this Change.
- ▶ Assign the Change to a Change Owner. The person who accepts the request becomes the owner of the Change.

3.3.5 Assessing a Change

The Assessment page contains both a summary of the assessment as well as an area for Technical Assessments as well as Business Assessments.

Technical Assessments

The Technical Assessment Result sub-section is used to capture assessment analysis performed by technical subject matter experts (SMEs), as well as implementation related information that the SME needs to pass along to those doing the implementation planning (Figure 3-96).

Each assessment entry contains the following fields for which the SME will provide values:

- ▶ **Assessment type:**
The area that was assessed, such as Storage, Network, Security, or others. If you want to add new values to the provided list, use the Domains application to add them to the PMCHGASSESSMENTTYPE domain.
- ▶ **Assessment description:**
The text entered by the expert performing the assessment.
- ▶ **Cost:**
The estimated monetary cost of this Change for the area being assessed. The sum of the costs entered by all assessors is displayed at the top of this sub-tab.
- ▶ **Effort:**
The estimated hours of effort required to implement this Change in the area being assessed. The sum of the effort entered by all assessors is displayed at the top of this sub-tab.
- ▶ **Impact:**
A summary rating of the impact, such as None, Low, Medium, or High. If you want to add new values to the provided list, use the Domains application to add them to the PMCHGIAIMPACT domain.
- ▶ **Owner:**
The user responsible for this assessment.

While performing the technical assessment, each expert needs to use the Implementation Note feature to record items that must be considered during the implementation phase. These notes have to include assessments of the outage requirements for the targets of the Change. This might be determined by delving into the details of the implementation, for example, by reading the installation notes provided with a software update to determine whether the server needs to be taken down during the update.

Also, examine the current IT configuration to understand the interaction between the implementation work and availability of the service being modified. For example, if only one server in an application server cluster will be affected by the implementation work, then the service might not have an outage, but simply a performance degradation. After the assessments have been completed, the Change Owner will convert these notes into implementation tasks. Adding tasks to the schedule is discussed in 3.3.7, “Change Management Schedule” on page 224.

Figure 3-96 shows the Impact Analysis - Technical Assessment Results window.

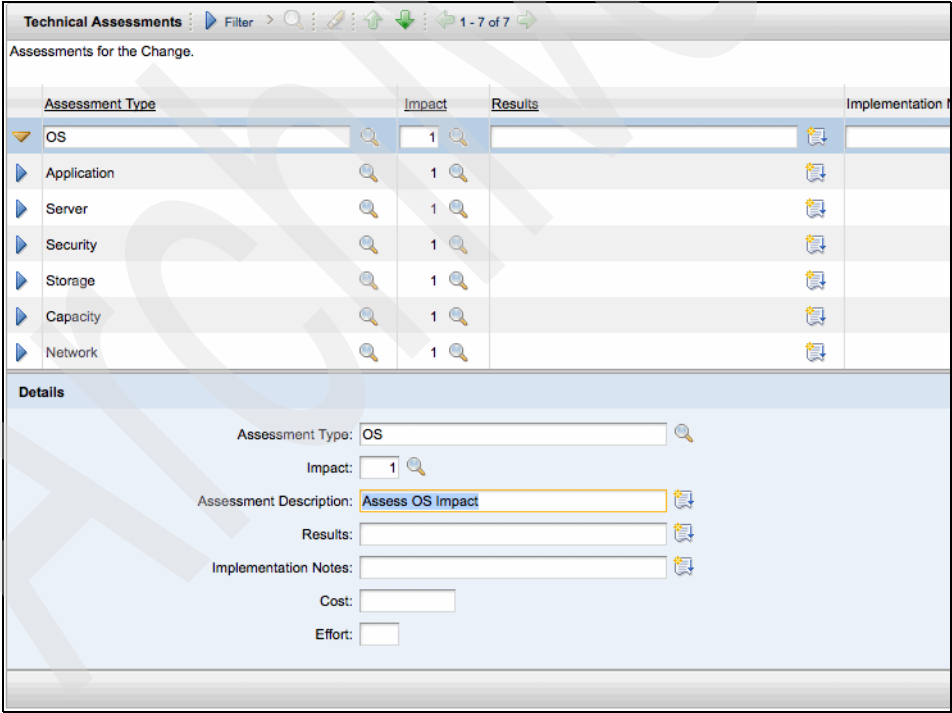


Figure 3-96 Technical Assessments Results

Business Assessments

The Business Assessment Results sub-section is used to analyze and record the business impacts of the Change. Business subject matter experts can use the Business Assessment Results sub-tab to record their assessments of the impact the Change will have in their areas, such as financial, operational, regulatory, or others (Figure 3-97 on page 215).

Each expert must complete the following fields:

- ▶ **Assessment type:**

The area that was assessed, such as Financial, Operational, or SOX. If you want to add new values to the provided list, use the Domains application to add them to the PMCHGBUSASSESTYPE domain.

- ▶ **Assessment description:**

The text entered by the expert performing the assessment.

- ▶ **Cost:**

The estimated monetary cost of this Change for the area being assessed. The sum of the costs entered by all assessors is displayed at the top of this sub-tab.

- ▶ **Effort:**

The estimated hours of effort required to implement this Change in the area being assessed. The sum of the effort entered by all assessors is displayed at the top of this sub-tab.

- ▶ **Impact:**

A summary rating of the impact, such as None, Low, Medium, or High. If you want to add new values to the provided list, use the Domains application to add them to the PMCHGIAIMPACT domain.

- ▶ **Owner:**

The user responsible for this assessment.

Figure 3-97 shows the Business Assessments section of the Assessments Tab.

The screenshot displays the 'Business Assessments' section. At the top, there's a header 'Business Assessments' with a filter icon and a status '1 - 3 of 3'. Below this is a sub-header 'Assessments for the Change.' followed by a table. The table has columns: Assessment Type, Impact, Results, Cost, Effort, Owner, and Owner Group. Three rows are visible: SOX, Financial, and Operational. The 'Operational' row is selected. Below the table is a 'Details' section. It contains fields for Assessment Type (Operational), Impact (4), Assessment Description (Systems Down from 8pm to midnight), Results, Cost, and Effort. On the right side of the details section, there are fields for Owner, Owner Group (PMCHGANA), Assessor (MAXADMIN), Date Created (6/8/10 1:07 PM), and Last Modified (6/8/10 3:24 PM). A 'New Row' button is at the bottom right.

Assessment Type	Impact	Results	Cost	Effort	Owner	Owner Group
SOX	4					PMCHGANA
Financial	4					PMCHGANA
Operational	4					PMCHGANA

Details

Assessment Type:

Impact:

Assessment Description:

Results:

Cost:

Effort:

Owner:

Owner Group:

Assessor:

Date Created:

Last Modified:

[New Row](#)

Figure 3-97 Business Assessment Results

3.3.6 Change impact analysis

Impact analysis is an essential activity of the Change Management process because it is responsible for ensuring that an RFC is evaluated from both business and technical perspectives and can be successfully implemented with a minimal impact to committed service(s). It identifies and records which systems, applications, or other configuration items that will be impacted or targeted by a proposed Change, including the type of effects on each CI.

Most Changes modify one or more configuration items. These modified configuration items are also called target CIs. An impacted CI is a CI that is not being modified as part of a Change implementation, but might suffer degradation of service due to that implementation.

Figure 3-98 shows the CIs target and impacted concept.

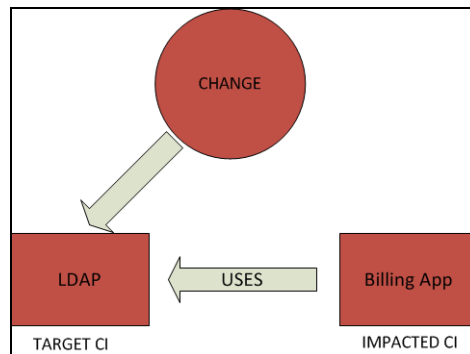


Figure 3-98 Target versus Impacted concept

When all the consequences are documented, the subsequent steps of the process will use this information. For example, authorizations and implementation scheduling will depend on accurate impact analysis data. Here is an example of impact analysis: Consider a Change that updates the version of WebSphere Application Server on a set of computer systems. Impact assessment might determine that the computers will have to be restarted as part of the upgrade process.

Further investigation might reveal that your company's accounting applications run on those servers. While those applications are not the target of the Change, they are impacted by it, and the effect on them must be taken into consideration. Relationships between configuration items are identified in the discovery process. Use these relationships as a starting point and use the judgment of technical experts to identify all the relevant relationships between the configuration items specifically targeted by a Change and others that will be affected.

Previewing impacts of a Change

By previewing the possible impacts of a Change, a user can identify a disruptive Change early on in order to plan more accordingly. The impact analysis engine can be invoked at any time.

To preview the impacts of a Change, perform the following steps:

1. With the Change open in the Changes application, select **Impact Analysis** → **Preview Impacts** from the Select Action menu. The Preview Impacts of the Change Window opens.

2. In the Preview Impacts of the Change Window, open one of the following tabs:
 - **Calculated:** Lists impacted CIs, as calculated by the engine using the impact assessment rules that you have configured.
 - **Historical:** Lists impacted CIs, as retrieved by the engine based on impacts that have historically been identified.

Figure 3-99 shows the Preview Impacts Results window.

Impacted CI	Impacted CI Name	Business Impact	Impacted Description	Target CI Causing Impact	Target CI Name
10003	MIALINUXDB2		COMPUTER	DB2 INSTANCE	MIAMI DB2
DB2 INSTANCE	MIAMI DB2		Software Image, DB2, 9.5	10003	MIALINUXDB2
10006	WEBSPPHERE ND		WebSphere app on Miami Linux WAS server	10004	MIALINUXWAS01

Figure 3-99 Impact results

The Impacts Engine can also be invoked by using a workflow and not manually as described. After the engine performs the impact analysis, navigate to the *Impacts* tab, if you are not already in this view. The results of the analysis are displayed. You can view a topology graph that shows task target CIs and impacted CIs; view a summary of the identified impacts; and perform a number of operation to manage the impacts.

Impacts tab

The Impacts tab is shown when a Change Record is selected. It contains two sub-tabs:

- ▶ Summary
- ▶ Manage Impacts

Figure 3-100 illustrates the Impact Summary Analysis tab.

Change: 1161

Summary: Patch Content Manager

Outage Impact:

Total Outages: 2

Details: Patch WAS instance of Content Manager

Owner: MAXADMIN

Owner Group:

Status: IMPL

SummaryManage Impacts

Targeted by Change 1161

Filter

1 - 4 of 4

Download

Configuration Item Number	Configuration Item Name	Asset	Location	Business Impact	Outage	Scheduled Start	Scheduled Finish	Task
10004	MIALINUXWAS01				Offline			1166
10004	MIALINUXWAS01				Offline			1167
10004	MIALINUXWAS01				Offline			1168
10004	MIALINUXWAS01				Offline			1169

Impacted by Change 1161

Filter

1 - 4 of 4

Download

Configuration Item	Configuration Item Name	Asset	Location	Business Impact	Outage	Assessment Done	Approval	Notify	Scheduled Start	Scheduled Finish	Task
10006	WEBSPHERE ND				Offline						1166
10006	WEBSPHERE ND				Offline						1167
10006	WEBSPHERE ND				Offline						1168
10006	WEBSPHERE ND				Offline						1169

Calculate Impacts

Detach Historical Impacts

Figure 3-100 Impact Summary Analysis

Summary sub-tab

The Summary sub-tab provides a high level view of the impact assessment results. It contains three sections: a summary of the impact, and a roll-up of the CIs identified as Targeted or impacted by this Change, as well as a graphical detailed or business diagram of the targeted or impacted CIs. The Summary section contains a text field where the Change Analyst can record a high level summary of the impact assessment. For simple Changes, this might be all the impact assessment data required. It also calculates the number of outages.

The target CIs section contains a table listing all the targets associated with any of the implementation tasks in this Change with information such as location, impact, and schedule information.

The Impacted CIs section contains a table listing all the impacted CIs associated with any of the implementation tasks in this Change with information about task schedules as well as individual outage information. There are also fields for assessment completion, authorizations and notifications in order to ensure impacted items are properly accounted for in the Change process.

Manage Impacts sub-tab

The Manage Impacts sub-tab lists all of the implementation tasks for the Change you have selected. As you click each task in the Implementation Tasks for Change are you will notice that both the Targets for Tasks and Impacts for Tasks sub areas update accordingly in relation to the task you have chosen. This is useful if you have possibly added or removed a target from the Change and want to see the affects on the impacted CIs as a result.

You also might want to calculate the impacts of individual target CIs or assets when a new target is added to the task, or when new CI relationships are added to the configuration management database (CMDB). After you calculate the target CI or asset impacts, you can select one or more of the impacts to record them as impacts for the target.

Figure 3-101 shows the Manage impacts tab.

SummaryManage Impacts

In order to reduce unplanned outages, use the tables below to identify CIs or Assets impacted by each implementation task.

Implementation Tasks for Change 1161

Filter > 🔍 ✎ ⬆ ⬇ ⬆ 1 - 4 of 4 ⬆

Choose a task from the table, then use the button below to identify CIs or Assets impacted by the chosen task.

Task	Task Id	Description	Estimated Duration	Sched
▶ 1166 >>	10	Assessment	2:00	
▶ 1167 >>	20	Approval	1:00	
▶ 1168 >>	30	Implement the Change	10:00	
▶ 1169 >>	40	Post Implementation Review	2:00	

Targets for Task 1166-10

Filter > 🔍 ✎ ⬆ ⬇ ⬆ 1 - 1 of 1 ⬆

This table contains targets associated with the task selected above. Use the button below to identify the impacts for the selected target.

Configuration Item Number	Business Impact	Asset	Location	Target Description	Outage
▶ 10004					Offline

Impacts for Task 1166-10

Filter > 🔍 ✎ ⬆ ⬇ ⬆ 1 - 1 of 1 ⬆

This table contains impacts associated with the task selected above. Use the button below to identify additional impacts.

Configuration Item	Business Impact	Asset	Target CI Causing Impact	Target Asset Causing Impact
▶ 10006			10004	

Figure 3-101 Manage Impacts tab

Viewing an Impact Analysis CI topology

On the Manage Impacts tab and beneath the Target and Impact lists, the diagram beneath these sections gives you a visual business or detailed impact analysis topology of your Change. Depending on the filter criteria, such as Tasks, Relationships, Classifications or Breadth (the number of relations to show in a view), you can modify the reach and scope of the topology in relation to your needs and analysis.

The difference between the detailed topology vs. the business topology is that The business view only shows CIs that have been designated to have a high business impact. When CIs are classified, you choose whether or not they are substantial enough to cause impact when taken out of service. For example, you cannot classify NICs or hot pluggable storage devices as high impact items, but for servers and services, you can. The detailed view will just show all CIs regardless of classification.

Figure 3-102 shows the CI topology View for impact analysis.

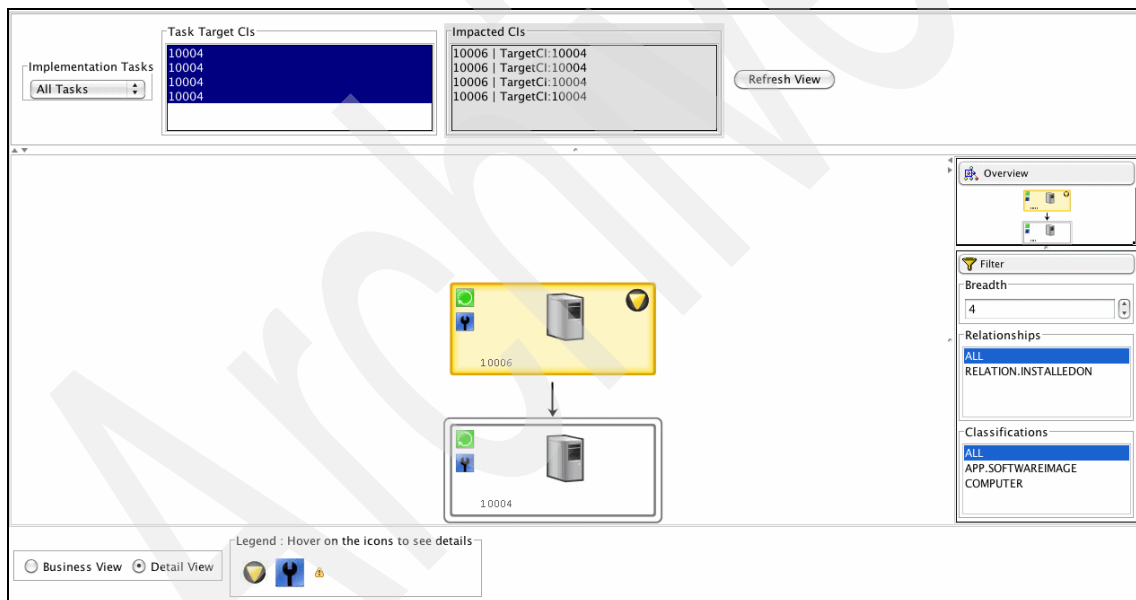


Figure 3-102 CI topology

Scheduling tasks

You can schedule tasks in either the Change Application or the Task Scheduler Application. Scheduling a task in the Change Application will reschedule all tasks, while if done in the Scheduler application, only the task will be scheduled.

Tasks for Change sub-section

The main purpose of the Tasks for Change sub-section of the scheduler is to use the implementation notes, created during technical assessment, to create implementation tasks. Implementation tasks and task targets must be created to translate the requirements identified during assessment, and captured in the implementation notes, into concrete process tasks.

Terms:

► Activity:

Activities are the main building blocks for processes. An activity is a collection of work breakdown elements, such as task descriptors, role descriptors, work product descriptors, and milestones. Activities can include other activities.

Activities can be presented in work breakdown structures and activity diagrams that graphically describe the flow of work by showing which activities precede other activities. Phase and iteration are special types of activities that define specific properties.

► Task:

A task is an assignable unit of work. Every task is assigned to a specific role. The duration of a task is generally a few hours to a few days. Tasks usually generate one or more work products.

Prior to creating implementation tasks, one or more child implementation activities must be created to contain the implementation tasks. Activities can be created in the Change - Activities and Tasks Tab using the **New Activity** button. Creating the activities in this manner will result in them automatically being created as children of this Change.

After implementation tasks are created on the Schedule Tab, targets are assigned to the tasks from the list of targets associated with the Change. As a convenience, targets can be added to implementation tasks without leaving this tab. To add targets to an implementation task, the task details twisty arrow must be opened to display enhanced view of each task, schedule for that task and the target list with a select button in the Task Targets area. When the **Select** button is selected, a dialog with a list of targets from the Change is displayed. The user selects one or more targets from this list and these targets are automatically added to the implementation task.

Figure 3-103 shows the selection of CIs for the implementation tasks.

Configuration Item Number	Configuration Item Name	Description	Asset	Site	Status
☐					OPERATING
☒ 10001	MIALINUX01	Miami Linux Server			OPERATING
☐ 10002	MIWINAD01	Miami Active Directory Server			OPERATING
☐ 10003	MIALINUXDB2	COMPUTER			OPERATING
☐ 10004	MIALINUXWAS01	Miami Linux WAS server			OPERATING
☐ 10005	MIWINSQL01	Miami Windows SQL Server			OPERATING
☐ 10006	WEBSPPHERE ND	WebSphere app on Miami Linux WAS server			OPERATING
☐ TPT400	THINKPAD T400	COMPUTER,Laptops, 4GB, 2.26MHZ			OPERATING
☐ WMOUSE	WIRELESS MOUSE	COMPUTER			OPERATING
☐ DB2 INSTANCE	MIAMI DB2	Software Image, DB2, 9.5			OPERATING

Figure 3-103 CI Selection for Tasks

As target CIs are added to the implementation tasks, the outage specification must be set. The outage specification indicates the degree of service outage caused by the implementation work on that target. During technical impact analysis, the SME must have identified the outage level caused by the implementation work. This can be determined by delving into the details of the implementation, for example, by reading installation notes provided with a software update to determine whether the server needs to be taken down during the update. Also, the current IT configuration needs to be examined during technical impact analysis in order to understand the interaction between the implementation work and availability of the service being modified. For example, if only one server in an application server cluster will be affected by the implementation work, then the service might not have an outage, simply a performance degradation.

The outage specification values are an ALN (alphanumeric) domain, so customers can define the specification values that meet their business needs.

Figure 3-104 points out the Outage field.

Configuration Item Number	Business Impact	Asset	Location	Outage	Assessment Done
10004				Offline	☐

Figure 3-104 Outage Field for Task Target

An Assessment Done check box is provided in the Task Target table so the Change Analyst can indicate that they have created all the implementation tasks needed to satisfy the work described in the implementation note. This is meant to be used as a reminder to indicates which ones still need to be translated into implementation tasks.

After Tasks are scheduled, they will also show up on the Impacts Tab under the Manage Impacts section as seen in Figure 3-105.

SummaryManage Impacts

In order to reduce unplanned outages, use the tables below to identify CIs or Assets impacted by each implementation task.

Implementation Tasks for Change 1161

Filter > < < > > > > 1 - 4 of 4

Download ?

Choose a task from the table, then use the button below to identify CIs or Assets impacted by the chosen task.

Task	Task Id	Description	Estimated Duration	Scheduled Start	Scheduled Finish	Status
>> 1166	10	Assessment	2:00	20	20	WAPPR
>> 1167	20	Approval	1:00	20	20	WAPPR
>> 1168	30	Implement the Change	10:00	20	20	WAPPR
>> 1169	40	Post Implementation Review	2:00	20	20	WAPPR

Impacts for All Targets of Selected Task

Targets for Task 1166-10

Filter > < < > > > > 1 - 1 of 1

Download ?

This table contains targets associated with the task selected above. Use the button below to identify the impacts for the selected target.

Configuration Item Number	Business Impact	Asset	Location	Target Description	Outage	Assessment Done
>> 10004					Offline	☐

Impacts for Selected Target

Impacts for Task 1166-10

Filter > < < > > > > 1 - 1 of 1

Download ?

This table contains impacts associated with the task selected above. Use the button below to identify additional impacts.

Configuration Item	Business Impact	Asset	Target CI Causing Impact	Target Asset Causing Impact	Assessment Done	Approval	Notify
>> 10006			10004		☐	☒	☒

Select Impacts

Figure 3-105 Implementation Tasks for Change

Impact for Tasks sub-section

Impacted CIs are associated with implementation tasks. There are a couple of reasons that the impacted CIs are scoped to the implementation task rather than just the Change. First, the implementation work done when completing an implementation task might affect service availability. Also, the time when the impact will occur is very important. The time of impact is the time when the implementation work is being done, so it is the scheduled dates on the implementation task that indicate the time of the impact, not the dates of the encompassing Change.

There are three ways to identify an impacted CI:

- ▶ Select the CIs that are related to an implementation task target CI.
- ▶ Select any CI from the CMDB.
- ▶ Select the CIs that are related to a previously identified impacted CI.

To select an impacted CI based on its relationship to a target of the implementation task, proceed as follows:

1. Select an implementation task from the list.
2. Select a target of that implementation task.
3. Use the **Impacts for Selected Target** button on the task target table.

This button will present a dialog listing all the CIs that have a relationship defined in the CMDB with the selected target CI. One or more CIs can be selected from this list. It might be that even though a CI does not have a relationship defined in the CMDB with one of the implementation targets, a Change Analyst might know it will be impacted. In this situation, use the **Select Impacts → Select from CIs** button on the impacted CI table to identify any CI from the CMDB as an impacted CI. It will present a dialog that can be used to select any CI from the CMDB using filter criteria. The CIs selected from the dialog will be added to the impacted CI list for the highlighted implementation task.

3.3.7 Change Management Schedule

The Change Management Schedule is the functionality in CCMDB that permits viewing Changes that have been authorized for implementation and scheduling.

During the assessment step, this function helps check conflicts between new tasks and ones that are already scheduled.

The Change Management Schedule also can be used anytime for anyone who wants to know what tasks are scheduled.

Reasons to use the Change Management Schedule

Following are reasons that you might need to use the Change Management Schedule:

- ▶ A CI owner wants to see all Changes scheduled over a time period on a particular CI of interest. The CI owner can be a business application owner or an infrastructure component owner.
- ▶ A Change Manager wants to see all Changes scheduled this weekend. Secondary use cases include critical Changes, late Changes, and so on.
- ▶ A Change Manager wants to see all business applications affected by Changes this weekend.
- ▶ A Change Manager wants to see all business applications impacted by Changes this weekend.

It is possible to view tasks for Changes in six ways, as we discuss next:

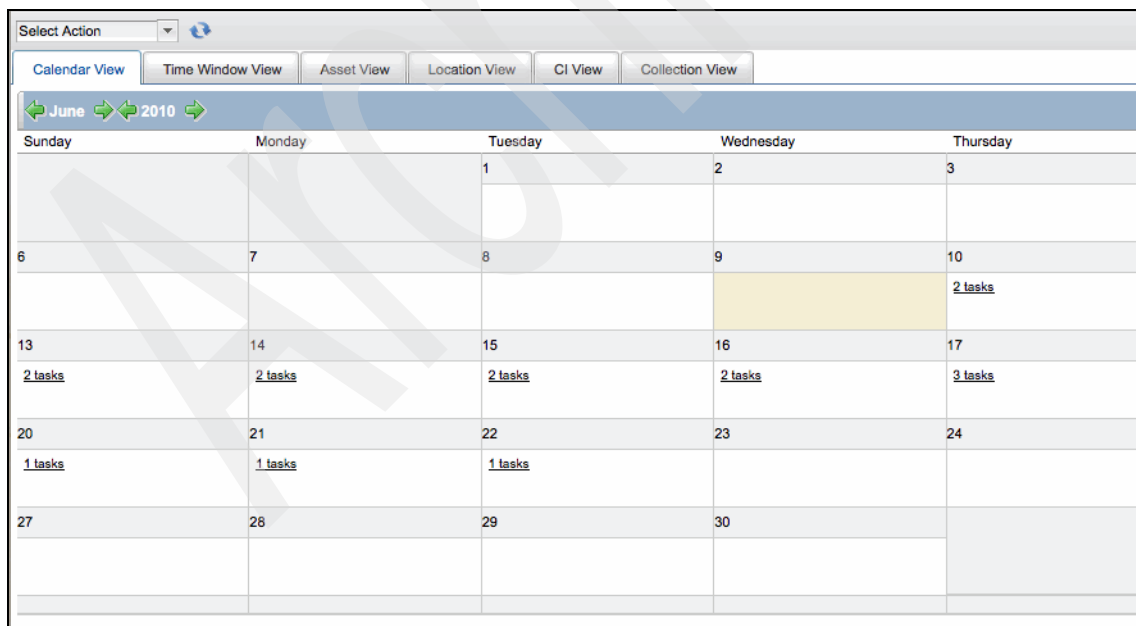
- ▶ Calendar view
- ▶ Time window view
- ▶ CI view
- ▶ Collection view
- ▶ Location View
- ▶ Asset View

Calendar View

This view shows the number of implementation tasks scheduled for each day on the calendar. If Change Windows were defined for CIs, the view will display a conflict icon on each day where the scheduled tasks do not conform to a CIs Change Window. To view the details of the tasks scheduled for a day, click the number displayed on that day in the calendar. A dialog box is displayed showing the scheduled start and end times, the Change number and description, the owner of the Change, and a description of the task. A few of these items will be links, which makes it possible open related records.

The Calendar View is broken down by single tasks and does not only list the Changes itself. Select **Go to** → **Change** → **Change Implementation Schedule** to open the Change Implementation Schedule and Select Calendar View tab.

Figure 3-106 shows the Calendar View of the implementation Schedule.



The screenshot shows a software interface for viewing a calendar. At the top, there is a 'Select Action' dropdown menu and a refresh icon. Below this is a row of tabs: 'Calendar View' (selected), 'Time Window View', 'Asset View', 'Location View', 'CI View', and 'Collection View'. Under the 'Calendar View' tab, there is a navigation bar with 'June' and '2010' and arrows for navigation. The main area is a calendar grid for June 2010. The days of the week are listed at the top: Sunday, Monday, Tuesday, Wednesday, Thursday. The calendar shows dates from 1 to 30. Some dates have a number of tasks listed below them, such as '2 tasks' on June 10, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 27, 28, 29, and 30. A yellow highlight is visible on the date June 10.

Sunday	Monday	Tuesday	Wednesday	Thursday
		1	2	3
6	7	8	9	10
				2 tasks
13	14	15	16	17
2 tasks	2 tasks	2 tasks	2 tasks	3 tasks
20	21	22	23	24
1 tasks	1 tasks	1 tasks		
27	28	29	30	

Figure 3-106 Calendar View

The tasks scheduled for a day are shown in Figure 3-107. They are obtained by clicking a particular day and the link to the tasks on that day.

Task	Task Description	Status	Conflicts	Process	Process Description
1166	Assessment	WAPPR	☐	1161	Patch Content Manager
1168	Implement the Change	WAPPR	☐	1161	Patch Content Manager
1169	Post Implementation Review	WAPPR	☐	1161	Patch Content Manager

Figure 3-107 Daily Task List

Time Window View

The Time Window View shows the number of scheduled implementation tasks each day during a period specified. All tasks scheduled for the time window are listed, with basic information for each task.

Select **Go to → Change → Change Implementation Schedule** to open the Change Implementation Schedule application.

Select the **Time Window View** tab, and specify the window start time and end Time

Figure 3-108 shows the tasks in the Time Window View for the implementation schedule.

Task	Task Description	Status	Conflicts
1166	Assessment	WAPPR	☒
1167	Approval	WAPPR	☒
1168	Implement the Change	WAPPR	☒
1169	Post Implementation Review	WAPPR	☒

Figure 3-108 Time Window View: Implementation Schedule

By default, the Time Window View shows the implementation schedule by tasks but it is also possible to view by Target CIs and Additional Impacted Tasks.

Figure 3-109 shows the targets in the Time Window View for the implementation schedule.

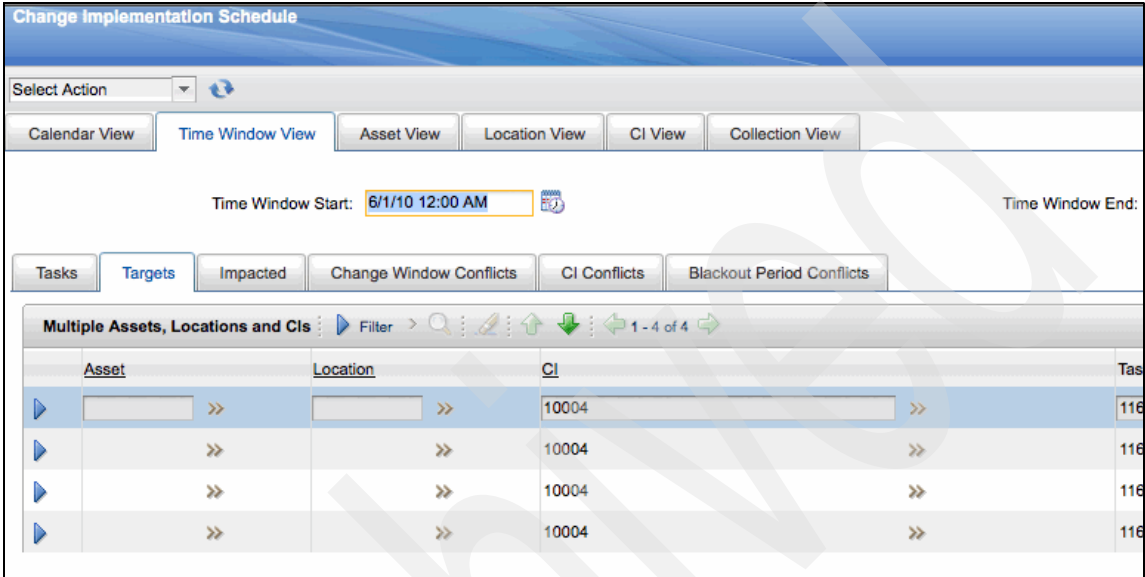


Figure 3-109 Time Window View by Targets

Figure 3-110 shows the Change Implementation Schedule: Time Window View by Additional Impacted CIs.

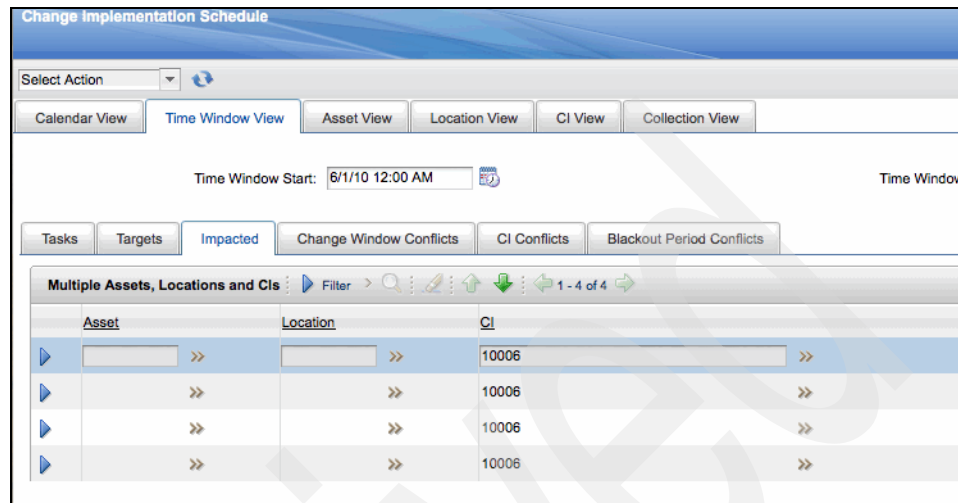


Figure 3-110 Time Window View by Impacted CIs

Change implementation (CI view)

This view provides fields in which to filter the view by a specific CI, and by a time range for which to view information about conflicts that involve that CI.

Open the **Tasks Targeting** tab to view the implementation tasks that have the specified CI as a target.

Open the **Tasks Impacting** tab to view tasks that impact the specified CI. The Change Window Conflicts, CI Conflicts, and Blackout Period Conflicts tabs display the same information that is displayed in the Tasks dialog, but for the specified CI only.

Select **Go to → Change → Change Implementation Schedule** to open the Change Implementation Schedule application.

Select the **CI View** tab and enter a CI number, time window start, and time window end.

Figure 3-111 shows the Change Implementation Schedule: CI View.

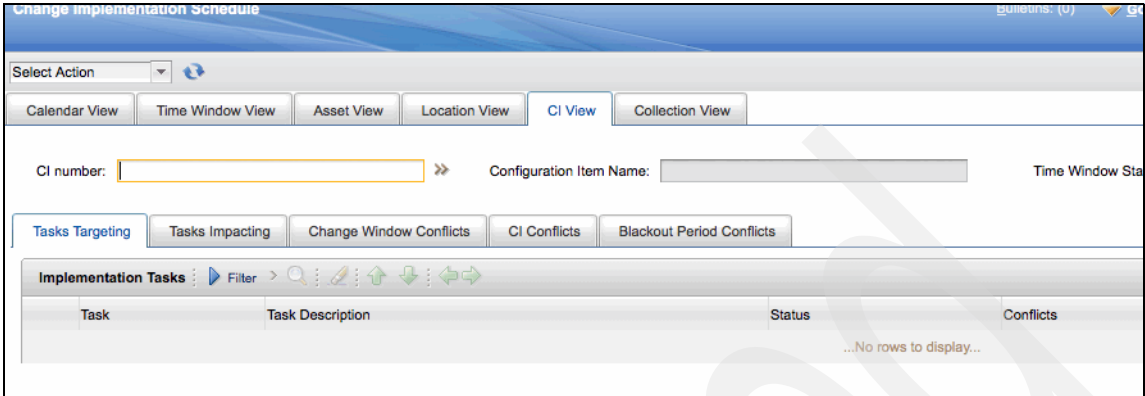


Figure 3-111 Implementation Schedule: CI view

By default, the implementation schedule by tasks targeting the CI is shown, but it is also possible to view by tasks impacting a CI.

Figure 3-112 shows the impacted CIs view.

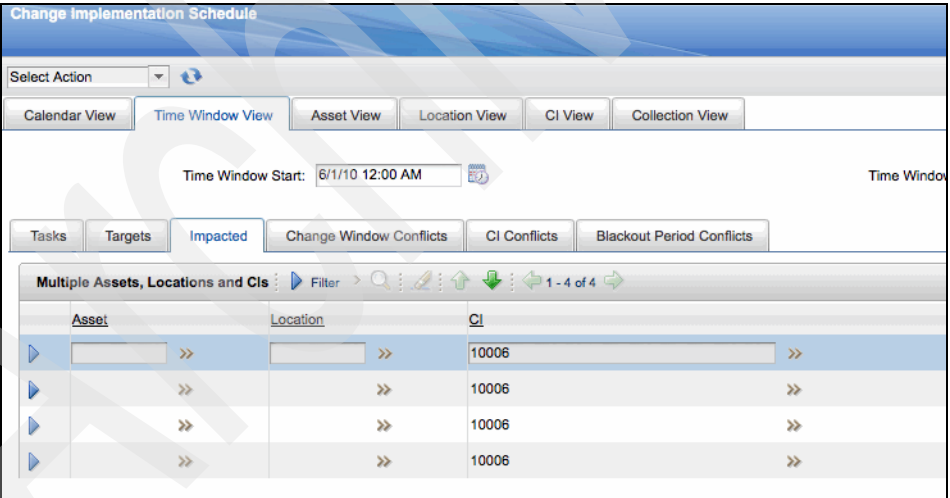


Figure 3-112 Impacted CI view for Implementation Schedule

CI Collection View

This view provides fields in which to filter the view by a collection of affected CIs, and by a time range for which to view information about conflicts that involve that CI. Open the Tasks Targeting tab to view the implementation tasks that have the specified CI collection as a target.

Open the **Tasks Impacting** tab to view tasks that impact the specified CI collection. The Change Window Conflicts, CI Conflicts, and Blackout Period Conflicts tabs display the same information that is displayed in the Tasks dialog, but for the specified CI collection only.

Select **Go to → Change → Change Implementation Schedule** to open the Change Implementation Schedule application.

Select the **CI Collection View** tab and enter a collection number, time window start, and time window end.

Figure 3-113 shows the Change Implementation Schedule: CI Collection View

The screenshot shows the 'Change Implementation Schedule' application window. At the top, there's a blue header bar with the title 'Change Implementation Schedule' and a 'Bulleins: (0)' indicator. Below the header, there's a 'Select Action' dropdown menu and a refresh icon. A row of tabs includes 'Calendar View', 'Time Window View', 'Asset View', 'Location View', 'CI View', and 'Collection View' (which is currently selected). Below the tabs, there are input fields for 'Collection Number' (with a search icon), 'Time Window Start' (set to '6/1/10 12:00 AM'), and 'Time Window End'. A row of sub-tabs includes 'Tasks Targeting', 'Tasks Impacting', 'Change Window Conflicts', 'CI Conflicts', and 'Blackout Period Conflicts'. Below these, there's a section for 'Implementation Tasks' with a 'Filter' button and several icons. At the bottom, there's a table with columns: 'Task', 'Task Description', 'Process', 'Asset', and 'Location'. The table is currently empty, with a message at the bottom right stating '...No rows to display...'.

Figure 3-113 CI Collection View for Implementation Schedule

Location View

This view provides the Location field in which to specify the location, along with Time Window Start and Time Window End fields in which to define a time range for which to view implementation tasks.

After you specify a location and a time range, the Tasks Targeting view displays a read-only table with a row for each task whose schedule overlaps the specified start and end times and for which this location is a target. The Tasks Impacting view displays a read-only table with a row for each task whose schedule overlaps the specified start and end times and by which this location is impacted. In each view, you can click links to access details.

Select **Go to → Change → Change Implementation Schedule** to open the Change Implementation Schedule application.

Select the **Location View** tab and enter a collection number, time window start, and time window end.

Figure 3-114 shows the Location View of the implementation schedule.

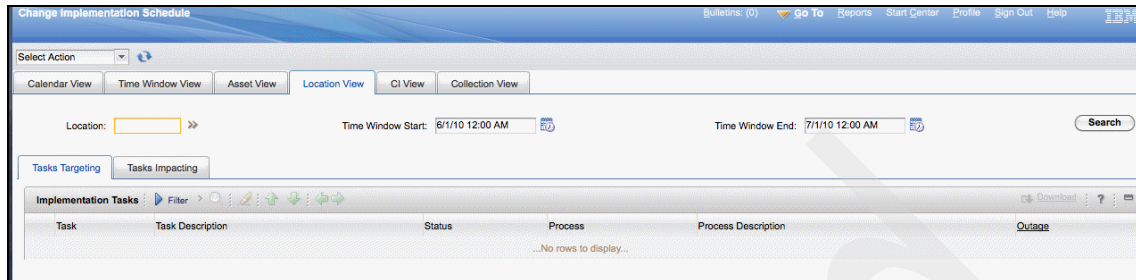


Figure 3-114 Location View for Implementation Schedule

Asset View

This view provides the Asset field in which to specify the asset number, along with Time Window Start and Time Window End fields in which to define a time range for which to view implementation tasks.

After you specify an asset and a time range, the Tasks Targeting view displays a read-only table with a row for each task whose schedule overlaps the specified start and end times and for which this asset is a target. The Tasks Impacting view displays a read-only table with a row for each task whose schedule overlaps the specified start and end times and by which this asset is impacted. In each view, you can click links to access details.

Select **Go to** → **Change** → **Change Implementation Schedule** to open the Change Implementation Schedule application.

Select the **Asset View** tab and enter a collection number, time window start, and time window end

Figure 3-115 shows the Asset View of the Change Implementation Schedule.

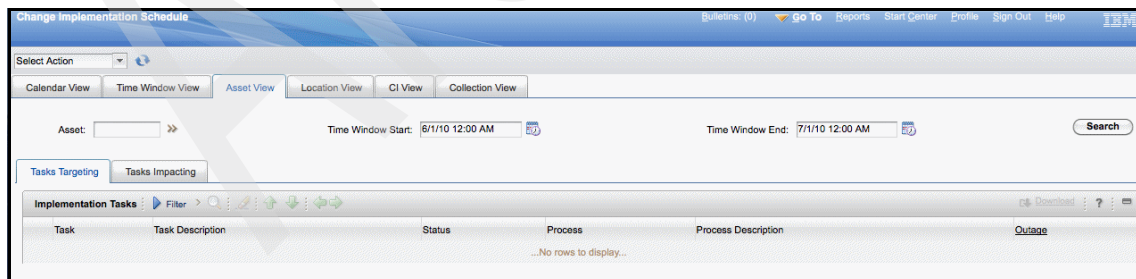


Figure 3-115 Asset View for Implementation Schedule

Running the Scheduler for all tasks

The scheduler takes blackout periods, Change Windows, and other criteria into account when it looks for a scheduling solution. Therefore, before you accept the assignment to schedule the implementation tasks, ensure these prerequisites:

- ▶ Blackout periods and associated CIs are defined.
- ▶ Change Windows are defined, along with CIs that can be taken out of service during the Change Windows.
- ▶ You have identified target and impacted CIs for each of the implementation tasks that are to be scheduled.
- ▶ You have identified any predecessor tasks for each implementation task that is to be scheduled. If a job plan was applied to the Change, this information is attached to the tasks that are in the job plan; if you have created tasks in addition to those in the job plan, ensure that predecessor tasks are specified. Certain tasks do not have predecessor tasks.
- ▶ Each task to be scheduled has an estimated duration. If a job plan was applied to the Change, this information is attached to the tasks that are in the job plan; if you have created tasks in addition to those in the job plan, ensure that you set estimated durations. If you do not set a task duration, the scheduler assumes that the duration of the task is one minute. If you are not sure how long a task will take to complete, specify your best guess.
- ▶ (Optional) An owner is assigned to each task to be scheduled, and owner work shifts are defined.

From within the Change on the Schedule Tab, you can see the tasks that are scheduled for your Change. From the Scheduler Project box, you can launch the Scheduler program by choosing Create a Project Schedule.

Figure 3-116 shows how to invoke the Scheduler program.

The screenshot displays the IBM Changes application interface. At the top, there's a navigation bar with tabs: List, Change, Assessments, Impacts, Authorization, Schedule, Workplan Map, Related Records, Actuals, and Log. The 'Schedule' tab is active. Below the navigation bar is a 'Progress Map' section with a sequence of steps: ACC_CAT, ASSESS, AUTH, SCHED, IMPL, INPRG, COMP, REVIEW, and CLOSE. The main form area contains various input fields for scheduling a change. A red box highlights the 'Scheduler Project' dropdown menu, which is open, showing options: 'Select Value', 'Go To Scheduler', 'Clear', and 'Create a Project Schedule'. The form also includes fields for 'Change' (MYCHANGE), 'Update payroll system', 'Supervisor', 'Status' (WAPPR), 'Job Plan', 'Response Plan', 'Requires Scheduling?' (checked), 'Fully Automated?' (unchecked), 'Parent', 'Predecessors', 'Target Start', 'Target Finish', 'Estimated Duration' (0.00), 'Start No Earlier Than', 'Finish No Later Than', 'Sched Start', 'Sched Finish', 'Actual Start', 'Actual Finish', 'Time Remaining', 'Include Tasks in Schedule?' (checked), and 'Remediation Plan'. At the bottom, there's a 'Children of Change MYCHANGE' section with a table of tasks. The table has columns: Sequence, Task, Summary, Estimated Duration, Status, Owner, and Owner Group. One task is listed: Sequence 10, Task 'Install DB patch', Summary 'Install DB patch', Estimated Duration 2.00, Status WAPPR, Owner, and Owner Group. Below the table is a 'Task Information' section for the selected task, showing 'Task: 10', 'Install DB patch', 'Under Flow Control?' (checked), and 'Attachments'.

Figure 3-116 Invocation of Scheduler Program

The Project Schedule is then created and you are brought to the summary display for that project schedule.

Figure 3-117 shows the initial summary page as a project Schedule is created.

The screenshot displays the IBM Tivoli Scheduler interface. At the top, there's a navigation bar with 'List', 'Schedule', and 'Gantt View' tabs. The 'Schedule' tab is active. Below the navigation bar, there's a form for creating or editing a schedule. The 'Schedule' field is set to 'CHANGE_9' and the 'Update payroll system' task is selected. Other fields include 'Calendar', 'Shift', 'Organization', 'Start Date', 'Public?' (checked), 'Last Saved Date' (5/13/10 14:48:32), 'Last Committed Date', 'Created By' (MAXADMIN), and 'Created Date' (5/13/10 14:48:32). There are also checkboxes for 'Preserve Schedule Start Dates?' (checked) and 'Refresh When Opening Gantt View?' (unchecked). Below the form, there's a section for 'Work Queries' with a table showing a single query named 'CHANGE_9' with a description 'Update payroll system' and a 'Where Condition' 'wonus=MYCHANGE'. Below this, there's a section for 'Additional Resources' and a section for 'Resource Availability' which currently shows '...No rows to display...'. The bottom of the window shows a status bar with 'Changes > Scheduler'.

Figure 3-117 Project Schedule Creation

The upper left pane of this display contains the work records and their subordinate tasks. When a task is selected, the resource constraints for it are displayed in the constraints view in the lower left quadrant of the window. There is shading on both the Gantt and Resource Constraints chart.

The gray sections identify times where the constraints are violated; the white areas are times when the constraints are met. The green bars in the Resource Constraints chart represent ranges of time where the resources for that specific task are available; in contrast, the red bars show areas where the constraints are violated.

One way to manually schedule a Change is to simply drag the Gantt chart slider to a non-shaded area as shown in Figure 3-118. If the slider turns from red to blue, the constraints are met and the Change can be scheduled for that time. For manual scheduling, expanding the view by zooming out helps when aligning the task with the available time periods.

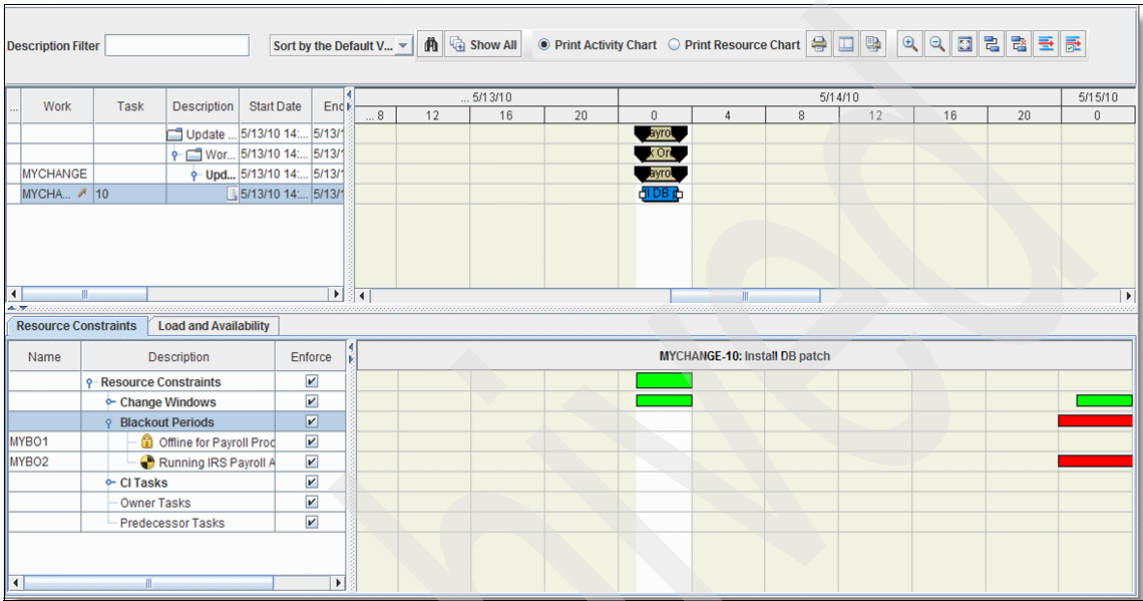


Figure 3-118 Project Schedule Gantt Chart

Another way to manually schedule a Change is to right-click a green bar. The Use for Schedule pop-up is displayed. When you click this, the task will be moved to this period. Figure 3-119 shows this one way of manually changing items in the Schedule.

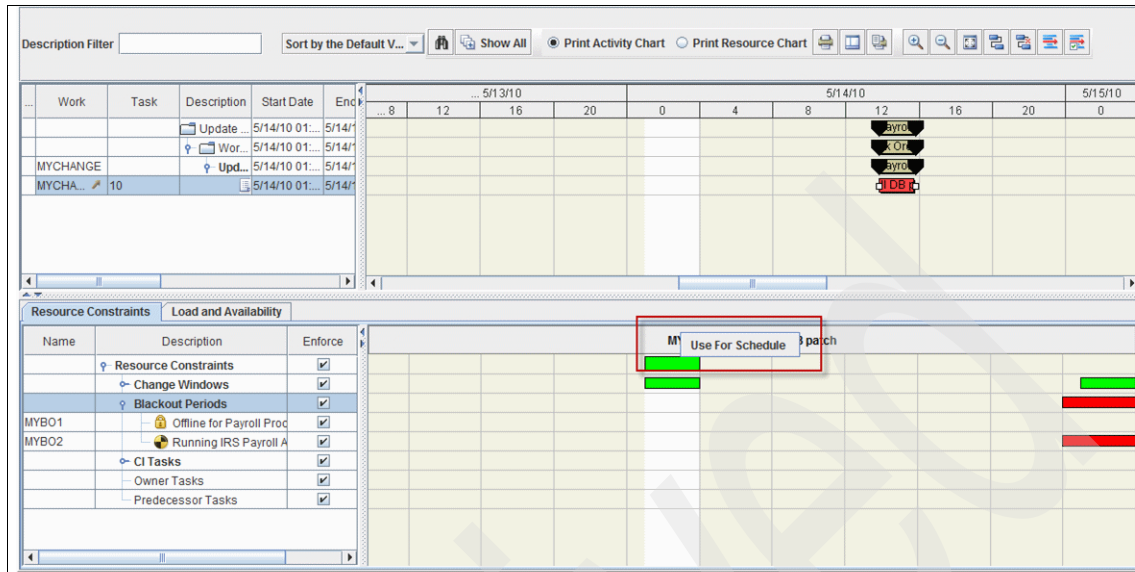


Figure 3-119 Rescheduling Items in Gantt Chart

To automatically schedule a Change, choose **Perform Critical Path Method** on All Rows. This will automatically find the next available time window available for this Change. Figure 3-120 points out the button used to automatically schedule the project to fit open time windows.

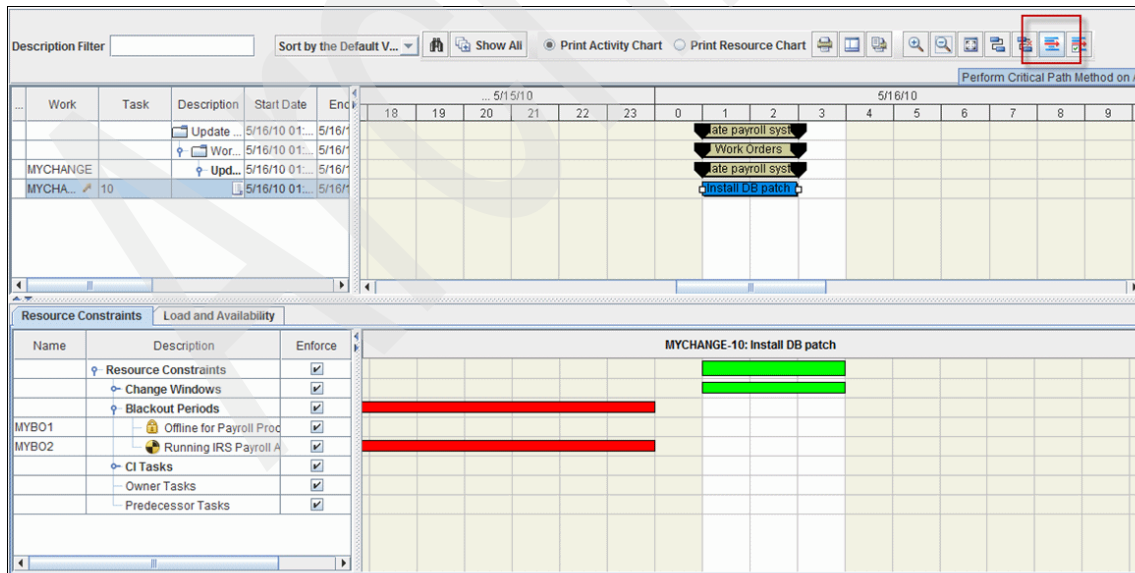


Figure 3-120 Automatically Scheduling Items in Gantt Chart

After the Change has been scheduled, click the **Commit** button on the toolbar to accept the project schedule for your Change. Figure 3-121 shows the button used to commit your Changes and save the project schedule to the Change itself.

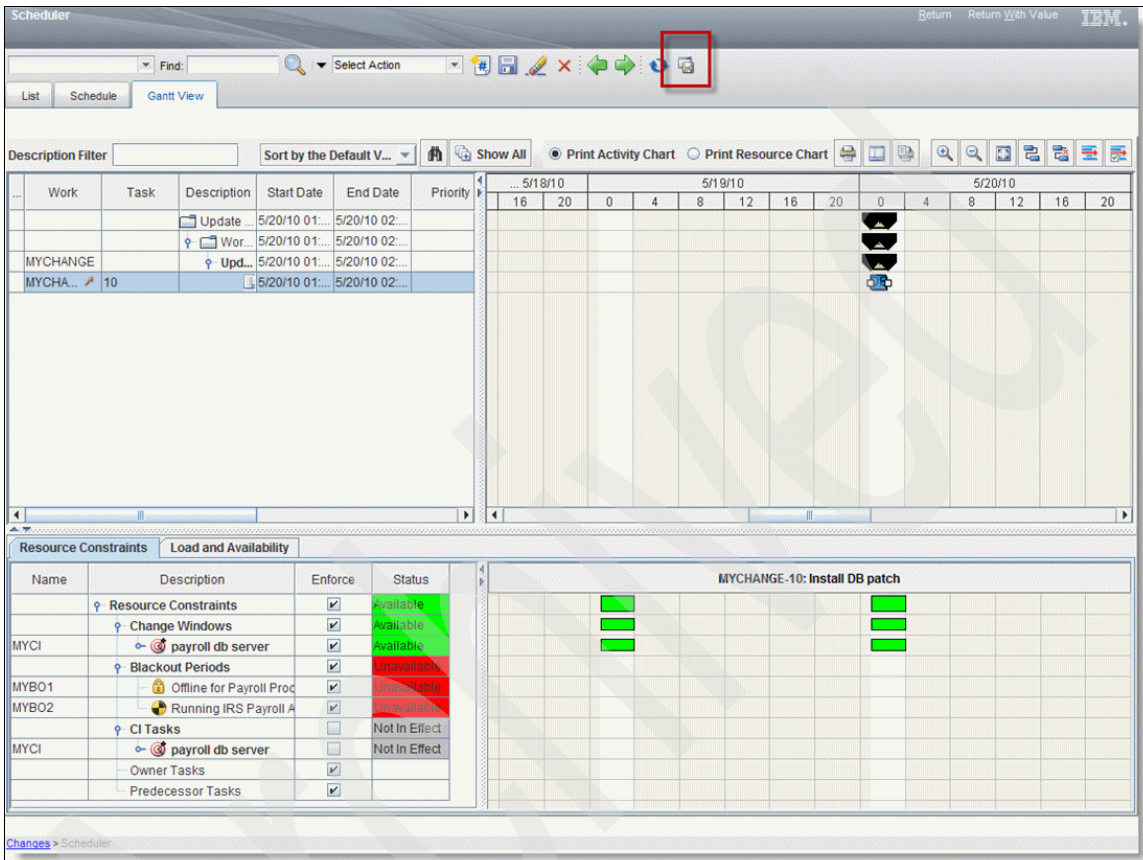


Figure 3-121 Submitting Final Schedule to Change

The scheduler will now have accurately updated your Change's scheduled start and end time as well as assigned it the name of the project schedule. You can now get to this schedule as well by choosing **Go To → Planning and Scheduling → Scheduler**.

Figure 3-122 shows the Change updated with the project schedule details.

The screenshot displays the IBM Tivoli CCM 'Changes' application. The 'Schedule' tab is active, showing a progress map with steps: ACC_CAT, ASSESS, AUTH, SCHED, IMPL, INPRG, COMP, REVIEW, and CLOSE. The 'SCHED' step is highlighted. Below the progress map, the 'Change' field is set to 'MYCHANGE' with the description 'Update payroll system'. Other fields include 'Supervisor', 'Status' (WAPPR), 'Job Plan', 'Lead', 'Site' (AUSTIN), 'Response Plan', 'Work Group', 'Remediation Plan', 'Requires Scheduling?' (checked), 'Fully Automated?' (unchecked), 'Parent', 'Predecessors', 'Target Start', 'Target Finish', 'Estimated Duration' (1:30), 'Start No Earlier Than', 'Finish No Later Than', 'Time Remaining', 'Include Tasks in Schedule?' (checked), 'Actual Start', 'Actual Finish', and 'Scheduler Project' (CHANGE_10). A red box highlights the 'Scheduled Start' (5/20/10 01:00:00) and 'Scheduled Finish' (5/20/10 02:30:24) fields. Below the form, the 'Children of Change MYCHANGE' section shows a table of tasks for 'MYCHANGE'.

Sequence	Task	Summary	Estimated Duration	Status	Owner	Owner Group
10	Install DB patch		1:30	WAPPR		

The 'Schedule Conflicts' section at the bottom shows 'Summary' and 'Tasks With Conflicts'.

Figure 3-122 Updated Change with Project Schedule details

3.3.8 Change Window

The Change Window is a central repository for negotiated maintenance windows, to be used to define when CIs can be taken out of service to have Changes made, or to modify or delete existing Change Windows.

It allows repeating and custom scheduling of Change Windows within the Change Window calendar, for example, Daily, Weekly, Monthly, and so on.

A Change Window Calendar can contain zero, one, or more Change Windows (Daily, Weekly, Monthly, Annual, or Custom/Ad-Hoc). A CI can be linked to one Change Window Calendar.

Figure 3-123 illustrates the Change Window concept.

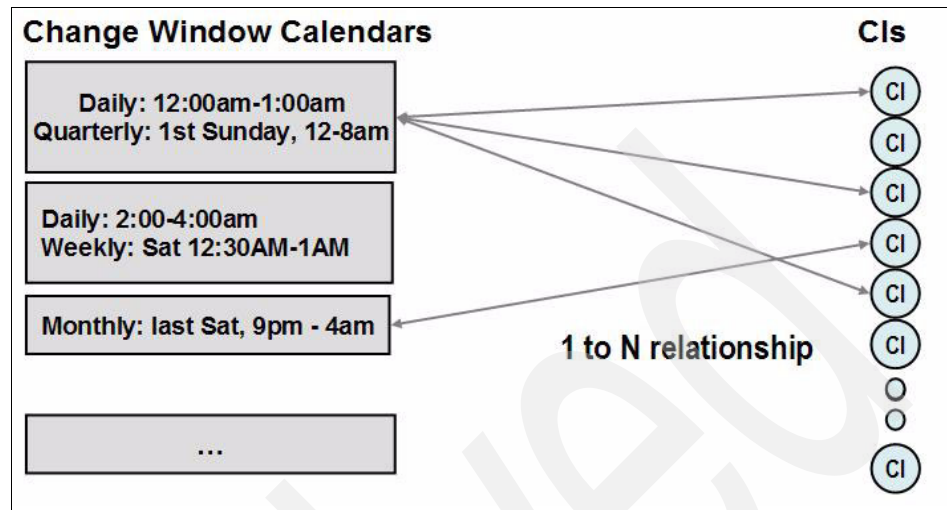


Figure 3-123 Change Window concept

Working with the Change Window

Follow these instructions to work with the Change Window:

1. Select **Change Application** → **Change Window Calendars** to open the Change Window application.
2. Click the **Change Window Calendar** tab to work with a calendar view or the **Change Window Schedule** tab to work with a tabular view of existing Change Windows.
3. To add a new Change Window, click **New Row** in either view. A dialog box is displayed in which you can specify the type, date, start, and stop times, and any notes you want to enter.

While using either of these views, you can also open a Change Window to modify it, or delete a Change Window. You can also schedule repeating Change Windows from the main Change Window Calendar and selecting the Action Schedule Change Windows.

Figure 3-124 shows the user using the Change Window Calendar and then using the select action dialog to Schedule new Change Windows.

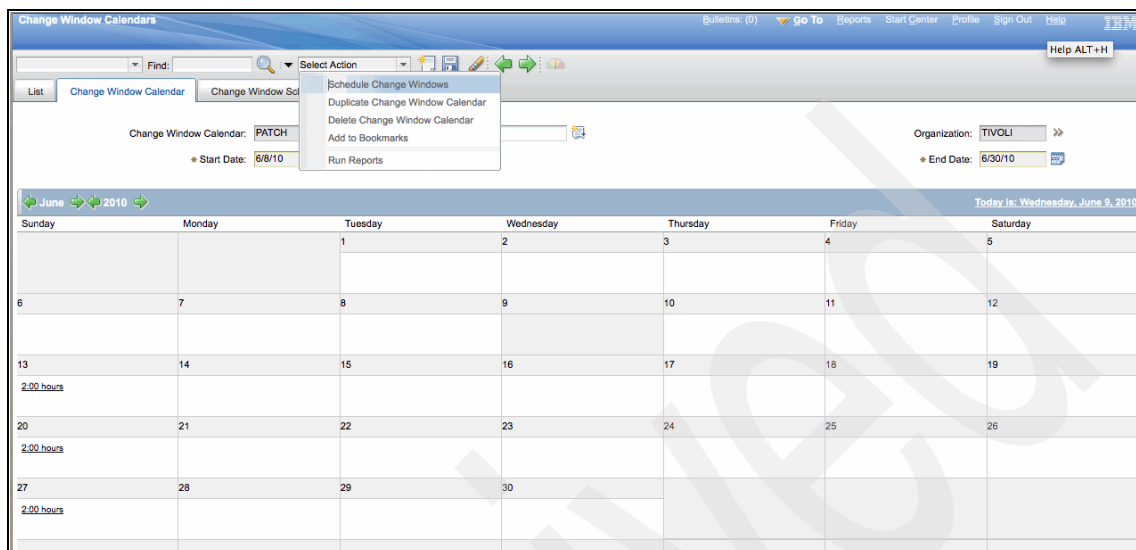


Figure 3-124 Change Window Calendar

The following dialog box will then pop up, allowing for more complex scheduling of Change Windows as seen in Figure 3-125.

The 'Set Schedule' dialog box is titled 'Set Schedule' and contains the following fields and options:

- Select a date interval and then Preview to see the dates.**
- Type:** WEEKLY
- * Duration:** 2:00
- * Start Date:** 6/8/10
- * End Date:** 6/30/10
- Notes:** (empty text field)
- Scheduling Options:**
 - ☐ Every [] day(s), at time []
 - ☒ Every **1** week(s), on day **Sunday** at time **1:00 AM**
 - ☐ Every [] month(s),
 - ☐ on day **01** at time []
 - ☐ on the **first** **Sunday** of the month, at time []
 - ☐ Every [] year(s),
 - ☐ on **January** on day **01** at time []
 - ☐ on the **first** **Sunday** of the year, at time []
- Date Preview:** A list showing dates and times: 6/13/10 1:00 AM, 6/20/10 1:00 AM, 6/27/10 1:00 AM.
- Buttons:** OK, Cancel, and a Preview button above the Date Preview list.

Figure 3-125 Complex Scheduling of Change Windows

Change Window Conflicts function

The Change Window Conflicts function permits identification of implementation tasks whose schedules do not conform to the Change Windows for the configuration items they will affect. Configuration items must have defined Change Windows to detect conflicts.

This function is enabled when viewing the Change Implementation Schedule in the Time Window view, CI view, or CI Collections view.

To use this function, click **Show Change Window Conflicts** as seen in Figure 3-126.

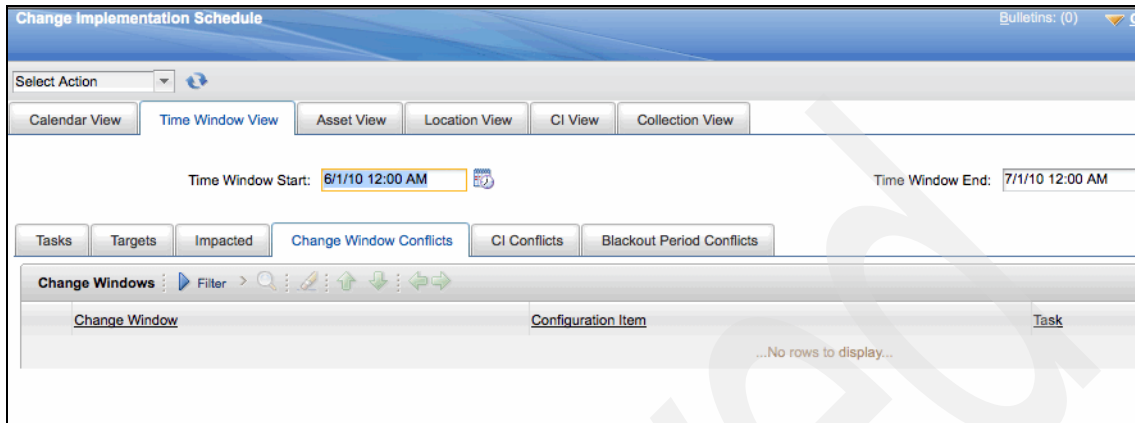


Figure 3-126 Change Window Conflicts

3.3.9 Tracking the progress of a Change

The progress of a Change is tracked through the following fields:

- ▶ Progress: This field shows what phase of the cycle is the record is in and its overall progress.
- ▶ Status: This field shows the sequence of the status.

The progress and status of a Change must be updated after each completed phase. Figure 3-127 lists the available Changes and their status.

Changes

Bullets: (0)Go ToReportsStart CenterProfileSign OutHelp

Find:

Select Action

</

Figure 3-127 Available Change List

After choosing a Change from the available list of Changes, you will see the Change detail information window as seen in Figure 3-128.

The screenshot shows the 'Changes' application interface. At the top, there is a navigation bar with tabs: List, Change, Assessments, Impacts, Authorization, Schedule, Workplan Map, Related Records, Actuals, and Log. The 'Change' tab is selected. Below the navigation bar is a 'Progress Map' section showing a sequence of steps: ACC_CAT, ASSESS, SCHED, AUTH, IMPL, INPRG, COMP, REVIEW, and CLOSE. The 'IMPL' step is highlighted. Below the progress map, there are fields for Change: 1161, Owner: MAXADMIN, Owner Group: (empty), Status: IMPL, and Attachments. The 'Change Details' section contains several fields: Summary: Patch Content Manager, Change Type: Normal, Reason for Change: Bug, Effect Of Not Implementing: Indexing Wont work, Verification Plan: Test Indexer, Remediation Plan: (empty), Risk: 2, Failure Probability: 2, Impact: 1, Urgency: 4, Priority: 4, and Change Category: Major. At the bottom, there is a 'Classification' field with the value PMCHG \ PMCHGSFW \ MIDDLEWARE \ JZEEMC and a 'Class Description' field with the value JZEE Middleware Database.

Figure 3-128 Change Detail Information

The progress of a Change can be updated in two ways:

1. Automatically: The Change Management job plan includes an automated task to update the progress value when certain activities are completed. See 3.1.3, “Job plan” on page 87, which explains how to configure it.
2. Manually: The Change Owner can modify the progress value using the Change Progress action from the Select Action menu.
 - a. Select **Go to** → **Change** → **Change**.
 - b. With the Change selected in the list, or while viewing the details of the Change, select **Change Progress** from the Select Action menu.
 - c. Choose a progress value from the list.
 - d. If necessary, enter a comment related to the modification
 - e. Click **OK**.

Figure 3-129 shows how to change the status of a Change to Accepted or Rejected as well as many others.

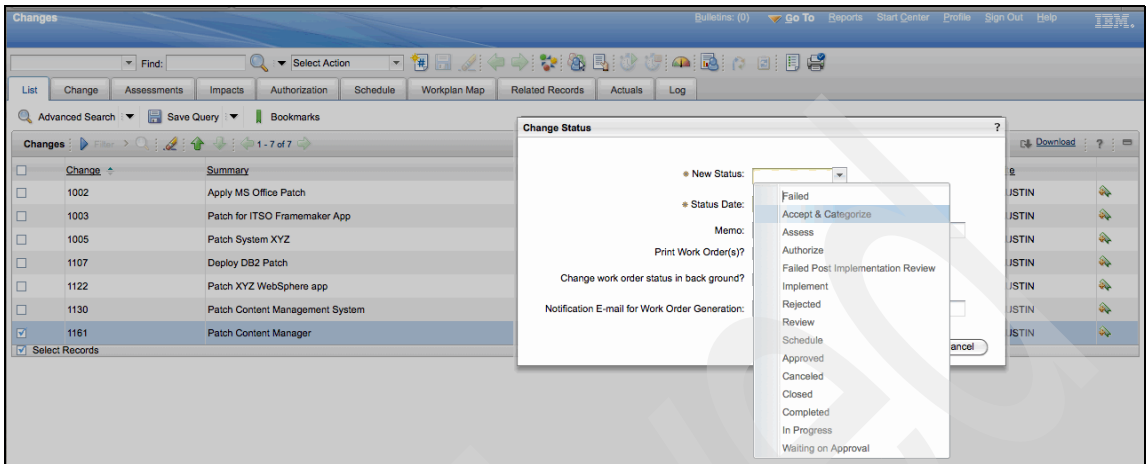


Figure 3-129 Changing status of a Change

The Changes application maintains a list of the modifications made to the progress value and the comments, if any, entered with each modification. To view the list, click **View** → **History**.

Figure 3-130 shows the history of modifications to the selected Change.

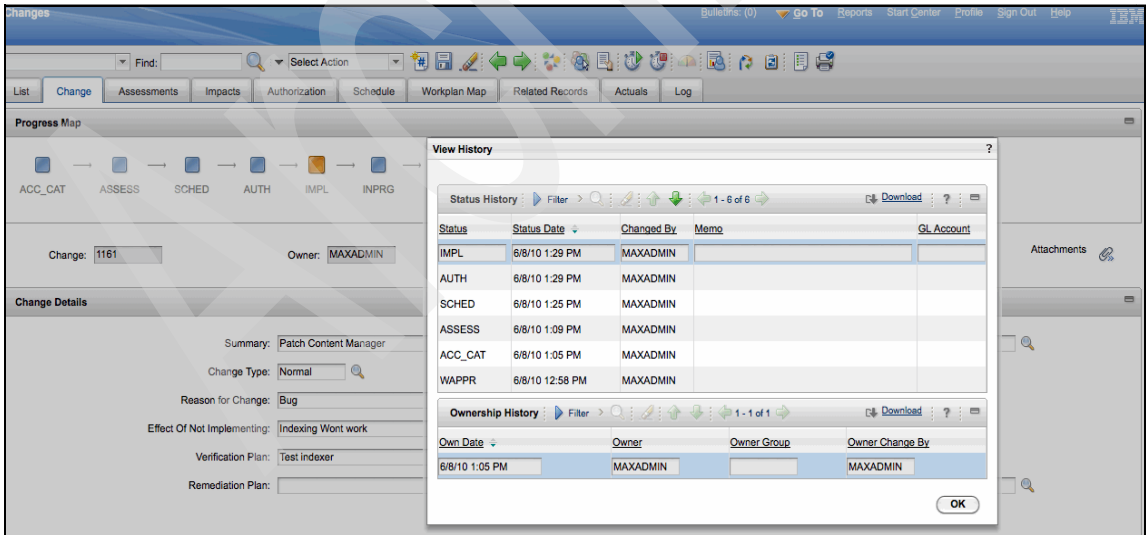


Figure 3-130 History of a Change

The list of values used for the Change progress attribute is stored in the PMCHGPROGRESS domain and can be modified using the Domains Application.

1. Select **Go to → System Configuration → Platform Configuration → Domains**.
2. Search and select the domain PMCHGPROGRESS.
3. The list of values available will be shown.
4. Add, edit, or delete values from the list as needed.

Figure 3-131 shows modifying the Change progress list in the domain application.

The screenshot shows the 'ALN Domain' application interface. At the top, the 'Domain' is set to 'PMCHGPROG' and the 'Data Type' is 'UPPER'. Below this, a table lists various change progress values. The table has columns for 'Value', 'Description', 'Organization', and 'Site'. The 'Value' column is highlighted, and the 'Description' column is also highlighted. The table contains 12 rows of data, including 'ACCEPTED', 'APPROVAL', 'APPROVED', 'ASSESSED', 'ASSESSMENT', 'CANCELED', 'CLOSED', 'FAILED', 'IMPLEMENTATION', and 'IMPLEMENTED'. At the bottom of the table, there are buttons for 'View/Modify Conditions' and 'New Row'. Below the table, there are 'OK' and 'Cancel' buttons.

Value	Description	Organization	Site
ACCEPTED	Accepted		
APPROVAL	Approval		
APPROVED	Approved		
ASSESSED	Assessed		
ASSESSMENT	Assessment		
CANCELED	Cancelled		
CLOSED	Closed		
FAILED	Failed		
IMPLEMENTATION	Implementation		
IMPLEMENTED	Implemented		

Figure 3-131 Change Progress List Modifications

3.3.10 Interaction with other processes

This section describes interactions between Change Management and two other popular PMPs: Configuration Management and Release Management.

Change and Configuration

The Change Management process uses the configuration items data provided by Configuration Management to perform the assessment of a Change, so data accuracy is very important to have a detailed analysis.

Otherwise, Change Management is the process that helps Configuration Management maintain the consistency of the CMDB, because Change Management controls the Changes that are made in the environment. Virtually all Changes modify CI attributes. These attribute Changes need to be reflected in the configuration database in order to have CI information available and up to date. It is important because other process use CI information. It allows the enterprise to be audit ready and avoid disparity between the CI data and the actual infrastructure.

Updating CI attributes after a Change implementation

Modifications to CI attributes must be specified in the **Move/Swap/Modify** option in the Select Action menu, and then a request must be opened from within the Change to have the CI records modified. When the modification is specified using this task, the CI is automatically updated when the Change is completed. For example, a Change that involves adding memory to a computer results in a modification to the memory attribute for the computer CI.

It is also possible to open a request using a Process Request application, but then the request is not associated with the Change record because the CI attributes update will occur in another time of the Change.

The Move/Swap/Modify dialog box provides other operations. To access a thorough description of this dialog box, along with instructions for performing Move/Swap/Modify tasks, click the question mark (?) in the upper right corner of the dialog box. This topic focuses on the modification of CI attributes.

Attention: If a Change was implemented and the database was not updated, the next discovery update will reflect the updated attribute values, but the Change in the CI record will not be associated to the corresponding RFC.

Procedural steps

The following steps must be performed by the Change Owner during the Change implementation in order to update the CI attributes:

1. With the Change open in the Changes application, select **Move/Swap/Modify** from the Select Action menu. The Move/Swap/Modify dialog box is displayed as shown in Figure 3-132.
2. Open the **Modify** tab, and then open the **Configuration Items** tab. The CIs section lists all of the CIs that are associated with the Change; CI specifications, including attributes, are displayed in the CI Specifications section.
3. With a CI row highlighted, select an attribute in the Specifications section that you want to modify, and click **Modify Attribute** to display the fields in which to modify the selected attribute. For example, type a new value for the Memory attribute if the Change adds memory to a computer.
4. After you have specified all of the CI attribute modifications that apply, click **Save as Plan**. Clicking **Save as Plan** causes the CI to be updated as specified when the Change moves to the COMPLETE status. Clicking **Execute Now** updates the CI now. Save as Plan is the preferred choice if you want to synchronize the Change and Configuration Management processes.

Figure 3-132 shows the move/swap/modify function for CI attribute Changes.

Move/Swap/Modify

Move/Swap **Modify** Users and Custodians

Asset Location **Configuration Item**

CIs Filter > 1 - 3 of 3

Configuration Item Number	Configuration Item Name
10004	MIALINUXWAS01
10003	MIALINUXDB2
10004	MIALINUXWAS01

CI Specifications Filter > 1 - 2 of 2

☐ Attribute	Description	Section	Data Type	Alphanumeric Value
☒ DATABASENAME			ALN	CTMGR
☐ INSTANCENAME			ALN	DBINST1

Planned Modifications Filter > 1 - 1 of 1

Configuration Item	Configuration Item Name	Attribute	Data Type	Old Value
10003	MIALINUXDB2	DATABASENAME	ALN	CTMGR

Figure 3-132 Move/Swap/Modify CI Attributes

5. From within the Change, create a process request to Configuration Management to update the CI. In the new process request (Figure 3-133), proceed as follows:
 - a. Specify a classification of PMCFGUR (CIs configuration update request).
 - b. Open the CHGRECD attribute.
 - c. In the Table Value field, type or select **YES**.

- d. In the Target CIs section, specify the CI that is modified, and save the request. When you specify the YES table value for the CHGRECD attribute, you are informing the Configuration Manager that receives the request that the modifications were specified in the Move/Modify/Swap dialog box. The Configuration Manager can then go to the Changes application and examine the modifications; in addition, the Configuration Manager will be aware that the CI was automatically updated as specified when the Change status moved to COMPLETE.

Figure 3-133 shows creating a process request to Configuration Management to update the CI.

The screenshot shows the 'Process Request' form with the following details:

- Process Request:** 1048
- Owner:** [Empty]
- Owner Group:** [Empty]
- Process State:** DRAFT
- Status:** NEW
- Attachments:** [Icon]
- User Information:**
 - Requestor: MAXADMIN
 - Name: MAXADMIN
 - Phone: [Empty]
 - E-mail: [Empty]
 - Reported Date: 6/9/10 2:55 PM
- Process Request Details:**
 - Description: CI's update configuration Request
 - Process Manager Type: Configuration
 - Site: AUSTIN
 - Requested Completion: [Empty]
 - Asset: [Empty]
 - Location: [Empty]
 - Configuration Item: 10003
 - Configuration Item Name: MIALINUXDB2
 - Response Plan: [Empty]
 - Priority: [Empty]
- Request Classification Details:**
 - Classification: PMCFGUR
 - Class Description: CIs Configuration Control / Update Request
- Classification Attributes:**

Attribute	Description	Value
CHGRECD	All Updates Recorded	
CIOWNER	Primary CI Owner Person ID	
REASON	Reason to Update	
REQNOTE	Request Note	
SRVNAME	Server Name	

Figure 3-133 Process Request for updating CI

Changes and Releases

Release Management is the process that rolls out a collection of approved related Changes together in the same maintenance window. The Changes can be related based on time, technology interdependencies, target, risk mitigation, organization, scale (multiple copies), or service dependencies. This collection of related Changes is called a *Release*.

The Release Management Process is mostly used for complex Changes such as large-scale software deployments that affect multiple application servers and client workstations, major business application updates, major Changes to the network infrastructure, and emergency software and hardware fixes.

Managing Changes as a Release provides extra control when introducing complex Changes in the environment and minimizes the associated risks because all related Changes are tested and planned together.

In order to use the Release Management Process integrated with CCMDDB, it is necessary to install IBM Tivoli Release Process Manager.

The interaction between the Change Process Manager and Release Process Manager is also based on Process Requests, as are the Configuration Process Manager and Change Process Manager.

The following sections describe basic actions that can be performed to integrate Changes and Releases. These actions can be accessed by selecting **Select Action** → **Release Requests**.

Adding a Change to a specific Release

The Add Change to Release dialog box must be used to request that a specific Release handle a Change (Figure 3-134).

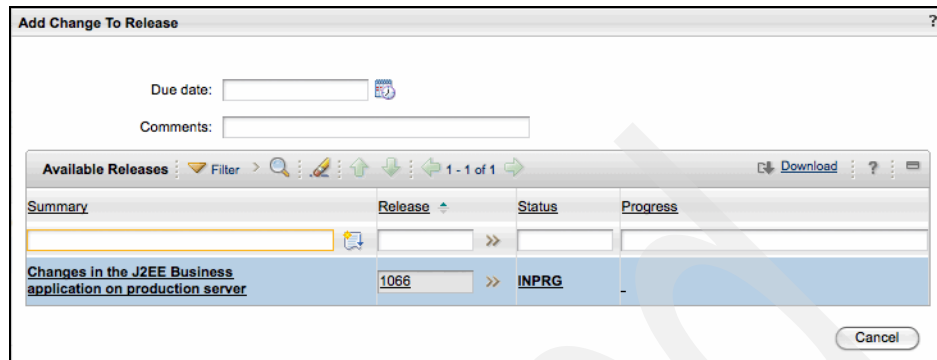
The dialog box lists all of the Releases that are available for accommodating a Change. After you select a Release for the current Change, all of the configuration items (CIs) that are associated with the Change are associated with the Release as well.

To request that a specific Release handle a Change:

1. Select the Release that you want to request for handling the Change.
2. (Optional) Specify a due date on which you want the Release to be completed, and add any comments that you think might be useful.
3. Click **OK**.

A Release Owner can respond to the request and accept the Change into a Release for which it is appropriate.

Figure 3-134 shows adding a Change to a Release.



The "Add Change To Release" dialog box features a title bar with a question mark icon. It contains a "Due date:" field with a calendar icon and a "Comments:" text area. Below these is a table titled "Available Releases" with a toolbar containing icons for filter, search, edit, and navigation, along with a "Download" link. The table has four columns: "Summary", "Release", "Status", and "Progress". The first row is highlighted in blue and contains the text "Changes in the J2EE Business application on production server", the release number "1066", the status "INPRG", and a progress bar. A "Cancel" button is located at the bottom right.

Summary	Release	Status	Progress
Changes in the J2EE Business application on production server	1066	INPRG	

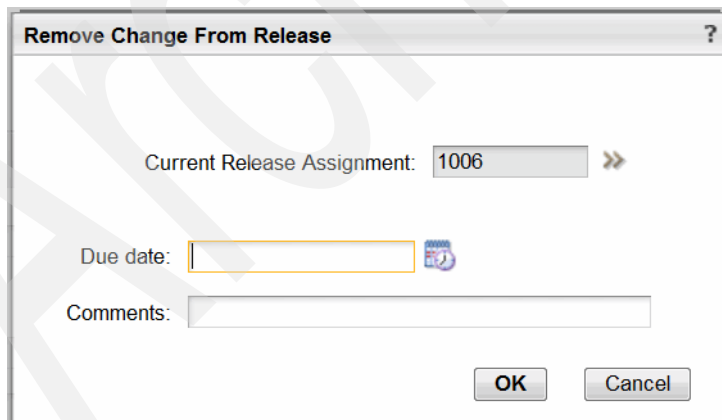
Figure 3-134 Add Change to Release

Removing a Change from a Release

The Remove Change From Release dialog box must be used to request that a Change be removed from a Release to which it is currently assigned. The dialog box shows the number of the Release to which the Change is currently assigned. It also displays a due date and comments, if this information was supplied when the Change was added to the Release.

To request that a Change be removed from a Release, click **OK**.

Figure 3-135 shows removing a Change from a Release.



The "Remove Change From Release" dialog box has a title bar with a question mark icon. It displays the "Current Release Assignment:" as "1006" with a double arrow icon. Below this are a "Due date:" field with a calendar icon and a "Comments:" text area. At the bottom are "OK" and "Cancel" buttons.

Figure 3-135 Remove Change from Release

Making a Change available for any Release

The Make a Change Available for Any Release dialog box must be used to request that a Change be made available for any Release that is defined in the environment. After making a Change available for any Release, a Release Owner can accept the request for a particular Release, thereby assigning the Change to that Release.

Tip: This functionality is useful when you have various groups working in the environment. For example, a team responsible for developing new Releases of a software or improvements might not know when they can release it in the environment. So they open a Request for Change that informs the Change Management team that it is available.

To make a Change available for handling by any Release, select **Release Requests** → **Make a Change Available for Any Release**. A dialog box will be displayed, where you can specify a due date field and enter comments into a field. Both of these fields are optional.

Figure 3-136 shows making a Change available for any Release.

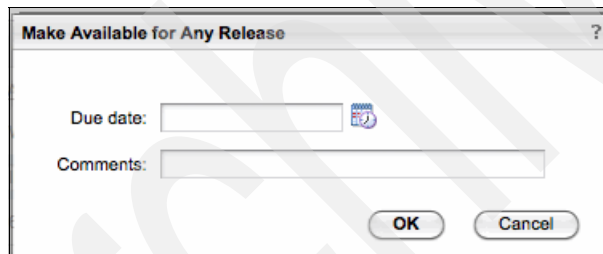


Figure 3-136 Making Change available for any Release

Canceling an outstanding request

The Cancel Outstanding Request dialog box must be used to cancel an outstanding request to add a Change to a Release, remove a Change from a Release, or make a Change available for any Release. The dialog box lists outstanding requests. The Specifications section indicates, for each request, the Change that is involved in the request and the actual or suggested Release that was requested. When cancelling an outstanding request, the request is voided and is no longer displayed in the Releases application. The user can then make a new request to add or remove the Change, as appropriate.

To cancel an outstanding Request to associate a Change with a Release or remove a Change from a Release, select the request that must be canceled and, from the Select Action Menu, select **Release Requests** → **Cancel Outstanding Request**.

Figure 3-137 shows the cancelling of an outstanding request.

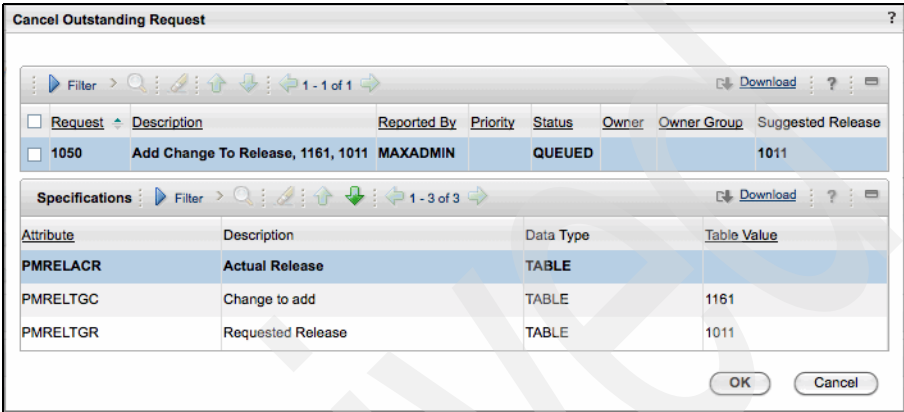


Figure 3-137 Cancel an Outstanding Request

3.4 Release Process Manager

In this section we take you through Release Manager 7.1.2. We also explain a scenario that will help users to understand how Change and Release management interact.

Release process is managed through the Release application which establishes standardized processes to ensure consistent management of Releases to IT infrastructure. This will help minimize business impact and improve operations. Releases allow a set of Changes to be added to a Release, which is then managed as a unit. Individual parts of the Release can also be managed separately.

It is also used to manage large-scale deployments of authorized versions of software to Release targets. Release contents include the tasks, scheduling, and people or groups that are involved in the operation. The flow of activities can be customized to ensure that the Release tasks are executed in the correct order by the authorized people.

3.4.1 IBM Tivoli Release Process Manager overview

IBM Tivoli Release Process Manager utilizes best-practice ITIL process. The main functionalities of IBM Tivoli Release are as follows:

- ▶ Plan and oversee the successful rollout of software and related hardware, centrally tracking the Releases.
- ▶ Configure the flow of Release activities to support your current processes.
- ▶ Agree to the exact content and rollout plan for a Release, through liaison with Change Management.
- ▶ Assign activities and tasks based on roles, ensuring that the right people own the Release steps.
- ▶ Organize and manage Releases using job plans that are based on ITIL best practices.
- ▶ Dynamically customize Release activities and tasks, responding to changing needs as they arise.
- ▶ View Release status and pending tasks in real time.
- ▶ Adapt existing software repositories as Definitive Media Libraries (DMLs) that can be tapped for software distribution during a Release.
- ▶ Schedule multiple rollouts targeting multiple configuration items (CIs) in a single Release.
- ▶ Receive reports that show, for all of your Releases, Release progress, Release status, and the CIs that are affected by the Releases.

Best of all, this product enables you to achieve incremental value by adopting a process, growing and adapting the process at your own pace, and adding task automation to further reduce labor costs around defined Release management pains.

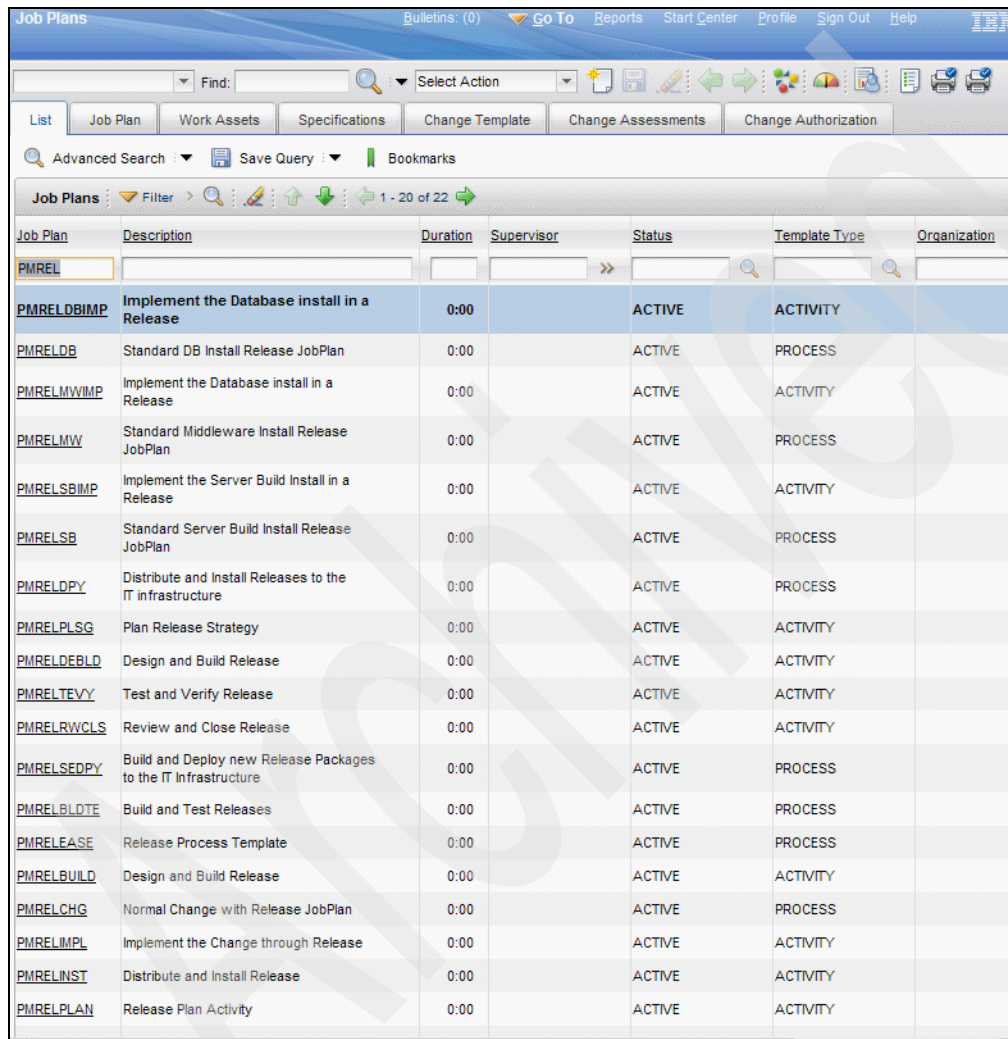
IBM Tivoli Release Process Manager is tightly integrated with two operational management products (OMPs): IBM Tivoli Configuration Manager and IBM Tivoli Provisioning Manager for Software. This OMP integration enables you to perform launch-in-context, software distribution, Tivoli Provisioning Manager logical device operations (LDOs), and Definitive Software Library (DSL)-related operations, provided that you install the Integration Module (IM) for the OMP that you want to use.

The OMPs can be used for automated software distribution of Releases and software updates. In this book we are covering only manual deployments. For automated deployment using OMPs, see the *End-to-End Service Management Using IBM Service Management Portfolio*, SG24-7677.

Important: Information in this chapter is based on IBM Tivoli Release Process Manager 7.1.2, which is the current version of the product. In the follow up release (Release Process Manager V7.2.1), Release will be considered a separate, licensable component of CCMDB, unlike the current version. At the time of writing this book, Release Process Manager V7.2.1 was not available.

IBM Tivoli Release Process Manager V7.1.2. provides three new job plans for managing Release packages; a new classification for easily selecting CIs for deployment; a Deployment Specialist security group; a Workplan Map for tracking the task-by-task progress of a Release; and new platform support, upgrade support, and installation options

New built-in job plans are available in the Release Process Manager component. Each job plan contains nested activity-level job plans. To view job plans for Release, from the main CCMDDB menu select **Go To** → **Planning** → **Job Plans**, then filter with **PMREL** (Figure 3-138).



Job Plan	Description	Duration	Supervisor	Status	Template Type	Organization
PMREL						
PMRELBIMP	Implement the Database install in a Release	0:00		ACTIVE	ACTIVITY	
PMRELOB	Standard DB Install Release JobPlan	0:00		ACTIVE	PROCESS	
PMRELMWIMP	Implement the Database install in a Release	0:00		ACTIVE	ACTIVITY	
PMRELMW	Standard Middleware Install Release JobPlan	0:00		ACTIVE	PROCESS	
PMRELSBIMP	Implement the Server Build Install in a Release	0:00		ACTIVE	ACTIVITY	
PMRELSB	Standard Server Build Install Release JobPlan	0:00		ACTIVE	PROCESS	
PMRELDPY	Distribute and Install Releases to the IT infrastructure	0:00		ACTIVE	PROCESS	
PMRELPLSG	Plan Release Strategy	0:00		ACTIVE	ACTIVITY	
PMRELDEBLD	Design and Build Release	0:00		ACTIVE	ACTIVITY	
PMRELTVY	Test and Verify Release	0:00		ACTIVE	ACTIVITY	
PMRELRWCLS	Review and Close Release	0:00		ACTIVE	ACTIVITY	
PMRELSEDPY	Build and Deploy new Release Packages to the IT Infrastructure	0:00		ACTIVE	PROCESS	
PMRELBIDTE	Build and Test Releases	0:00		ACTIVE	PROCESS	
PMRELEASE	Release Process Template	0:00		ACTIVE	PROCESS	
PMRELBUILD	Design and Build Release	0:00		ACTIVE	ACTIVITY	
PMRELCHG	Normal Change with Release JobPlan	0:00		ACTIVE	PROCESS	
PMRELIMPL	Implement the Change through Release	0:00		ACTIVE	ACTIVITY	
PMRELINST	Distribute and Install Release	0:00		ACTIVE	ACTIVITY	
PMRELPLAN	Release Plan Activity	0:00		ACTIVE	ACTIVITY	

Figure 3-138 Job plans for Release

3.4.2 Phases of Release process

Release Process Manager V7.1.2 supports a Release process that is closely aligned with the process defined in ITIL and ITUP processes.

The main phases of Release process are as follows:

1. Plan:

After a new Release is created, you determine which Changes will go into the Release. Multiple Changes can be added to a Release. When you add a Change to a Release, the source and target configuration items for the Change are also associated with the Release. An impact analysis identifies dependencies that might affect additional related CIs across the data center. A basic task list is defined. You can apply a job plan that contains activities and tasks that meet the requirements of the Release. Certain activities might be empty at this stage. For example, detailed rollout tasks are defined during the plan rollout phase. The overall structure of the Release plan is the final goal of this stage.

2. Design and build:

The installation scripts and packages to be deployed are designed and created during this phase. The software is not created, but it is packaged for deployment. Installation scripts and mechanisms, communication and education plans, and backout procedures are also developed.

3. Test and accept:

During this phase, the Release package is tested to ensure that is free of errors, and the Release is reviewed to determine whether it can be accepted for deployment. Copies of the package are added to the Definitive Software Library. The Definitive Hardware Store is used for any hardware that is needed for deployment.

4. Plan rollout:

Detailed plans, including Release dates and deliverables, are created for rolling out the Release to each site. For each site, delivery is scheduled for any new assets that are part of the Release, and the Release deployment is also scheduled. If training is needed, it is scheduled, as are any communications that are needed.

5. Communicate and prepare:

During this phase, you ensure that all stakeholders, users, and support personnel are aware of the Changes that will occur when the Release is deployed. To ensure that a site is ready for a Release, you might schedule site-specific testing, system shutdowns, reminders about the Release, and so on.

6. Distribute and install:

In this phase, distribution and installation occurs. With CCMDB, you verify the status of CIs that will be targets of Release operations. If the component used for Release deployment is Tivoli Provisioning Manager or Tivoli Provisioning Manager for Software, the Release delivery can be the deployment of a Logical Device Operation (LDO) or a software distribution package. If the component used for Release deployment is Tivoli Configuration Manager, then the Release delivery is a Software Distribution Package.

Each of the phases of the Release are performed by the people who are aligned with ITIL roles.

3.4.3 The Change and Release process interaction

The Release process makes Changes to one or more configuration items. CI Changes are managed through the Change Management process.

The Change Management process and the Release Management process are tightly integrated.

Often simple Changes are not associated with Releases. For complex Changes, such as a large-scale software deployment, the Release process guarantees an ordered and auditable Change.

A Release must be associated with one or more approved Changes. A Change that is associated with a Release cannot be completed until the Release is completed. Multiple Changes can be associated with a Release, but a Change cannot be associated with more than one Release.

Various operations are used to associate Changes with a Release. For example, it is possible create a Change from within a Release to handle CI Changes that the Release is planned to make. Additionally, a Change Manager can add approved Changes to a Release from within the Changes application. This operation generates an *add Change to Release* request. An add Change to Release request can be accepted either within the Releases application or from the ISM Request application. The request can also be rejected, and Changes can be removed from a Release.

The following operations are used to associate Changes with a Release:

► Creating an RFC from a Release:

This task can be used to create a Request for Change (RFC) from a Release. When an RFC is created from a Release, it is necessary to specify the RFC completion date that is required by the Release. It is also required to specify any other information that defines the requested Change.

► Adding Changes to a Release:

This task is used to request one or more scheduled Changes to be added to a Release. If an existing Release is defined so that it can accommodate the Change, you can specify this Release. Otherwise, the Change can be made available to any Release.

► Responding to add Change Requests:

When a request is made to add one or more Changes to a Release, the requests are available for acceptance or rejection from within the Releases application. If the request is to add the Change to a specific Release, the Release Owner of that Release is notified, and he accepts or rejects the request. If the request makes the Change available for any Release, all Release Owners are notified, and one of them accepts the task of accepting or rejecting the request.

► Removing Changes from a Release:

It is possible to remove Changes from a Release in two ways:

- From the Change that has been added to the Release
- From the Release to which the Change was added

When a Change that was previously added to a Release is removed, all of the configuration items or software packages that are associated with the Change are also removed from the Release.

► Responding to a remove Change Request:

When requests to remove Changes from a Release are submitted, the requests are available for acceptance or rejection from within the Releases application. When a remove Change Request is accepted, the configuration items associated with the Change are no longer associated with the Release. When a request is rejected, the CIs retain their association with the Release.

► Transferring a Change to another Release:

The Transfer Change to another Release dialog box is used to transfer one or more Changes that are assigned to a Release to another Release. It is possible to transfer a Change to another Release because the currently assigned Release has been cancelled because another Release is more appropriate for a Change.

► Using process requests for Change and Release interaction:

Rather than using the Change and Release applications, it is possible to submit requests through the process requests application to add a Change to a Release or remove a Change from a Release. After one of these requests is submitted, the request is evaluated by a Release Owner and either accepted or rejected.

► Wait for Release Task:

The goal of a Wait for Release task is to automatically pause the Change process until the Release containing the Change completes successfully. The task must only be used when the Change is implemented by a Release. The Change process has certain steps which can only be completed after the Change is implemented (for example, the PIR). The Wait for Release task can be used to prevent those tasks from being executed until the Change is implemented.

3.4.4 The process flow in Release Manager V7.1.2

To better show you the process flow in Release Manager V7.1.2, let us consider a scenario where a user raises an RFC for a requirement that will need Change in the business applications. The Change Manager receives the process request and accepts it. He does the impact analysis and finds out that this Change is a complex one and requires to be implemented as a part of Release. He assigns the CR to a particular Release. Also the Change Manager will define tasks for Change that will assigned and completed.

The Release Owner accepts the Release request for that Change. The Change record gets added to the Release content of that Release. Release Owner assigns a job plan to the Release and initiates the Release. This Release will now go through the activities such as Plan, Design, Build, Test and Accept, Communicate and Prepare, and Distribute and Deploy. Each activity will have the owners assigned who will do their assigned tasks

The whole scenario goes through the following stages:

- Creating a Release
- Creating an RFC
- Accepting the RFC and creating tasks
- Assigning the RFC to Release
- Adding the Change for Release
- Initiating a Release
- Taking the Release through various stages of the Release process
- Completing the Release
- Completing the Changes associated with this Release

Creating a Release

The Release can be created in two ways:

- ▶ The Release manager can create a Release based on the need in the infrastructure Changes.
- ▶ Any user can raise request for a new Release.

Release creation by Release manager

The Release manager will create a new Release based on the requirement in the infrastructure that requires Changes. For example, suppose that a particular business application has a group of RFCs that are raised by various users for their requirement to be implemented during a period. These RFCs will require various downtimes, are complex, and cannot be implemented straightforwardly, hence all Changes have to be implemented as a Release.

To create a Release, it is necessary to specify certain attributes, such as its classification, the type and scope of the Release, the CIs that are source and target of the Release, and so on.

1. To create a new Release, from the CCMDB menu, select **Go To** → **Release** → **Releases** (see Figure 3-139).

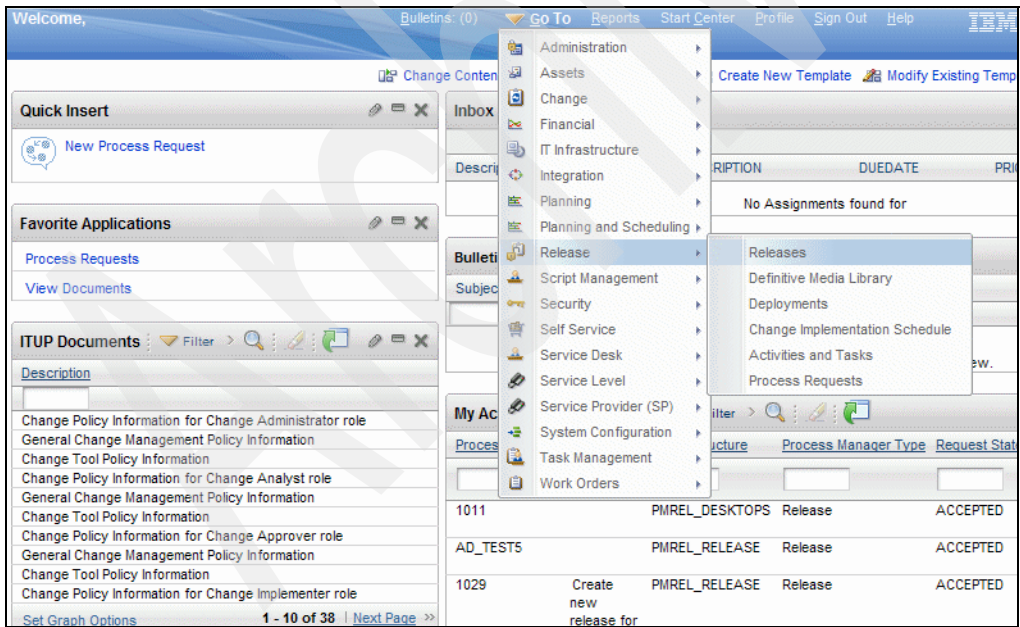


Figure 3-139 Release application

2. Inside the Release application, choose the number button (#) on the right of the Select Action window (Figure 3-140).

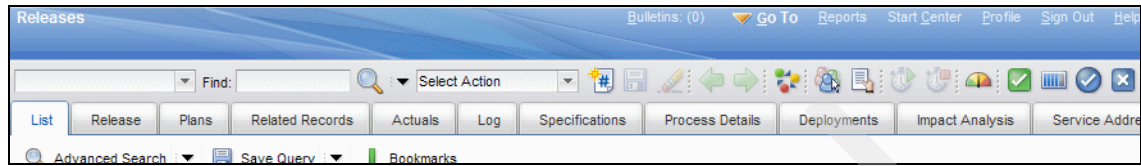


Figure 3-140 New Release

3. Add a short description of the Release details in the Summary field.
4. Select **Classification** → **Classify** and you will see the classification tree. Here choose the classification value that best describes this Release (Figure 3-141).

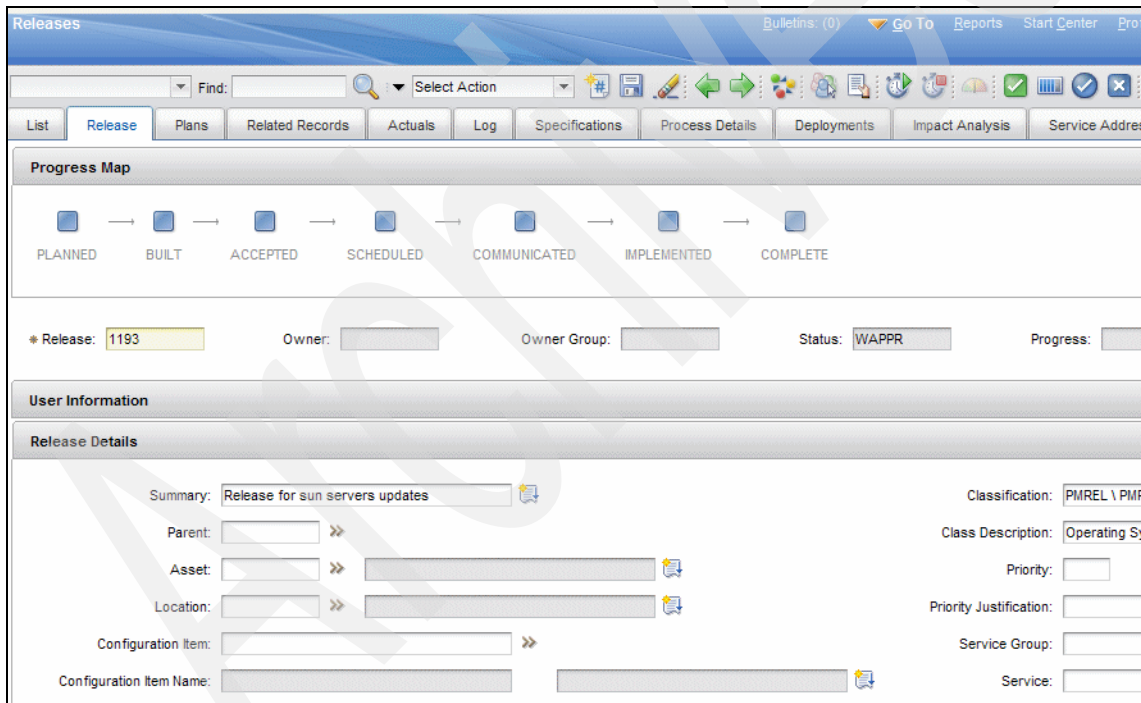


Figure 3-141 Release record

5. Save the record.

This creates a Release record with Release number as shown in the Releases box.

Release creation by the users

Any users can raise a request for new Release. For example, the Change Manager might want to add a Change to a particular Release. But the Release might not be available in the current list. Or, if the user wants a new Release, they can raise a process request to create a new Release:

1. To raise a process request, click **Goto** → **Change** → **Process Requests**.
Select the New Process Request icon (CTRL+ALT+I) and specify the required information to define a new process request:
 - a. Enter a summary in the Description field.
 - b. Enter the Process Manager Type as Release.
 - c. Select Classification as PMREL \ PMREL_SOFTWARE \ PMREL_OPERATING_SYSTEM with Description as Operating System.
 - d. Click **Submit**.

See Figure 3-142.

The screenshot displays the 'Process Requests' web application interface. At the top, there is a navigation bar with links for 'Bullets (0)', 'Go to', 'Reports', and 'Start Center'. Below this is a search bar with a 'Find:' label and a 'Select Action' dropdown. A toolbar with various icons is also present. The main content area is divided into several sections:

- Process Request:** 1031
- Owner:** [Empty field]
- Owner Group:** [Empty field]
- Process State:** SUBMITTED
- Status:** QUEUED

User Information

- Requestor:** MAXADMIN
- Name:** MAXADMIN
- Phone:** [Empty field]
- E-mail:** [Empty field]
- Reported Date:** 6/15/10 4:03 PM

Process Request Details

- Description:** Add Change To Release, 1149, 1148
- Process Manager Type:** Release
- * Site:** DEARBORN
- Requested Completion:** [Empty field]
- Details:** [Empty text area]

Figure 3-142 Submit process request

2. On clicking **Submit**, the Release Manager will receive this request for creation of a new Release (Figure 3-143).
3. The Release Manager accepts the Release request by clicking **Select Action** → **Accept**.

The screenshot displays the 'Process Requests' web application interface. At the top, there is a navigation bar with links for 'Go To', 'Reports', 'Start Center', 'Profile', and 'Sign'. Below this is a search bar with a 'Find:' field and a 'Select Action' dropdown. A toolbar with various icons is also present. The main content area is divided into several sections:

- Process Request:** 1029
- Owner:** MAXADMIN
- Owner Group:** (empty)
- Process State:** ACCEPTED
- Status:** INPROG

User Information

Requestor: MAXADMIN

Name: MAXADMIN

Phone: (empty)

E-mail: (empty)

Reported Date: 6/15/10 2:38 PM

Process Request Details

Description: Create new release for Sun server changes

Process Manager Type: Release

* Site: DEARBORN

Requested Completion: (empty)

Details: (empty text area)

Figure 3-143 Accept process request

This process will automatically create the Release and is shown in the Related Records tab of the process request (Figure 3-144).

Process Requests

Bullets: (0) Go To Reports

Find: Select Action

List Process Request Related Records Log Service Address

Process Request: 1029 Create new release for Sun server changes Site: DEARBORN

Related Tickets Filter

Related Record Key Description Process Type Process State Class

...No rows to display...

Related Work Orders Filter 1 - 1 of 1

Work Order	Description	Class
1148	release for Sun server changes	RELEASE

Figure 3-144 Release in related records

This new Release will now appear in the Release application.

Creating a Request for a Change

A process request is raised for a Change when the user requires Changes to the IT environment. For example, a business application has to be modified to include the functionality required by end users.

To create a Request for Change:

1. Click **Go To** → **Change** → **Process Request**. Select **New Process Request** and specify the following information to define the new process request:
 - a. Type a short description of the requested Change in the Description field. The description must clearly identify the needed Change so that other users can easily identify what is requested.
 - b. Select the **Process Manager Type** field as **Change**.

- c. Select **Classify**, and click the blue box beside the classification value that best describes the purpose of the request. The Classification and Class Description fields are populated based on the selection, and any attributes associated with the classification are displayed in the Classification Attributes section. Optionally, specific information can be entered to define the attributes (Figure 3-145).

Process Requests

Bulletins: (0) Go To Reports Start Center

Find: Select Action

E-mail: Reported Date: 6/15/10 3:27 PM

Process Request Details

Description: Update OS fixpack to latest

* Process Manager Type: Change

* Site: DEARBORN

Requested Completion:

Asset: >>

Location: >>

Configuration Item: >>

Configuration Item Name:

Response Plan: >>

Impact:

Urgency:

Priority:

Customer: >>

Customer Charge Account:

Customer Cost Center:

Request Classification Details

* Classification: PMCHG \ PMCHGSFW \ OPSYSTEM >>

Class Description: Operating System

Figure 3-145 new RFC

- d. Enter the values in other optional fields as appropriate and then **Submit** the process request (Figure 3-146).

The screenshot shows a web application interface titled "Process Requests". At the top, there is a blue header bar with "Bullets: (0)", "Go To", "Reports", and "Start Center" links. Below the header, a status message reads "BMXAA441E - Process ISMSUBMIT started.". A toolbar contains various icons for actions like search, select, and save. Below the toolbar, there are tabs: "List", "Process Request" (which is active), "Related Records", "Log", and "Service Address". The main content area displays form fields: "Process Request:" with the value "1030", "Owner:" (empty), "Owner Group:" (empty), "Process State:" with the value "SUBMITTED", and "Status:" with the value "QUEUED". At the bottom, there is a section labeled "User Information".

Figure 3-146 Submitted New RFC

Accepting the Request

The Change Manager will receive the Request for Change. This person does the feasibility analysis and accepts the process request.

After a process request is accepted, it will create a Change record. The Change Manager will update the appropriate fields and the target CIs that are going to be affected. Because this Change is complex and requires downtime, it has to go through Release for implementation.

The Change Manager then adds the Change to a particular Release which is already scheduled and affects the same IT infrastructure.

To accept a Change, complete the following steps:

1. Go to that particular process Request for Change and click the **Accept** button.
2. After a process request is accepted, it will pop up a window. Here, click **OK** (Figure 3-147).

Process Requests | Guidelines (0) | Go To | Reports | Start Center

Find: [] Select Action: []

List | **Process Request** | Related Records | Log | Service Address

Process Request: 1030 | Owner: [] | Owner Group: [] | Process State: SUBMITTED | Status: QUEUED

User Information

Requestor: MAXADMIN
 Name: MAXADMIN
 Phone: []
 E-mail: []
 Reported Date: 6/15/10 3:27 PM

Process Request Details

Description: Update OS fixpack to latest
 Process Manager Type: Change
 Site: DEARBORN
 Requested Completion: []
 Asset: [] >> []
 Location: [] >> []
 Configuration Item: [] >> []
 Configuration Item Name: []
 Response Plan: [] >> []
 Impact: []

Manual Input

☒ Review and refine the new Change Work Order
☐ Continue working on Process Requests

Memo: []

OK Cancel

Figure 3-147 Accepted RFC

- On clicking **OK**, it will redirect to a Change record (Figure 3-148). Update the appropriate fields in Change.

Changes Bulletins: (0) [Go To](#) [Reports](#) [Start Center](#)

Find: Select Action

[List](#)
[Change](#)
[Assessments](#)
[Impacts](#)
[Authorization](#)
[Schedule](#)
[Workplan Map](#)
[Related Records](#)
[Actuals](#)
[Log](#)
[Service Address](#)

Progress Map

☐ → ☐ → ☐ → ☐ → ☐ → ☐ → ☐ → ☐ → ☐ → ☐

ACC_CAT ASSESS SCHED AUTH IMPL INPRG COMP REVIEW CLOSE

Change:
 Owner:
 Owner Group:
 Status:

Change Details

Summary:
 Risk:

Change Type:
 Failure Probability:

Effect Of Not Implementing:
 Impact:

Verification Plan:
 Urgency:

Remediation Plan:
 Priority:

Reason for Change:
 Change Category:

Priority Justification:

Classification:
 Class Description:

Figure 3-148 New Change

4. The CI that are affected can be selected in the source CI, primary targets and additional targets (Figure 3-149). In the Lab environment, no CIs were selected. Hence these fields are blank here.

Changes Bulletins: (0) [Go To](#) [Reports](#) [Start Center](#) [Prof](#)

Find: Select Action Classification: Class Description:

Specifications Filter > 1 - 2 of 2

Attribute	Description	Data Type	Alphanumeric Value	Numeric Value	Unit of Measure
PKGNAME	Name	ALN			
VERSION	Version	ALN			

Source CIs Filter >

CI number	Configuration Item Name	Configuration Item Description	Classification	Source Description
...No rows to display...				

Primary Targets

Configuration Item: Asset:

Configuration Item Name: Location:

CI Business Impact: Asset/Location Priority:

Customer:

Target Description:

Additional Targets Filter >

Asset	Location	Configuration Item Number	Business Impact
...No rows to display...			

Figure 3-149 Specify target CI in Change

Creating Wait for Release tasks

The Change Manager, after accepting the Change, then creates the tasks or attaches a job plan to the current Change.

There will be a Wait for Release task type which can be placed in a job plan or work plan. The task will remain in INPRG state until the Release completes successfully, at which point it will be set to COMPLETE so that the Change process can continue. If something happens which will prevent the Change from being successfully implemented by the Release, then the task will be set to FAIL state. This will get the attention of the Change Owner so they can take action to recover.

The Change Manager will go to the **Schedule** Tab to create tasks for this Change (Figure 3-150).

Changes Bulletins: (0) [Go To](#) [Reports](#) [Start Center](#)

Find: Select Action

List Change Assessments Impacts Authorization **Schedule** Workplan Map Related Records Actuals Log Service Address

Progress Map

ACC_CAT → ASSESS → SCHED → AUTH → IMPL → INPRG → COMP → REVIEW → CLOSE

Change: Supervisor:

Job Plan: Lead:

Response Plan: Work Group:

Fully Automated? ☐ Remediation Plan:

Parent:

Predecessors:

Target Start: Scheduled Start: Actual Start:

Target Finish: Scheduled Finish: Actual Finish:

* Estimated Duration: Time Remaining: Scheduler Project:

Start No Earlier Than: Include Tasks in Schedule? ☒

Finish No Later Than:

Children of Change 1149

Tasks for Change 1149

Sequence	Task	Summary	Estimated Duration	Status	Owner
...No rows to display...					

Figure 3-150 Tasks for Change

Here the Change Manager will create a task for this Change in the *Task for Change* section. He can also attach a job plan if it is already created and active.

For this scenario we are adding three tasks for this Change (Figure 3-151).

Changes | Bulletins: (0) | Go To | Reports | Start Center

Find: | Select Action:

List | Change | Assessments | Impacts | Authorization | **Schedule** | Workplan Map | Related Records | Actuals | Log | Service Address

Progress Map

ACC_CAT → ASSESS → SCHED → AUTH → IMPL → INPRG → COMP → REVIEW → CLOSE

Change: | | Supervisor: >>

Job Plan: >> | >> | Lead: >>

Response Plan: >> | >> | Work Group: >>

Fully Automated? ☐ | Remediation Plan: >>

Parent: >>

Predecessors: >>

Target Start: | Scheduled Start: | Actual Start:

Target Finish: | Scheduled Finish: | Actual Finish:

Estimated Duration: | Time Remaining: | Scheduler Project:

Start No Earlier Than: | Include Tasks in Schedule? ☒

Finish No Later Than:

Children of Change 1149 | Filter > | 1 - 3 of 3

Sequence	Task	Summary	Estimated Duration	Status	Owner
1	10	Technical and business assessment	1:00	WAPPR	>>
2	20	Change implementation	1:00	WAPPR	>>
3	30	Post implementation Review	1:00	WAPPR	>>

Figure 3-151 Tasks created

Sequence	Task	Summary	Estimated Duration	Status	Owner
1	10	Technical and business assessment	1:00	WAPPR	
2	20	Change implementation	1:00	WAPPR	
3	30	Post implementation Review	1:00	WAPPR	

Task Information

* Task: 20 Change implementation

Sequence: 2

Status: WAPPR

Classification: PMCHG \ PMCHGTSK \ PMCHGWFR

Classification Description: Wait For Release task

Implementation Task? ☒

Under Flow Control? ☒

Flow Action:

Flow Action Assist? ☐

Assisted Workflow:

Launch Entry Name:

In the Predecessors field of the last two tasks, mark the previous task as the predecessor.

After the tasks are created, change the status of *Change* to *In Progress* (Figure 3-153).

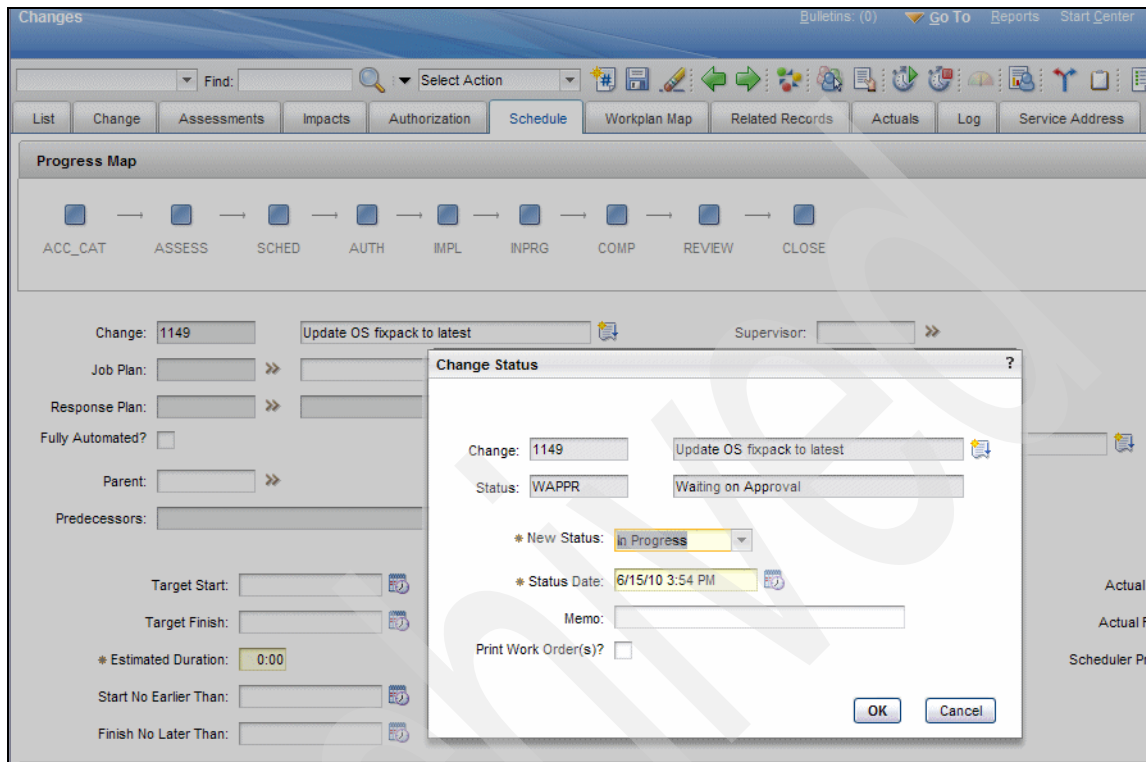


Figure 3-153 Change status

The first task will get in progress (Figure 3-154).

Changes Bulletins: (0) Go To Reports Start Center

Find: Select Action # Save Undo Redo Print Help

List Change Assessments Impacts Authorization **Schedule** Workplan Map Related Records Actuals Log Service Address

Progress Map

ACC_CAT → ASSESS → SCHED → AUTH → IMPL → **INPRG** → COMP → REVIEW → CLOSE

Change: Supervisor: >>

Job Plan: >> Lead: >>

Response Plan: >> Work Group: >>

Fully Automated? ☐ Remediation Plan: >>

Parent: >>

Predecessors: >>

Target Start: Scheduled Start: Actual Start:

Target Finish: Scheduled Finish: Actual Finish:

* Estimated Duration: Time Remaining: Scheduler Progress:

Start No Earlier Than: Include Tasks in Schedule? ☒

Finish No Later Than:

Children of Change 1149 Filter > [Icons]

Tasks for Change 1149 Filter > [Icons] 1 - 3 of 3

	Sequence	Task	Summary	Estimated Duration	Status	Owner
▼	1	10	Technical and business assessment	1:00	INPRG	>>
▶	2	20	Change implementation	1:00	WAPPR	>>
▶	3	30	Post implementation Review	1:00	WAPPR	>>

Figure 3-154 Change and Task in progress

On completing the first task, the second task automatically changes to *In Progress*. After the Release is complete, the second task will get completed automatically. In case Release fails, then this task will also fail.

Adding a Change to a Release

To add the Change to a Release, complete the following steps:

1. In the Changes panel, choose **Select Action** → **Release Requests**.
To add a Change to Release, there are four available options (Figure 3-155):
 - a. **Add to a specific Release:** If the Change Manager knows to which Release this Change will be implemented, or wants to request it for a particular Release, the Change Manager can select that particular Release in the list of available Releases. After selecting the particular Release, a process request is raised to the Release Owner to add the Change to Release.
 - b. **Make available for any Release:** If the Change Manager does not know to which Release, out the of available ones, this Change will go, then select this option. It will notify all the Release Owners to add the Change to Release.
 - c. **Remove from a Release:** If the Change Manager or Change Owner wants to remove the Change from a Release, then select this option. It will send a request to the Release Owner for Change removal from the Release.
 - d. **Cancel outstanding Requests:** This option enables users to cancel any requests that are pending for acceptance by the Release Owners.

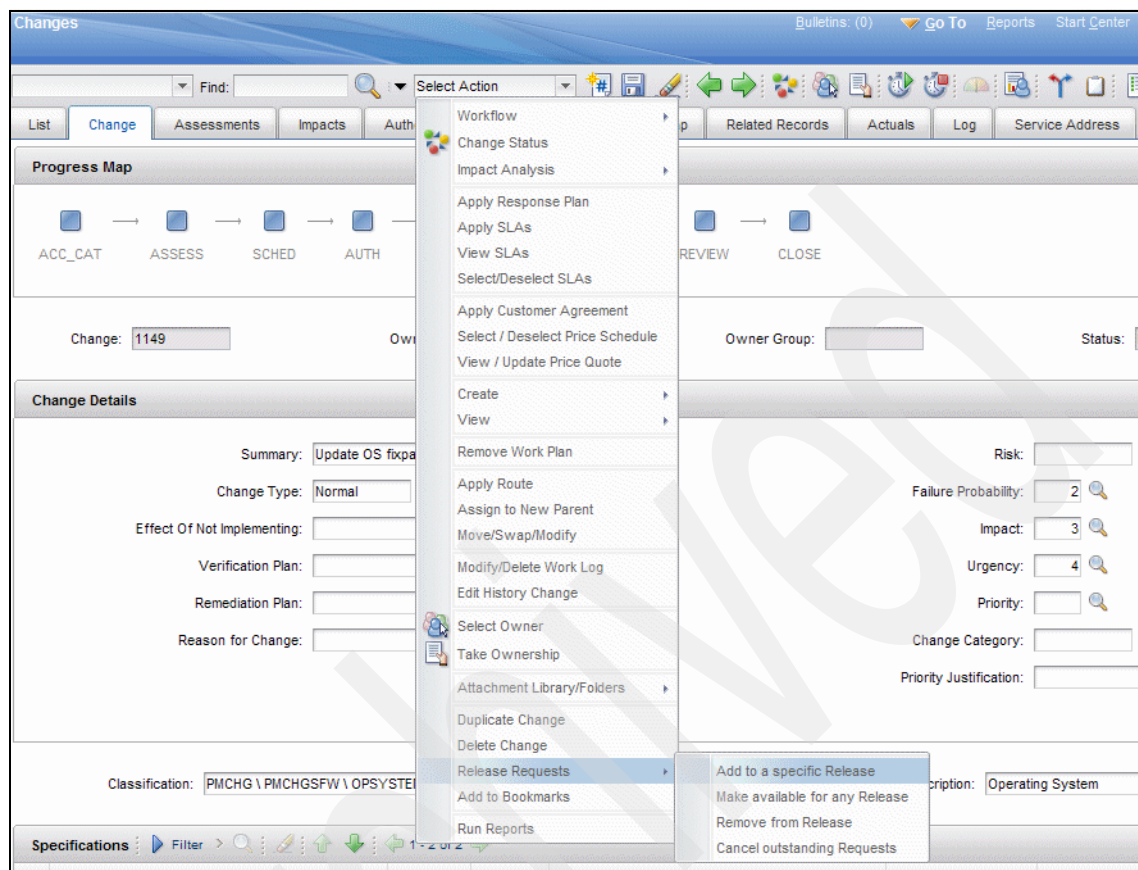


Figure 3-155 Release options

2. For our scenario, select **Add to a specific Release** (Figure 3-156). After a Release is selected, it will trigger a request to the Release Owner.

While selecting these options, certain workflows are triggered in the background to take it through a process flow.

Add Change To Release?

Due date:

Comments:

Available Releases

Filter > 1 - 9 of 9 Download ?

Summary	Release	Status	Progress
test	1002 >>	WAPPR	OPEN
test	1003 >>	WAPPR	SCHEDULED
	1011 >>	WAPPR	
	1012 >>	WAPPR	COMPLETE
	1137 >>	WAPPR	
ds rel 1	1142 >>	WAPPR	
release for Sun server changes	1148 >>	WAPPR	
	AD_TEST >>	WAPPR	
Build and Deploy new Release Packages to the IT Infrastructure	AD_TEST JP >>	INPRG	

Cancel

Figure 3-156 Available Releases

Workflows

In the Release process, there are certain predefined workflows which trigger at various points in the process flow.

For example, in the previous step, when the Change Manager selects a particular Release to be added to a Change, workflows *PMRELCRWF* and *ISMSUBMIT* are triggered, which creates a process request, sets the appropriate value of the process request and, submits it to the Release Owner.

To view these workflows, go to the **Workflow application**, search, and open the respective workflows. Figure 3-157 shows the *PMRELCRWF* workflow.

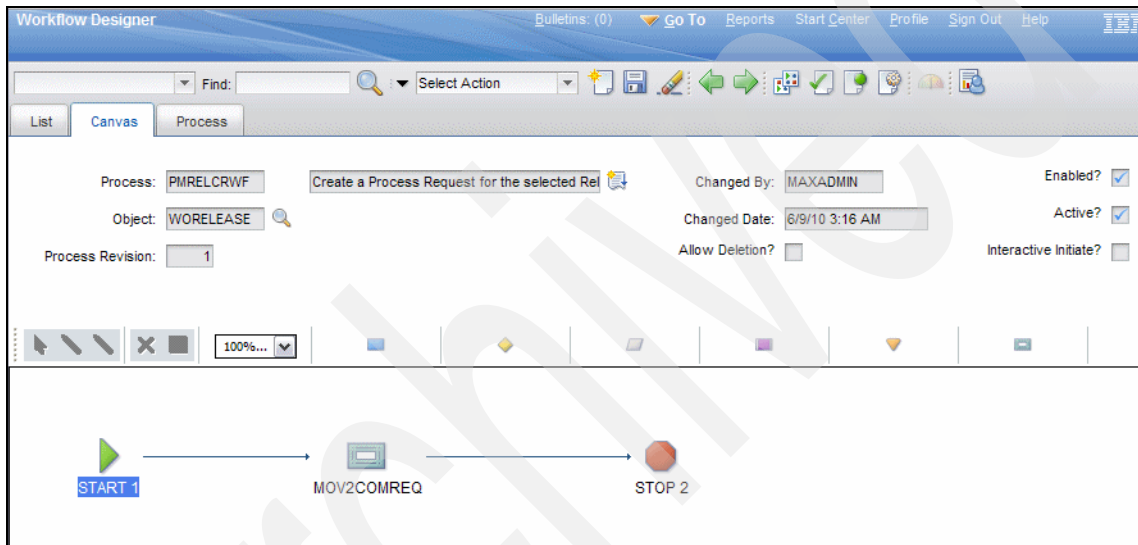


Figure 3-157 Create process request workflow

The process request is created (by *PMRELCRWF* process) and the *ISMSUBMIT* workflow process is triggered.

Figure 3-158 shows the *ISMSUBMIT* workflow. This workflow has a sub-workflow defined in it, called *PMRELREQSB*.

The *RELEASESUB* node on the *ISMSUBMIT* workflow has the subprocess *PMRELREQSB* defined in it, which can be viewed by opening the properties of this node.

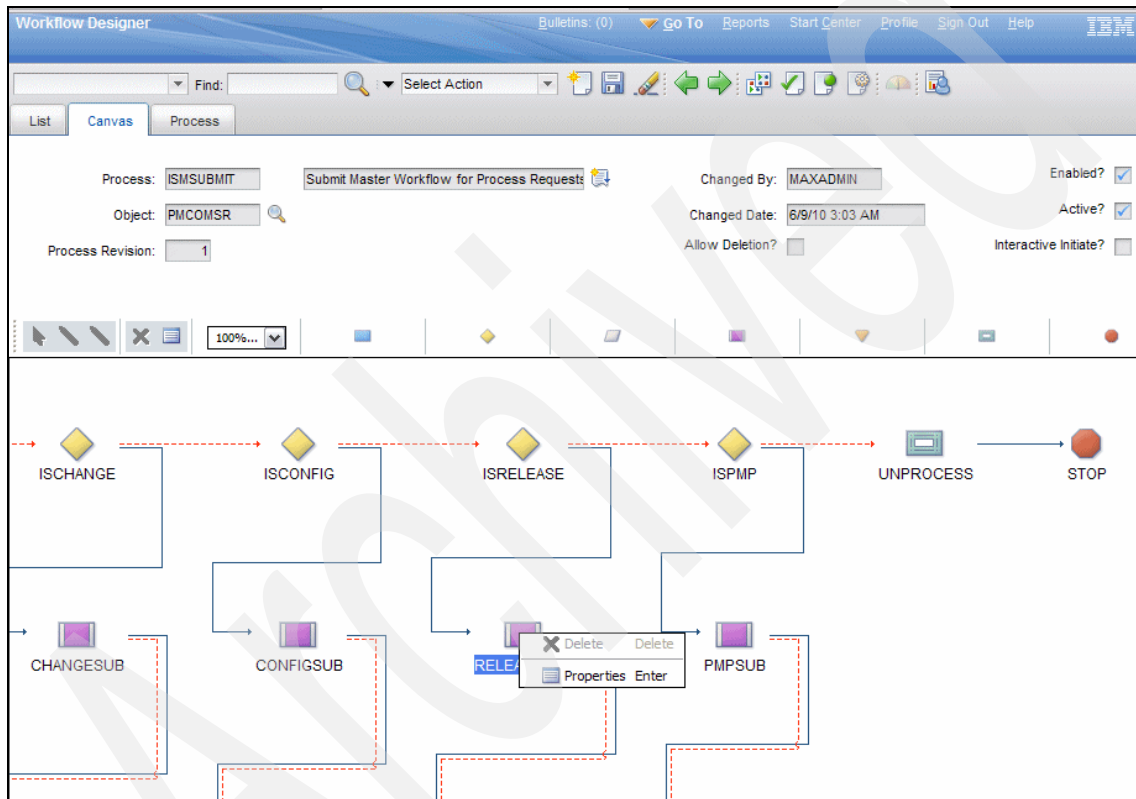


Figure 3-158 *ISMSUBMIT* workflow

Figure 3-159 shows the *PMRELREQSB* workflow. The workflow consists of certain condition nodes that check for the classification of the request. The condition nodes check whether the request is for add Change to Release or remove Change from Release. Based on this request type, certain actions are triggered. To see these actions, right-click the **positive arrow** next to the condition node.

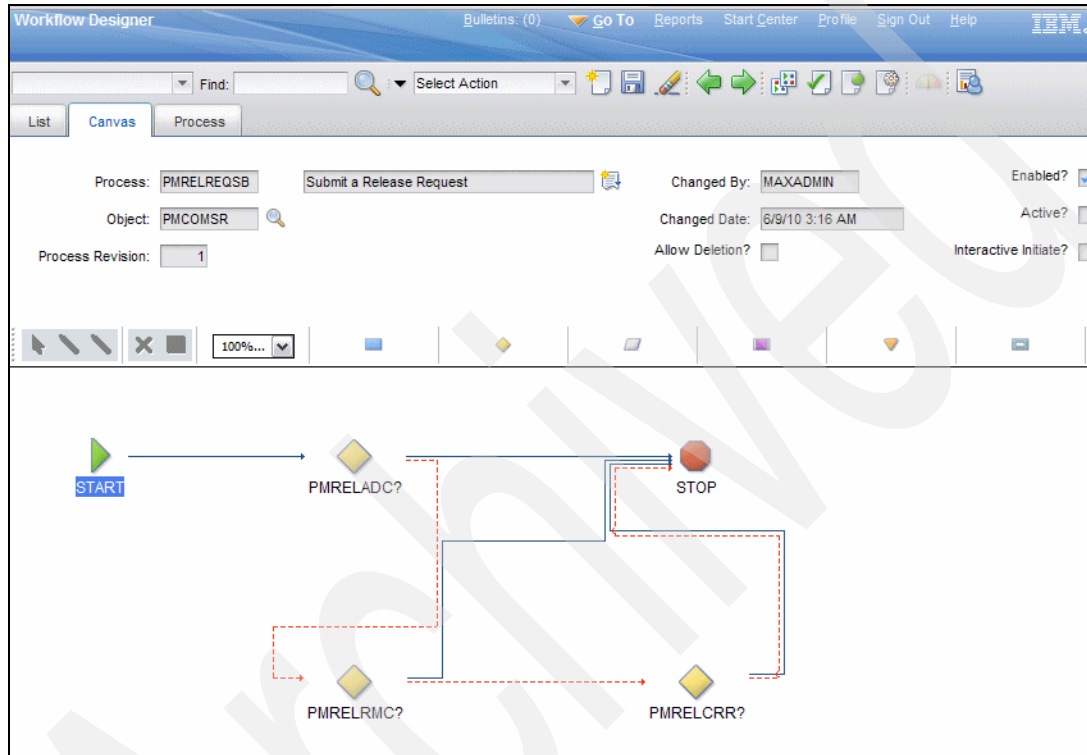


Figure 3-159 WorkFlow Submit process request for add Change to Release

Figure 3-160 shows the action properties window of the positive arrow, where *PMRELADDRELREQGRP* action is defined, when the request is to add Change to a Release. To view this action, click the **arrow** next to Action field (from the positive action arrow) and select **Go To Actions**.

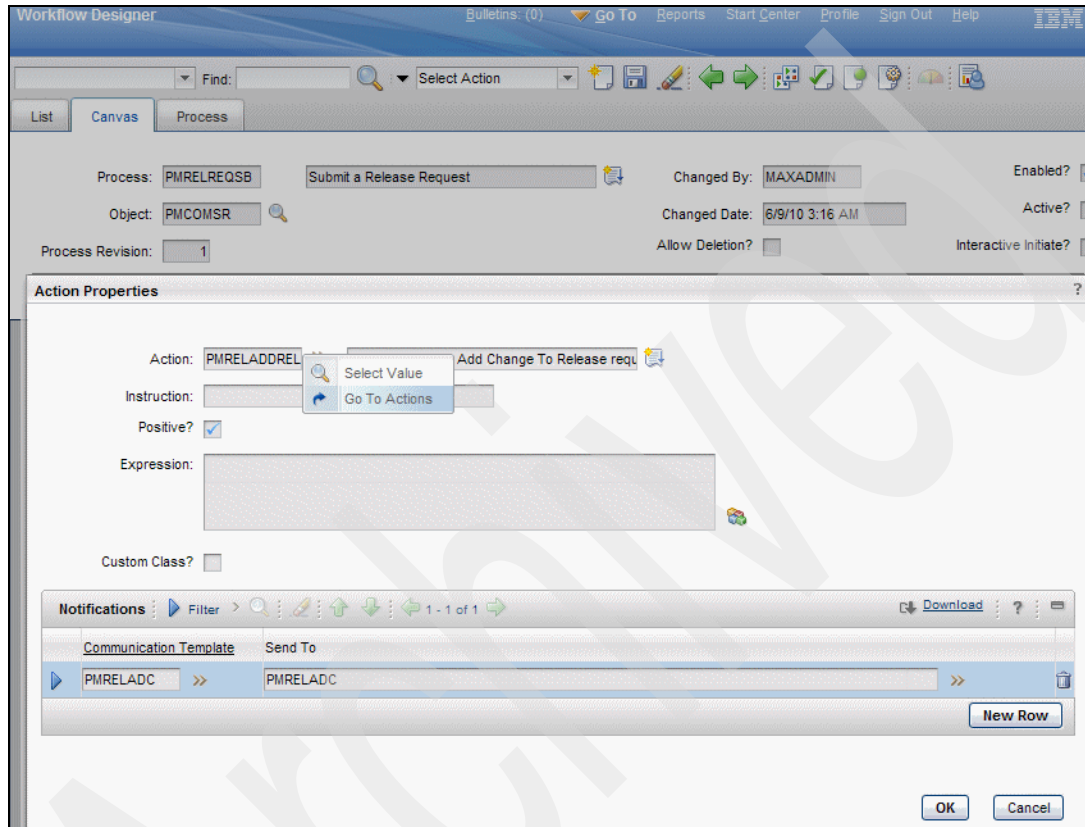


Figure 3-160 Action in workflow

Figure 3-161 shows the action definition. There are two custom actions *PMRELADDRELREQRELATEREC* and *PMCOM_REQUEST_SUBMIT*.

The *PMRELADDRELREQRELATEREC* action creates the related record entries into the Process Request, and *PMCOM_REQUEST_SUBMIT* action changes the status of this process request to submitted.

The screenshot displays the IBM Actions configuration window. At the top, there's a search bar with 'Find:' and a 'Select Action' dropdown. Below this are tabs for 'List' and 'Action'. The 'Action' tab is active, showing configuration fields for an action group. The 'Action' field is set to 'PMRELADDREL' and the 'Object' is 'PMCOMSR'. The 'Type' is 'Action Group'. The 'Value' field is empty. The 'Parameter/Attribute' and 'Memo' fields are also empty. The 'Classification' and 'Class Description' fields are empty. The 'Accessible From' dropdown is set to 'ALL'. Below these fields is a 'Members' section with a table listing two actions.

Action	Description	Object	Type	Value
PMRELADDRELREQRELATEREC	Create the Related Record Entries.	PMCOMSR	CUSTOM	com.ibm.ism.pmrel.workorder.PmRelAddngeToReleaseRequestAction
PMCOM_REQUEST_SUBMIT	Process Request Submit Status Action	PMCOMSR	CUSTOM	com.ibm.ism.pmcom.PmComSrSubmitStatusAction

Figure 3-161 Action definition for submit process request

Likewise, for other request types such as Change getting added to any Release, removing Change from Release, and canceling outstanding requests, there are workflows that are auto initiated.

After the Process Request is submitted, based on the definitions of Release process, they are auto initiated for authorizations to the Release manager or process owner.

Based on the actions done on the Process Request, such as accept or cancel, the ISM workflows corresponding to these actions are triggered. This is as shown in Table 3-8 on page 284.

For example, the *ISMACCEPT* workflow is triggered when Process Requests are accepted. This workflow initiates the subprocess *PMRELREQAC*. This subprocess is as shown in Figure 3-162.

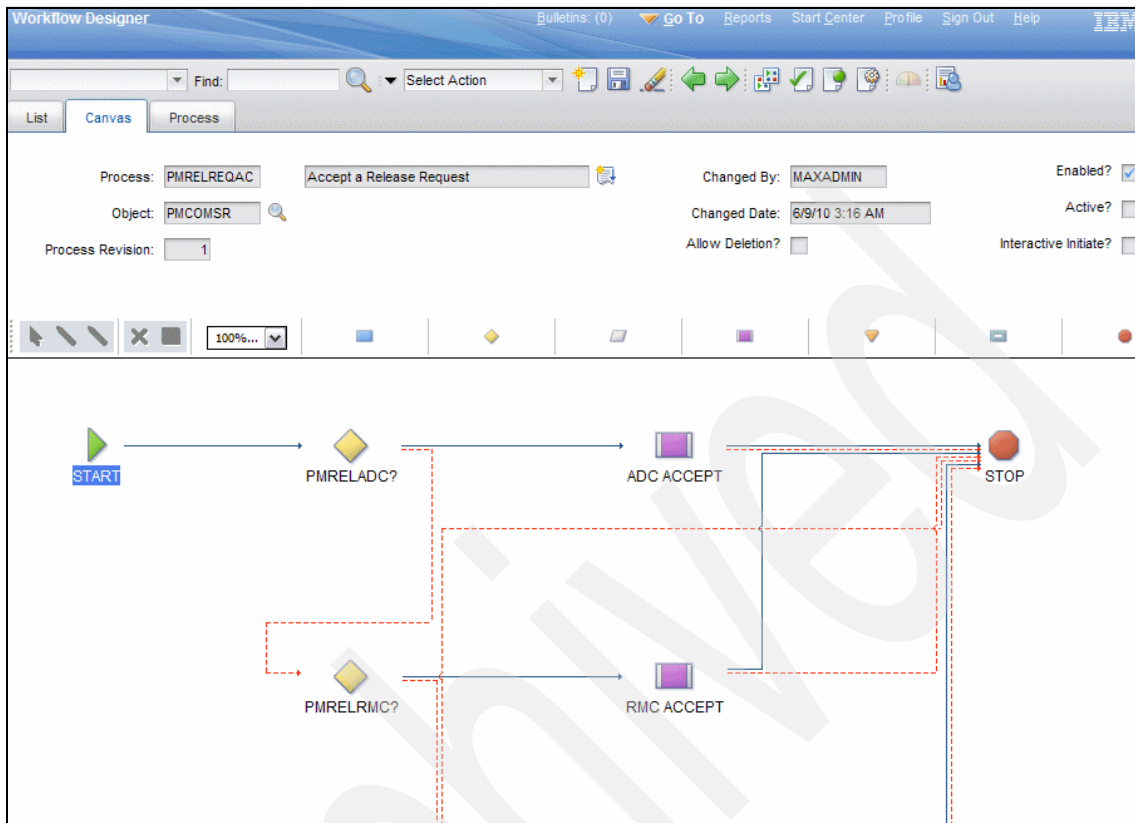


Figure 3-162 Workflow for accept process request

Likewise for other actions such as cancel, close, reject etc. the corresponding Release process workflows are triggered. This mapping is shown in Table 3-8.

Table 3-8 Process Request default actions

SI.No	Process	Description	Release Process
1	ISMACCEPT	Accept Master Workflow for Process Requests	PMRELREQAC
2	ISMCANCEL	Cancel Master Workflow for Process Requests	PMRELREQCA
3	ISMCLOSE	Close Master Workflow for Process Requests	PMRELREQCL
4	ISMREJECT	Reject Master Workflow for Process Requests	PMRELREQRJ
5	ISMREPLACE	Process Request Not Handled	PMRELREQSB
6	ISMSUBMIT	Submit Master Workflow for Process Requests	NA

The subprocess *PMRELREQAC* again checks whether this request is for adding Change to Release or remove Change from Release or for creating a Release. Depending on the condition, a subprocess is initiated again. For example, the *PMRELADCAC* subprocess is initiated when request is for adding a Change to Release. This sub task is as shown in Figure 3-163.

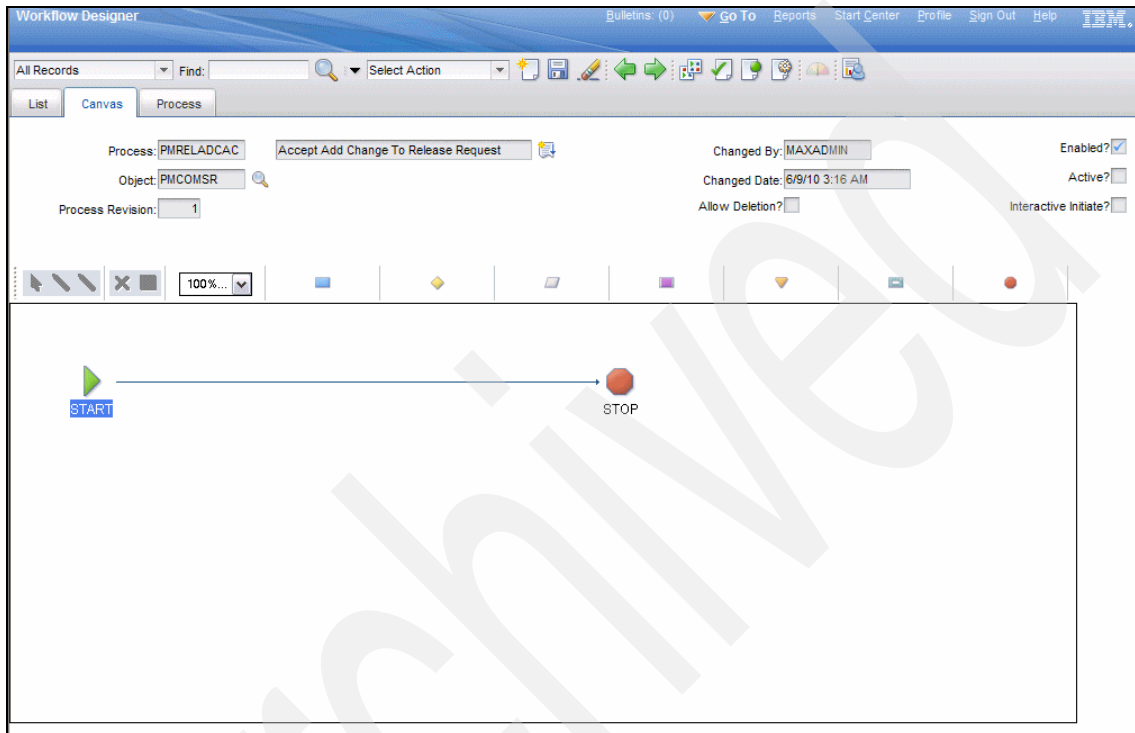


Figure 3-163 *PMRELADCAC* subprocess

Users can modify these workflows to include functions supported in their organizations, such as an additional level of authorization required. In the *PMRELADCAC* workflow process, we can add a task node to initiate an authorization before accepting this add request. The modified workflow is shown in Figure 3-164.

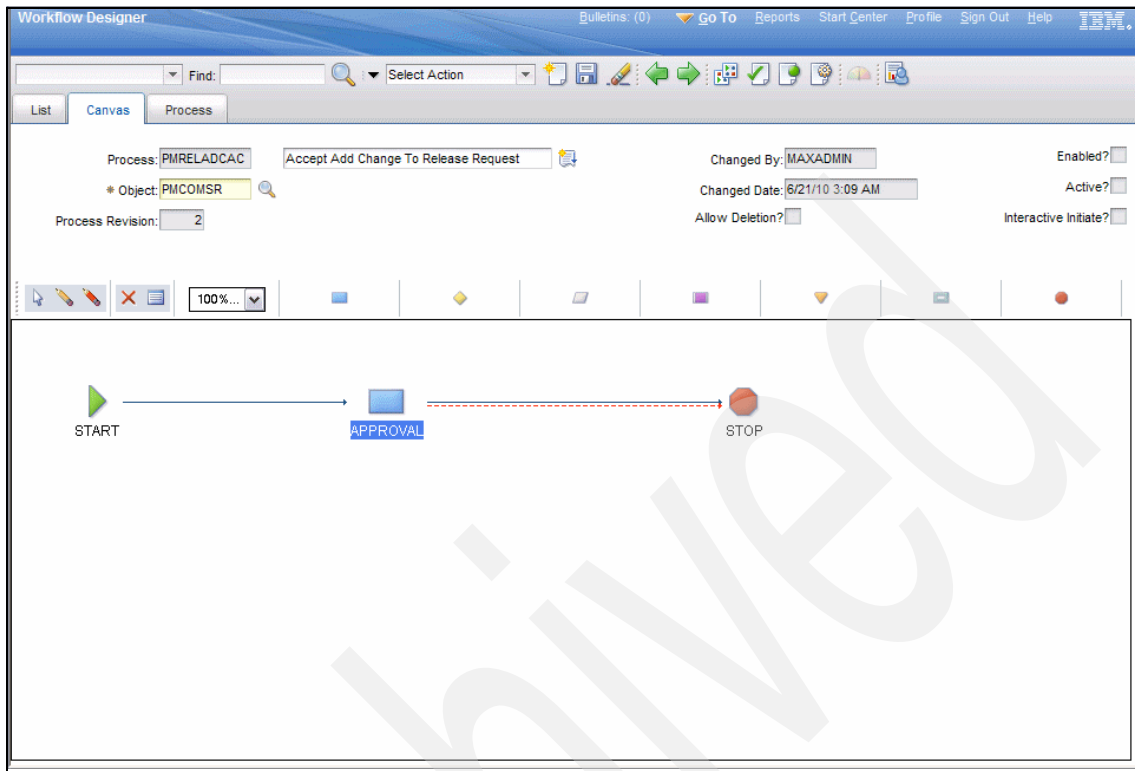


Figure 3-164 Modified workflow

Accepting an add Change to Release

The Release Owner will receive the process request to add the Change to their Release and will check the feasibility to include this Change in this Release implementation. The Release Owner can add the Change to the requested Release or to another Release.

Accepting the add Change to Release can be done in two ways:

1. Go to the process request raised for addition of Change to Release, and assign the actual Release to accept this request (Figure 3-165 and Figure 3-166).

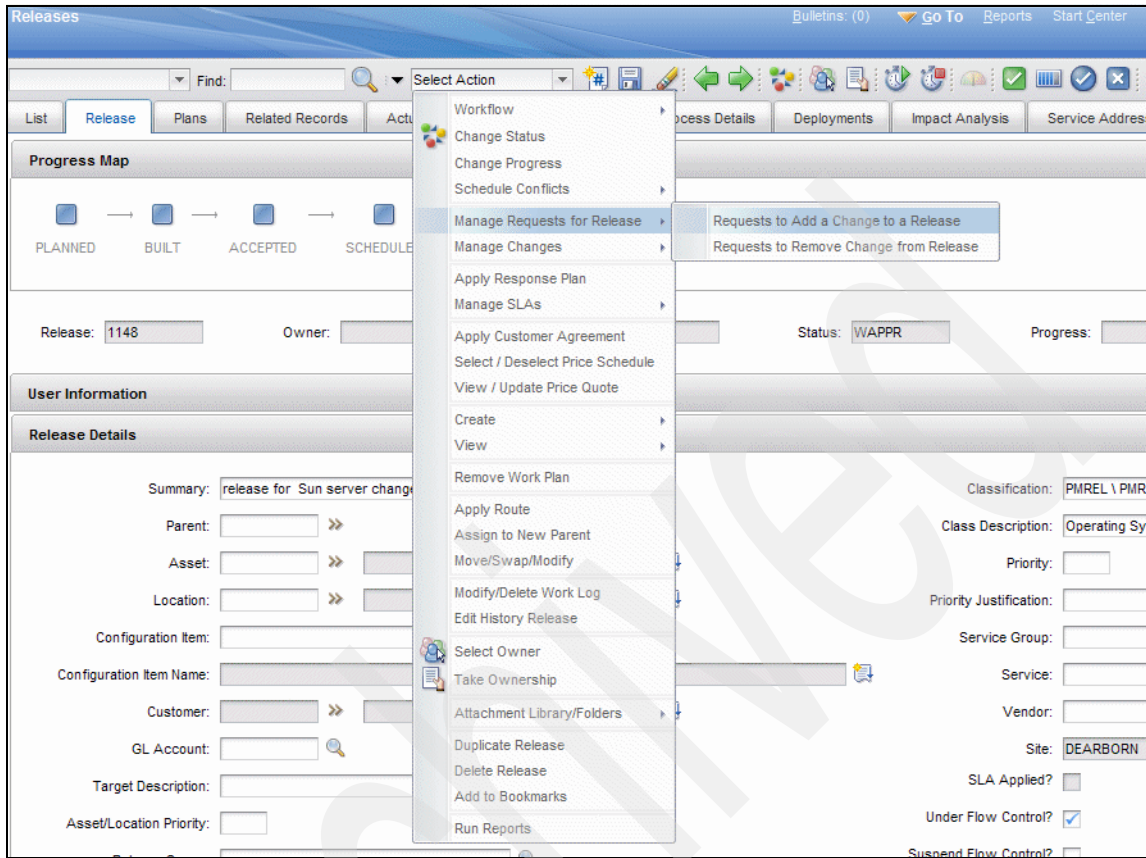


Figure 3-167 Respond to Release request

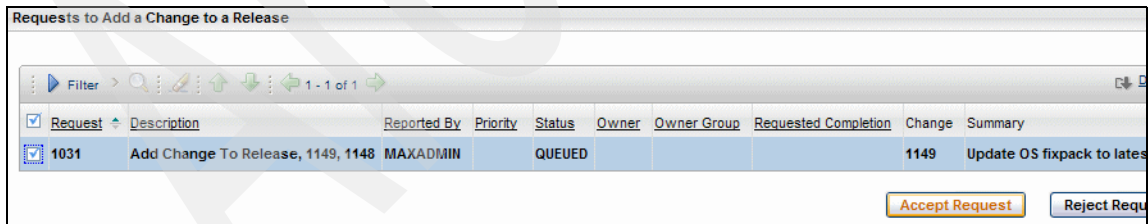


Figure 3-168 Select Change

3. The Change, after being accepted, can be seen in the Release contents of the particular Release (Figure 3-169).

Releases

Bullets: (0) Go To Reports Start Center

Find: Select Action

Impact:

Urgency:

Release Type:

Verification:

Risk Assessment:

Back Out Plan:

Back Out Plan Executed? ☐

Customer Charge Account:

Flow Action:

Flow Action Assist? ☐

Launch Entry Name:

Predecessors:

Response Plan:

Customer Cost Center:

Release Content

Changes Implemented by Release Filter 1 - 1 of 1

Change Number	Summary	Owner	Status
1149	Update OS fixpack to latest	MAXADMIN	INP

Figure 3-169 Change in Release contents

Initiate Release

Certain milestones characterize each of the major phases of the ITIL-aligned Release process. For actual Releases in your data center, you might emphasize one or more steps, while de-emphasizing or even skipping others. For example, if a Release does not involve software distribution, the Distribute and Install phase might not apply. Hence based on the requirement in the process, the Release manager will decide on what tasks will be done in a Release. The tasks are pre-defined in the job plan which can be attached to the Release.

To attach a job plan complete the following steps:

1. **Go To** → **Release** → **Releases** and open the Release record.
2. Go to the Plans tab and click the **arrow** next to the job plan to select the job plan (Figure 3-170).

Releases Bulletins: (0) [Go To](#) [Reports](#) [Start Center](#)

Find: Select Action

[List](#)
[Release](#)
[Plans](#)
[Related Records](#)
[Actuals](#)
[Log](#)
[Specifications](#)
[Process Details](#)
[Deployments](#)
[Impact Analysis](#)
[Service Address](#)

Progress Map

☐ → ☐ → ☐ → ☐ → ☐ → ☐ → ☐

PLANNED BUILT ACCEPTED SCHEDULED COMMUNICATED IMPLEMENTED COMPLETE

Release: Supervisor: >> Status:
 Parent: >> Lead: >> Progress:
 Job Plan: Work Group: >> Site:
 Quote Type: Go To Job Plans
 Quoted Price:

Children of Release 1148 Filter >

Sequence	Record	Record Class	Summary	Asset	Location	Configuration Item
...No rows to display...						

Tasks for Release 1148 Filter >

Sequence	Task	Summary	Estimated
...No rows to display...			

[Labor](#)
[Materials](#)
[Services](#)
[Tools](#)
[Fees and Charges](#)

Figure 3-170 Plan for Release

3. Here you select a job plan for this Release (Figure 3-171). We have used the default *PMRELEASE* job plan for our scenario. You can select or create your own job plans based on your requirement.

Select Value

Show Job Plans for the Work Order's Asset and Location Only? ☐

Show Job Plans with No Classes Defined? ☒

WO Class: Release

Refresh

Filter > 1 - 8 of 8

Job Plan	Description	Template Type	Organization	Site
PMRELEDB	Standard DB Install Release JobPlan	PROCESS	-	-
PMRELMW	Standard Middleware Install Release JobPlan	PROCESS	-	-
PMRELSB	Standard Server Build Install Release JobPlan	PROCESS	-	-
PMRELDPY	Distribute and Install Releases to the IT Infrastructure	PROCESS	-	-
PMRELSEDPY	Build and Deploy new Release Packages to the IT Infrastructure	PROCESS	-	-
PMREBLDTE	Build and Test Releases	PROCESS	-	-
PMRELEASE	Release Process Template	PROCESS	-	-
JP9130502	TC AddJobPlanToRelease	MAINTENANCE	-	-

Cancel

Figure 3-171 Job plans available

4. On selecting the job plan, it gets attached to the Release (Figure 3-172).

Releases Bulletins: (0) Go To Reports

Find: Select Action [Icons]

List Release **Plans** Related Records Actuals Log Specifications Process Details Deployments Impact Analysis Service Address

Progress Map

☐ PLANNED → ☐ BUILT → ☐ ACCEPTED → ☐ SCHEDULED → ☐ COMMUNICATED → ☐ IMPLEMENTED → ☐ COMPLETE

Release: Supervisor: >> Status: WAP
 Parent: >> Lead: >> Progress:
 Job Plan: >> Work Group: >> Site: DEA
 Quote Type:
 Quoted Price:

Children of Release 1148 Filter > [Icons] 1 - 5 of 6 >

Sequence	Record	Record Class	Summary	Asset	Location	Configuration Item
▶ 10	1153	>> ACTIVITY	Release Plan Activity			
▶ 20	1162	>> ACTIVITY	Design and Build Release			
▶ 30	1166	>> ACTIVITY	Test and Accept Release			
▶ 40	1171	>> ACTIVITY	Plan Release Rollout			
▶ 50	1180	>> ACTIVITY	Communicate, Prepare, and Train for Release			

Figure 3-172 Job plan tasks in Release

Default insert site: If a user's default insert site is not added, then that user might not be able to do certain tasks. Ensure that each user has a default insert site.

- The Release Manager now initiates the Release by changing the status to *In Progress*. This will change the status of the first tasks to *In Progress* (Figure 3-173).

Releases Bulletins: (0) [Go To](#) [Reports](#)

Find: Select Action










[List](#)
[Release](#)
[Plans](#)
[Related Records](#)
[Actuals](#)
[Log](#)
[Specifications](#)
[Process Details](#)
[Deployments](#)
[Impact Analysis](#)
[Service Address](#)

Progress Map

 →
  →
  →
  →
  →
  →
 

Release:  Supervisor: >> Status:
 Parent: >> Lead: >> Progress:
 Job Plan: >>  Work Group: >> Site:
 Quote Type:
 Quoted Price:

Children of Release 1148      1 - 5 of 6 

Sequence	Record	Record Class	Summary	Asset	Location	Configuration Item
▶ 10	1153 >>	ACTIVITY	Release Plan Activity 	>>	>>	
▶ 20	1162 >>	ACTIVITY	Design and Build Release 	>>	>>	
▶ 30	1166 >>	ACTIVITY	Test and Accept Release 	>>	>>	
▶ 40	1171 >>	ACTIVITY	Plan Release Rollout 	>>	>>	
▶ 50	1180 >>	ACTIVITY	Communicate, Prepare, and Train for Release 	>>	>>	

Figure 3-173 Release and task in progress

Progress of the Release

After the job plan is assigned to the Release and initiated, the task owners, who are defined in the job plan, will receive notification to complete their assigned tasks.

The job plan, which we had attached, has nested job plans within each task (Figure 3-174).

Releases Bulletin: (0) Go To Reports

Find: Select Action

Release: 1148 release for Sun server changes Supervisor: Status: INPR
Parent: Lead: Progress: PLA
Job Plan: PMRELEASE Release Process Template Work Group: Site: DEA
Quote Type:
Quoted Price:

Children of Release 1148 Filter > 1 - 5 of 6

Sequence	Record	Record Class	Summary	Asset	Location	Configuration Item
10	1153	ACTIVITY	Release Plan Activity			
20	1162	ACTIVITY	Design and Build Release			
30	1166	ACTIVITY	Test and Accept Release			
40	1171	ACTIVITY	Plan Release Rollout			
50	1180	ACTIVITY	Communicate, Prepare, and Train for Release			

Child Information

Sequence: 10 Status: CO
Record: 1153 Release Plan Activity GL Account:
Record Class: ACTIVITY Priority:
Asset: Inherit Status Changes? ☐
Location: Accepts Charges? ☒
Configuration Item: Under Flow Control? ☒
Job Plan: PMRELPLAN Suspend Flow Control? ☐
Verification: Flow Action:

Figure 3-174 Nested Job plan in task

The nested job plans can be seen by going to **Go to** → **Planning** → **Job Plans**. Open the nested job plan *PMRELPLAN* (Figure 3-175).

Sequence	Task	Description	Nested Job Plan	Duration
50	50	Determine Staffing		1:00
60	60	Define Scheduling		2:00
70	70	Approve Release Schedule		1:00
80	80	Update Release Progress		0:01

Figure 3-175 Nested job plan tasks

The nested job plan has the tasks defined, which are flow controlled with owners assigned and predecessors specified. Each task in the process will change to *In Progress* after the previous task is complete. Each nested job plan also has the last task as *Update Release Progress*. After all the tasks are completed during the process of this nested job plan, this task will set the progress state of the Release using the Flow Action that is defined (Figure 3-176).

When the task changes to *In Progress*, the action defined in the flow action field is triggered.

tails

Organization:

Site:

Task:

Sequence:

Nested Job Plan:

Classification:

Class Description:

Launch Entry Name:

Implementation Task? ☐

* Duration:

Meter:

Owner:

Owner Group:

Flow Controlled? ☒

Suspend Flow Control? ☐

Flow Action:

Flow Action Assist? ☐

Assisted Workflow:

Predecessors:

Figure 3-176 Update progress state task in job plan

When you click **Go To Actions**, the action definitions can be seen. The action sets the value of progress state of the Release to *Planned* (Figure 3-177).

Actions Return

Find:

List

Action:

Object:

* Type:

Classification:

Class Description:

Value:

Parameter/Attribute:

Memo:

* Accessible From:

Members: 1 - 2 of 2

Action	Description	Object	Type	Value
PMRELPROGTOPLAN	Change Release Progress to Planned	WOACTIVITY	SETVALUE	'PLANNED'
PMCHGCOMPTASK	Set the current task to complete	WOACTIVITY	CHANGESTATUS	COMP

Figure 3-177 Action group to change the progress state

After this task is complete, it will set the progress value of the Release to *Planned* (Figure 3-178).

The screenshot shows the 'Releases' application window. At the top, there's a header bar with 'Releases' and 'Bulletins: (0)'. Below it is a toolbar with various icons. The main form area contains fields for 'Release: 1148', 'Parent:', 'Job Plan: PMRELEASE', 'Quote Type:', and 'Quoted Price:'. To the right, there are fields for 'Supervisor:', 'Lead:', 'Work Group:', 'Status: INPR', 'Progress: PLA', and 'Site: DEA'. Below this is a section titled 'Children of Release 1148' with a 'Filter' button and a table. The table has columns: 'Sequence', 'Record', 'Record Class', 'Summary', 'Asset', 'Location', and 'Configuration Item'. It lists five activities: 'Release Plan Activity' (Sequence 10, Record 1153), 'Design and Build Release' (Sequence 20, Record 1162), 'Test and Accept Release' (Sequence 30, Record 1166), 'Plan Release Rollout' (Sequence 40, Record 1171), and 'Communicate, Prepare, and Train for Release' (Sequence 50, Record 1180). Below the table is a 'Child Information' form for the selected record (Sequence 10, Record 1153, Record Class: ACTIVITY). This form includes fields for 'Asset:', 'Location:', 'Configuration Item:', 'Job Plan: PMRELPLAN', and 'Verification:'. On the right side of the 'Child Information' form, there are several checkboxes: 'Status: CO', 'GL Account:', 'Priority:', 'Inherit Status Changes?', 'Accepts Charges?' (checked), 'Under Flow Control?' (checked), 'Suspend Flow Control?', and 'Flow Action:'.

Figure 3-178 First task in Release complete and progress state Planned

After all the tasks defined in the Release job plans are complete, the Release record status changes to *Complete*. After the Release completes, the Wait for Release task, which we had defined earlier) in the corresponding Change, also completes (Figure 3-179).

Changes

Find: Select Action

List Change Assessments Impacts Authorization **Schedule** Workplan Map Related Records Actuals Log Service Address

Progress Map

ACC_CAT → ASSESS → SCHED → AUTH → IMPL → INPRG → COMP → REVIEW → CLOSE

Change: 1149 Supervisor: Status: INPRG
 Job Plan: Lead: Site: DEARBORN
 Response Plan: PMCHGNORMIL Work Group: Quote Type:
 Fully Automated? ☐ Remediation Plan: Quoted Price:
 Parent:
 Predecessors:

Target Start: Scheduled Start: Actual Start: 6/15/10 3:54 PM
 Target Finish: Scheduled Finish: Actual Finish:
 Estimated Duration: 0.00 Time Remaining: Scheduler Project:
 Start No Earlier Than: Include Tasks in Schedule? ☒
 Finish No Later Than:

Children of Change 1149

Tasks for Change 1149

Sequence	Task	Summary	Estimated Duration	Status	Owner	Owner Group
1	10	Technical and business assessment	1.00	COMP		
2	20	Change implementation	1.00	COMP		
3	30	Post implementation Review	1.00	INPRG		

Figure 3-179 Wait for Release completes automatically

The Change Manager can now complete the last task in the task list. When all the tasks in the Change are complete, the status of Change records changes to *Completed* (Figure 3-180).

Changes Guidelines (0) [Go To](#) [Reports](#) [Start Center](#) [Profile](#) [Sign Out](#) [Help](#)

Find: [Select Action](#) [New](#) [Save](#) [Print](#) [Refresh](#) [Back](#) [Forward](#) [Home](#) [Search](#) [Help](#) [Log Out](#)

Progress map

ACC_CAT → ASSESS → SCHED → AUTH → IMPL → INPRG → **COMP** → REVIEW → CLOSE

Change: Supervisor: >> Status:
 Job Plan: >> >> Lead: >> Site:
 Response Plan: >> >> Work Group: >> Quote Type:
 Fully Automated? ☐ Remediation Plan: >> Quoted Price:
 Parent: >>
 Predecessors: >>

Target Start: Scheduled Start: Actual Start:
 Target Finish: Scheduled Finish: Actual Finish:
 Estimated Duration: Time Remaining: Scheduler Project: >>
 Start No Earlier Than: Include Tasks in Schedule? ☒
 Finish No Later Than:

Children of Change 1149 [Filter](#) [New](#) [Save](#) [Print](#) [Refresh](#) [Back](#) [Forward](#) [Home](#) [Search](#) [Help](#) [Log Out](#) [Download](#) [Help](#)

Tasks for Change 1149 [Filter](#) [New](#) [Save](#) [Print](#) [Refresh](#) [Back](#) [Forward](#) [Home](#) [Search](#) [Help](#) [Log Out](#) [Download](#) [Help](#)

Sequence	Task	Summary	Estimated Duration	Status	Owner	Owner Group
1	10	Technical and business assessment	1.00	COMP	>>	>>
2	20	Change implementation	1.00	COMP	>>	>>
3	30	Post implementation Review	1.00	COMP	>>	>>

[New Task](#)

Figure 3-180 Change status completes

This changes all the process requests associated with this Change to *Resolved* (Figure 3-181).

Changes

Go To Reports Start Center Profile Sign Out Help

Find: Select Action

List Change Assessments Impacts Authorization Schedule Workplan Map **Related Records** Actuals Log Service Address

Progress Map

ACC_CAT → ASSESS → SCHED → AUTH → IMPL → INPRG → COMP → REVIEW → CLOSE

Change: 1149 Update OS fixpack to latest
Parent: >>> Originating Record: 1030 Status: COMP
Originating Record Class: SR Site: DEARBORN

Related Work Orders Filter > 1 - 1 of 1 Download

Work Order	Description	Class	Status	Relationship
1194 >>	release for sun servers	RELEASE	COMP	FOLLOWUP

Select Work Orders New Row

Related Tickets Filter > 1 - 4 of 4 Download

Related Record Key	Description	Class	Status	Relationship
1030 >>	Update OS fixpack to latest	SR	RESOLVED	ORIGINATOR
1031 >>	Add Change To Release, 1149, 1148, 1148	SR	RESOLVED	FOLLOWUP
1032 >>	Remove Change From Release, 1149	SR	RESOLVED	FOLLOWUP
1033 >>	Add Change To Release, 1149, 1194, 1194	SR	RESOLVED	FOLLOWUP

Select Ticket New Row

Figure 3-181 All related process requests are resolved

Customizing Change in Tivoli Change and Configuration Management Database

In this chapter, we describe how to customize the configurations in the IBM Tivoli Change and Configuration Management Database to meet the specific requirements of an organization's Change Management process.

Although the product ships with a set of Change workflows aligned with the best practices described in the Information Technology Infrastructure Library (ITIL)¹, Version 3, as well, other users might want to customize the existing workflows or create new ones to fit the organization's process better. Particular organizations might have highly complex processes requiring many steps and the participation of multiple roles, whereas other organizations might have much simpler processes that do not require all of the functions implemented in the workflows that ship with the product.

¹ Information Technology Infrastructure Library and ITIL are both registered trademarks of the UK Office of Government Commerce.

The end-to-end design of the entire Change Management process for an organization is beyond the scope of this book. We focus primarily on those areas of the Change Management process that impact tool configuration and how you can best configure Tivoli Change and Configuration Management Database to support your organization's process needs.

Readers looking for more comprehensive coverage of the entire Change Management process have various sources from which to choose:

- ▶ IBM Integrated Service Management website:
<http://www.ibm.com/ibm/servicemanagement/us/en/>
- ▶ Tivoli Unified Process Composer website:
<http://www-01.ibm.com/software/tivoli/products/unified-process-composer/>
- ▶ ITIL documentation (note that a subscription is required; this is not free content)
 - ITIL Service Strategy, ISBN 0113310455
 - ITIL Service Design, ISBN 0113310471
 - ITIL Service Transition, ISBN 011331048X
 - ITIL Service Operation, ISBN 0113310463
 - ITIL Continual Service Improvement, ISBN 0113310498

We cover the following topics:

- ▶ 4.1, "Process role identification" on page 303
- ▶ 4.2, "Change Management workflows" on page 305
- ▶ 4.3, "Standardizing Changes with job plans" on page 313
- ▶ 4.4, "Configuration item considerations for Changes" on page 327

4.1 Process role identification

In this section, we explain important steps to take when starting out with Tivoli Change and Configuration Management Database. The concept of the *role* is central to any process discussion. Any process will include a number of discrete roles to be performed by process participants. Each specific role has associated with it certain actions or tasks that anyone in the role must perform as part of executing the process.

4.1.1 Individuals and roles

A role is not the same as a specific individual; a given individual can perform one or more roles within the scope of an entire process. For example, suppose that Chuck is the manager of both Sarah and Franz. Suppose as well that Chuck owns certain configuration items (CIs) to be affected by a Change that Franz is planning to implement.

In this scenario, Franz can perform both the *Change Owner* and *Change Implementer* roles; Franz is responsible for the end-to-end management of the specific Change, and will be performing the actual technical implementation tasks. Chuck in turn might be performing both the *CI Owner* and *Change Manager* roles. As the CI Owner, Chuck has a stake in the Change outcome as the owner of an IT resource to be affected by the Change. As the Change Manager, Chuck oversees the end-to-end process execution for all Changes and ensures the continued quality of the organization's Change Management process.

The actual roles, including both the names and definitions, can vary from organization to organization. Tivoli Change and Configuration Management Database implements a number of default roles based on those defined in the *IBM Tivoli Unified Process (ITUP)*, which is itself strongly aligned with the best practices described in ITIL.

Be aware that we distinguish between process *roles* and the *Roles* application in Tivoli Change and Configuration Management Database. Tivoli Change and Configuration Management Database does implement a few of the Change Management process roles as MAXROLE objects, which are the objects managed through the Roles application. But Tivoli Change and Configuration Management Database implements roles variously as one or more of MAXROLE objects, Security Groups, and Person Groups. You can combine these as well (for example, defining a MAXROLE to map to a Person Group), depending on the requirements of the process.

Most organizations will begin by mapping their identified roles onto those provided by the product. Not every organization will have every role that Tivoli Change and Configuration Management Database provides, and various organizations might have more roles, which tool administrators will need to configure in the product.

4.1.2 Process role considerations

Tivoli Change and Configuration Management Database administrators need to consider various questions when deciding how precisely to implement a newly identified process role in the tool:

- ▶ Does the role have specific access requirements, such as being the only role to have the ability to modify certain objects?
- ▶ Does the role require access to a well-defined set of information, such as might be captured in a Start Center?
- ▶ Does the identity of the individual performing the role vary with each iteration through the process, or can it be tied consistently to a specific individual or group of individuals?
- ▶ In the case that the individual specified to perform a given role is unavailable, who performs the role?
- ▶ Will the role be assigned specific tasks within the flow of a process?

The answer to each of these questions can help guide you in a specific direction. For example, if the role will need to perform specific tasks which can be assigned to the role during a process iteration, then you probably need to create a corresponding MAXROLE object using the Roles application. If the individual to perform the role will vary from iteration to iteration (Change Owner is a good example), then the MAXROLE has to be defined for dynamic evaluation, but if the role will always be performed by one of a defined set of individuals, then you might want to define the MAXROLE so that it evaluates to a Person Group containing the Person objects for those individuals.

The identification, definition, and configuration of the objects required to represent the roles required by any given process are activities common to the implementation of any product based on Tivoli's process automation engine, such as Tivoli Change and Configuration Management Database or Tivoli Service Request Manager. For step by step instructions on performing each of the tasks required for these activities, see the CCMDB 7.2.1 documentation found in your product's InfoCenter:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.ccmdb.doc_721/ccmdb-homepage.html

4.2 Change Management workflows

Tivoli Change and Configuration Management Database, Version 7.2.x, implements significant changes from Version 7.1. One of the most significant changes in this version is the way in which the Change Management process is automated. In Versions 7.2 and later, the majority of the process flow is implemented by Tivoli's process automation engine workflows.

The workflows that ship with the product are robust and fairly complex. Although we do not describe in detail every single step in the workflows, in this section we describe the high level flow as implemented in the PMCHGITLV3 workflow, which ships with Tivoli Change and Configuration Management Database, versions 7.2. and later. We also illustrate a few points in key subflows where administrators can modify the workflows to tailor them to the organization's specific process. We also discuss a few areas, although we do not explore these as fully, where implementers might want to make Changes.

4.2.1 The PMCHGITLV3 workflow

Although you can create your own Change Management workflows from scratch and route the Change through them using the Route Workflow action button, found at the top of the window in the Changes application, we advise most users to begin by using the default PMCHGITLV3 workflow that ships with the product and adapt it and its subflows to their specific requirements. The PMCHGITLV3 workflow is based on the best practices described in ITIL V3 and can serve as a sound starting point for most organizations. You can route a Change specifically through this workflow by clicking the ITIL V3 Change button in the Changes application when a specific Change is open, as shown in Figure 4-1.

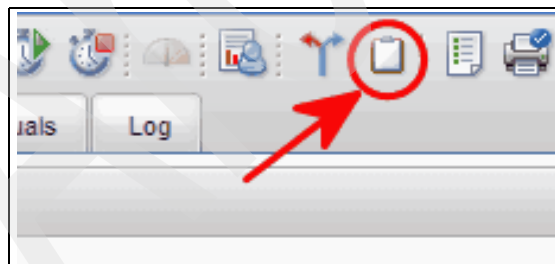


Figure 4-1 ITIL V3 Change button in Changes application

This workflow will step the Change through the following process activities using the specified subflows:

1. Accept and Categorize Change: Implemented by the PMCHGACCV3 workflow
2. Assess Change: Implemented by the PMCHGASSV3 workflow
3. Schedule Change: Implemented by the PMCHGSCHV3 workflow
4. Authorize Change: Implemented by the PMCHGAUTH workflow
5. Implement Change: Implemented by the PMCHGIMPV3 workflow
6. Review and Close Change: Implemented by the PMCHGREVV3 workflow

Although not all organizations will have precisely these steps within their Change Management process, most will have activities that correspond roughly to those listed here, making it advisable for Change Management architects to investigate and understand the content that ships with the product before embarking on a lengthy customization project.

To open the **PMCHGITLV3** workflow, whether you want to clone it, modify it, or simply examine it, first access the *Workflow Designer* application by selecting **Go To → System Configuration → Platform Configuration → Workflow Designer** or **Go To → System Configuration → Platform Configuration → Workflow Designer (Advanced)**.

Although the Workflow Designer (Advanced) application does provide additional capabilities over the standard application, these are not required for our discussion, and users can choose whatever version of the application they are most comfortable for the following activities. We use captures from the Advanced version, because the interface is slightly cleaner, and the Canvas tab displays additional workflow details that we find useful.

After accessing your chosen version of the Workflow Designer, enter the name of the workflow into the Process column in the search filter and press Enter. See Figure 4-2.

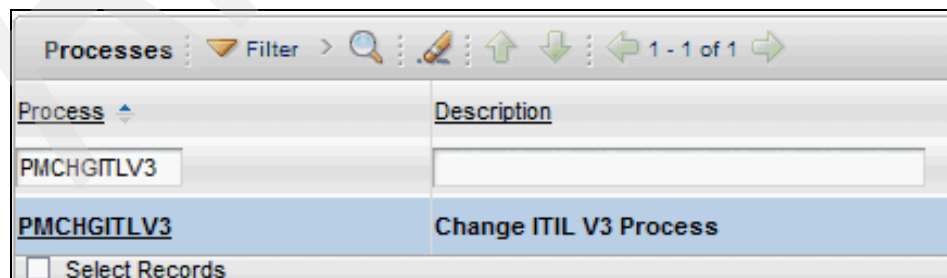


Figure 4-2 Selecting the PMCHGITLV3 workflow in the Workflow Designer

Click the workflow name to open it and display the process flow in the Canvas tab. See Figure 4-3 and Figure 4-4.

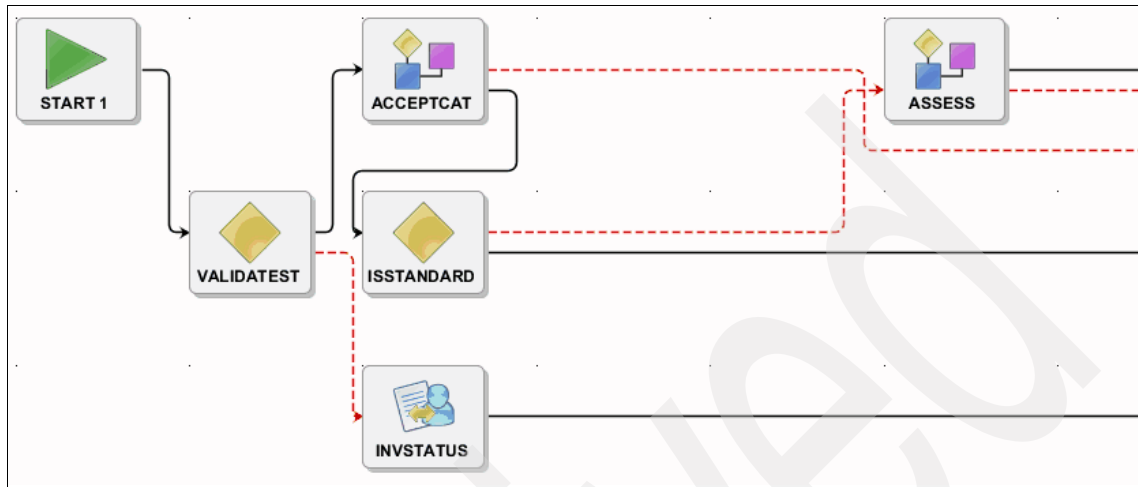


Figure 4-3 First half of PMCHGITLV3 workflow viewed using Canvas in Workflow Designer (Advanced)

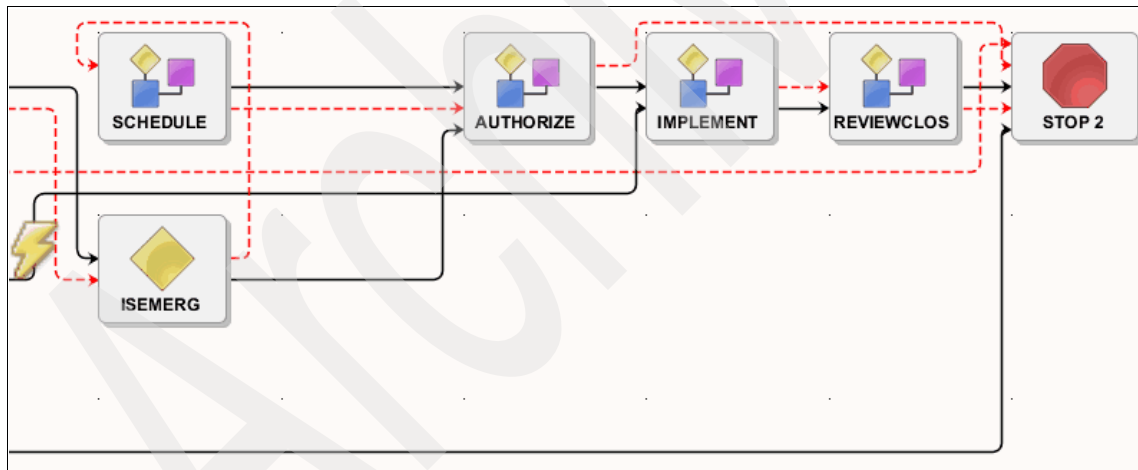


Figure 4-4 Second half of PMCHGITLV3 workflow viewed using Canvas in Workflow Designer (Advanced)

Notice that the majority of the workflow logic is actually implemented by subprocesses corresponding to each of the activities we listed earlier in this section. Open and examine each of the subprocesses using the same method as that used to open PMCHGITLV3. Administrators will probably find the majority of workflow customizations they initially make will apply to these subprocesses, unless their organization's Change Management process differs so drastically from that in the tool that the configured activities do not apply at all.

4.2.2 Determining Change priority

One of the first tasks that a good Change Management process must do is to determine the priority of each planned Change. When there is a conflict in scheduling or allocating resources to Changes, the organization needs a mechanism to determine which Change takes priority. The Accept and Categorize Change activity includes the task of assigning each Change a priority. In the Tivoli Change and Configuration Management Database, the PMCHGCALPR workflow performs this task and is called as a subprocess by the PMCHGACCV3 workflow, which is responsible for the entire Accept and Categorize Change activity. Figure 4-5 shows the PMCHGCALPR workflow as it displays in the Canvas tab of the Workflow Designer (Advanced) application.

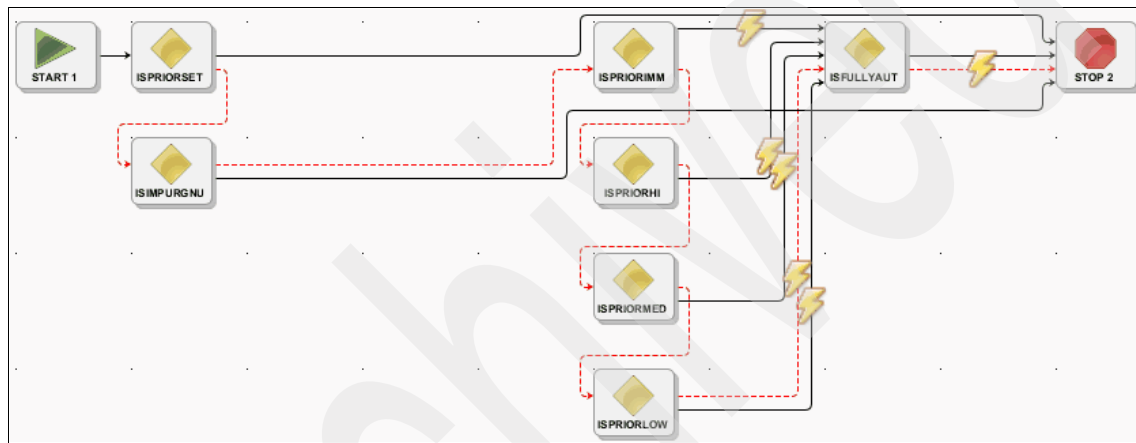


Figure 4-5 PMCHGCALPR workflow viewed using the Canvas in Workflow Designer (Advanced)

This workflow requires no direct user involvement and assigns no tasks; it just implements a basic algorithm for automatically calculating and assigning a value for the priority of a Change. The one exception to the assignment process is also a good example of a point at which an administrator can make a simple modification to accommodate the specific requirements of an organization.

The first node in the workflow is a *condition node*. This node checks whether the Change Owner has already assigned a priority manually. If the Change Owner specifies a priority manually, the workflow is routed around all of the calculations based on risk and impact. Certain organizations might not want to allow Change Owners to assign a priority manually, but rather require that priority only be calculated on the basis of agreed-upon criteria, so that priority is assigned consistently. Modifying the PMCHGCALPR workflow for this purpose is a simple Change that illustrates the method by which administrators can modify any out-of-the-box workflow to meet specific requirements.

4.2.3 Modifying the PMCHGCALPR workflow

The steps required to modify an existing workflow and put the modification into effect are as follows:

1. Create a new revision of the existing workflow.
2. Make the desired Changes to the new revision.
3. Verify the validity of the modified workflow.
4. Disable the previous revision.
5. Enable the new revision.
6. If necessary, activate the new revision after deactivating the previous one.
This is only necessary for top-level workflows, not for those only called as subprocesses.

Creating a new workflow revision

To create a new revision of an existing workflow, whether the workflow shipped with the product or was developed specifically for the organization, open the workflow according to the steps described in 4.2.1, “The PMCHGTLV3 workflow”. Then either click the **Create Process Revision** button as shown in Figure 4-6 or select **Select Action** → **Create Process Revision**.

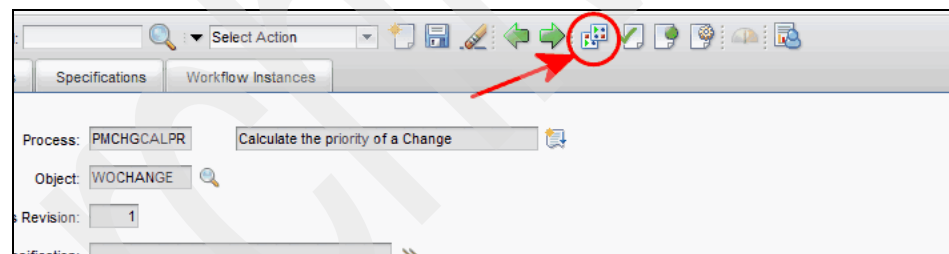


Figure 4-6 Create Process Revision button

This action creates an exact copy of the existing workflow and updates the revision attribute, as shown in Figure 4-7 on page 309.

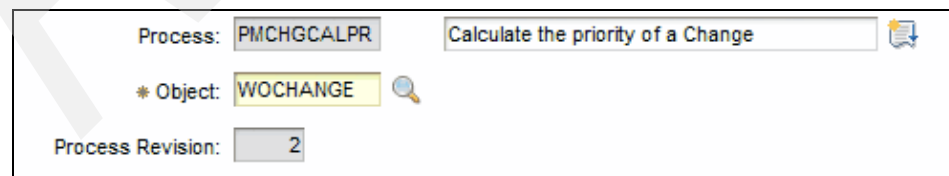


Figure 4-7 Workflow Designer showing the updated Process Revision attribute

In Figure 4-8 we can see that the Process Revision attribute has been updated, and we are now ready to make Changes. In our scenario, we want to force all Changes to be assigned a priority value calculated based on impact and risk, rather than allowing this value to be assigned manually. In a real implementation, we can probably make additional modifications; for example, we might want to disable the manual entry of values into the Priority field in the Changes application. In this book, however, we limit our discussion to making a simple workflow modification.

Make sure that you are working in the **Canvas** tab. In the workflow diagram, click the **ISPRIORSET** node to select it, then click the **Delete Selection** button.

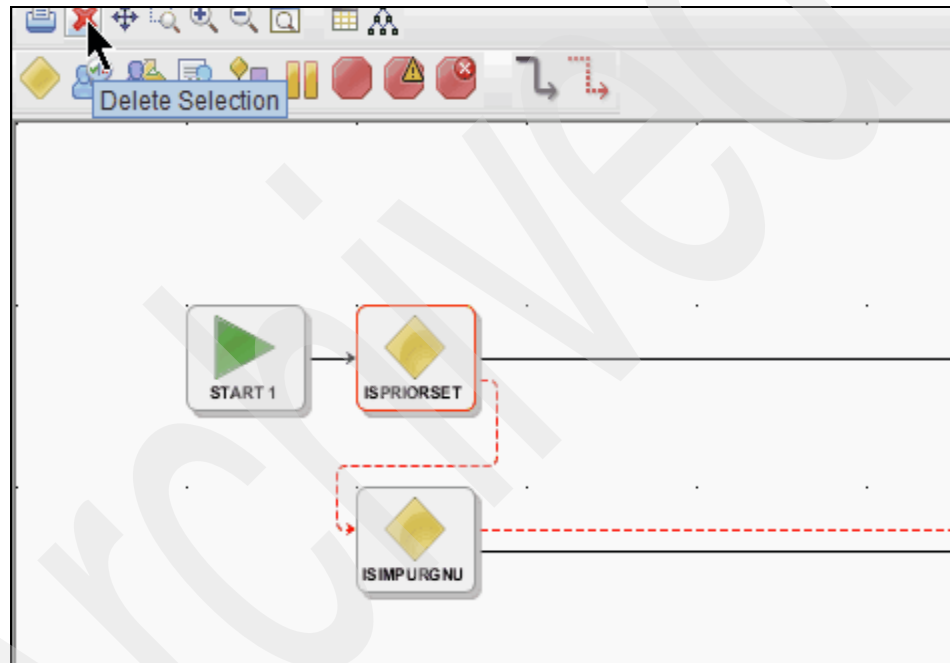


Figure 4-8 Deleting a selection in the Workflow Designer Canvas

When you delete the ISPRIORSET node, all incoming and outgoing connections are deleted as well. Our next step is to connect the START 1 node with which the workflow belongs to the ISIMPURGNU node. There are two types of connections one can make in Workflow Designer, positive and negative. For this scenario, we only need a positive connection. In the Canvas, positive connections are represented as solid black lines with arrowheads at the terminating point, and negative connections are represented as dashed red lines with terminating arrowheads.

To create the connection, click the **Positive Connection** button, then click the **START 1** node and drag to the ISIMPURGNO node. This will create the new connection path. One thing to note is that you will have the Positive Connection tool enabled until you click the **Positive Connection** button again, which will return you to the normal cursor as well. See Figure 4-9.

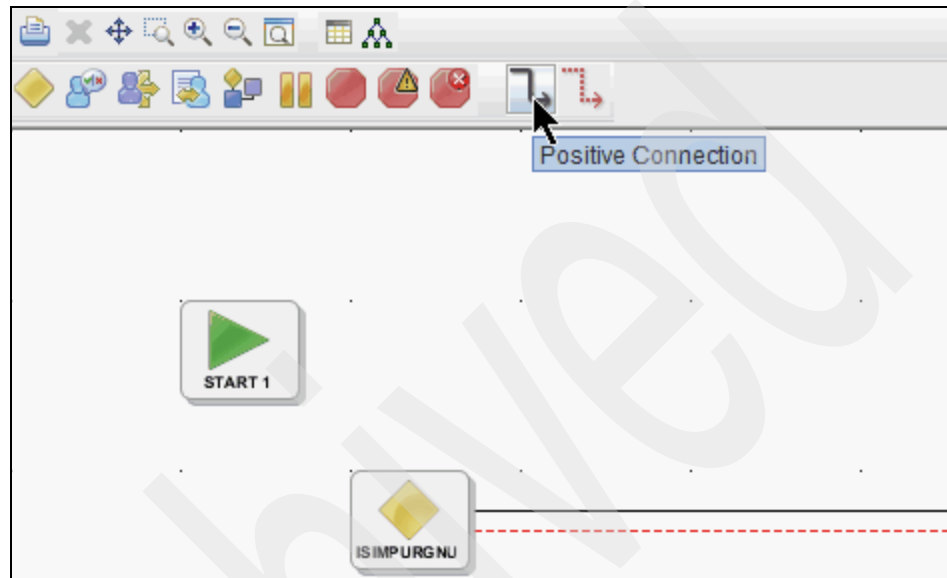


Figure 4-9 Adding a new Positive Connection to a workflow

After you have added the new connection, click the **Save Process** button or click **CTRL+ALT+S** to save your work. Being sure to do this after each modification will save you headaches if you let your session time out, because the Workflow Designer does not do any autosaving.

Priority considerations

In the original workflow, a Change has its priority assigned either manually (presumably by the Change Owner) or automatically based on a combination of risk and impact. But in both the original workflow and in our modified workflow, there is a case where no priority will be assigned within the PMCHGCALPR workflow. The ISIMPURGNO condition node checks if *either* impact *or* urgency is null (which means that the attribute has not been set). If either of these two conditions evaluates to true, then the ISIMPURGNO node will route the flow directly to the STOP 2 node, which terminates the entire workflow, and no priority is assigned to the Change. Perhaps, rather than leaving priority unset, we want to assign a default priority value to any Change for which the priority cannot be calculated. We do this by associating an action (which, if one does not exist, we can define using the Actions application) with the connection.

Although the Workflow Designer (Advanced) application will display any action associated with a connection directly in the Canvas (it is displayed as a lightning bolt icon over the connection path, if any action is assigned, and a mouse-over will display the details), the standard Workflow Designer does not. In either version of the Workflow Designer application, you can navigate to the Process tab to view the action objects associated with connections.

To see if a given connection has an action specified, navigate to the **Process** tab, then, in the **Process Nodes** pane, find the node from which the connection *originates* and select it to display the action details. In the table at the bottom of the window will be one row for each connection originating from the node, and any connection which has an action associated with it will have that specified in the leftmost column.

Figure 4-10 shows this view with the ISPRIORHI node selected. This node has two outbound connections. For one of these, the PMCHGSETPRIORITYHIGH action has been specified.

Process: PMCHGCALPR Calculate the priority of a Change Process Revision: 2 Object: WOCHANGE

Process Nodes Filter > 1 - 8 of 8

Title	Description	Node Type	Pointed To By
START 1	START 1	START	
STOP 2	STOP 2	STOP	
ISPRIORIMM	Is the priority of the Change immediate?	CONDITION	
ISPRIORHI	Is the Priority High	CONDITION	
ISPRIORMED	Is the Priority Medium?	CONDITION	
ISPRIORLOW	Is the priority of the Change Low?	CONDITION	
ISIMPURGNU	Is Impact or Urgency Null?	CONDITION	
ISFULLYAUT	Is fully automated	CONDITION	

New Node

ISPRIORHI Actions Filter > 1 - 2 of 2

Action	Instruction	To Node	Positive
>>	CONDITION 5	ISPRIORMED >>	☐
PMCHGSETPRIK >>	Is fully automated	ISFULLYAUT >>	☒

New Action

Figure 4-10 Process tab in Workflow Designer (Advanced)

To add our specific action to assign a default priority, select the row for the ISIMPURGNU process node. Two rows are displayed in the lower table. The row for which we want to add an action has the value STOP 2 in the To Node column. In the Action column click the double angle bracket to bring up a menu, then select **Select Value**. We are going to choose an existing action, but if you wanted to create an entirely new action or modify an existing action before selecting it, you can choose instead **Go To Actions**. See Figure 4-11.

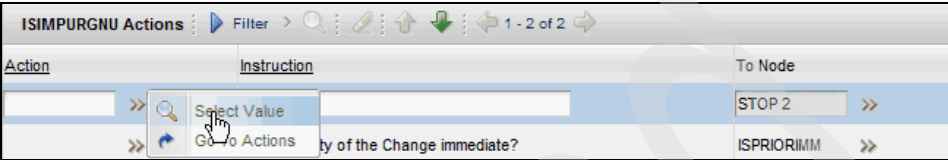


Figure 4-11 Selecting an action for a connection path

In the search dialog which displays, enter PMCHGSETPRIORITY in the Actions column and press Enter. You will see a set of pre-configured actions, each of which will assign a specific value to the Change priority attribute. You can pick whichever action best reflects what your organization wants as a default priority value; in our example, we chose PMCHGSETPRIORITYMED. See Figure 4-12.

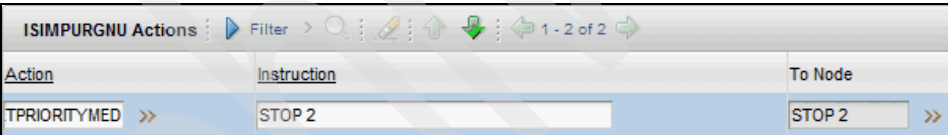


Figure 4-12 Details of connections showing selected action

4.3 Standardizing Changes with job plans

Very few Changes are wholly unique. As an organization implements more and more Changes over time, patterns emerge, enabling the organization to recognize a typology of commonly used Change types. Each type of Change is characterized by a set of activities, tasks, and roles required for the implementation and deployment of the Change.

As stated previously, most of the general roles and activities are part of the overall Change Management process, and we have discussed how these can be implemented using Roles, Person Groups, Start Centers, and Workflows in the Tivoli Change and Configuration Management Database.

But we have not yet addressed how to define a common set of implementation tasks used for all occurrences of a given type of Change. Next we discuss how to create a job plan to use as a template for any Change of a certain type and explain how to specify the tasks required for the implementation of the Change. In addition, we describe how certain process roles can be assigned depending on the type of Change.

4.3.1 Identifying common activities and tasks

Before defining a job plan, we must first select a type of Change and identify the activities tasks that are common to all instances of that Change type. When you begin this activity, ask yourself a set of questions and be sure to begin with a top-down decomposition; begin with the Change and work down through the activities required, then determine the tasks. Doing this ensures that you are pursuing a coherent approach across each of the Change types you identify and will help to ensure that each of the job plans that you create adheres to a consistent level of detail.

In 4.2.1, “The PMCHGITLV3 workflow”, we discussed the major activities in the Change Management process. Each Change that your organization manages will require a similar set of activities, but there will be a variation. Certain types of Changes, for example, require a more stringent assessment to determine the risk involved; others might require authorization by upper management, or might require little or no authorization at all. You can capture these characteristics, which include not only how the activity is performed, but in various cases, who performs it, as you define the types of Changes to be managed, and you will likely discover additional types of Changes over time.

You can also capture the implementation *tasks*. In this context, we use the work task to refer specifically to those steps required for the technical implementation, deployment, and activation of the Change. As with activities and roles, the tasks can vary widely from one Change type to the next. A very trivial Change might require only a single task which includes all of the technical steps required. A more complex Change, or a Change with very complex dependencies, might require additional steps which must be represented as additional tasks.

For our purposes, we are going to assume that we have identified a Change type for installing an operating system patch. We assume that a Change of this type, which poses very little to no risk, requires authorization only from the IT Operations Manager; one posing a moderate risk requires authorization from the IT Management Board; and one posing a fairly severe risk requires authorization from the executive layer.

Additionally, we assume that the following technical tasks are required for each Change of this type:

1. Stop applications
2. Install patch
3. Reboot system
4. Verify system function
5. Start applications
6. Verify application function

Although any individual performing one of the foregoing tasks can be considered a Change Implementer as far as their role in the broader Change Management process goes, this does not mean that the same individual can or even ought to perform all of these tasks. In addition to identifying the tasks that we need to perform, we might also want to identify which persons or groups own the responsibility for each task. We assume for our example that the System Administrators team owns tasks 2, 3, and 4, and the Application Administrators team owns tasks 1, 5, and 6.

4.3.2 Creating the job plan

In this section, we provide an overview of the Job Plans application and describe the structure of a job plan. Next, we describe the steps to create a job plan in Tivoli Change and Configuration Management Database and configure it with the information we gathered in the previous section. In the example that follows, we have already created the Person Groups for the System Administrators, Application Administrators, and IT Management Board, and we do not describe the steps for creating these objects.

Job plan overview

As we described in 2.2.5, “Job plans” on page 48, in the Tivoli Change and Configuration Management Database, job plans are created to organize activities, tasks, and participant roles required to execute work of a given type; in this case, Changes. The Job Plans application is used in many products based on Tivoli’s process automation engine, such as Tivoli Service Request Manager, not only in the Tivoli Change and Configuration Management Database, and in each one, the type of work activities defined using job plans might differ. This distinction is crucial; a job plan is not the same as a Change, but in this product, we create a job plan object to serve as a template Change Plan for one of a set of standardized Changes.

The Job Plans application

Figure 4-13 shows the Job Plans application. As in any other application in the product, a set of tabs provides access to windows for configuring specific aspects of the associated object. In addition to the two standard tabs, the Job Plans application also provides tabs for Work Assets and Specifications, but the tabs seen in the application can vary from job plan to job plan. What tabs are visible depends on the value specified for the Default WO Class attribute.

As shown in Figure 4-13, the job plan displayed has a value of CHANGE set, and because of this, three additional tabs are made available. Each of these additional tabs allows you to configure attributes of the job plan which are specific to managing a Change, but which are not be required for a job plan used for another kind of work.

To access the Job Plans application, select **Go To → Planning → Job Plans**, as shown in Figure 4-13.

The screenshot displays the 'Job Plans' application window. At the top, there is a header bar with 'Job Plans' on the left and 'Bulletins: (0)' and 'Go To Report' on the right. Below the header is a toolbar with a 'Find' search bar, a 'Select Action' dropdown, and various icons for file operations and navigation. A row of tabs is visible: 'List', 'Job Plan' (selected), 'Work Assets', 'Specifications', 'Change Template', 'Change Assessments', and 'Change Authorization'. The main content area shows the 'Job Plan' configuration for 'PMCHGNORML'. A text field contains 'Normal Change JobPlan'. To the right is an 'Organization' field with a search icon. Below this is a 'Details' section with a 'Status' dropdown set to 'ACTIVE', a '* Template Type' dropdown set to 'Process', and a '* Duration' field set to '4:00'. To the right of these are 'Default WO Class' set to 'CHANGE', 'WO Priority' (empty), 'Interruptible?' (unchecked), and 'Flow Controlled?' (checked). On the far right is a 'Responsibility' section with checkboxes for 'Supervisor', 'Crew', and 'Lead', all of which are currently unchecked.

Figure 4-13 Job Plans application

The *Change Template* tab is the first Change-specific tab. The fields on this tab provide a core set of attributes common to all Changes. When defining a job plan, be sure to specify the correct value for the Change Type attribute. The following values are available values:

- ▶ Normal: Changes of this type are handled by the regular Change Management process and receive no special processing.
- ▶ Standard: A standard Change is a frequently-performed Change whose risk and impact are well understood, often one whose implementation is automated, and which is therefore allowed to be deployed without the same scrutiny applied to other Changes,
- ▶ Emergency: An Emergency Change is one whose urgency is such that it cannot be routed through all of the normal Change Management activities and must receive priority processing. Typically such Changes are performed to resolve Incidents or Problems, but that is not necessarily always the case.

The fields in the Details section are not populated when you define the job plan; the Change Owner will fill these fields out when the job plan is applied to an actual Change. Fields here represent standard attributes for Changes within the attribute, and administrators might want to customize the application with any additional attributes defined for the WOCHANGE object class.

Use the *Change Assessments* tab to provide a list of required assessments for the Change type as well as to provide an estimate of the default risk of failure. The administrator defining the job plan will enter any technical assessments (an example is ensuring that the impact is probably understood) and business assessments (an example is assessing the risk, perhaps due to an audit exposure or due to losses which might occur in the case of an outage) into the required sections. For each required assessment, the administrator will also specify either an owner or owning group which determines who is responsible for performing the assessment. The specified owners are assigned assessment tasks by the PMCHGASSV3 workflow when the Change is routed through the main PMCHGITLV3 workflow.

The *Change Authorizations* tab provides a list of approvals required for a Change of the specified type. The Change Authorizations tab looks superficially similar to the Change Assessments tab, but there is one crucial difference. Before we discuss this difference, let us first return to the workflows used to manage a Change, specifically, the PMCHGAUTH workflow (Figure 4-14).

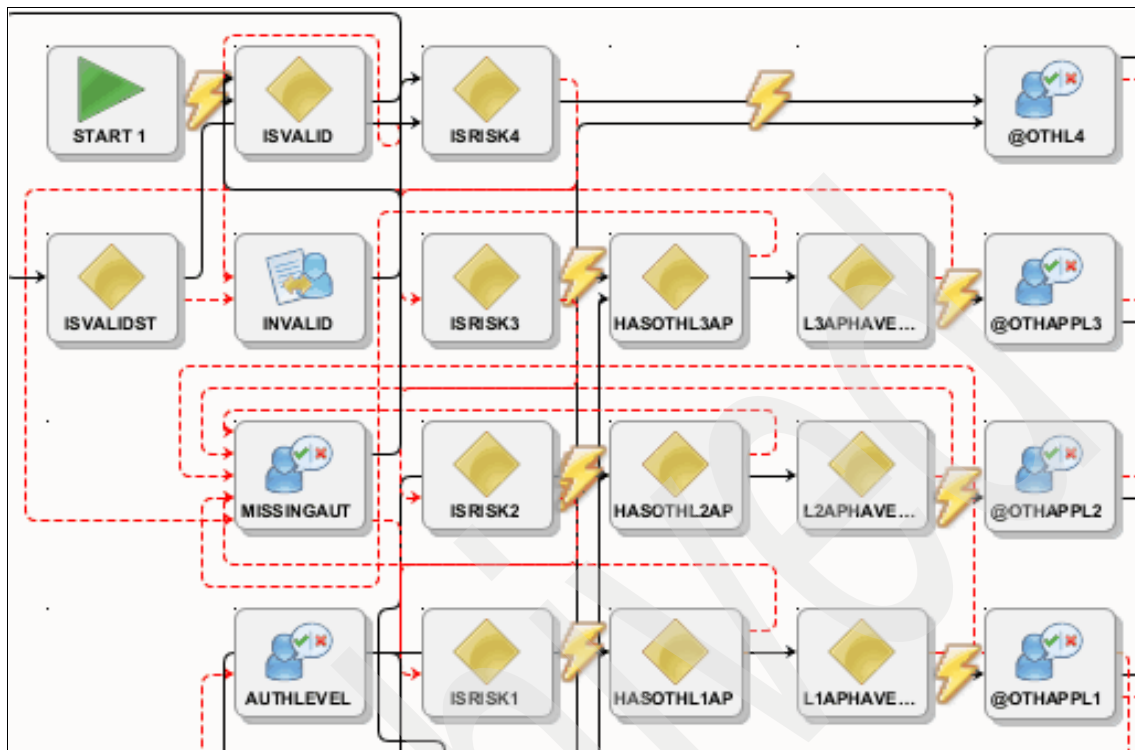


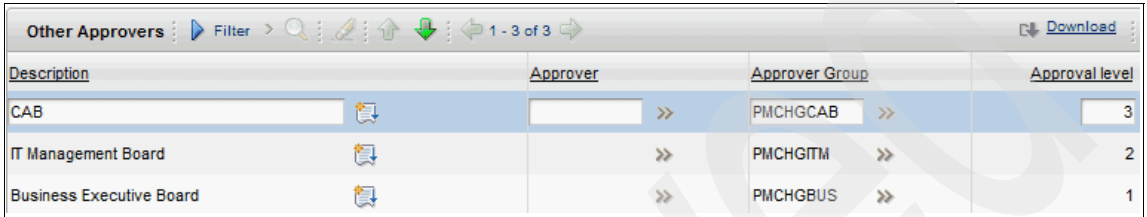
Figure 4-14 PMCHG AUTH Workflow Detail

Figure 4-15 shows a section of the PMCHGAUTH workflow in the Canvas Tab of the Workflow Designer. Note that the third column from the left includes a set of condition nodes, each of which checks the risk of the Change against a specific value. Depending on the risk, an approval task is assigned. Here the assignments are specified for a Change with a risk value of 2.

Assignments				
Role ID	Description	Send E-mail	Task Description	Time Limit
PMCHGAPPL2	Areas to Approve for Level 2	☐	L2 - Approve or Reject the Change	0:00
PMCHGAPL2G	Finds the default person in the group for approval level 2	☐	L2 - Approve or Reject the Change	0:00

Figure 4-15 Approval assignments for a Change with a risk of 2

Now return to the Job Plan application. In Figure 4-16 we see the approvers specified for a sample job plan. Each row defines a type of approval and the person or group who must provide the approval in order for the Change to proceed. Crucially, each row also includes an Approval level attribute. The value specified here determines how the approval tasks are assigned by the PMCHGAUTH workflow. So for a Change using this job plan with a risk of 2, the IT Management Board is required to provide approval. If the Change had a risk of 3, then the full Change Advisory Board (CAB) is required to approve the Change before implementation can proceed.



Other Approvers			
Description	Approver	Approver Group	Approval level
CAB		PMCHGCAB	3
IT Management Board		PMCHGITM	2
Business Executive Board		PMCHGBUS	1

Figure 4-16 Required Change approvals specified in a job plan

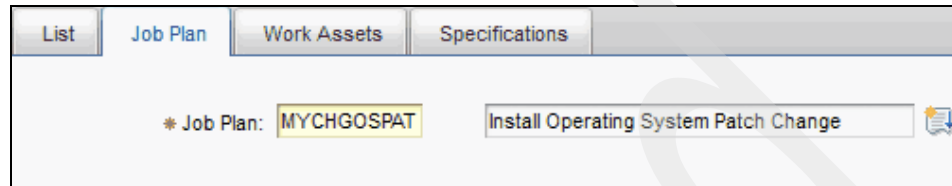
It is not always possible to know everyone required to approve a given Change in advance; these are the required approvals for *any* Change of the specified type and risk. When the actual Change record is created using the Changes application, the Change Owner can and often does specify additional approvers, such as the project manager or application owner, in the Authorization tab of the Changes application.

Job plan creation

Now we walk through the steps to create a sample job plan. We continue to use the sample Change type we described in 4.3.1, “Identifying common activities and tasks” on page 314 for installing a patch onto an operating system.

First we create a new job plan. When creating a new job plan, you can use the same two methods common to most Tivoli’s process automation engine applications. You can create an entirely new object from scratch by clicking the **New Job Plan** button, or, with an existing job plan open, you can create a copy of it to modify and save as a new job plan by selecting **Take Action → Duplicate Job Plan**. We use the latter method to avoid having to populate every attribute, and we advise administrators beginning with the Tivoli Change and Configuration Management Database to do the same, as the product ships with useful job plans that can serve as templates to help you get started.

We start with the PMCHGNORMAL job plan, which is a good default object to use for most Changes not provided for by other job plans. First, we open this job plan, then create a copy by selecting **Take Action** → **Duplicate Job Plan**. We enter MYCHGOSPAT for the name in the Job Plan field and enter a brief description as well; we have entered Install Operating System Patch Change in the example shown in Figure 4-17.



The screenshot shows a software interface with a tabbed menu at the top containing 'List', 'Job Plan', 'Work Assets', and 'Specifications'. The 'Job Plan' tab is selected. Below the tabs, there is a label '* Job Plan:' followed by a text input field containing 'MYCHGOSPAT'. To the right of this is another text input field containing 'Install Operating System Patch Change'. A small icon of a document with a plus sign is visible to the right of the second input field.

Figure 4-17 Duplicating an existing job plan

Notice that in Figure 4-18, the additional Change-specific tabs are not available. These do not appear until after you first save your new job plan (assuming that you have either duplicated an existing Change job plan or created one and specified the **Default WO Class** to be CHANGE).

Next fill in the job plan details. These attributes are populated in the **Details** pane of the **Job Plan** tab. In Figure 4-18 we show this pane with the most important attributes highlighted in red:

- ▶ **Template Type:** This attribute defines the type of template. For a Change, this attribute must always be set to *Process*.
- ▶ **Duration:** If you are defining a job plan to be used for a specific kind of Change with known implementation details, you might want to specify a default duration, which can assist in planning.
- ▶ **Default WO Class:** The same job plan can be used with various types of workorders, as all workorders share common characteristics (and extend a common parent class). While you might or might not want to enable your job plan to be used with other workorder types, such as Activity or PMCFGWO (the workorder type used for performing Configuration Management tasks), you must definitely make the default class (only one class can be the default) CHANGE.
- ▶ **Progress Map Sequence:** Determining the correct progress map sequence is a useful, if tricky, part of defining a job plan. As the Change flows through the process, its state changes to signify where in the process the Change currently sits. You define the sequence of progress states using this attribute. Preferably, refer to one of the existing job plans for use with Changes and copy the sequence used there initially, but you might find that your organization has a unique sequence entirely.

The screenshot shows a 'Details' window with the following fields and values:

- Status: DRAFT
- Template Type: Process
- Duration: 4:00
- Default WO Class: CHANGE
- Classification: (empty)
- Class Description: (empty)
- Launch Entry Name: (empty)
- Include Tasks in Schedule? ☒
- Start Constraint Offset: (empty)
- Finish Constraint Offset: (empty)
- Progress Map Sequence: (empty)
- WO Priority: (empty)
- Interruptible? ☐
- Flow Controlled? ☒
- Suspend Flow Control? ☐
- Flow Action: (empty)
- Flow Action Assist? ☐

Figure 4-18 Job plan details

Configuring the values for most of these attributes is relatively straightforward, but configuring the values for the Progress Map Sequence is a bit trickier and requires planning. Before you start creating job plans, ideally you have identified the set of states through which a Change passes within the organization's Change Management process and the order in which the states occur. As we stated previously, a good way to start is by copying the sequence used in the job plans that ship with Tivoli Change and Configuration Management Database.

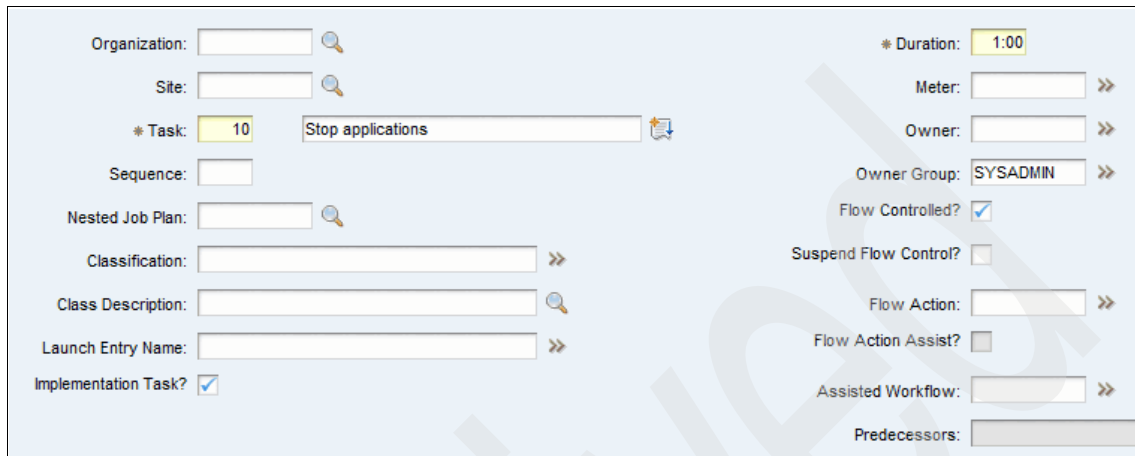
Creating job plans can vary from being very simple to very complex. For job plans intended to be used with Changes, these are the most important steps:

1. Specifying the implementation tasks and owners
2. Specifying required assessments
3. Specifying required approvals

Specifying implementation tasks

In the section of the *Job Plan* tab labelled *Job Plan Tasks*, click the **New Row** button to begin entering implementation tasks. Users of Tivoli Change and Configuration Management Database prior to Version 7.2 might remember specifying tasks for obtaining technical assessments, business assessments, and approvals, as well as scheduling the Change. Beginning with Version 7.2, *only implementation tasks* are listed in the Job Plan Tasks section. This is a major change and users need to be aware of it. Limit the tasks entered in this section to those directly related to the technical implementation and deployment of the Change; assessments and approvals are entered using the tabs specifically for those items, respectively, which is a topic we describe shortly.

After clicking **New Row**, enter a description for the task and either a Person or Person Group to own the task. In Figure 4-19 we show the first task in our Install Operating System Patch Job Plan.



The screenshot shows a web form for defining a job plan task. The form is divided into two main sections. The left section contains fields for 'Organization', 'Site', '* Task' (with a dropdown menu showing '10'), 'Sequence', 'Nested Job Plan', 'Classification', 'Class Description', 'Launch Entry Name', and 'Implementation Task?' (checked). The right section contains fields for '* Duration' (set to '1:00'), 'Meter', 'Owner', 'Owner Group' (set to 'SYSADMIN'), 'Flow Controlled?' (checked), 'Suspend Flow Control?', 'Flow Action', 'Flow Action Assist?', 'Assisted Workflow', and 'Predecessors'. The form has a light blue background and a white border.

Figure 4-19 Job plan task details

We have added the Stop applications task and assigned it to the SYSADMIN Person Group. We have left the duration as the default value of one hour. When you specify implementation tasks, try to specify reasonably accurate values for the durations, because this will make planning for Changes much more effective.

You do not need to specify a classification, flow action, or other value for your job plan, and you probably need to postpone doing so until you are comfortable with the basics. The Flow Action attribute can be useful for job plans, because it allows you to trigger automated actions as each task is marked complete, but for most Changes, this is probably not necessary.

As you grow your inventory of job plans, you might find that your organization essentially reuses portions repeatedly, perhaps as a common part of larger job plans. In that case, you might want to use a *Nested Job Plan*, which allows you to specify that one job plan is composed of one or more subordinate plans. You might find this useful for complex Changes that require coordination across multiple towers, but, again, we advise not doing this initially; instead, develop a good set of basic job plans that work for your most common Changes first.

Specifying required assessments

Now you need to identify and specify any assessments, whether technical or business, which your organization requires when performing this type of Change. Again, users of the product prior to Version 7.2 might recall doing this by adding assessment tasks to the *Job Plan Tasks* table, but because versions of Tivoli Change and Configuration Management Database beginning with Version 7.2 manage the Change flow primarily through workflows, this is no longer necessary. Instead, move to the *Change Assessments* tab of your job plan.

First, specify the Change type, which can be one of Emergency, Normal, or Standard. Then specify the failure probability. This attribute can have values of High, Medium, or Low. Failure probability is an estimate of the risk that the Change will not succeed. It is often difficult to estimate this accurately for infrequently performed Changes, and even so, the actual failure probability will likely vary over each use of the job plan, depending on variables such as who is implementing the Change, the scope of the Change, and other factors which you cannot know when creating the job plan. For any Change that the organization will be performing frequently enough to merit creating a job plan as a template, you can probably specify Medium or Low.

Next, determine if your job plan will require any technical assessments. If yes, then press the **New Row** button in the appropriate section and enter the assessment details. First, enter a brief description (not more than a short sentence, at most), then specify the assessment type.

Figure 4-20 shows the available types for technical assessments:

The 'Select Value' dialog box displays a table of technical assessment types. The table has two columns: 'Value' and 'Description'. The 'Value' column lists various assessment types, and the 'Description' column provides a more detailed description for each. The 'Application' type is highlighted in blue.

Value	Description
<u>Application</u>	<u>Application assessment type</u>
<u>Capacity</u>	<u>Capacity assessment type</u>
<u>Database</u>	<u>Database assessment type</u>
<u>Network</u>	<u>Network assessment type</u>
<u>OS</u>	<u>Operating System assessment type</u>
<u>Security</u>	<u>Security assessment type</u>
<u>Server</u>	<u>Server assessment type</u>
<u>Storage</u>	<u>Storage assessment type</u>

Figure 4-20 Technical assessment types

We specify just one assessment. For the description, we enter Operating System assessment, then we select OS as the type. Next we need to specify who will own performing the assessment. Remember that the workflow for assessments will assign a task for each assessment we enter, so we need to specify an owner to whom the assessment will be assigned. In our example, shown next, we have selected the OSLEAD Person Group, which we have created to represent a team of lead OS admins. Whom you specify for performing assessments will vary, depending on your organization's structure and responsibilities, but be sure that you have identified in cooperation with your Change Process Owner or Change Manager who needs to own assessments before the job plan can be made available for use with actual Changes. See Figure 4-21 on page 324.

The 'Details of Operating System Assessment' form shows the 'Assessment Type' field set to 'OS'. The 'Owner' field is empty, and the 'Owner Group' field is set to 'OSLEAD'.

Assessment Type	Owner	Owner Group
OS		OSLEAD

Figure 4-21 Details of Operating System Assessment

You specify the business assessments in exactly the same fashion as you do the technical assessments, but the available assessment types differ. By default, they are Financial, Operational, and SOX (which means an assessment for Sarbanes-Oxley compliance/exposure). If your organization has other regulatory or compliance requirements (whether these are internal or external requirements), you can add these as additional values which can be selected for the type of the business assessment.

Adding additional business assessment types

In many cases where a product based on Tivoli's process automation engine presents a user with a list of values from which the user can select, the available choices are controlled by a *Domain*. Domains in Tivoli Change and Configuration Management Database and related products are objects used to group related values which can be specified as the value of an attribute, typically. In many cases, administrators can modify the values within a domain to customize the choices in Select Value dialogs.

To add an additional business assessment type, first go to the Domains application by selecting **Go To → System Configuration → Platform Configuration → Domains** (after saving your job plan first, of course). In the search filter, enter PMCHGBUS for the value of the Domains column and press Enter. You will only have one Domain available, the PMCHGBUSASSESTYPE Domain. Next to this domain is the **Edit Detail** button (circled in red in Figure 4-22). Click this button.

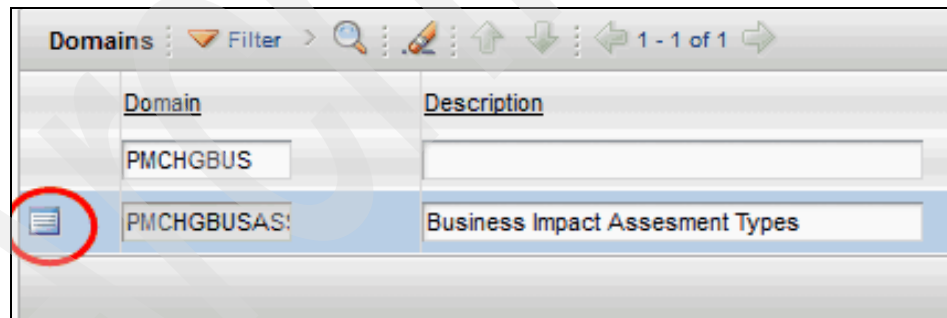


Figure 4-22 Domains application

In the dialog that displays, click **New Row** and enter a value and description for your assessment type. Then click **OK**. Now return to the Job Plans application and find your draft job plan, then return to the **Change Assessments** tab. Add a business assessment and click the **Select Value** button for the type. You can see the value you added to the Domain as an available choice. Figure 4-23 shows the type we added, 714B, to represent compliance with a hypothetical organizational standard.

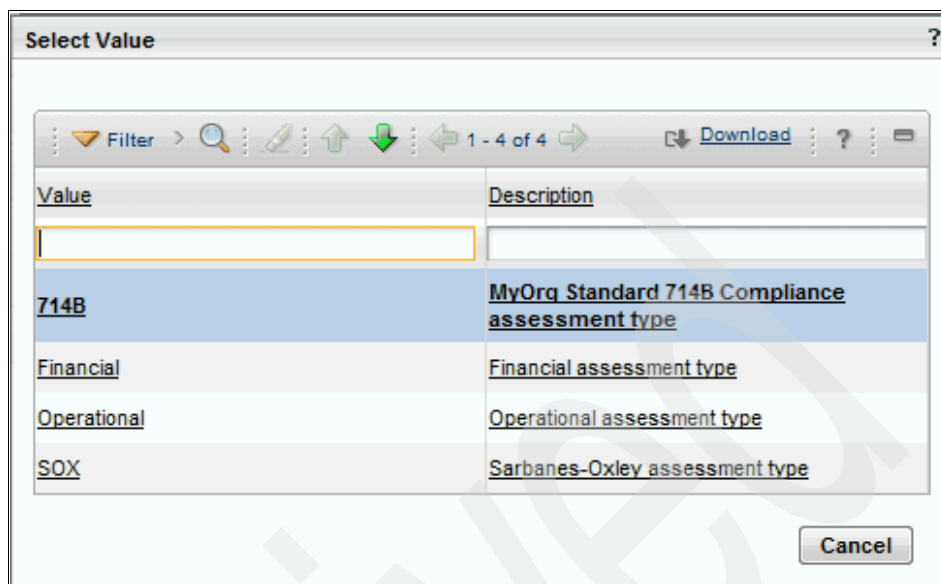


Figure 4-23 Business assessment types showing newly added value for PMCHGBUSASSESSTYPE Domain

Specifying required approvals

We discussed a few of the crucial details for specifying approvals earlier in this section, but now we revisit this topic so that we can actually add a required approval to our job plan. In the **Job Plans** application click the **Change Authorization** tab. Note that the section in which we add approvers is labeled **Other Approvers**. Approvals we specify here will be required for any Change to which this job plan is applied, even if the Change already has approvers specified directly.

So when deciding whether to specify an approval in the job plan or only in Changes, think whether the approval might be required for *any* instance of such a Change - if this is true, then you probably need to add the approval in the job plan. If not, you might want to let Change Owners specify the approvals in the Change itself, because they can vary from Change to Change, even for the same type of work. We assume here that all Changes for installing an operating system patch require approval from an operations manager, so we add an approval to our job plan.

In the Other Approvals section, click the **New Row** button. Enter a description and either an owner (which is a Person) or owner group (which is a Person Group). Recall next that the level of approval is related to the risk of the Change; if you want the same person or group to approve a Change regardless of the risk,

you can enter three rows and assign the same ownership for each row, being sure to set the Approval level to 1 for one entry, 2 for the next, and 3 for the third.

Before other users can apply your job plan to a Change, you will need to change the status. All newly created job plans have a status of DRAFT. You will need to change this to *ACTIVE* to make your job plan available. To do this, click the **Change Status** button, select **ACTIVE** as the new status, click **OK**, and then save your job plan.

4.4 Configuration item considerations for Changes

Often, when creating a Change, the Change Owners will specify one or more configuration items (CIs) affected or impacted by the Change. If your organization is using Tivoli Change and Configuration Management Database to implement your Configuration Management process, you might want to work with your Configuration Librarian or whichever role owns management of CIs in your organization to ensure that certain values which play a role in managing a Change are set when the CI is stored. Change workflows use these values to perform various evaluations and routing operations, so neglecting them can make your Change Management less effective.

Figure 4-24 shows selected attributes from the configuration items application, which you use to manage authorized CIs.

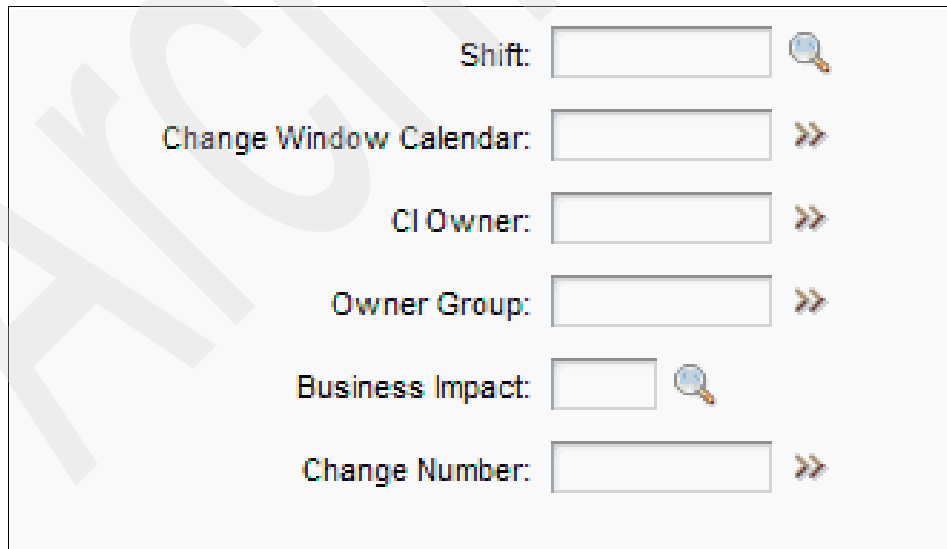
A screenshot of a web-based form for configuring items. The form contains several input fields with labels to their left. The labels are: 'Shift:', 'Change Window Calendar:', 'CI Owner:', 'Owner Group:', 'Business Impact:', and 'Change Number:'. Each label is followed by an input field. To the right of each input field is a small icon: a magnifying glass for 'Shift:', 'Change Window Calendar:', and 'Business Impact:'; and a double right-pointing arrow for 'CI Owner:', 'Owner Group:', and 'Change Number:'. The background of the form is light gray, and the text is black.

Figure 4-24 Configuration item attributes

4.4.1 Change Window calendars and blackout periods

Many organizations restrict the times when Changes can be performed, whether for all CIs or for selected CIs, to specific intervals known as *Change Windows*. A Change Window is a specific predetermined period which usually occurs at regular intervals during which Changes can be scheduled. Changes scheduled outside the Change Window often require special processing and are only allowed under very specific circumstances, such as when the Change is required to resolve a critical Incident. An authorized CI can be associated with a Change Window by specifying a value for the **Change Window Calendar** attribute.

An organization might also want to prohibit *any* Changes to particular CIs during certain periods. Many organizations will prohibit Changes to financial systems during key accounting periods, when workloads are heavy and schedules are extremely tight, as the risk of an outage due to a Change cannot be tolerated. You can implement this in Tivoli Change and Configuration Management Database by specifying a *blackout period* for a CI.

Creating and specifying both Change Windows and blackout periods is an important part of ensuring effective Change Management, but because these are specified in part by associating them with CIs, it is easy for a Change Administrator to overlook them. The **PMCHGSCHV3** workflow in Tivoli Change and Configuration Management Database, versions 7.2. and later, checks CIs to be affected by a Change for both of these objects and uses them to determine if the scheduled time and date for the Change are valid in accordance with organization policies. It is best that Change and Configuration Administrators work closely together to ensure that the appropriate objects get created and specified for any important CIs.

In this section, we discuss the following topics:

- ▶ Creating a Change Window Calendar and associating it with a CI
- ▶ Viewing the Change Window Calendar
- ▶ Creating a Blackout Period and associating it with a CI

Creating a new Change Window calendar

To specify a Change Window, click the **Detail Menu** button next to the Change Window Calendar attribute. You can then either select a defined Change Window or create a new one using the **Change Window Calendars** application. In this example, we create a new window and set it as the value of the attribute for a CI. To create a new Change Window, select **Go To Change Window Calendars** from the detail menu. If you are not currently working with an authorized CI, you can reach the application by selecting **Go To → Change → Change Window Calendars**. Then click the **New Change Window Calendar** button to create a new object. See Figure 4-25 on page 329.

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Figure 4-25 Creating a new object in the Change Window Calendars application

Figure 4-25 shows the view after clicking the **New Change Window Calendar** button. The application presents you with a blank calendar showing the current month and a few attributes. First, name your calendar. If a group of CIs will always be subject to the same Change Window restrictions, then create one calendar to use with all of them and name it in as clear a fashion as possible. In our example, we call our calendar CWCSTAND, indicating that this is the standard Change Window calendar to be applied to a CI if it is not subject to further restrictions. Next, enter a description if desired and specify start and end dates.

You must specify both dates before you can save the calendar. If the calendar is already in effect, you can likely use the current date as the start date. Certain organizations reevaluate their Change Window policies on a periodic basis, and you might want to set your end date to coincide with the next review cycle. The danger with doing so is that, if the cycle is quite short, and you forget to update the calendar, it will no longer be valid for evaluating Change schedules. So you might want to set the calendar out further. However, it is best to establish a regular cycle for revisiting the configured Change Window calendars and ensuring that they still align with your organization's Change policies, whichever method you choose. After choosing the dates, save your calendar.

Next you must specify the actual Change Windows. Click the **Change Window Schedule** tab. In the details section, you will see two tabs, Administrative and Merged. The application shows all scheduled windows under the Administrative tab. Where two or more windows overlap within the same calendar, the application will display them as a single merged window in the Merged tab. You can create both regularly scheduled and custom Change Windows. Most Change Windows will be scheduled, meaning that the organization has planned them ahead of time regardless of whether any actual Changes are planned and scheduled or not. Scheduled Change Windows. You can use a custom window for a one time window that will not reoccur. You can use such a window, for example, for a large release that requires a number of interlocking Changes.

Most users will primarily use scheduled Change Windows which reoccur at regular intervals. To create such a window, select **Select Action → Schedule Change Windows**. You will see the **Set Schedule** dialog, which you use to create a scheduled window. You will need to set the frequency, which you specify using both unit (day, week, etc.) and unit count. For a biweekly window, for example, you can click the radio button in the row for weeks and enter the value 2 in the field between the words **Every** and **week(s)** in the Set Schedule dialog. You must also enter a start time, which you actually type into the field, rather than using a selection control, as one might normally expect. You can use either twelve hour or twenty-four hour formats - the tool understands both and will convert to the appropriate locale format for display.

Next enter the length of the window in the **Duration** field. By default, the Start Date and End Date will be set to the same values specified for the calendar, and most users will probably not need or want to change these values, but if you want the window schedule to be valid only for a portion of the calendar, then set that here. After entering all of the attributes, click the **Preview** button at the right of the dialog to see a projected schedule, which you can examine for correctness before actually creating the schedule. If you are satisfied with the schedule, click **OK**.

Figure 4-26 shows the schedule we specified; we are scheduling a weekly window that starts every Saturday evening at 10:00 PM and lasts nine hours.

Set Schedule

Select a date interval and then Preview to see the dates.

* Duration: 9:00

* Start Date: 10/8/10

* End Date: 12/31/10

Notes:

☐ Every day(s), at time

☒ Every 1 week(s), on day Saturday at time 10:00 PM

☐ Every month(s),

☐ on day 01 at time

☐ on the first Sunday of the month, at time

☐ Every year(s),

☐ on January on day 01 at time

☐ on the first Sunday of the year, at time

Preview

Date Preview

10/9/10 10:00 PM
10/16/10 10:00 PM
10/23/10 10:00 PM
10/30/10 10:00 PM
11/6/10 10:00 PM
11/13/10 10:00 PM
11/20/10 10:00 PM
11/27/10 10:00 PM
12/4/10 10:00 PM
12/11/10 10:00 PM
12/18/10 10:00 PM
12/25/10 10:00 PM

OK Cancel

Figure 4-26 Creating a Change Window schedule

After you have returned to the application, your Change Windows will now appear in the tables under both tabs. In Figure 4-27 you can see how the Change Windows appear under the Administrative tab. Note that if you want, you can modify either or both of the Start Time and End Time fields for any scheduled window individually, and you can also delete individual windows without modifying the underlying schedule. This is particularly useful when, for example, a regularly scheduled window needs to be rescheduled or cancelled.

Change Windows Filter 1 - 12 of 12				
	Change Window Type	Start Time	End Time	Duration
▶	WEEKLY	10/9/10 10:00 PM	10/10/10 7:00 AM	9:00
▶	WEEKLY	10/16/10 10:00 PM	10/17/10 7:00 AM	9:00
▶	WEEKLY	10/23/10 10:00 PM	10/24/10 7:00 AM	9:00
▶	WEEKLY	10/30/10 10:00 PM	10/31/10 7:00 AM	9:00

Figure 4-27 Scheduled windows as displayed in the Administrative tab

All of the windows you create will also appear in the Merged tab, but you cannot modify them there.

You can also navigate to the *CI*s tab in the Change Window Calendars application to see which CIs use the currently open window, but if you are working with a newly created calendar, even if you reached the application by navigating directly from the Change Window Calendar attribute of a CI, no CIs are displayed yet.

Viewing the Change Window Calendar

Now click back on the **Change Window Calendar** tab. You will see the same calendar format view as before, but each window is displayed as an entry on the day on which it occurs, listed by the duration. Be aware that a window that overlaps two or more consecutive days will show as an entry on both days. For example, ours is a nine hour window spanning two hours on Saturday and seven hours on Sunday, so the application displays it as 2:00 on Saturday and 7:00 on Sunday. You can see the entire window by clicking either of the entries to bring up a Merged view as shown in Figure 4-28.

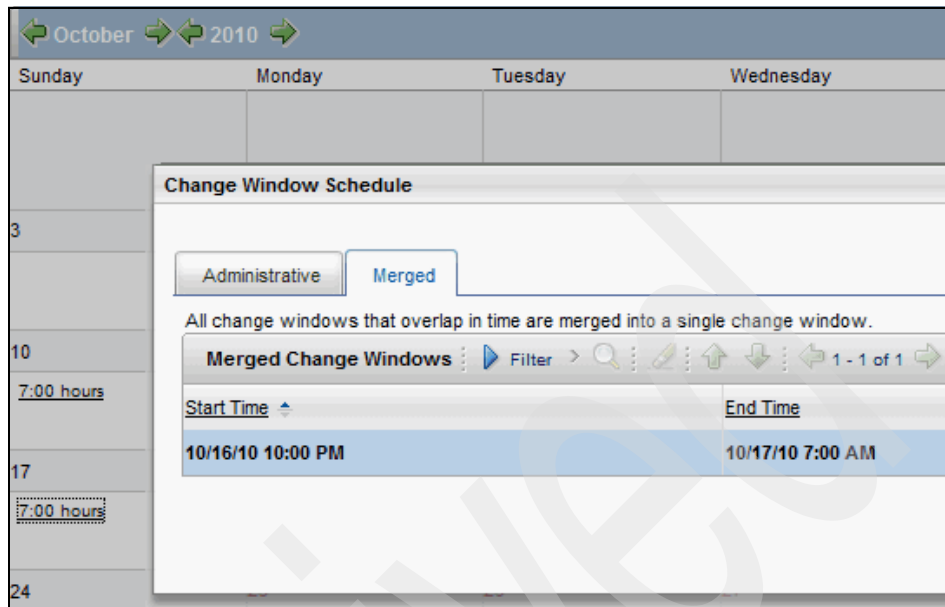


Figure 4-28 Updated Change Window Calendar showing scheduled entries and merged schedule view

The view here is precisely the same as what you see by going to the Change Window Schedule tab, except that it is limited to just the single window, and you can even modify the window by clicking the **Administrative** tab within the dialog.

To return to the configuration items application and the CI you were modifying, click **Return With Value** at the top right of the window.

Creating a blackout period

To create a blackout period, access the application by selecting **Go To → Change → Blackout Periods**. Then click the **New Blackout Period** button. You specify four primary aspects of a blackout period:

- ▶ **Duration:** Indicates when the blackout period is in effect and for how long
- ▶ **Type:** Indicates one of two types, LOCKDOWN and RESTRICTED, that blackout periods can have
- ▶ **Approvers;** Indicates who must approve any Change scheduled during the blackout period
- ▶ **Scope:** Indicates to which CIs the blackout period applies, which can be either all CIs or a selected set of CIs

Begin by naming the blackout period and entering a description. Use the description field to provide a reason for the blackout period. Because blackout periods are normally set centrally according to organizational policy, individual CI and Change Owners might not always be aware of them, and you can reduce confusion by ensuring that your description provides a brief description of the policy reason behind the blackout period. If your organization publishes policies, perhaps to an internal website, you might want to consider including a link to the policy documentation here as well.

In our sample scenario, we create a blackout period called 4QRESULTS that represents the period during which the company provides quarterly earnings results to investors, which is often a very critical and high-visibility period. In our description, we enter Financial results for 4Q 2010 - Corporate Std 76F, which provides both a justification and a reference to the corporate standards document containing the relevant policy. Keep in mind that blackout periods, unlike Change Windows, cannot be configured to reoccur periodically; you cannot, for example, schedule a regular quarterly blackout period. If you needed blackout periods for each quarterly earnings report within a year, you have to create four separate blackout period objects, one for each report.

After specifying start and end dates and times, we need to specify the type of blackout period. Tivoli Change and Configuration Management Database provides two types of blackout periods, LOCKDOWN and RESTRICTED. A restricted blackout period must only allow the scheduling of critical Changes, for example, a Change to fix a critical security vulnerability, but one that has not yet been exploited. During a *lockdown* blackout period, by comparison, Change Owners must not schedule *any* Changes. Typically, only Changes to resolve a critical incident will be allowed during a lockdown period, but this can vary from organization to organization. Due to the visibility associated with earnings reports, we choose the type LOCKDOWN for our blackout period.

Next, we need to specify who must approve any Changes scheduled during our blackout period. For each approval required, click **New Row** in the approvers section and select either an individual approver or an approver group. You do not need to enter any description or specify an approval level; *every* specified approver must give approval for *any* Change scheduled during the blackout period.

The last part of creating a blackout period is specifying the scope. Figure 4-29 shows the scope details section of the Blackout Periods application. By default, the box for **Applies to all CIs** is checked on every new blackout period; if this attribute is selected, then the blackout period will be applied to every authorized CI in the system.

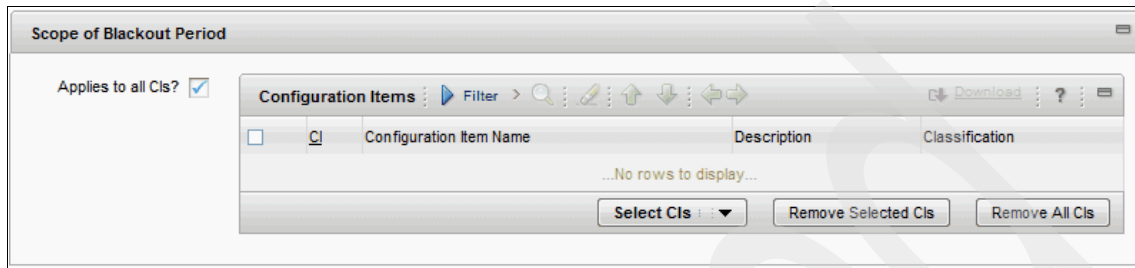


Figure 4-29 Scope of blackout period section in Blackout Periods application

While many blackout periods will be this broad in scope, others might apply to selected CIs or sets of CIs. In the latter case, uncheck the **Applies to all CIs** box and then click the **Select CIs** button as shown in Figure 4-30.

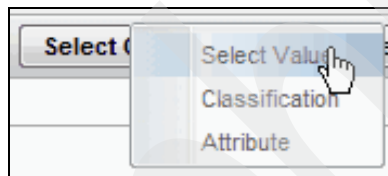


Figure 4-30 Options for selecting CIs in Blackout Periods application

You can select the CIs to which the blackout period will apply according to the following methods:

- ▶ **Select Value:** By selecting them individually from a list of all authorized CIs
- ▶ **Classification:** By browsing to the desired portion of your classification structure, then selecting from the CIs having that classification
- ▶ **Attribute:** By specifying one or more attributes and the values to match, then selecting from the CIs having the specified values

In our case, we apply our blackout period to our business applications. To do this, we choose the **Classification** option for selecting the CIs. After selecting this option, the application gives us a dialog that includes a classification browser and a CI table.

In Figure 4-31 we have clicked the BUSINESS_APP classification, giving us a list containing the LOAN MANAGEMENT application. To apply the period to this CI, we click the box at the left of the row containing the CI and then click **OK**. If there had been more CIs in this classification, we might have selected the others at the same time simply by checking the appropriate boxes before clicking **OK**.

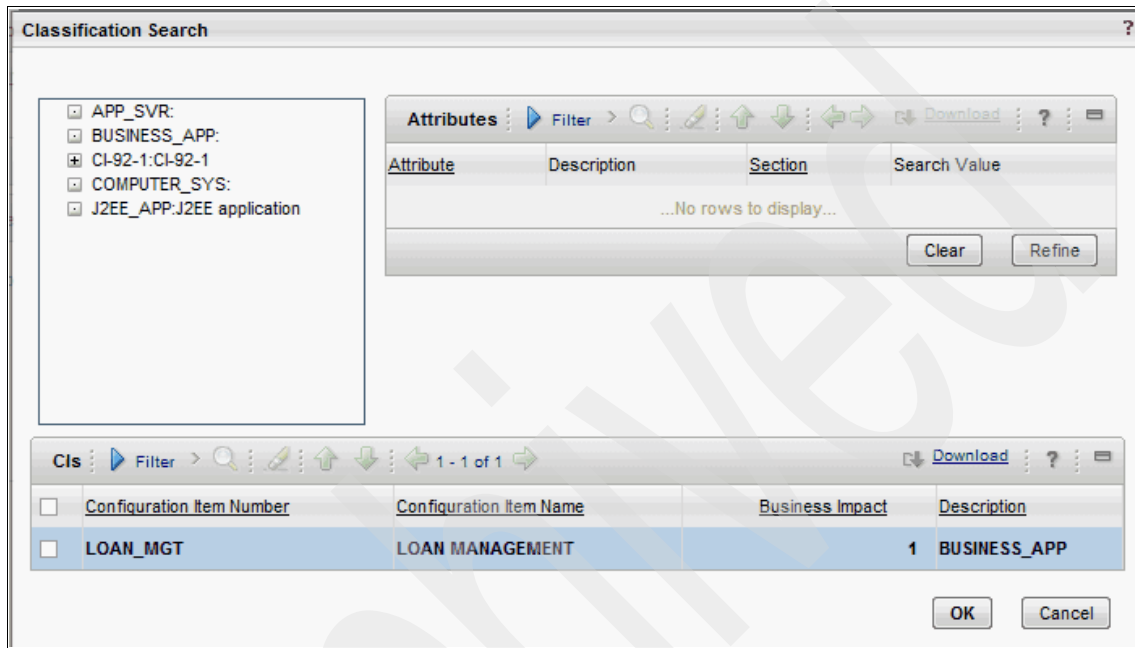


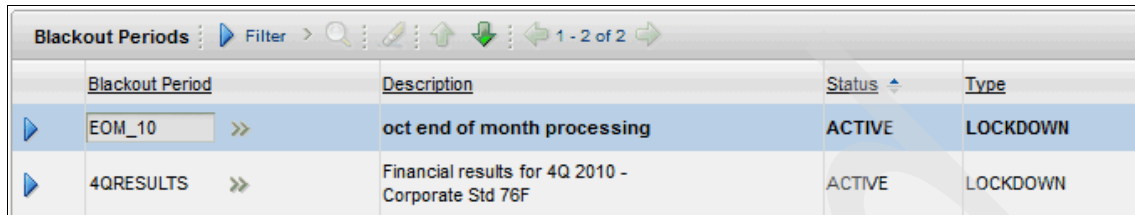
Figure 4-31 Selecting CIs by Classification

After returning to the application, we finish by changing the status to ACTIVE using the **Change Status** button, then saving our blackout period.

Viewing the blackout periods for a specific CI

Now that we have specified that the blackout period 4QRESULTS applies to the LOAN MANAGEMENT CI, we want to see what, if any, other blackout periods apply to the same CI. Although we can browse through all configured blackout periods and look at each one individually, it is easier and more effective simply to use the configuration items application, to which we return by selecting **Go To** → **IT Infrastructure** → **Configuration Items**.

In the configuration items application, we enter LOAN MANAGEMENT into the first column and press Enter. This returns just the single result for the application, and after selecting it, we see the CI details in the Configuration Item tab. At the bottom of the window is the Blackout Periods section, shown Figure 4-32.



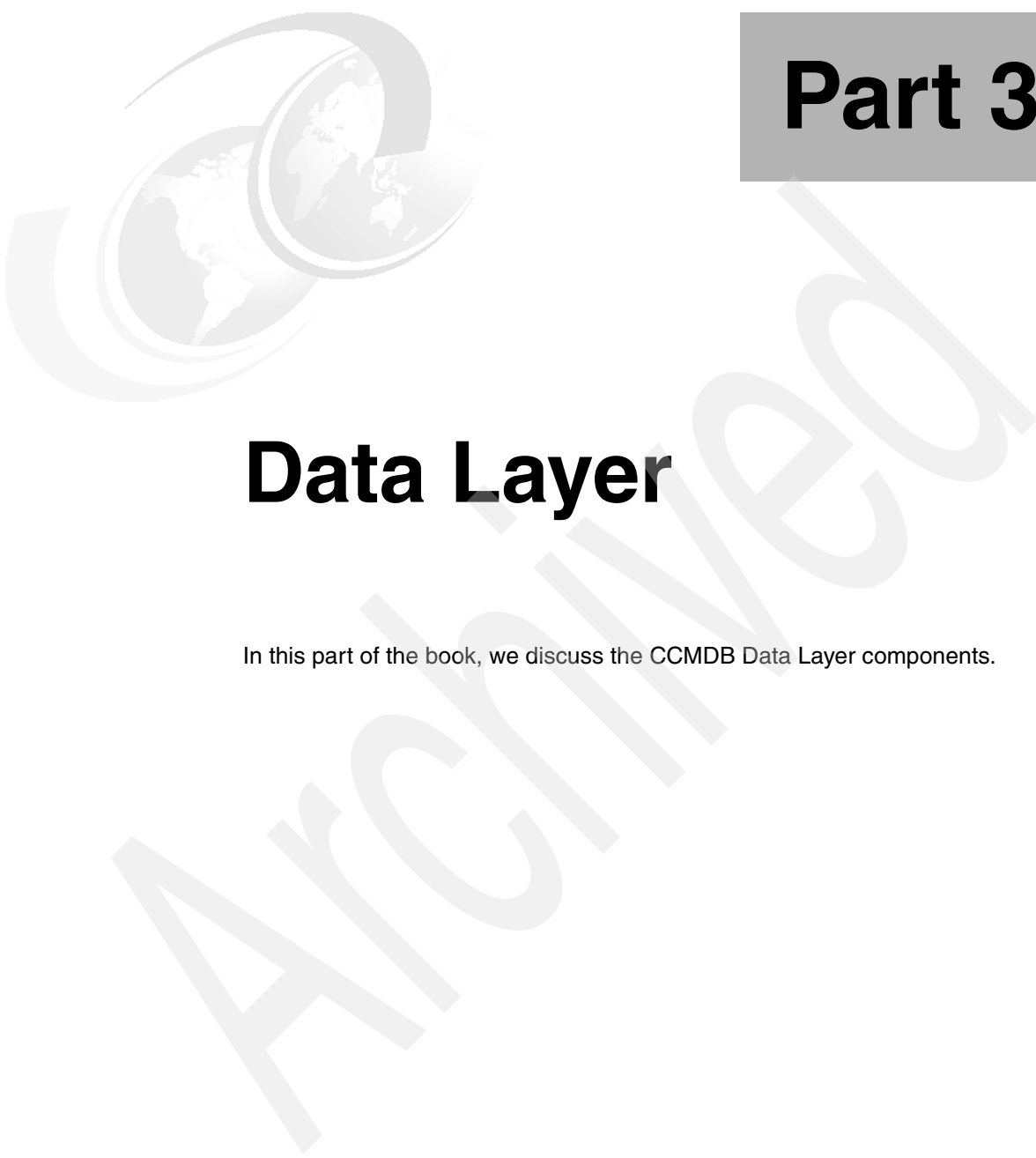
Blackout Period	Description	Status	Type
EOM_10	oct end of month processing	ACTIVE	LOCKDOWN
4QRESULTS	Financial results for 4Q 2010 - Corporate Std 76F	ACTIVE	LOCKDOWN

Figure 4-32 Blackout Periods for Loan Management application

You can see that two blackout periods apply to our CI, one of them is the 4QRESULTS blackout period we just created, the other is the end of month processing period for October.

4.4.2 Specifying a CI Owner

We finish our discussion by addressing a very simple but often neglected topic, that of the CI Owner. The CI Owner ensures that the CI is in a known and supported state and is therefore a stakeholder in any action (such as a Change) that might change the operating state of the CI. Determining ownership of a given CI often requires organizational analysis; for a server system, is the CI Owner the business owner of hosted applications, or is it the team responsible for supporting the system operationally? Understand your organization's policies on CI ownership and work with your Configuration Librarian to have an owner assigned to as many of your authorized CIs as possible. and ensure that the CI Owners are aware of what roles the CIs that they own can play in the business.



Part 3

Data Layer

In this part of the book, we discuss the CCMDB Data Layer components.

Configuration item hierarchy

In this chapter we provide an overview of the configuration item hierarchy, which is a key component of the Data Layer of the CCMDB architecture.

We cover the following topics:

- ▶ 5.1, “CCMDB Data Layer” on page 342
- ▶ 5.2, “Thinking about your business processes” on page 343
- ▶ 5.3, “Discovered configuration items” on page 343
- ▶ 5.4, “Actual configuration items” on page 345
- ▶ 5.5, “Authorized configuration items” on page 348
- ▶ 5.6, “CIs in the CDM” on page 351
- ▶ 5.7, “How CIs are represented in the CCMDB” on page 352
- ▶ 5.8, “Working with CIs in CCMDB” on page 357

5.1 CCMDB Data Layer

The CCMDB Data Layer contains three data spaces that hold CIs, process artifacts, and relationships between these objects. This layer provides everything from a dependency mapping of the discovered environment to a specification of authorized CIs that define the specific aspects and characteristics of CIs that you want 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. Figure 5-1 reflects the three CI data spaces of the CCMDB solution, its interoperability, and its relationship to other data structures, such as Process Artifacts.

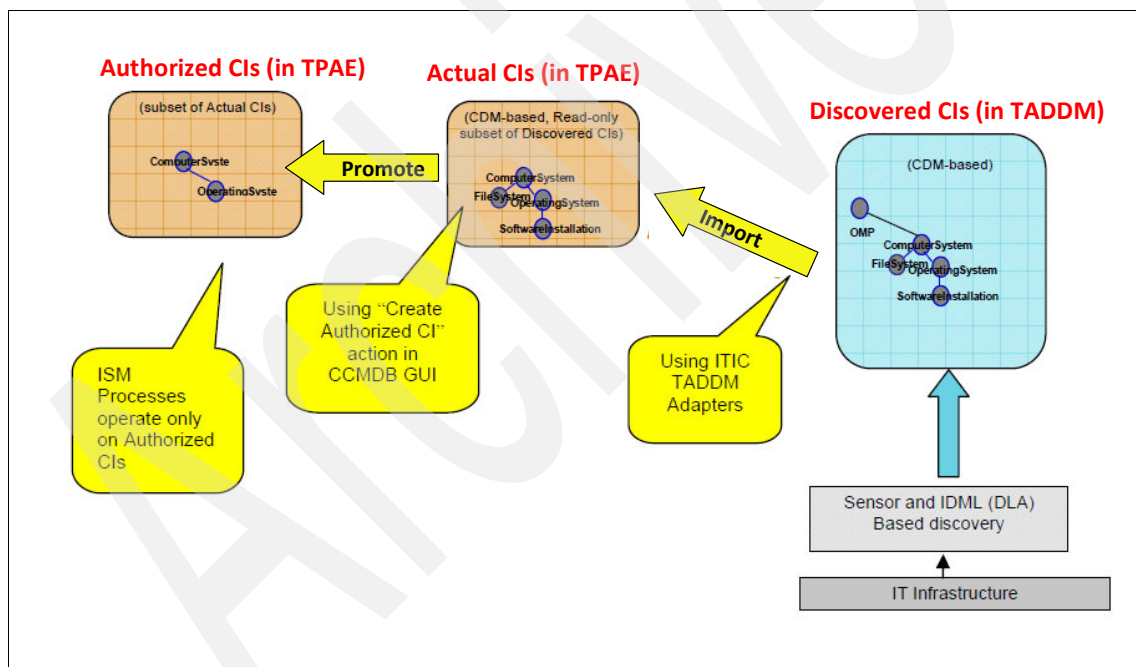


Figure 5-1 CCMDB Data Layer contains

In Figure 5-1, the high granularity of details in reference to the discovered CIs implies that the data discovered might be more detailed than needed for actual or authorized CIs. When the data is imported as an actual CI, the unnecessary details can be filtered out.

Before you can use your CCMDB solutions to manage your IT systems and processes, you design, configure, and manage your configuration items (CIs).

5.2 Thinking about your business processes

Before you can decide what CI data you're interested in, you must think about the business processes you intend to automate:

- ▶ Are you planning to start with Configuration Management and then add Change Management to that?
- ▶ Will you need Release Management or Service Desk capabilities?
- ▶ Within each of these larger areas, which specific processes do you plan to automate first?

Normally an expert in Information Technology Infrastructure Library (ITIL), or an expert in your existing business processes, will help you make these decisions. The IBM Tivoli Unified Process (ITUP) and ITUP Composer tool, which are included with CCMDB, can also help you through this phase.

5.3 Discovered configuration items

The discovered CI data space of the CCMDB, also referred to as the discovered CI space, 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 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 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 space.

A discovered CI and its relationships can be quite deep and complex, as indicated in Figure 5-2. 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 contains relationships with other CIs.

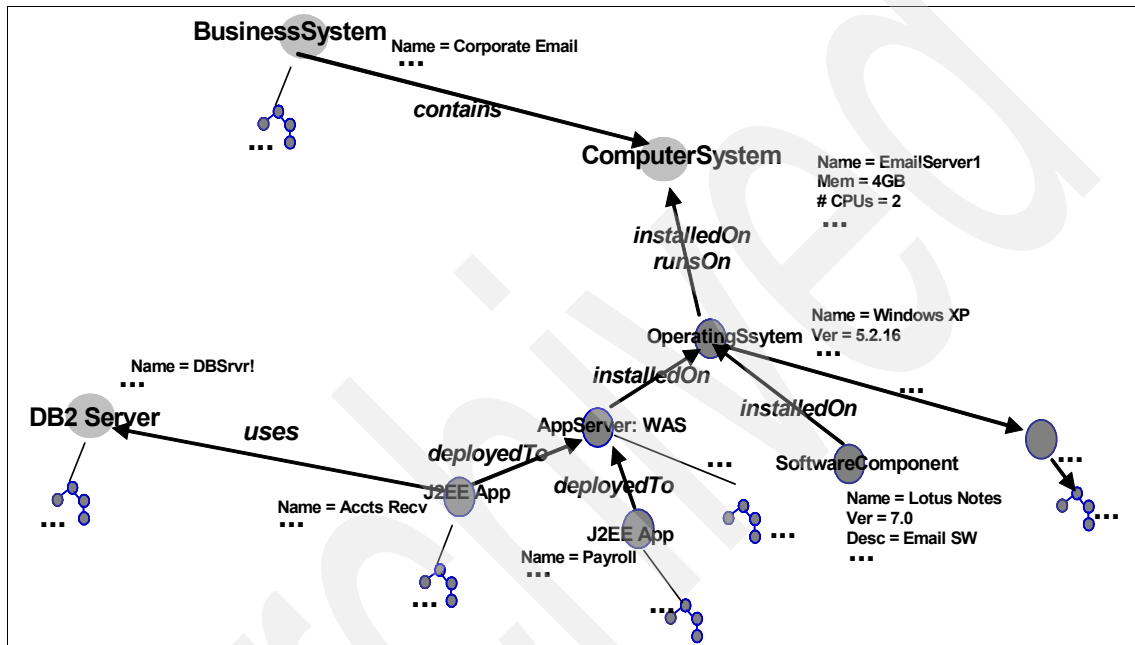


Figure 5-2 Interconnected graph of CI dependencies

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.

5.4 Actual configuration items

A filter can be applied that results in a subset of the CIs and relationships of the discovered CI space being copied over into the actual CI data space. Doing 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 can 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 come under Process Management. The server systems can 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 space.

5.4.1 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.

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 Integration Composer 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.

5.4.2 Example of CI types

Figure 5-3 shows an example of what CIs and relationships are copied over from the discovered CI space if the ComputerSystem CI type was specified in the filter to bring computer systems into the actual CI space,

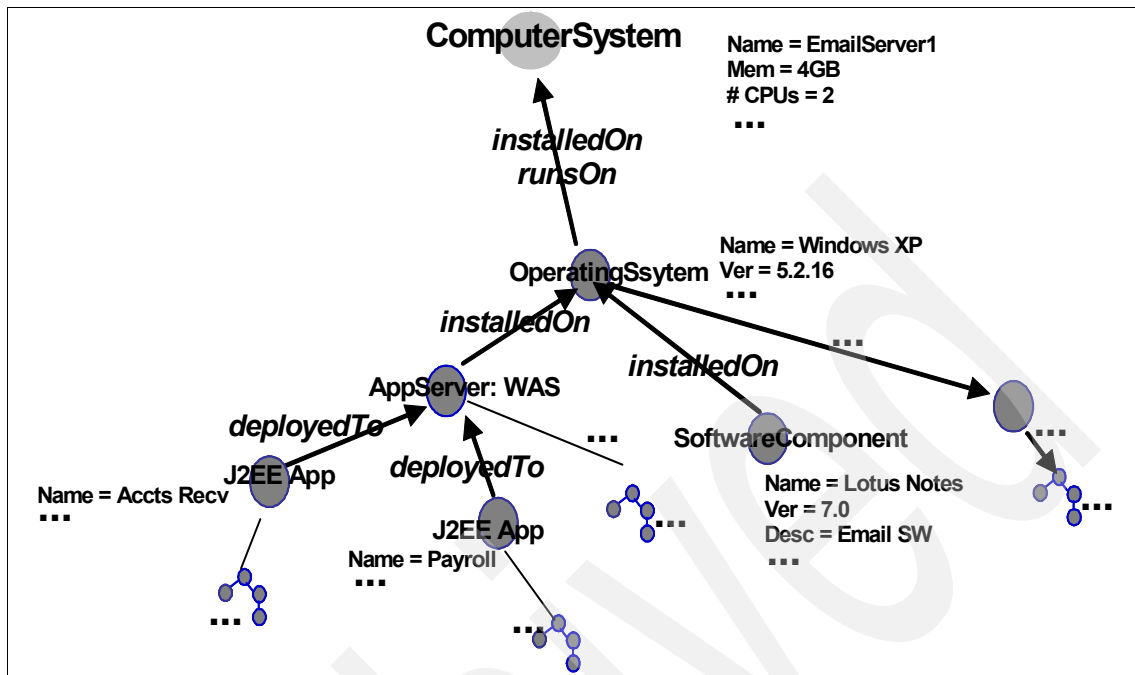


Figure 5-3 Filtered CI data in actual CI data space

Note that the business service called Corporate Email and the database server called DBSrvr1 (and their relationships to EmailServer1) shown in Figure 5-2 on page 344 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 can contain a significant number of objects and attributes, it is best 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.

Whereas 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.

By default, the depth level is set to 3. This means for any discovered CI instance that is being copied into the actual CI space it will only follow relationships in a downward notion to the third level in the CI child tree. In Figure 5-4, when bringing a computer system from the discovered CI space into the actual CI space, this equates to everything above the dashed line.

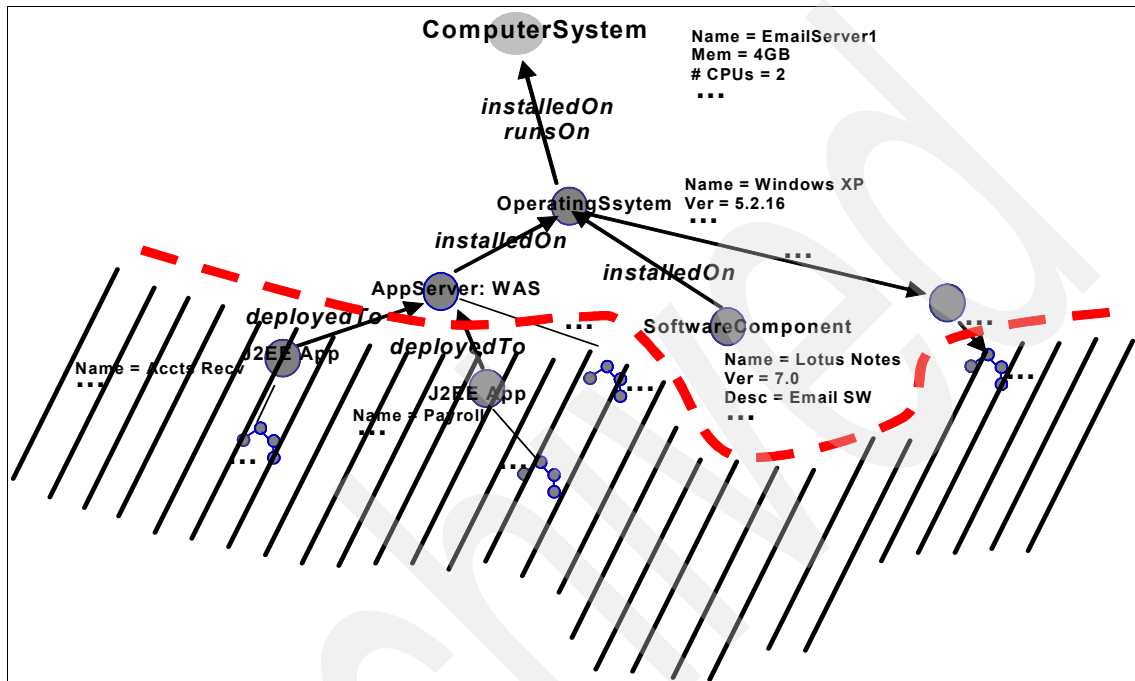


Figure 5-4 Actual CI filter depth

None of the relationships, objects, or their attributes that are below the dashed line are brought over into the actual CI space, which will result in a significant savings of movement of data. Depth level 3 will allow you to determine the software running on a computer system, so if this contains all that you are looking to manage in the authorized CI space (discussed in 5.5, “Authorized configuration items” on page 348), this can be an appropriate depth level setting.

However, for example, if you are looking to manage J2EE applications running in a web application server on a computer system, you need to set a deeper depth level filter to ensure that you get the J2EE applications brought over into the actual CI space as part of the computer system.

In case you are working with a scaled deployment of the CCMDB discovery solution, where you have an Enterprise Discovery Server (eCMDB) and a number of domain discovery servers, make sure that the data that you intend to synchronize into the actual CI data space is physically kept in the eCMDB database. If not, the ITIC adapter data request must 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.

5.5 Authorized configuration items

After you have specified which CIs must 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 CCMDB 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 contains only the attributes or values and relationships to objects that you want to bring under Change control.

For example, to meet corporate standards for a production email 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 email computer system called EmailServer1, you build the setup shown in Figure 5-5 as a set of authorized CIs, attributes, and relationships in the authorized CI space. This reflects your authorized state.

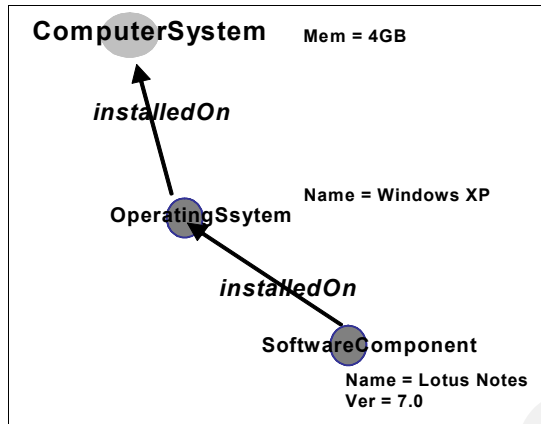


Figure 5-5 Sample of an authorized CI view

Notice 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 want 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.

5.5.1 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 might 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 that 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 (attributes and values, relationships to other authorized CIs, and 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 might 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 can create an authorized CI for each computer system and drive them through the Change process to get them deployed into your IT environment. When 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 might 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 want 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 an authorized CI space (called a CI Hierarchy in the previous versions of CCMDb). An authorized CI space 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 as with an authorized CI, the authorized CI space 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 want to promote them to a given authorized CI space, 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 authorized CI space selected during promotion.

Note that when promoting actual CIs to an authorized CI space, if you do not specify copying the attributes during the promotion, then it will use the attribute value that is in the authorized CI space 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). an authorized CI space 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, see *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 another structure. Usually you might extend the authorized CI scheme with attribute definitions to persist or federate data that is not discoverable.

5.5.2 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 CI, 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 such as IT- or Enterprise Asset Management while CI representations are foremost oriented to supporting more technically oriented processes such as Configuration or Change Management.

5.6 CIs in the CDM

All CIs are grouped within the CDM into entities that correspond to items in the real world.

These groups are called classes (in the CDM). CCMDB uses the ITIL term, CI Types, to refer to these groups. CCMDB also uses classifications (a Maximo construct) to represent these groups.

Each CI Class contains the following key information:

Hierarchy:

The structure that the classes are put in to provide a single-inheritance hierarchy that enables attributes to be shared among classes. Note that the CDM uses hierarchy to represent inheritance; not to show a data tree that represents parent and child CI classes. CCMDB uses hierarchy to represent a data tree with parent and child CI types.

Attributes:	Atomic data that is at the most basic level of granularity. Each attribute has a data type, along with an attribute name. Attribute names are unique among all classes that are in the same inheritance hierarchy. Therefore if one class has the name 'Memory', none of its parent or child classes can use that attribute name. If it appears that two classes in the same inheritance hierarchy have the same attribute, they both inherited the attribute from the same class.
Relationships:	The link between two CIs. Each relationship has a source class, target class and relationship type (InstalledOn, DeployedTo). Each relationship also has a direction and cardinality.
Naming rules:	The CDM formally defines the ways in which each class is identified, in order to foster consistent identification of CIs in the CMDB. To do so, the model uses naming rules. Naming rules are rules that specify which attributes should be used to name CI. Naming rules are defined for a CI class. Therefore all CIs of the same class have the same naming rules. In most CI classes, there is more than one naming rule. When there is more than one, the source of the data determines which rule will be used, by which attributes are populated and the priority of the naming rule.

The entire CDM and associated documents can be viewed at the Tivoli CDM Web site, which is included on the TADDM product DVD. To access the Web site, unzip the following file, which is located in the TADDM installation directory:
<install_root>/cmdb/dist/sdk/doc/model/CDMWebsite.zip

5.7 How CIs are represented in the CCMDB

CCMDB uses classification objects (MBOs) and relationship objects (MBOs) to represent the Common Data Model. The CI Type adapter imports the TADDM's representation of the CDM into CCMDB. It does this by using the TADDM API to get the metadata that describes the CDM.

5.7.1 Classification objects and relationship objects

The CI Type adapter creates a classification for each CI class. CCMDB requires you to create a top CI classification before the CDM is imported. Each CI classification sets its parent classification to that top CI classification, so CI classifications can be distinguished from other classifications in CCMDB (such as Asset classifications). Each CI classification will be created with the attributes that the TADDM metadata specified. The TADDM metadata returns all attributes for a CI class, even those that are inherited. Therefore CCMDB does not copy TADDM inheritance class structure, nor does it directly represent the CDM hierarchy.

Relationships are represented as relationship objects in CCMDB. The relationship name in CCMDB is set to the relationship type from TADDM (i.e. InstalledOn). Relationship rules are then created that specify which CI classifications are valid for that relationship. Each relationship rule includes a source classification, target classification, and a set of CDM flags (described below).

All relationships that are imported from TADDM, both implicit and explicit, are created as explicit relationships in CCMDB.

5.7.2 CDM flags used in CCMDB

CCMDB uses several flags in its representation of the CDM, to allow for data to be traversed, for key use cases in CCMDB.

Top level CI type

Top level CI types are CI types that exist at the top of data hierarchies. Many use cases in CCMDB use top level CI types to be able to traverse through a single data tree, starting at the top and moving down the tree.

The top level flag is set by the CI Type adapter when it gets the metadata from TADDM.

TADDM stores the top level flag as a metadata property (TopLevel) for every CI class. In CCMDB, the top level flag is an attribute on a CI type that is accessed using the CI type application. It is also an attribute on the “Use With” CI and ACTCI records, when they are added as rows to a classification.

You can also look at the CI type in the CI Type application in the CCMDB GUI to determine if a CI is top level. (There is a check box that shows the value of this flag).

There might be cases when you have a CI type that you consider top level because it is at the top of the CI data tree you have defined. In these cases, it is okay to update the top level flag. This has to be done with caution because if there is data that is in the same data tree that is above this CI type, there are scenarios (for example, promotion, full CI comparison) in CCMDDB that won't be able to reach that data.

Containment

A containment relationship shows that one CI contains another CI. Sometimes this is called a parent-child relationship, but there are some parent-child relationships that are not containment relationships (such as inheritance).

Within the CDM, there is a notion of containment depending on how the resource and container are linked in terms of their lifecycles. In each case, the instance that represents the container is the source of the relationship; the instance that represents the contained resource is the target of the relationship.

The first containment relationship is called “contains”. (This particular relationship is also known as a composition association.) The “contains” relationship means the source contains the target resources, and there is a lifecycle dependency between the container and the resources within the container. When the relationship “contains” is used, it means the lifecycles of both the source and targets are dependent on each other; one cannot exist without the other. In a non-IT example, a “circle” contains a “center”.

The other containment relationship is called “federates”. (This particular relationship is also known as an aggregation association.) The “federates” relationship means the source contains the target resources, and there is no lifecycle dependency between the container and the resources within the container. For “federates”, the lifecycles of both the source and targets are independent of each other; one can exist without the other. In a non-IT example, a “car” federates a “wheel” (A car exists without wheels, but it does not drive very well!)

There is one other relationship that is commonly used to represent containment and the Common Data Model does not apply a statement about lifecycle dependency: “memberOf”. When using the relationship “memberOf”, the source is the resource and target is the container. It simply means the resource is a member of the container.

If the resources have a true “containment” relationship, they will use one of the relationships mentioned above.

The containment flag in CCMDB is calculated and set by the CI Type adapter. The adapter stores a list of containment relationships. Then when the adapter is importing a relationship from CCMDB, it checks to see if the relationship is in the containment list. If the relationship is not in the containment list, the containment flag is set to false. For the relationships that are in the list, the adapter will also check to see if the parent of the relationship is used in a naming rule for the child. If it is, the containment flag will be set to true.

In CCMDB, containment is an attribute of a relationship rule that is accessed using the Relationship application. There might be cases when it is helpful to query relationship rules directly in the CCMDB. The table that stores relationship rules is RELATIONRULES, and the containment flag is the column "CONTAINMENT."

Is Target Parent flag

The "Is Target Parent" flag states whether the target class of a relationship rule is the parent of the relationship. When this flag is false, it is assumed that the source class is the parent of the relationship. Many CCMDB scenarios traverse a data tree starting at the top and then going down the tree, following relationships from parent to child. This flag is used to determine which CI is below the other in the data tree, to determine which direction to follow the relationship.

The "Is Target Parent" flag is set by the CI Type adapter when it gets the metadata from TADDM. It uses an algorithm that uses the TADDM naming rules and TADDM's reverse relationship flag (which exists on implicit relationships in TADDM), to determine if the source or target of a relationship is the parent. Currently, it also requires the relationship to be containment, using the list of containment relationships the adapter stores.

In CCMDB, "Is Target Parent" is an attribute of a relationship rule that is accessed using the Relationship application. There might be cases when it is helpful to query relationship rules directly in the CCMDB database. The table that stores relationship rules is RELATIONRULES, and the "Is Target Parent" flag is in the column "REVRELATIONSHIP"

Figure 5-6 shows the "OperatingSystem InstalledOn ComputerSystem" relationship. ComputerSystem is the parent and is also the target of a relationship and therefore "Is Target Parent" is true.

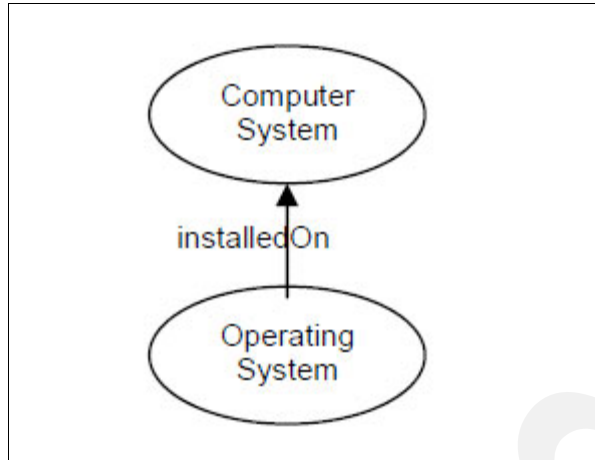


Figure 5-6 “OperatingSystem InstalledOn ComputerSystem” relationship

When you are designing your authorized CI template for promotion, you might occasionally want to follow a relationship from child to parent. In these cases, you can change the “Is Target Parent” flag on the authorized CI relationship so the promotion will follow the relationship in the opposite direction.

Cardinality

Cardinality defines the numeric relationships between occurrences of the CIs on either end of the relationship. CDM relationship rules define the cardinality for each relationship type in the CDM.

The cardinality flag is set by the CI Type adapter when it gets the metadata from TADDM. For implicit relationships, the adapter uses an algorithm that determines the cardinality based on what the attributes look like. For example, if an Array of ModelObjects is the target of a relationship, and a single required ModelObject is the source of a relationship, the adapter would set the cardinality to 1:N.

In the CCMDB, cardinality is an attribute of a relationship rule that is accessed using the Relationship application. There might be cases when it is helpful to query relationship rules directly in the CCMDB database. The table that stores relationship rules is RELATIONRULES and the cardinality flag is the column “CARDINALITY.”

Cardinality has been added to a couple of relationships in the CCMDB database. For example, for the relationship type ‘OperatingSystem installedOn ComputerSystem’, the cardinality is N:1. N stands for 0.*. Therefore 0, 1, or many operating systems can be installed on a single computer system.

5.8 Working with CIs in CCMDB

There are a number of CCMDB applications that are used for administering CIs. Here is a list of them, for reference.

Administrator applications (under the Administration Go To menu):

1. CI Types:
 - Activate CI types, for importing actual CIs.
 - View top level flag for CI Types.
2. Classifications:
 - Create new authorized CI classifications.
 - Update “Use With” records that specify which objects the classification applies to.
 - Add and remove attributes for the classification.
 - Set default values for classification attributes.

Operator applications (under the IT Infrastructure Go To menu):

1. Actual Configuration Items:
 - View and query Actual CIs.
 - View relationship instances between Actual CIs.
 - Run reports on Actual CIs.
 - Promote Actual CI to an Authorized CI (creating Authorized CI from an Actual CI).
2. Configuration Items:
 - Create, View, Query, Update and Delete Authorized CIs.
 - Run reports on Authorized CIs.
 - Change status of an Authorized CIs.
 - Add CIs to a Collection.
 - Add CIs to a Change Window.
 - View, Update and Delete relationship instances between Authorized CIs.
3. Relationships:
 - Create, View, Query, Update, and Delete relationships.
 - Note that a relationship can be considered a relationship type or relationship class (for example, RunsOn). This application is not used to administer relationship instances that exist between two CIs.
4. Collections:
 - Create, View, Query, Update and Delete collections.
 - Run reports on collections.
 - Collections are groups of CIs.

CI space development

Designing and developing the CI space enables promotion from actual CIs to authorized CIs (manageable). A good implementation of the CI space improves its management and promotion task performance, as well as reducing the amount of authorized CIs to manage.

In this chapter, we cover the following topics:

- ▶ 6.1, “What a CI space is” on page 360
- ▶ 6.2, “Designing your CI space with Deployer’s Workbench” on page 361
- ▶ 6.3, “Setting up Tivoli Integration Composer according to the Deployer’s Workbench” on page 377
- ▶ 6.4, “Promotion at a glance” on page 378

6.1 What a CI space is

A CI space is the set of authorized CI classifications (or hierarchies) and promotion scopes, which are used by the promotion task on CCMDB.

The authorized CI hierarchy is the data hierarchy that you want for your authorized CIs. This must be a subset of your actual CI. Include only CI types, attributes, and relationships that you need in your ISM processes (including those to which you want to audit against and apply Change control). CCMDB Audit uses reconciliation to compare an authorized CI to an actual CI. If your authorized CI hierarchy is very large, your auditing results will be too large to manage. Therefore it is important to take the time to build your authorized CI hierarchy so that it only includes what you need. You can also always add to it at a later time. Therefore, the general rule is to start small and slowly add to the hierarchy as needed.

Figure 6-1 shows a sample authorized hierarchy diagram for the actual CI diagram:

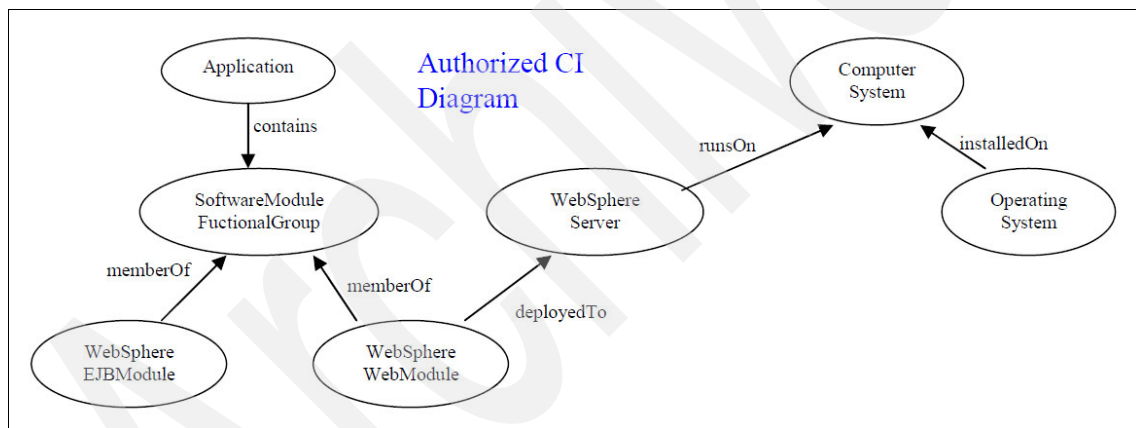


Figure 6-1 Authorized Hierarchy Diagram

Important: The concept of a “Dual class” hierarchy is no longer supported in CCMDB 7.2.1.

In CCMDB V7.2.1, the best practice is to use the Deployer's Workbench for designing the authorized CI space. The following sections cover a simple end to end scenario that goes through common tasks in designing and developing a CI space through Deployer's Workbench.

6.2 Designing your CI space with Deployer's Workbench

A good CI space is expected to have the actual CI classifications / authorized CI classifications mapping according to the user's needs and a very well defined promotion scope, also according to the user's interest. Knowing well the CIs you want to promote also improves CCMDB promotion performance and reduces the risks of bad performance for end users during a promotion task.

It is a good idea to study the Common Data Model (CDM) prior to working with Deployer's Workbench, so you can decide what objects, relationships, and attributes might interest you.

Refer to *IBM Tivoli Common Data Model: Guide to Best Practices*, REDP-4389 or to the following TADDM InfoCenter for additional information about CDM:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.2/SDKDevGuide/c_cmdbsdk_database.html

Data models: The Common Data Model is a standardized definition of how system solutions and technologies represent resources and their relationships. The Common Data Model is a logical data model that brings together various industry data models, acting as the dictionary and grammar for consistently describing details and identity of resources.

Keep in mind that you are looking for the best organization for your authorized CIs in such a way that it is more convenient for your needs. It might make more sense in your context to have a hierarchical structure unlike that of the default, and that is all what the Deployer's Workbench is all about. Configuring your authorized CIs the best way possible to meet your needs.

The following sections assume that you have a clean environment and are aware of configuration items and promotion concepts.

6.2.1 Setting up a project with the Deployer's Workbench

Proceed as follows:

1. Open the Deployer's Workbench. The tool will prompt a workspace selection as you can see in Figure 6-2. If you already have a workspace defined, it is best to create a new workspace for this scenario.

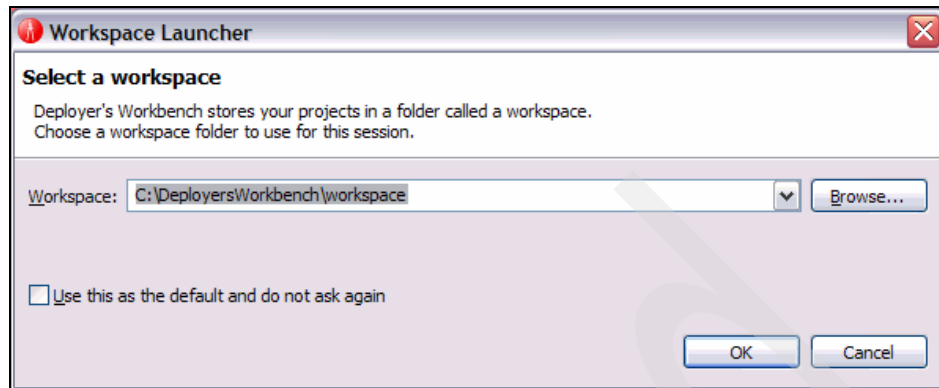


Figure 6-2 Select workspace Deployer's Workbench dialog

Deployer's Workbench is Eclipse based, therefore you might be familiar with the user interface. The next step is to create a Deployer's Workbench project, which does have it is own type.

2. Click **New** → **Project**. The dialog in Figure 6-3 is shown. Select **Deployer's Workbench Project**.

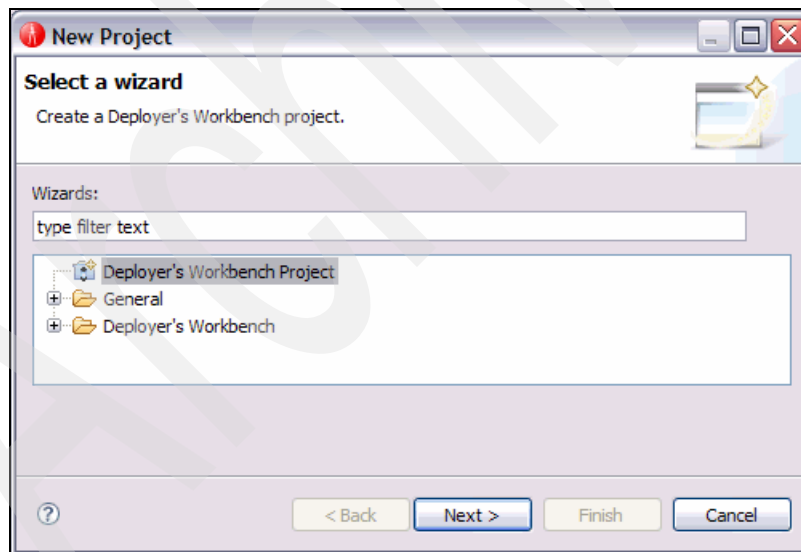


Figure 6-3 Select Deployer's Workbench Project

3. Set the project name to IBMRedbooks and click **Finish**.

4. Expand the whole tree under the Model Navigator tab (left-top), notice that you have the Server element and the CI space element, both with a red mark that means you have pending issues to resolve, ignore that by now.
5. Expand the **CCMDB Best Practices** in your Library tab (left-bottom) as shown in Figure 6-4.

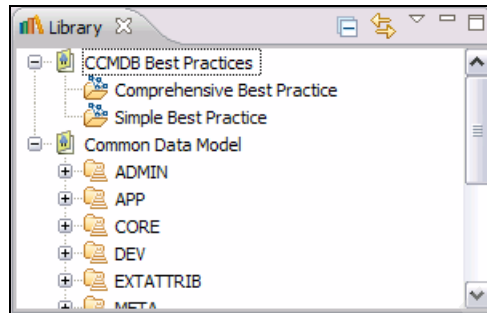


Figure 6-4 Library view

6. Drag and drop Comprehensive Best Practice into the CI space element under your Model Navigator tab (left-top), to copy the actual and authorized classifications. A progress dialog displays. Wait for it to finish.

At this point you have all the classifications ready, but prior to start working with that you need to fix the red marks.

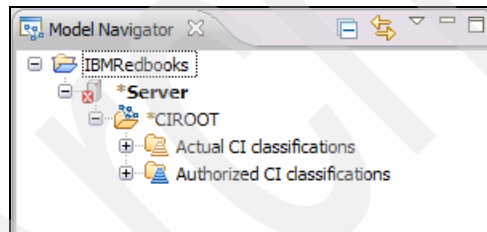


Figure 6-5 Server element in error

It is important to understand that each Deployer's Workbench Project can have only one Server element, therefore only one CCMDB can be connected per Deployer's Workbench Project.

7. Right-click the Server element and select **Edit Properties**. Define a hostname, port and User name and click **OK** (Figure 6-6).

Important: Ensure that the user that you have entered in the User name field in the CCMDB server properties is authorized to create Classifications and Object Structures. A few out of the box users such as *maxadmin* and *mxintadm* are already granted these authorities.

For more information about Classifications, see Chapter 18 in the following guide, available at this website:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.ccmdb.doc_721/reference/mam71_sys_admin_guide.pdf

Regarding information about Object Structures, see this website:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.ccmdb.doc_721/reference/mam71_integration_guide.pdf

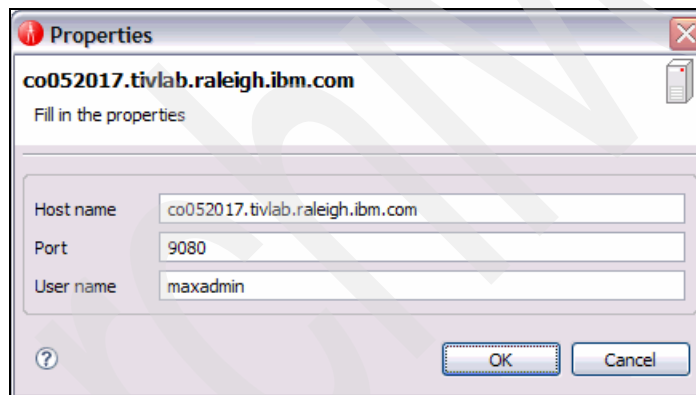
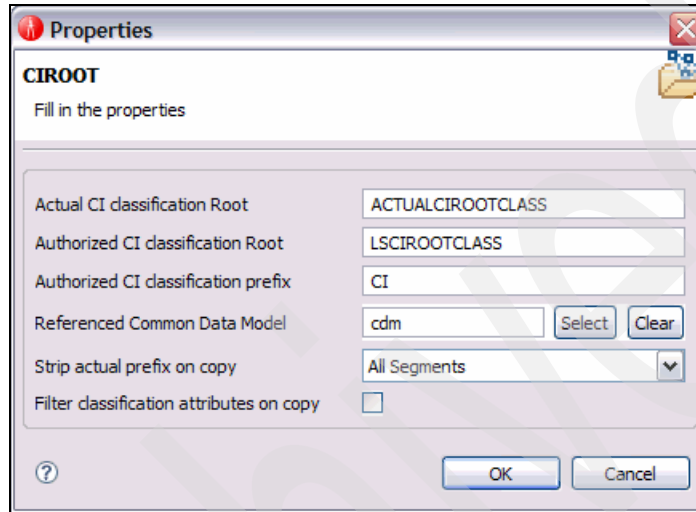


Figure 6-6 Setting up CCMDB server on Deployer's Workbench

A warning icon () might appear on the server element if the Deployer's Workbench is unable to resolve the host name (this does not apply for the other fields).

8. Modify your CIROOT element (CI space) by right-clicking the CI space element and select **Edit Properties**. Define the fields as Figure 6-7 and click **OK**.



The image shows a 'Properties' dialog box for a 'CIROOT' element. The title bar says 'Properties' with a red 'X' icon. Below the title bar, it says 'CIROOT' and 'Fill in the properties'. The dialog contains several fields and controls:

- 'Actual CI classification Root' with a text box containing 'ACTUALCIROOTCLASS'.
- 'Authorized CI classification Root' with a text box containing 'LSCIROOTCLASS'.
- 'Authorized CI classification prefix' with a text box containing 'CI'.
- 'Referenced Common Data Model' with a text box containing 'cdm', a 'Select' button, and a 'Clear' button.
- 'Strip actual prefix on copy' with a dropdown menu showing 'All Segments'.
- 'Filter classification attributes on copy' with an unchecked checkbox.

At the bottom, there is a help icon (question mark), an 'OK' button, and a 'Cancel' button.

Figure 6-7 Setting up CI space

Tip: For additional information about the field “Strip actual prefix on copy,” see: http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.dw.doc_721/dw/t_dw_class_name.html

Notice that the CI space element is now named LSCIROOTCLASS as you defined in Figure 6-7. At this point you are all set to update the Comprehensive Best Practices library to your specific needs.

6.2.2 Configuring your own set of authorized CI classifications

Let us consider a scenario where you are interested only in tracking infrastructure CIs such as these:

- ▶ Operational systems
- ▶ Application servers
- ▶ Databases

The purpose of Deployer's Workbench is to set up your CI space according to your specific requirements, and in this scenario, we assume that you do not have the following operating systems in your IT infrastructure:

- ▶ IBM AIX®
- ▶ HP-UX
- ▶ SUN

It might sound as if you really do not need those classifications and you want to remove them, but at this point they are just a mapping structure imported from the Comprehensive Best Practices library to the authorized CI classifications, with defined attributes and relationships. If there is any possibility that in the future, you might want to use these classifications, do not remove them. For example, if in the future you want to implement these operating systems in your IT infrastructure, you have to add them back, which is not a very trivial task.

Best practice: It is highly desirable to take notes of the Changes made to the imported Comprehensive Best Practices in your project. Your notes can be useful if you lose your actual data, when migrating to another system, and most importantly, if you plan to redesign your CI space and want to start from the beginning.

A good tool to preserve your Changes is the Archive tool. For additional information, see:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.dw.doc/dw/t_dw_archive.html

For the sake of this scenario, let us assume that it is very unlikely that these operating systems will be used in our infrastructure in the future (for example, this can be a company mandate), so we have decided to remove them.

Removing authorized CI classifications

Proceed as follows:

1. Expand the **Authorized CI classifications** → **CI.COMPUTERSYSTEM**. Right-click the **CI.AIXCOMPUTERSYSTEM** and select the option **Delete CI.AIXCOMPUTERSYSTEM** as shown in Figure 6-8.

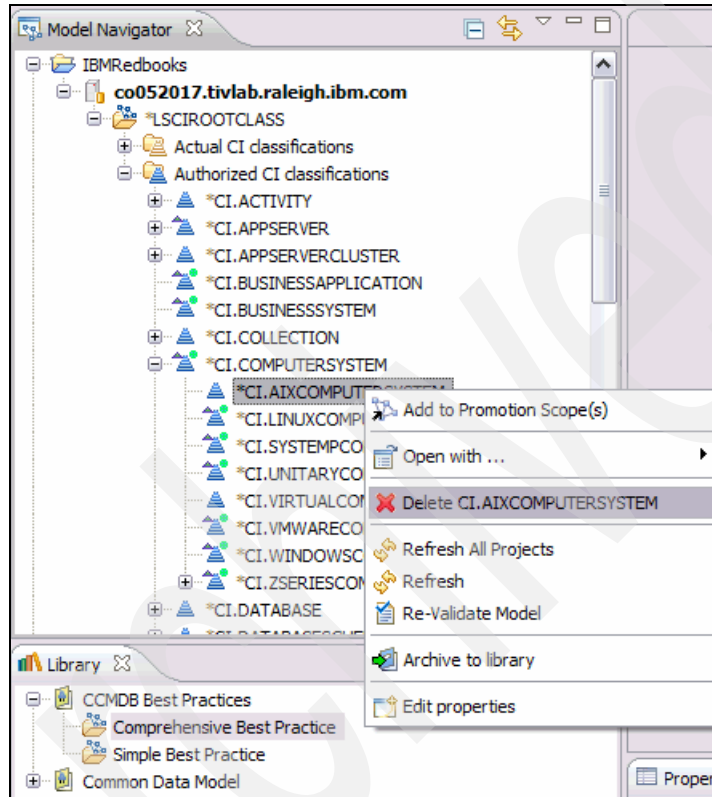


Figure 6-8 Steps to remove an authorized CI classification

A prompt will display stating that CI.AIXCOMPUTERSYSTEM is part of one or more promotion scopes and that deleting the classification will also remove it from the promotion scope. Answer *Yes*.

Another prompt will display stating that your deletion will remove related objects as well. Answer *Yes*.

A dialog displays showing the progress of the removal (total of 125 models and 344 relationships).

2. Follow the same steps to remove CI.HPUXCOMPUTERSYSTEM and CI.SUNCOMPUTERSYSTEM.

Tip: Notice the various marks on each CI. The arrow pointing to the top represents top-level classifications, and the green mark represents the classifications that contain promotion scope ().

For more information regarding the Deployer's Workbench user interface, see:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.dw.doc_721/dw/c_dw_ui.html

Reorganizing CI hierarchies

Reorganizing CI hierarchies can be very useful to make your CI classifications easier to use according to your needs. Given the infrastructure scenario described previously, having the CI.COMPUTERSYSTEM as it is might not be intuitive when using the CCMDB in the future. As you can see in Figure 6-9, the actual hierarchy is quite flat and does not contain many children.

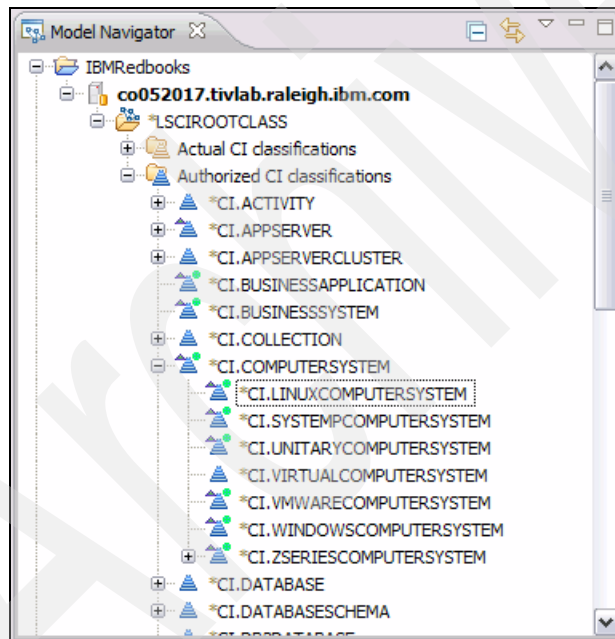


Figure 6-9 Actual CI.COMPUTERSYSTEM hierarchy

That is exactly the scenario that requires changes to the CI space. A computer system, for instance CI.LINUXCOMPUTERSYSTEM, usually has an operating system (OS) running, but looking at the classifications shown in Figure 6-9, it is not intuitive that a CI.LINUXCOMPUTERSYSTEM has an OS installed.

Adding the Linux OS to the computer system hierarchy is just a matter of drag and drop:

1. Expand the **CI.OS classification** → **drag and drop CI.LINUXOS into CI.COMPUTERSYSTEM** → **CI.LINUXCOMPUTERSYSTEM**. Now you have the hierarchy **CI.COMPUTERSYSTEM** → **CI.LINUXCOMPUTERSYSTEM** → **CI.LINUXOS** as Figure 6-10 shows.

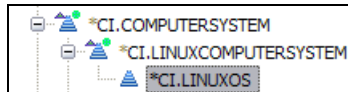


Figure 6-10 Added CI.LINUXOS to CI.LINUXCOMPUTERSYSTEM

Tip: It might take a while for your authorized CI classifications to be updated after a Change. To update it immediately, you can go to the project root (IBMRedbooks in this case) and press **F5** or right-click it and select **Refresh**.

Important: As soon as you have synchronized your Deployer's Workbench with CCMDDB, you cannot change the authorized CI classifications hierarchy any longer, unless you create a new CIROOT. For additional information, see:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.dw.doc/dw/c_dw_synch.html

2. You can apply the previous steps for any authorized CI classifications. Again, the idea is to configure your hierarchy according to your needs. At this point none of the Changes provided by this scenario affects how CCMDDB works.

Modifying CI attributes

Removing the CI attributes will free you from having undesired attributes that might be confusing for CCMDDB users and improve the promotion task as well, while adding attributes enables you to track specific attributes that are not being covered by the Comprehensive Best Practices library.

Following the scenario we have been using:

1. Simply double-click CI.COMPUTERSYSTEM or right-click the **CI.COMPUTERSYSTEM** → **Open with...** → **Authorized CI classification editor**. A tab named CI.COMPUTERSYSTEM displays (Figure 6-11).

2. Notice the two main sections, Details and Attributes. Also notice on the bottom of this tab that you have a more tabs:
 - Authorized CI classification Editor
 - Authorized Relationship Viewer
 - Promotion Scopes

Tip: In the top of the authorized CI classification editor, you can see a breadcrumb bar with the path to the opened CI classification.

The focus now is on the Attributes section, actually displaying 9 attributes. Notice that 4 of the 18 attributes start with **MODELOBJECT_**. Those attributes are metadata used by TADDM and do not add any value to the current CI classification, indeed using TADDM metadata is not desirable. Because they are for TADDM use only, they can be changed or even removed without previous notification.

3. Select the four items that start with **MODELOBJECT_** and click the **Delete** button.

Details:

Edit the details for this classification.

Name: CI.COMPUTERSYSTEM

Description: CI.COMPUTERSYSTEM

Generate Description: ☐

Use Classification: ☐

Top Level: ☒

Organization: Select Clear

Site: Select Clear

Actual CI classification: SYS.COMPUTERSYSTEM Select Clear

Attributes:

Edit the 18 attributes for this classification.

Attribute	Type	Domain Id	Default Value	Description
MODELOBJECT_OBJECTTYPE	ALN			MODELOBJECT_OBJECTTYPE
MODELOBJECT_LABEL	ALN			MODELOBJECT_LABEL
MODELOBJECT_DISPLAYNAME	ALN			MODELOBJECT_DISPLAYNAME
MODELOBJECT_CONTEXTIP	ALN			MODELOBJECT_CONTEXTIP
COMPUTERSYSTEM_VMID	ALN			COMPUTERSYSTEM_VMID
COMPUTERSYSTEM_VIRTUAL	ALN			COMPUTERSYSTEM_VIRTUAL
COMPUTERSYSTEM_TYPE	ALN			COMPUTERSYSTEM_TYPE
COMPUTERSYSTEM_SERIALNUMBER	ALN			COMPUTERSYSTEM_SERIALNUMBER
COMPUTERSYSTEM_PRIMARYMACADDRESS	ALN			COMPUTERSYSTEM_PRIMARYMACADDRESS
COMPUTERSYSTEM_NUMCPUS	NUMERIC			COMPUTERSYSTEM_NUMCPUS
COMPUTERSYSTEM_NAME	ALN			COMPUTERSYSTEM_NAME
COMPUTERSYSTEM_MODEL	ALN			COMPUTERSYSTEM_MODEL

Authorized CI classification Editor | Authorized Relationship Viewer | Promotion Scopes

Figure 6-11 Authorized CI classification Editor

4. Click the **Save** button or press **Ctrl + S**. The removed attributes improve the promotion task performance and are out of scope for this scenario.

The main goal of removing attributes is to clean up those that are not candidates for tracking, meaning those that are not relevant for a Change.

For example, if you have a server and are going to upgrade the memory (RAM) from 4 Gb to 8 Gb, the attribute MEMORYSIZE (computersystem) is something you probably want to track. On the other hand, you might not be interested in the CPUType attribute (for example, pentimII_Xeon, amdDuron, ultraSparc) because your company policy is not to change the processors (a third party support team replaces the computer itself if you have problems with the processor). So, it really depends on your needs, policies, and items you want to track.

- Following these examples, double-click **CI.COMPUTERSYSTEM** or right-click the **CI.COMPUTERSYSTEM** → **Open with...** → **Authorized CI Classification Editor**. As you did earlier, in the attributes section select the attribute **COMPUTERSYSTEM_CPUYPE** and click the **Delete** button. Then click the **Save** button or press **Ctrl + S**.

Important: The Changes made on attributes of a specific classification do not affect child classifications. So, for example, if you have removed the attribute **COMPUTERSYSTEM_CPUYPE** from **CI.COMPUTERSYSTEM**, the attribute on **CI.LINUX COMPUTERSYSTEM** will not be affected.

Also, you might want to add new attributes to your classifications, to track something that is not part of the Comprehensive Best Practices library. Again, it is best that you study the CDM to find which attributes or objects best fit your needs. For example, you might need to keep track of the ROMVersion (BIOS version of the motherboard in the ComputerSystem) of your server if the manufacturer says to keep it up to date.

- Double-click **CI.COMPUTERSYSTEM** or right-click the **CI.COMPUTERSYSTEM** → **Open with...** → **Authorized CI Classification Editor**. Click the **Add** button in the Attributes section. See Figure 6-12.

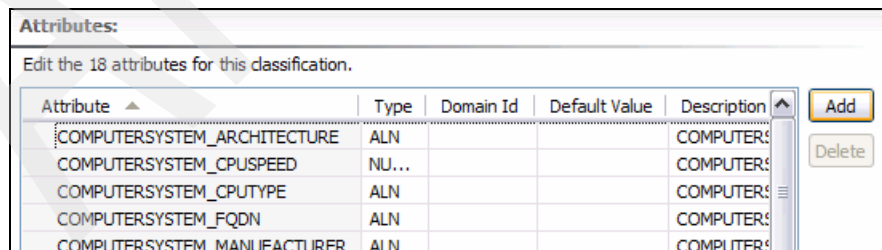


Figure 6-12 Adding attributes

A dialog to select or create an attribute displays. Notice that all attributes on the CMD will show, so you have to filter them. The CDM attributes have a prefix with the name of the classification that holds them.

7. On the Filter String field, write “computersystem_”, which filters the attributes related to the computersystem.
8. Click the **COMPUTERSYSTEM_ROMVERSION** check box (if you check multiple check boxes many attributes are added at once) and click the **OK** button. The COMPUTERSYSTEM_ROMVERSION is added to the CI.COMPUTERSYSTEM attributes.
9. Click the **Save** button or press **Ctrl + S** to save your Changes.

6.2.3 Defining a promotion scope

The promotion scope is a set of classifications and relationships that are going to be promoted from actual CI to authorized CI. The authorized CIs (or simply CIs) are the ones which you want to track (or control) through Change Requests. Therefore, the promotion scope is one of the most important terms in this chapter. At this point you will define what is sent to CCMDB to be promoted and available for Changes and tracking.

Keep in mind that you need a well defined set of CIs in your promotion scope to prevent having unused CIs available for Changes and as well to avoid performance issues (during the promotion task).

As mentioned in 6.2, “Designing your CI space with Deployer’s Workbench” on page 361, the scenario proposed uses the Comprehensive Best Practices library which already have a promotion scope defined for most of the top-level CIs.

Important: Remember that only top-level CIs can have a promotion scope!

Proceed as follows:

1. Expand CI.COMPUTERSYSTEM and right-click **CI.LINUXCOMPUTERSYSTEM → Open With... → Promotion Scope Editor**. A tab named CI.LINUXCOMPUTERSYSTEM opens and displays two sections:
 - Scoped Classifications
 - Relationships
2. Select a classification (CI.LINUXOS, for example) in the Scoped Classifications section. Notice that the Relationships section is updated, and now shows the Relationships between the selected classification and other

classifications. It also exposes the type of classification, cardinality, and few other columns (Figure 6-13).

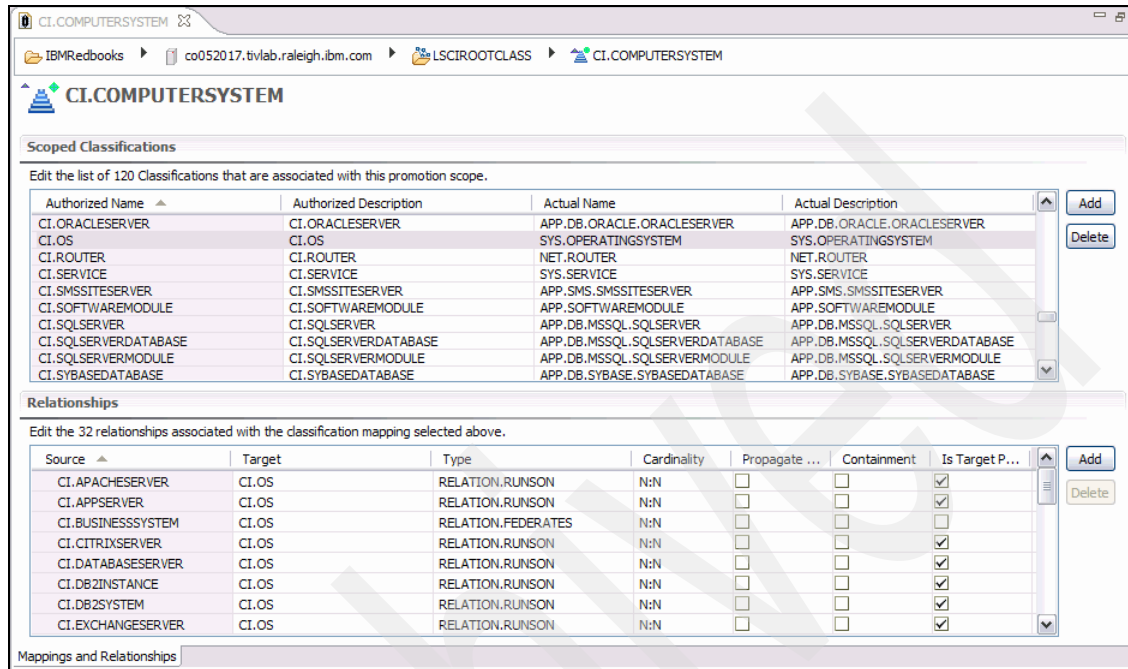


Figure 6-13 Promotion Scope Editor

All the 121 classifications in this promotion scope are targets to be promoted, but to simplify the results for this section scenario, you can create a new scope for CI.LINUXCOMPUTERSYSTEM as follows.

- a. First close all the tabs you have opened, then on the Model Navigator right-click the **CI.LINUXCOMPUTERSYSTEM** → **Delete Promotion Scope**. This removes the whole promotion scope for the top-level CI.LINUXCOMPUTERSYSTEM.
- a. Repeat those steps for CI.BUSINESSAPPLICATION, CI.BUSINESSSYSTEM and CI.SOFTWAREIMAGE, as well. In this scenario, none of those classifications ought to be promoted.
- a. Now right-click the **CI.LINUXCOMPUTESYSTEM** → **Create Promotion Scope** (). A new tab named CI.LINUXCOMPUTERSYSTEM will show up containing only one CI: CI.LINUXCOMPUTERSYSTEM.

Tip: This is one of many approaches to redefine your CI promotion scope. One other alternative is to delete CIs from an existing promotion scope.

- Click the **Scoped Classifications** section and click the **Add button**. A dialog to select the classification will open. Expand the **Authorized CI classifications** item, then filter by CI.OS on the Filter String field. Now select **CI.OS** and click **OK**.
- Click **Save** button or press **Ctrl + S**. Your CI.LINUXCOMPUTERSYSTEM under the Model Navigator will soon show a green mark.

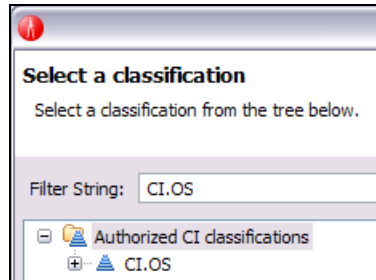


Figure 6-14 Add a CI.LINUXOS classification to the promotion scope

At this point you have CI.LINUXCOMPUTERSYSTEM and CI.LINUXOS classifications (Figure 6-15). It is a small promotion scope focused only in keeping track of computer systems and operating systems related to them.

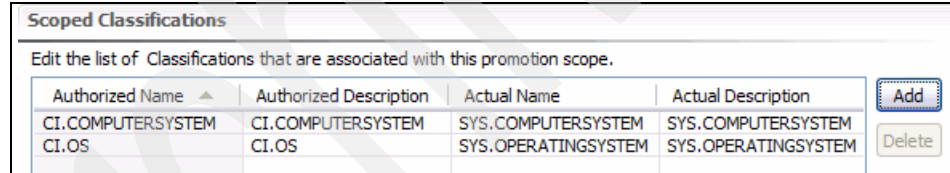


Figure 6-15 Promotion scoped CIs

Tip: You can also add CIs to your promotion scope by right-clicking them and selecting the option **Add to Promotion Scope(s)**. A list of the top-level CIs is displayed for you to choose the promotion scopes to which you want to add this CI.

- Define your promotion scope as wide as you need. This scenario is just a simple walkthrough to enable bigger scopes.
- Right-click **LSCIROOTCLASS** CI space and export Changes back to the server.

A dialog displays asking for the password for *maxadmin* (or for the user you have defined).

7. Enter the password and click **OK**. Another dialog displays asking if you want to create the LSCIROOTCLASS, because there is no CI root classification with this name. Answer **Yes** and a progress dialog displays.
8. Go to your CCMDB Classifications application, and search for LSCIROOTCLASS.
9. Open the record and click the **Detail menu** (**>>**) on the **Parent Classification**.
10. Select **Parent Classification**, expand **LSCIROOTCLASS** → **CI.COMPUTERSYSTEM** → **CI.LINUXCOMPUTERSYSTEM** → **CI.LINUXOS**. As Figure 6-16 and Figure 6-17 show, the Changes made in Deployer's Workbench regarding the CI.LINUXCOMPUTERSYSTEM were reflected in CCMDB. The CI.LINUXOS is now a child of the CI.LINUXCOMPUTERSYSTEM and the classifications CI.AIXCOMPUTERSYSTEM, CI.HPUXCOMPUTERSYSTEM and SUNCOMPUTERSYSTEM are not there.

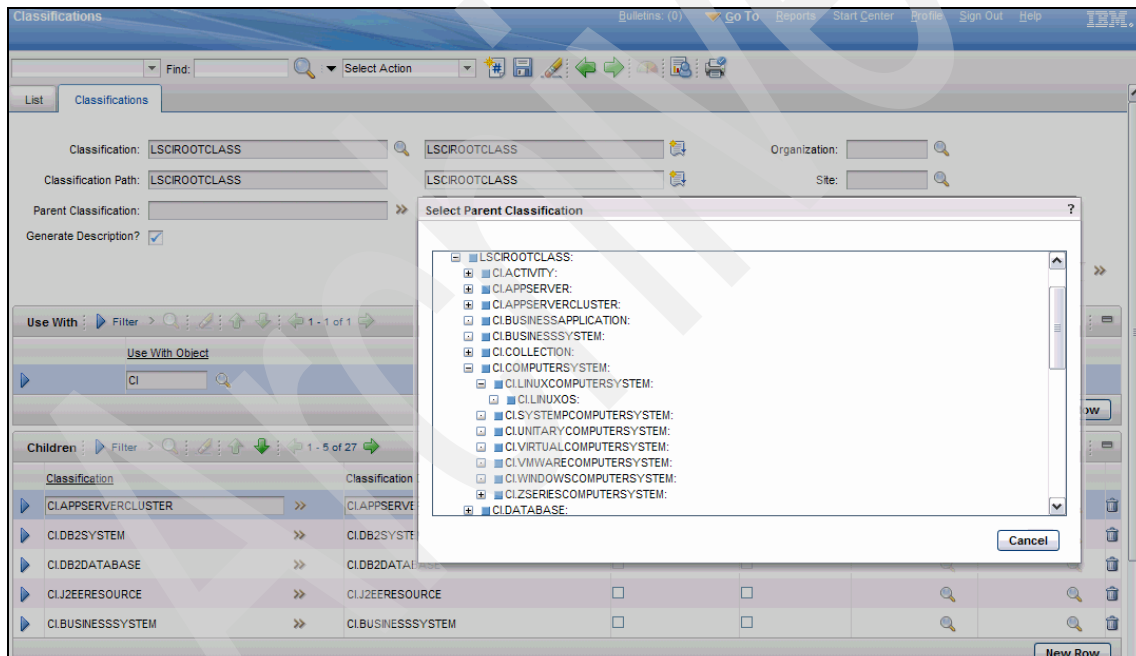


Figure 6-16 Exported classifications from Deployer's Workbench

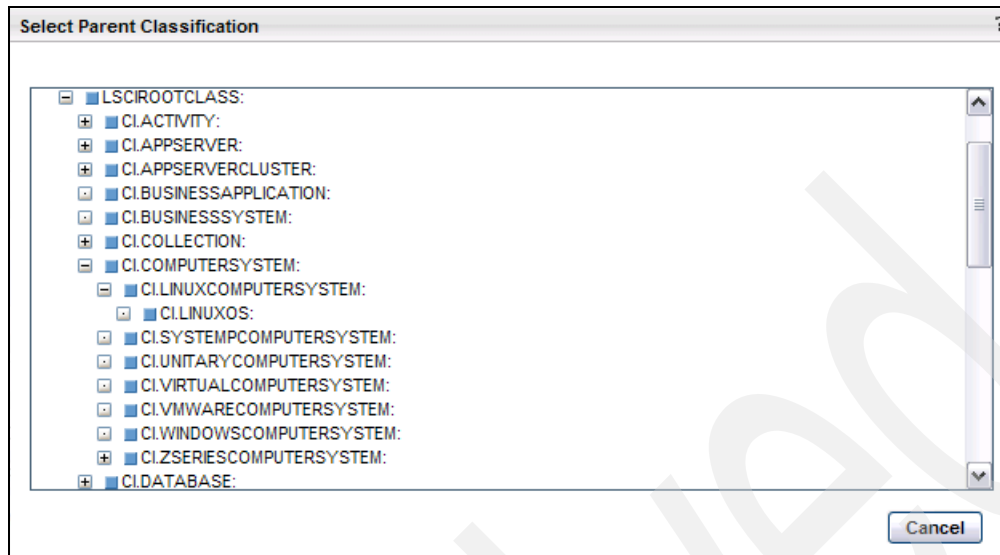


Figure 6-17 Exported classifications from Deployer's Workbench drilldown

The steps followed previously were just to show you the Change customizations done in the Deployer's Workbench affect your CCMDB. Click the **Cancel** button to close the Select Parent Classification dialog.

Remember that the classifications we are listing here are not the classifications in the promotion scope,. The scoped classifications can be found on the Deployer's Workbench.

Tip: Instead of using Deployer's Workbench to work with the promotion scope you can use the authorized CI hierarchies. For additional information refer to "Step 3: Map the entire authorized classification structure to the actual class structure" on page 431.

6.3 Setting up Tivoli Integration Composer according to the Deployer's Workbench

You need to set up your Tivoli Integration Composer (ITIC) according to the configuration you made in Deployer's Workbench, because the promotion task is essentially to promote actual CIs to authorized CIs, given a promotion scope. ITIC imports the discovered CIs from TADDM into CCMDB, and Deployer's Workbench defines the promotion scope. Therefore, synchronizing both configurations and reducing your set of CIs to be imported (considering the depth of CIs and which ones you are interested in) improves the promotion and import performance.

To synchronize your CI space (Deployer's Workbench) and your ccmdb.properties (ITIC import) you must design your CI space as discussed in 6.2, "Designing your CI space with Deployer's Workbench" on page 361 and define the depths and classifications of your ITIC properties (ccmdb.properties) as follows.

Given the scenario in this chapter, an example ccmdb.properties setup is shown in Example 6-1.

Example 6-1 Example ccmdb.properties

```
ccmdb.classification.depth.SYS.COMPUTERSYSTEM=3
ccmdb.classification.depth.SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM=2
ccmdb.classification.depth.SYS.SYSTEMPCOMPUTERSYSTEM=2
ccmdb.classification.depth.SYS.UNITARYCOMPUTERSYSTEM=2
ccmdb.classification.depth.SYS.WINDOWS.WINDOWSCOMPUTERSYSTEM=2
ccmdb.classification.depth.SYS.VMWARE.VMWAREUNITARYCOMPUTERSYSTEM=2
ccmdb.classification.depth.SYS.ZOS.ZSERIESCOMPUTERSYSTEM=2
```

Attention: Notice that the classifications removed from the CI space previously (CI.AIXCOMPUTERSYSTEM, CI.HPUXCOMPUTERSYSTEM and CI.SUNCOMPUTERSYSTEM) are not included in the ccmdb.properties.

See Chapter 7, "Importing configuration items" on page 383 for setting up the ITIC and import the actual CIs properly. These proposed properties are only in reference to this specific scenario; you can design them to reach the same classifications that your CI space defines through Deployer's Workbench.

6.4 Promotion at a glance

This section introduces an example on how to promote actual CIs to authorized CIs with the purpose of completing the scenario covered in this chapter. For more information about how to promote refer to Chapter 8, “CI promotion” on page 417.

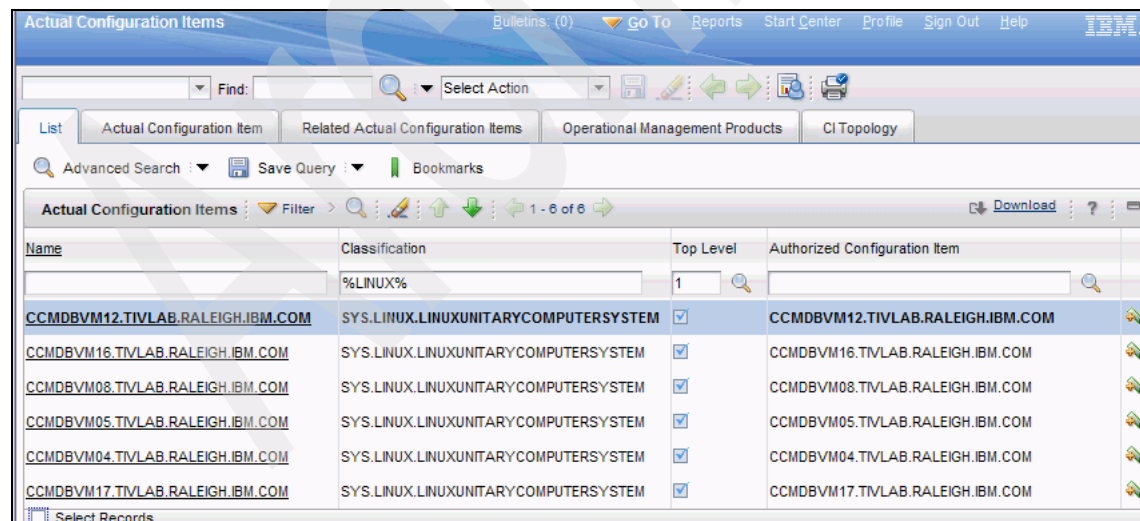
This section also assumes that you have already gone through the import procedures (using ITIC) covered or referenced in section 6.3, “Setting up Tivoli Integration Composer according to the Deployer’s Workbench” on page 377. Therefore, you already have the expected data in your actual configuration items table.

Important: The scenario covered in this chapter assumes you have Linux boxes discovered by Tivoli Application Dependency Discovery Manager (TADDM).

Proceed as follows:

1. In the CCMDB Start Center, click **Go To** → **IT Infrastructure** → **Actual Configuration Items**. On the List tab, filter the Classification field with “linux” and the Top Level field with 1.

As Figure 6-18 shows, the filtered list only results in SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM actual CIs.



Name	Classification	Top Level	Authorized Configuration Item
	%LINUX%	1	
CCMDBVM12.TIVLAB.RALEIGH.IBM.COM	SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM	☒	CCMDBVM12.TIVLAB.RALEIGH.IBM.COM
CCMDBVM16.TIVLAB.RALEIGH.IBM.COM	SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM	☒	CCMDBVM16.TIVLAB.RALEIGH.IBM.COM
CCMDBVM08.TIVLAB.RALEIGH.IBM.COM	SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM	☒	CCMDBVM08.TIVLAB.RALEIGH.IBM.COM
CCMDBVM05.TIVLAB.RALEIGH.IBM.COM	SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM	☒	CCMDBVM05.TIVLAB.RALEIGH.IBM.COM
CCMDBVM04.TIVLAB.RALEIGH.IBM.COM	SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM	☒	CCMDBVM04.TIVLAB.RALEIGH.IBM.COM
CCMDBVM17.TIVLAB.RALEIGH.IBM.COM	SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM	☒	CCMDBVM17.TIVLAB.RALEIGH.IBM.COM

Figure 6-18 Filtered actual configuration items

- On the left bottom, mark the check box, **Select Records**, and select all the actual CIs you are planning to promote. Then click **Select Action** → **Create Authorized Configuration Items** as shown in Figure 6-19.

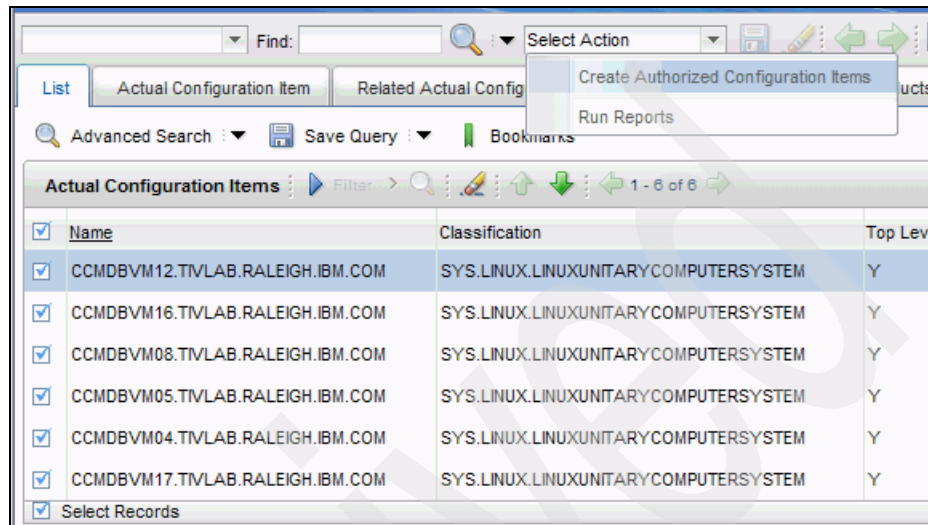


Figure 6-19 Select the actual configuration items to be promoted

A new dialog displays.

- Click **OK** keeping the default options, as shown in Figure 6-20.

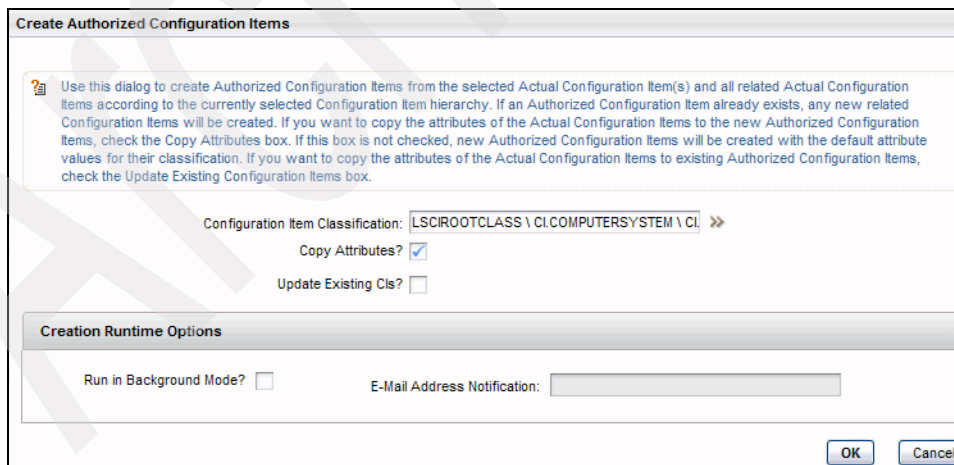


Figure 6-20 Promote dialog

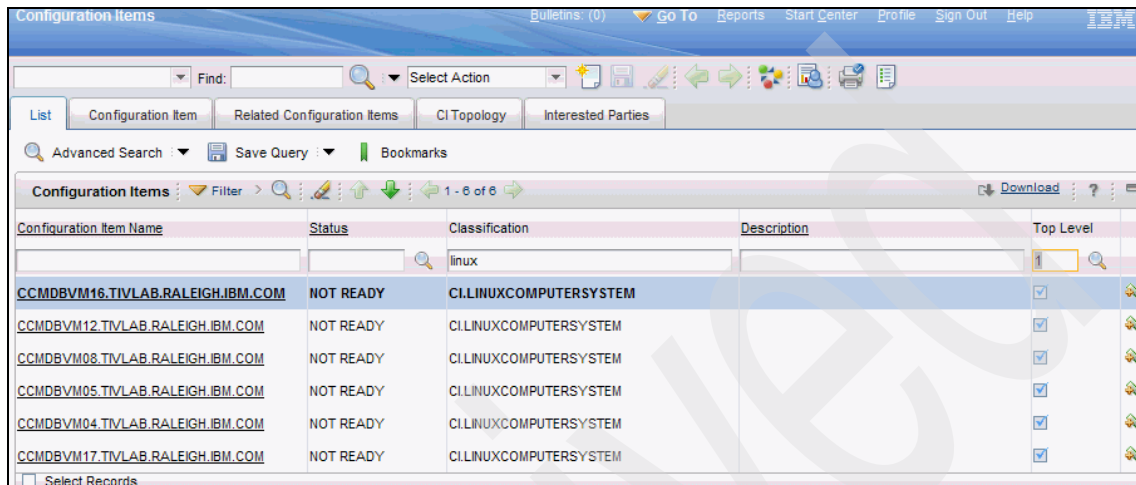
4. As soon as the promotion completes (depending on the number of actual CIs and depth imported, the time to promote varies), check how the Changes made in Deployer's Workbench affected the promotion.
5. Select **Go To** → **IT Infrastructure** → **Configuration Items**. On the List tab, filter the Classification field with "CI.COMPUTERSYSTEM", you can see a list of the imported/promoted CIs within the CI.COMPUTERSYSTEM.
6. Now click one the items filtered. The Configuration Item tab is displayed. In the Specifications table you can see something similar to Figure 6-21. Notice that as defined in Deployer's Workbench, your COMPUTERSYSTEM_ROMVERSION attribute exists and that the COMPUTERSYSTEM_CPUTYPE is not there.

Specifications Filter 1 - 9 of 9				
Attribute	Description	Data Type	Alphanumeric Value	
COMPUTERSYSTEM_MEMORYSIZE	COMPUTERSYSTEM_MEMORYSIZE	NUMERIC		
COMPUTERSYSTEM_MODEL	COMPUTERSYSTEM_MODEL	ALN	9131	
COMPUTERSYSTEM_PRIMARYMACADDRESS	COMPUTERSYSTEM_PRIMARYMACADDRESS	ALN	00145E966BCA	
COMPUTERSYSTEM_MANUFACTURER	COMPUTERSYSTEM_MANUFACTURER	ALN	IBM	
COMPUTERSYSTEM_SERIALNUMBER	COMPUTERSYSTEM_SERIALNUMBER	ALN	0645CCG	
COMPUTERSYSTEM_FQDN	COMPUTERSYSTEM_FQDN	ALN	itsmaix3.tivlab.raleigh	
COMPUTERSYSTEM_TYPE	COMPUTERSYSTEM_TYPE	ALN	ComputerSystem	
COMPUTERSYSTEM_CPUSPEED	COMPUTERSYSTEM_CPUSPEED	NUMERIC		
COMPUTERSYSTEM_ROMVERSION	COMPUTERSYSTEM_ROMVERSION	ALN	4.0.19	

Figure 6-21 CI specifications table

Attention: Your results might not look the same as shown in Figure 6-21. The values, attributes, and number of attributes might differ in your environment.

7. Now that the attributes have been checked, select **Go To** → **IT Infrastructure** → **Configuration Items**. On the List tab filter the Classification field with “linux” and Top Level field with 1 as shown Figure 6-22.



The screenshot shows the 'Configuration Items' web application interface. At the top, there is a navigation bar with links like 'Bullets: (0)', 'Go To', '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', 'Configuration Item', 'Related Configuration Items', 'CI Topology', and 'Interested Parties'. The 'List' tab is active, showing a table of configuration items. The table has columns for 'Configuration Item Name', 'Status', 'Classification', 'Description', and 'Top Level'. The 'Classification' column is filtered with 'linux'. The 'Top Level' column is filtered with '1'. The table shows six records, all with a status of 'NOT READY' and a classification of 'CL.LINUX.COMPUTERSYSTEM'. The first record is highlighted in blue. At the bottom of the table, there is a 'Select Records' checkbox.

Configuration Item Name	Status	Classification	Description	Top Level
CCMDBVM16.TIVLAB.RALEIGH.IBM.COM	NOT READY	CL.LINUX.COMPUTERSYSTEM		☒
CCMDBVM12.TIVLAB.RALEIGH.IBM.COM	NOT READY	CL.LINUX.COMPUTERSYSTEM		☒
CCMDBVM08.TIVLAB.RALEIGH.IBM.COM	NOT READY	CL.LINUX.COMPUTERSYSTEM		☒
CCMDBVM05.TIVLAB.RALEIGH.IBM.COM	NOT READY	CL.LINUX.COMPUTERSYSTEM		☒
CCMDBVM04.TIVLAB.RALEIGH.IBM.COM	NOT READY	CL.LINUX.COMPUTERSYSTEM		☒
CCMDBVM17.TIVLAB.RALEIGH.IBM.COM	NOT READY	CL.LINUX.COMPUTERSYSTEM		☒

Figure 6-22 Filtered (authorized) configuration items

- Click **CI Topology** tab, on the bottom left of the page, select **Detail View** on the radio buttons, and set the Breadth to 1 on the right top of the diagram toolbar as shown in Figure 6-23.

The box in the middle named Linux is of classification CI.LINUXCOMPUTERSYSTEM, and the preceding box is CI.LINUXOS (keep the mouse pointer over the boxes for a few seconds to get their descriptions).

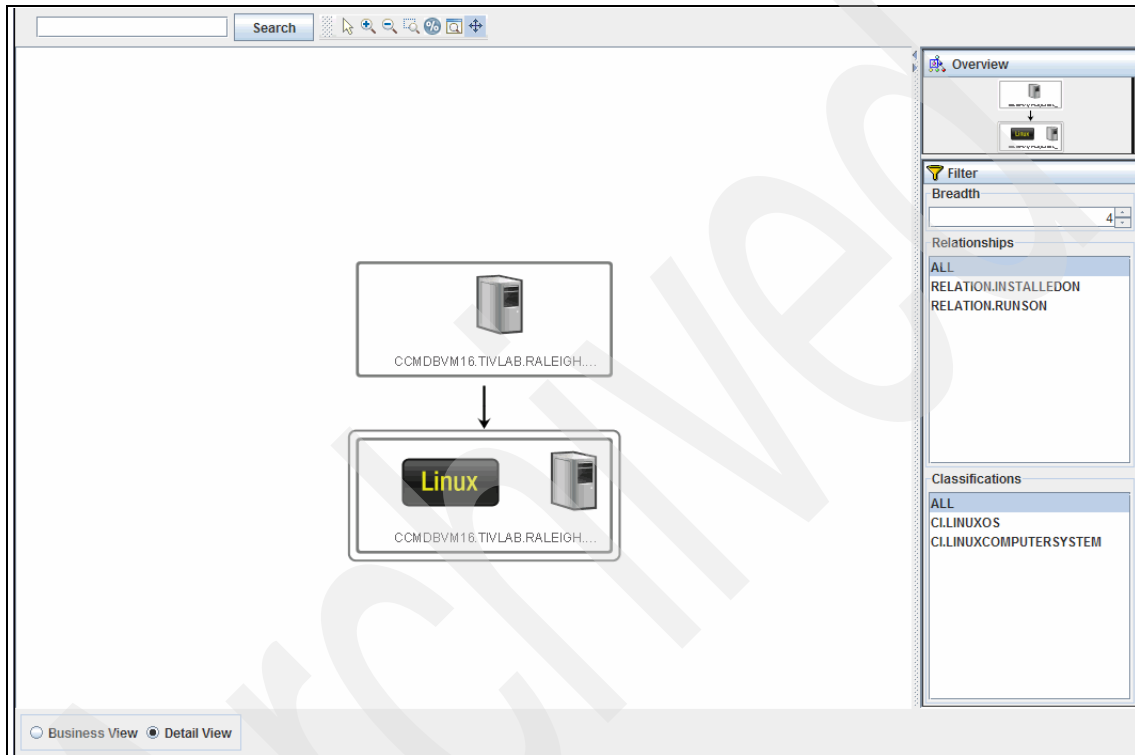


Figure 6-23 Detailed view of the related CIs

Notice that on the toolbar on the right side you can filter the Relationships and Classifications. This is a useful tool when you have a wide promotion scope.

At this point you have checked that the hierarchy scoped on the Deployer's Workbench have been promoted as defined previously and the scope was reduced to promote only CI.LINUXCOMPUTERSYSTEM and CI.LINUXOS.

In a wider promotion scope, the results on the Topology tab under Configuration Items result in many CIs and many relationships as well. You might want to consider going back to this chapter to improve or redesign your CI space.

Importing configuration items

In this chapter we cover the necessary steps to import configuration items from IBM Tivoli Application Dependency Discovery Manager (TADDM) into the IBM Change and Configuration Management Database. We explain important concepts of Integration Composer, its interactions with TADDM and CCMDB, and usage of the product in a Configuration Management project.

We cover the following topics:

- ▶ 7.1, “Integration Composer overview”
- ▶ 7.2, “Import process” on page 385
- ▶ 7.3, “Installing Integration Composer” on page 386
- ▶ 7.4, “Configuring Integration Composer” on page 397
- ▶ 7.5, “References” on page 415

7.1 Integration Composer overview

Integration Composer is an integration tool that imports hardware and software inventory data from external data sources into Tivoli's process automation engine database tables for deployed assets and configuration items. With Integration Composer, an enterprise can aggregate data collected by discovery tools and integrate it, creating a central repository for enterprise IT asset management, reporting, and decision support.

Integration Composer supports integration with various discovery and system management tools, such as Tivoli Application Dependency Discovery Manager. To gather data about deployed assets, the discovery tools scan computers, network devices, and network printers deployed in an enterprise and store information about the hardware and software installed on those assets in a database. Integration Composer transforms the data and imports it into Tivoli's process automation engine database.

Before you import data from an external data source into Tivoli's process automation engine target database, you use Integration Composer to create a mapping to transform data from the source format to the target format. A mapping is a set of expressions that tell 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 into the target. When you execute a mapping, Integration Composer transforms the collected data and imports it into the target.

If you have installed Tivoli Change and Configuration Management Database, you can view and manage imported hardware and software data in the IT Infrastructure applications: configuration items and actual configuration items.

Integration Composer connects to data sources using either Java Database Connectivity (JDBC) technology-enabled drivers or an application programming interface (API).

7.2 Import process

Integration Composer has a Java-based user interface application and an engine.

The user interface is used to create and manage object models as well as rules for data transformation. The engine transforms source database instances and transfers them to the target according to a set of defined rules. Classes from the source database are mapped to target class properties using the mapping rules. Properties from more than one source class can be mapped to a single target class or single target property. The mapping rules can be as simple as a one-to-one mapping or can involve a Java expression.

When Integration Composer executes a mapping, it reads source database instances, applies the rule set, compares the transformed data against the associated target instances (if any exist), and updates the target database in the following ways:

- ▶ If, during the data transfer process, the engine finds a new instance, it inserts that instance into the target.
- ▶ If the engine encounters an existing target instance, it evaluates the instance to determine if newer data exists. If new data exists, the engine updates the target instance. Otherwise, the instance is skipped.
- ▶ If data exists in the target but does not exist in the source, the instance can be deleted from the target. Deletion depends on qualifiers.

Most discovery agents store a data collection time stamp. For increased performance during mapping execution, Integration Composer compares the source time stamp for a top-level instance against an internally stored time stamp, a last scan time stamp. If time stamps differ, the top-level instance along with its child and reference class instances are processed, and the internally stored time stamp is updated. Otherwise these instances are skipped.

Figure 7-1 illustrates process flow.

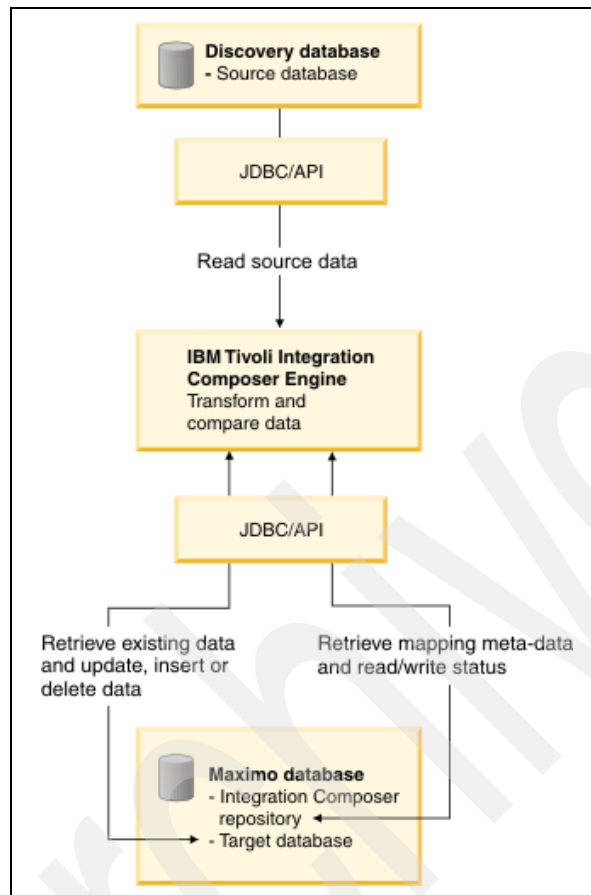


Figure 7-1 Integration Composer process flow

7.3 Installing Integration Composer

There are three ways to install Integration Composer, depending on the platform that you are using:

- ▶ Installing Integration Composer on 32-bit Windows using the launchpad:
For CCMDB installations using 32-bit Windows operating systems, the best way to install Integration Composer is from the launchpad.
- ▶ Installing Integration Composer on 64-bit Windows operating systems:
In this method, you perform this installation from the command line.

- Installing Integration Composer on 64-bit UNIX® operating systems:
In this method, you perform this installation from the command line.

In this section, we describe the Integration Composer installation steps. We use the launchpad install (first of the foregoing options listed), because our CCMDB installation is using a 32-bit Windows operating system. For the other methods of installing Integration Composer, see CCMDB InfoCenter, section called “Performing the Integration Composer installation” at the following website:

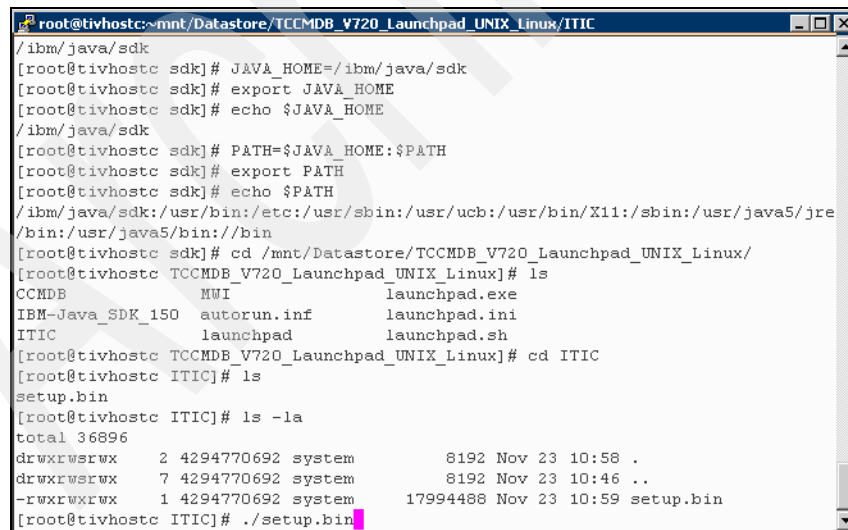
http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/index.jsp?topic=/com.ibm.ccmdb.doc/install/t_itic_performing_the_itic_installation.html

Proceed as follows:

1. Copy and extract IBM Java SDK:

```
cd /opt/IBM/java
gunzip j564redist.tar.gz
tar -xvf j564redist.tar
Change JAVA_HOME and PATH variables
JAVA_HOME=/opt/IBM/java/sdk
export JAVA_HOME
PATH=$JAVA_HOME:$PATH
export PATH
```

2. Execute `./setup.bin` from ITIC installation directory (Figure 7-2).



```
root@tivhostc:~mnt/Datastore/TCCMDB_V720_Launchpad_UNIX_Linux/ITIC
/ibm/java/sdk
[root@tivhostc sdk]# JAVA_HOME=/ibm/java/sdk
[root@tivhostc sdk]# export JAVA_HOME
[root@tivhostc sdk]# echo $JAVA_HOME
/ibm/java/sdk
[root@tivhostc sdk]# PATH=$JAVA_HOME:$PATH
[root@tivhostc sdk]# export PATH
[root@tivhostc sdk]# echo $PATH
/ibm/java/sdk:/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:/usr/java5/jre
/bin:/usr/java5/bin:/bin
[root@tivhostc sdk]# cd /mnt/Datastore/TCCMDB_V720_Launchpad_UNIX_Linux/
[root@tivhostc TCCMDB_V720_Launchpad_UNIX_Linux]# ls
CCMDB      MWI          launchpad.exe
IBM-Java_SDK_150  autorun.inf  launchpad.ini
ITIC        launchpad    launchpad.sh
[root@tivhostc TCCMDB_V720_Launchpad_UNIX_Linux]# cd ITIC
[root@tivhostc ITIC]# ls
Setup.bin
[root@tivhostc ITIC]# ls -la
total 36896
drwxrwxrwx  2 4294770692 system      8192 Nov 23 10:58 .
drwxrwxrwx  7 4294770692 system      8192 Nov 23 10:46 ..
-rwxrwxrwx  1 4294770692 system 17994468 Nov 23 10:59 setup.bin
[root@tivhostc ITIC]# ./setup.bin
```

Figure 7-2 `setup.bin`

3. Select **English** as installation language. Click **OK** to continue.

4. Click **Next** in the introduction and select the installation type (Figure 7-3 and Figure 7-4).

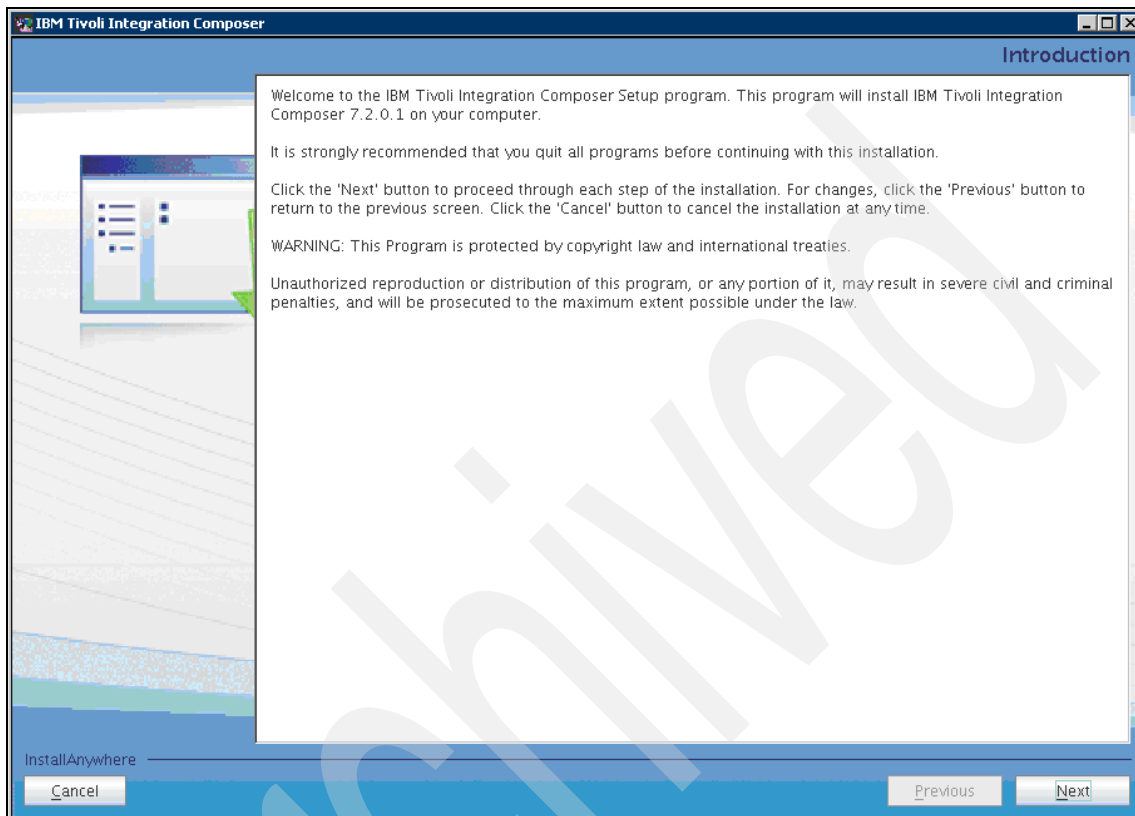


Figure 7-3 Installation Introduction

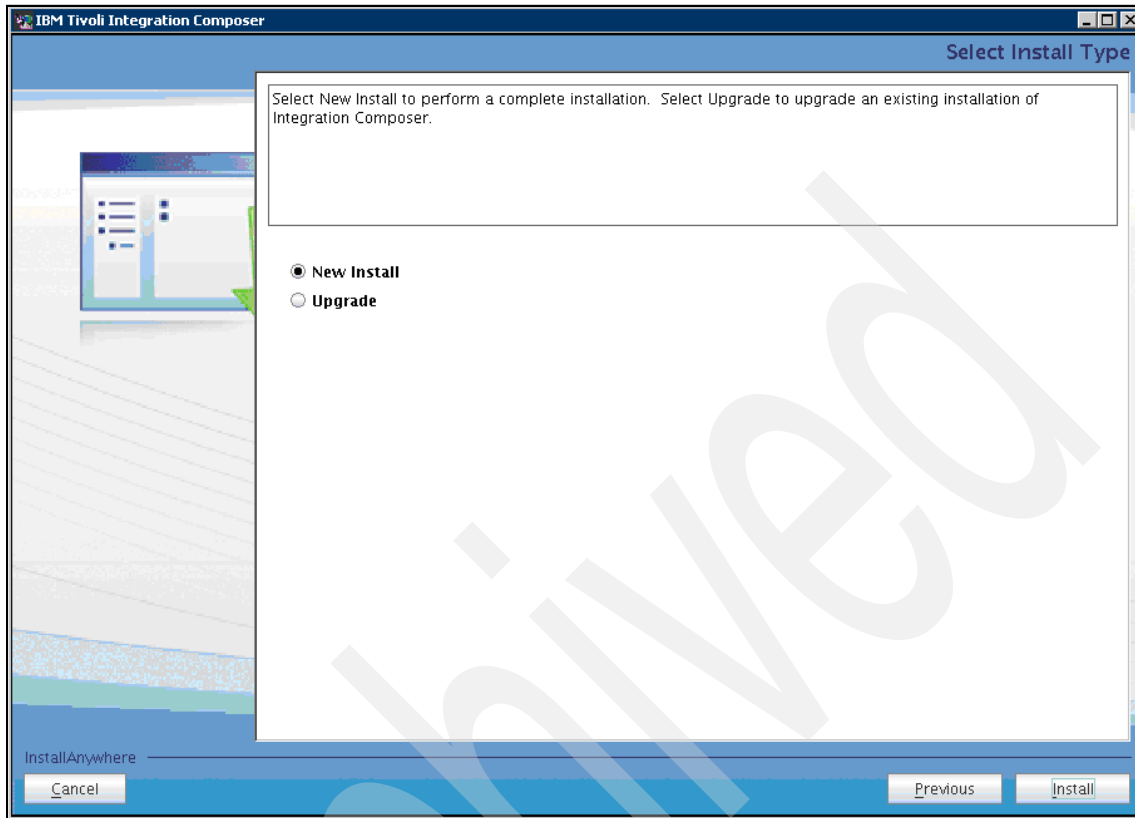


Figure 7-4 Select Install Type

5. If this is a new Installation, select **New Install**, if you are upgrading Integration Composer 7.2 from a previous version, select **Upgrade**. Click the **Install** button to start the installation (Figure 7-5).

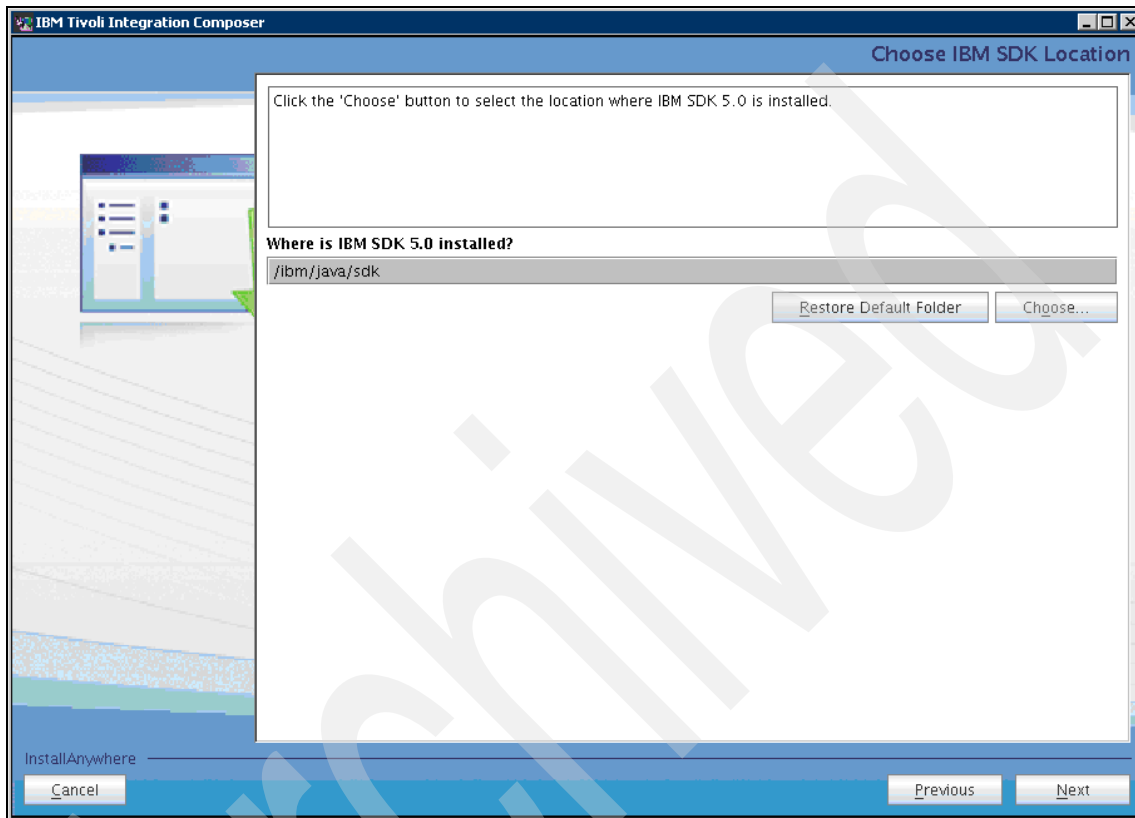


Figure 7-5 Choose IBM SDK Location

6. Select the location where IBM Java SDK 1.5 is installed. Click **Next** (Figure 7-6).

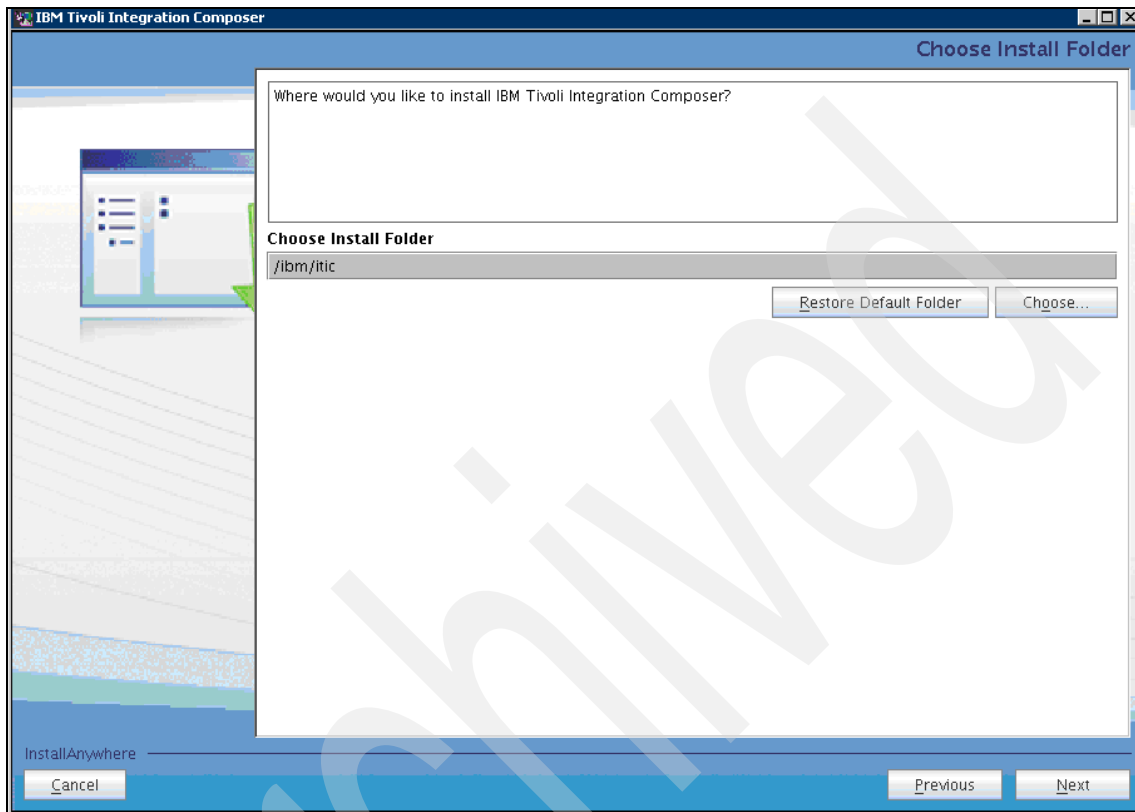


Figure 7-6 Choose Install Folder

7. Select the directory where Integration Composer executable and configuration files will be deployed. Click **Next** to continue.
8. Select the database type that you will be using with Integration Composer.

9. Integration composer will share the same database with IBM Change and Configuration Management database. Select the type accordingly. Click **Next** to continue (Figure 7-7).

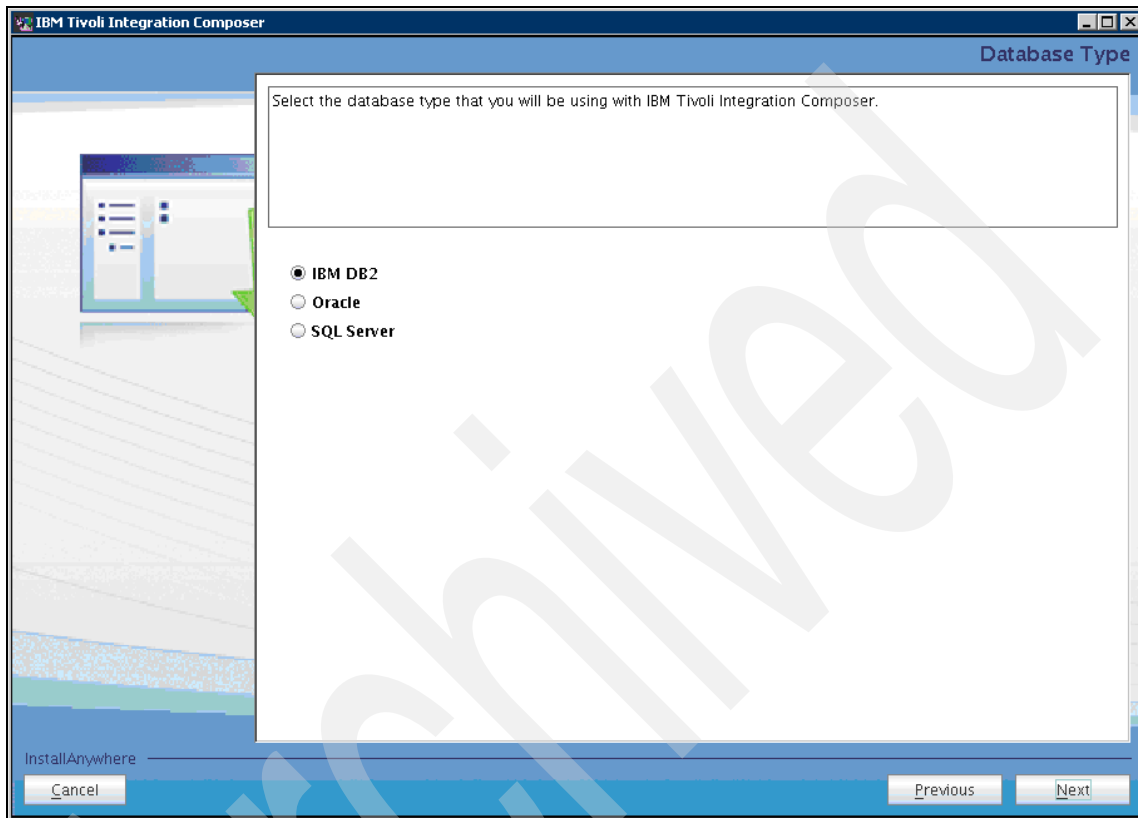


Figure 7-7 Select Database Type

10. Enter the Database Login information (Figure 7-8):

– DB2 example:

Database Server Name	hostname.domain.com
Port Number	50005
Database Name	maxdb71
Database Username	maximo
Database/Schema Owner	maximo

– ORACLE example:

Database Server Name	hostname.domain.com
Port Number	1521
Database Name (SID)	CCMDB
Database Username	maximo
Database/schema Owner	maximo

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: tivohostc.ic.tr.ibm.com

Port Number: 50005

Database Name: maxdb71

Database Username: maximo

Database/Schema Owner: maximo

InstallAnywhere

Cancel Previous Next

Figure 7-8 Database Login Information for DB2

11.The installation progresses (Figure 7-9).

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: tivhostc.ic.tr.ibm.com

Port Number: 1521

Database Name (SID): maxdb71

Database Username: maximo

Database/Schema Owner: maximo

InstallAnywhere

Cancel Previous Next

Figure 7-9 Database Login Information for ORACLE

12.Select **Yes, Disable software updates**.

Attention: If you disable software updates, Integration Composer inserts and/or deletes software records but does not update existing records. *Yes* is the preferable setting for performance reasons.

See Figure 7-10 for an example.

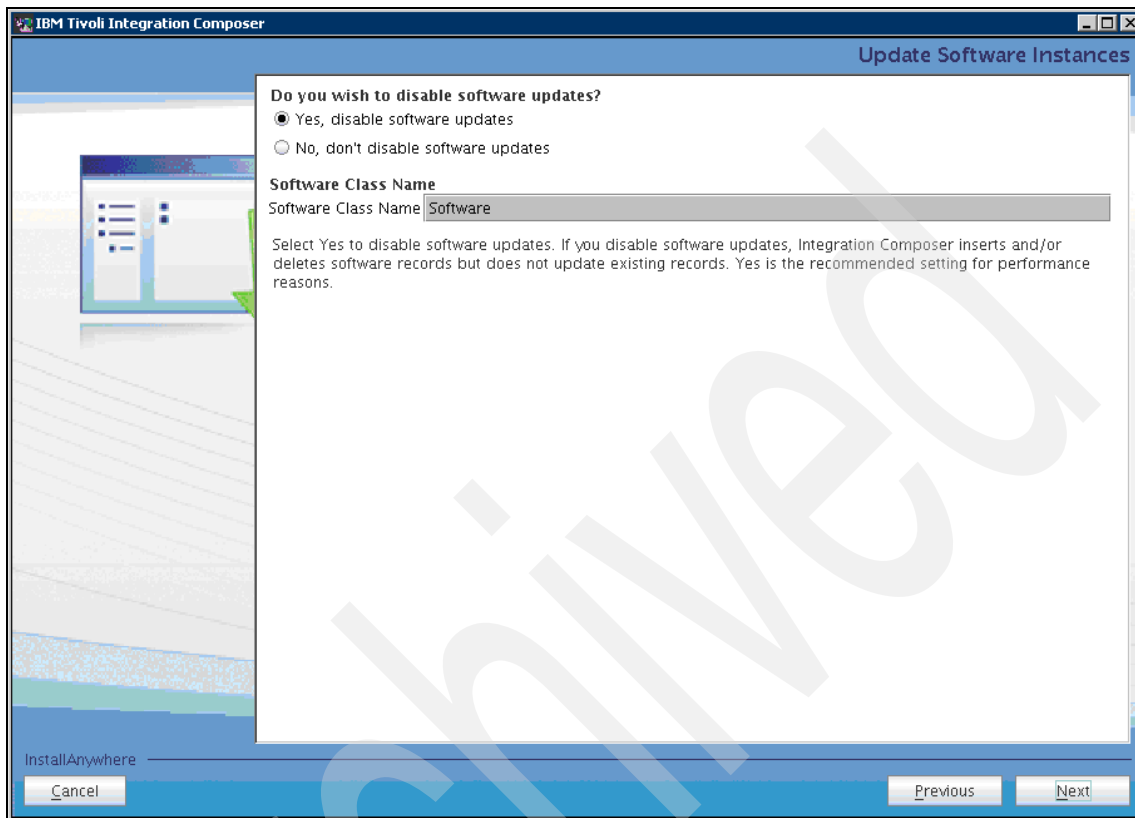


Figure 7-10 Update software Instance

13. Click the **Choose** button to select the main application file for the Internet browser to be used by Integration Composer online help (Figure 7-11).

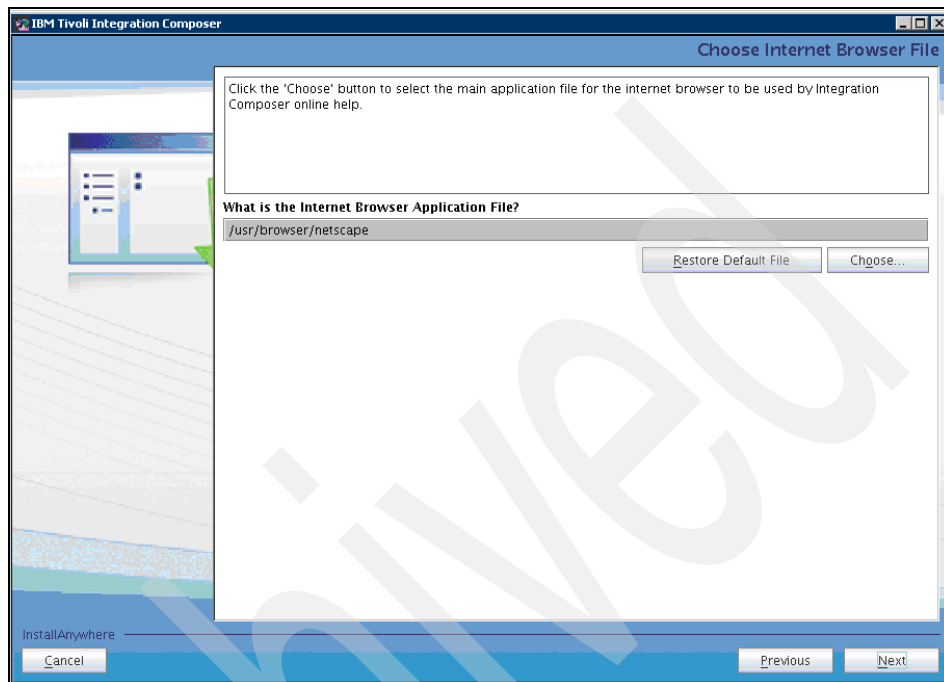


Figure 7-11 Choose Internet Browser File

14. Examine the Pre Installation Summary. Click the **Install** button to start installation, as shown in Figure 7-12.

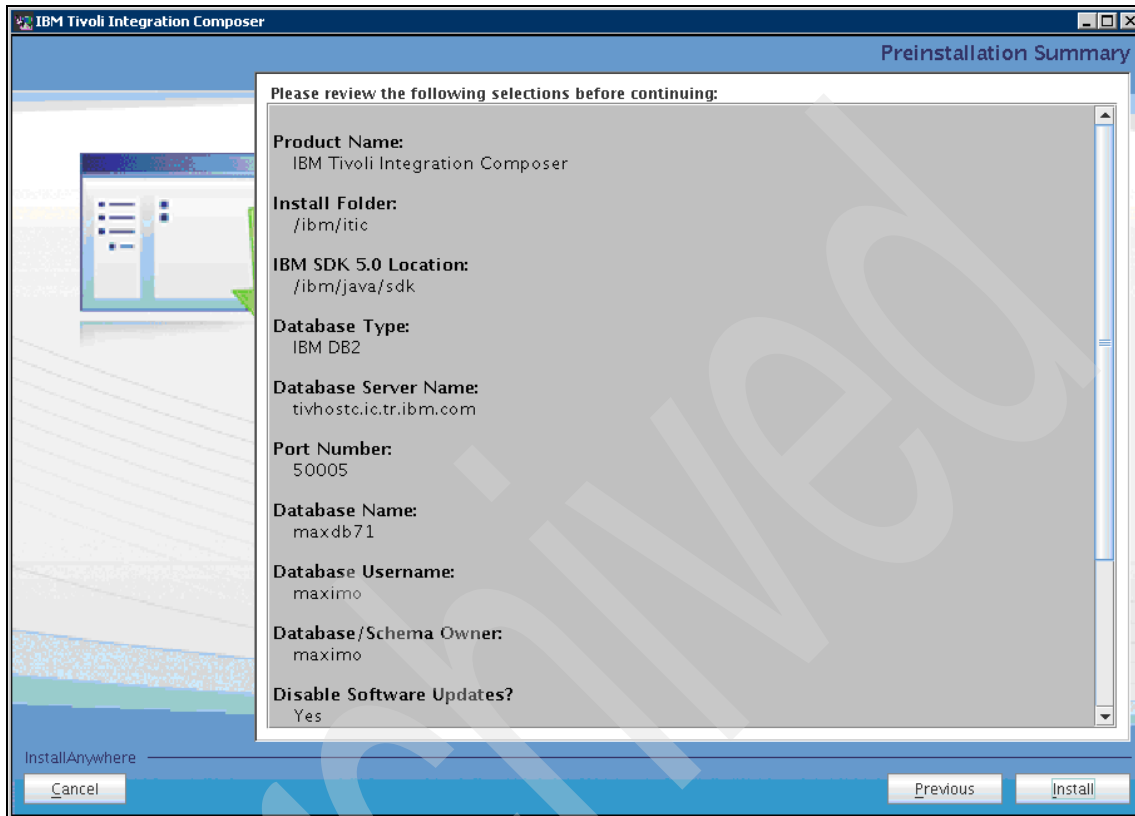


Figure 7-12 Pre Installation Summary

15. Click **Done** button to close the installation window.

7.4 Configuring Integration Composer

Verify that the TADDM server is running before starting the Integration Composer configuration, as the integration will be using TADDM API to import content from the database. Also check for network accessibility. Integration Composer will attempt to connect port 9530 of TADDM server by default:

1. Copy two jar files from Tivoli Application Dependency Discovery Manager to Integration Composer:

Copy taddm-api-client.jar and platform-model.jar files from /opt/IBM/cmdb/sdk/dist/lib of TADDM server to /ibm/itic/lib folder of ITIC server:

```
scp  
cmdbuser@taddm_host:/opt/IBM/cmdb/sdk/dist/lib/taddm-api-client.jar  
/ibm/itic/lib  
  
scp  
cmdbuser@taddm_host:/opt/IBM/cmdb/sdk/dist/lib/platform-model.jar  
/ibm/itic/lib
```

2. Start Integration Composer:
/ibm/itic/bin/startFusion.sh
3. Enter username and password.

Tip: This user is the database user you have created while you installed CCMDB.

4. Define the following new Data Sources:
 - TADDM72CType
 - TADDM72ActualCI
 - a. Click **Define New Data Source**.
 - b. Enter TADDM72CType in Data Source field. See Figure 7-13.

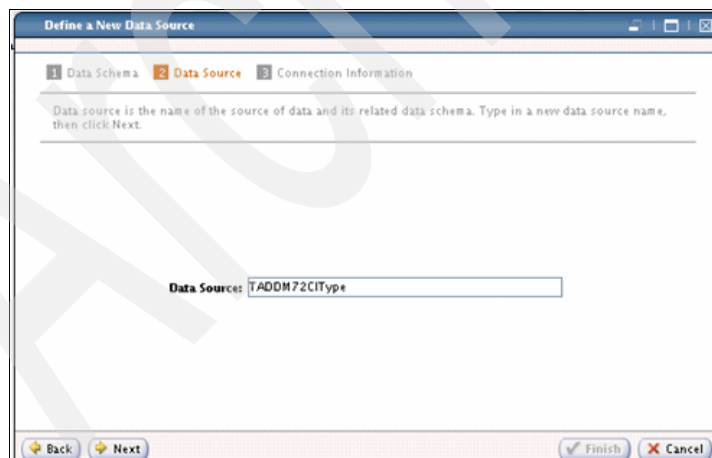


Figure 7-13 Define TADDM72CType Data Source

- c. Click **Next**.

d. Fill in the connection information form (Figure 7-14).

Define a New Data Source

1 Data Schema 2 Data Source 3 **Connection Information**

Connection Method IBM Configuration Discovery and Tracking API

Host Name: tivhostc.ic.tr.ibm.com

Host Port: 9530

Use SSL: ☐

Trusted Location:

User Name: administrator

Password: ****

Bidi Layout Format Clear Test Connection

Back Next Finish Cancel

Figure 7-14 Connection Information TADDM72CIType Data Source

Connection Method	IBM Configuration Discovery and Tracking API
Host Name	taddm_host.domain.com (Enter Fully Qualified Domain Name or IP address of TADDM Server.)
Host Port	9530
Use SSL	Unchecked
Trusted Location	Leave Blank
User Name	Administrator
Password	collation

- Click **Test Connection** button. If you do not see connection successful message, check the connection information.
- Click **Finish** button.
- Click **Define New Data Source** menu item.

- h. Enter TADDM72CIType in Data Source field.
 - i. Repeat steps c to f.
5. Define two new CCMDB data sources:

- CCMDB72CIType
- CCMDB72ActualCI

- a. Click **Define New Data Source** menu item.
- b. Enter CCMDB72CIType in Data Source field.
- c. Fill in the connection information form.

DB2 Example:

Connection Method	IBM DB2 JDBC Driver (select appropriate database driver)
Host Name	maximo_database_host.domain.com (enter Fully Qualified Domain Name or IP address for your Tivoli's process automation engine Database server).
Database	MAXDB71
User Name	maximo
Password	maximo
Table Domain	maximo

ORACLE Example :

Connection Method	Oracle JDBC Thin Driver
Host Name	maximo_database_host.domain.com
Host SID	CCMDB
User Name	MAXIMO
Password	maximo
Table Owner	maximo

- d. Click **Test Connection** button. If you do not see connection successful message, check the connection information.
- e. Click **Finish** button.
- f. Click **Define New Data Source** menu item.
- g. Enter CCMDB72ActualCI in Data Source field.
- h. Repeat steps c to e.

7.4.1 Mapping data sources

1. Create CI TYPE mapping:
 - a. Click **Mapping** → **Create New Mapping** menu item.
 - b. Fill in the mapping form: (see Figure 7-15 on page 401).

Source	TADDM72CIType
Target	CCMDB72CIType
Mapping Name	taddm2ccmdb_ci_type

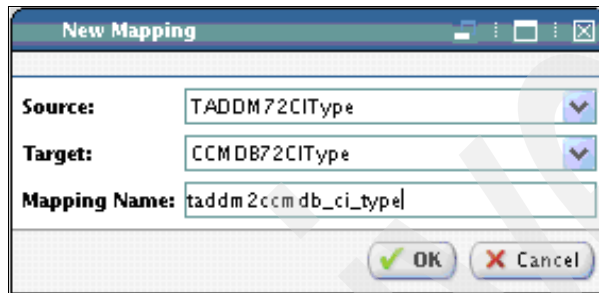


Figure 7-15 New Mapping

- c. Click **OK**.

- d. Check the information in the Open Source Data Source: TADDM72CIType dialog box (Figure 7-16).

Open Source Data Source: TADDM72CIType

1 Connection Information

Connection Method IBM Configuration Discovery and Tracking API

Host Name: tivhostc.ic.tr.ibm.com

Host Port: 9530

Use SSL: ☐

Trusted Location:

User Name: administrator

Password: ****

Back Next Bidi Layout Format Clear Test Connection Finish Cancel

Figure 7-16 Open Source Data Source

- e. Fill in the Password field with the TADDM administrator user's password.
- f. Click the **Test Connection** button. If the test does not succeed, check the connection information.
- g. Click **Finish**.

- h. Check the information in the Open Target Data Source: CCMDB72CIType dialog box (Figure 7-17).

Open Target Data Source: CCMDB72CIType

1 Connection Information

Connection Method: IBM DB2 JDBC Driver

Host Name: tivhostc.ic.tr.ibm.com

Host Port: 50005

Database: MAXDB71

User Name: maximo

Password: [masked]

Table Owner: maximo

Buttons: Bidi Layout Format, Clear, Test Connection, Back, Next, Finish, Cancel

Figure 7-17 Open Target Data Source

- i. Fill in the Password field with Tivoli's process automation engine user's password.
- j. Click the **Test Connection** button. If the test does not succeed, check the connection information.
- k. Click **Finish**.

- I. An empty Integration Composer Mapping record is created with one source and one target data source. But field by field mapping is not yet defined (Figure 7-18). We need to match each field in the source database to a field in target database and define how to convert data correctly. Fortunately, we do not have to deal with this detail because an adapter to convert TADDM data into CCMDB is already provided with the product.

IBM Tivoli Integration Composer

Integration Composer

Mapping: taddm2ccmdb_ci_type

Source: TADDM72CIType

Class

Key	Relation	Name	Type	Length	Value
		Classname	String	192	
		Parent	String	192	
		Haschildren	String	192	
		Hierarchylevel	String	192	
		Istoplevel	String	192	
		Isrelationship	String	192	

Target: CCMDB72CIType

Classification

Key	Relation	Name	Type	Length	Value
		Classificationid	String	192	
GV		Classificationuid	int	10	
		Description	String	100	
		HasId	int	10	
		Orgid	String	8	
		Siteid	String	8	

Figure 7-18 Integration Composer Mapping

- m. Click **Select Action** → **Import** menu item.

- n. Select the **TADDM72CITypeToCCMDB72Classification.fsn** file (Figure 7-19).

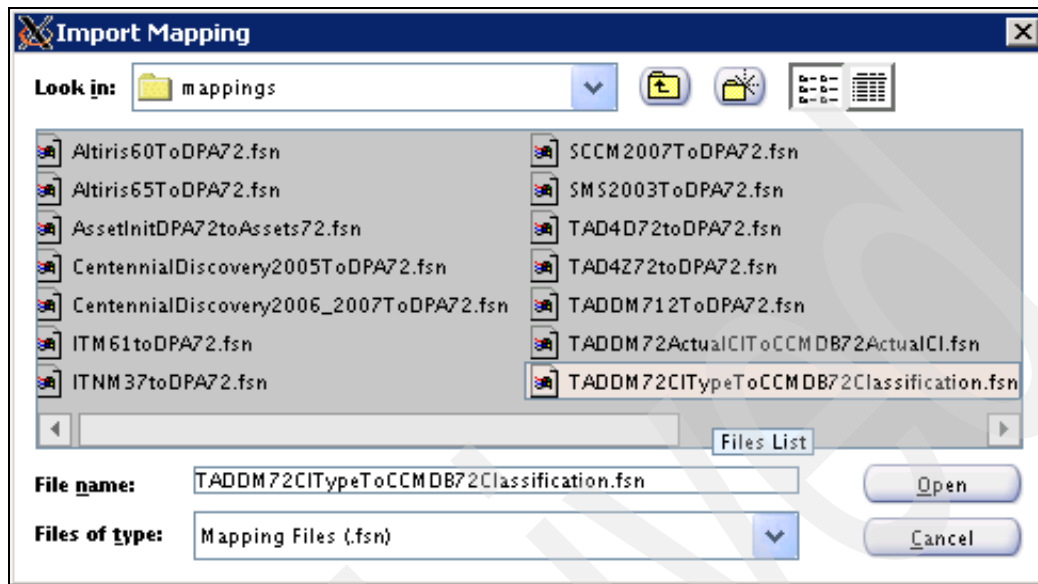


Figure 7-19 Import mapping file

- o. After mapping is imported, click Select **Action** → **Save** menu item.
- p. Click Select **Action** → **Close** menu item.
2. Create ACTUAL CI mapping.
 - a. Click **Mapping** → **Create New Mapping** menu item
 - b. Fill in the mapping form:

Source	TADDM72ActualCI
Target	CCMDB72ActualCI
Mapping Name	taddm2ccmdb_actual_ci
 - a. Click **Mapping** → **Create New Mapping** menu item
 - b. Click **OK** button.
 - c. Check the information in the Open Source Data Source: TADDM72ActualCI dialog box.
 - d. Fill in the Password field with the TADDM administrator user's password.
 - e. Click Test Connection button. If test does not succeed, check the connection information.

- f. Click **Finish**.
 - g. Check the information in the Open Target Data Source: CCMDB72ActualCI dialog box. Fill in the Password field with the user's password for Tivoli's process automation engine.
 - h. Click the **Test Connection** button. If test does not succeed, check the connection information.
 - i. Click **Finish**.
 - j. An empty Integration Composer Mapping record is created with one source and one target data source. But field by field mapping is not yet defined.
 - k. Click **Select Action** → **Import** menu item.
 - l. Select **TADDM72ActualCIToCCMDB72ActualCI.fsn** file.
 - m. Import the mapping file.
 - n. After mapping file has been imported, click **Select Action** → **Save** menu item.
 - o. Click **Select Action** → **Close** menu item.
3. Create batch scripts to execute mappings (optional).

Here we create shell scripts (UNIX) or batch files (Windows) to make the execution of mappings easier. Be sure to make the necessary corrections in the directory names, user names, and credentials to match your environment before using the example scripts given.

Parameters for **commandLine.sh** or **commandLine.bat**:

Mapping name	Mapping name defined in ITIC
ITIC user name	User name to connect ITIC
ITIC password	Password of ITIC user
TADDM user name	TADDM administrator user name
TADDM password	TADDM administrator password
Maximo db user name	User name to connect maximo database
Maximo db password	Password of maximo database user

UNIX example:

- a. Create a text file named :

```
/opt/IBM/itic/bin/execute_ci_type_map.sh
```
- b. Edit the file with a text editor (for example vi) and enter:

```
/opt/IBM/itic/bin/commandLine.sh taddm2ccmdb_ci_type maximo  
maximo administrator collation maximo maximo
```
- c. Save file and change permission of the file to make it executable.

```
chmod 777 execute_ci_type_map.sh
```
- d. Create a text file named :

```
/opt/IBM/itic/bin/execute_actual_ci_map.sh
```
- e. Edit the file with a text editor (for example vi) and enter:

```
/opt/IBM/itic/bin/commandLine.sh taddm2ccmdb_actual_ci maximo  
maximo administrator collation maximo maximo
```
- f. Save file and change permission of the file to make it executable.

```
chmod 777 execute_actual_ci_map.sh
```

WINDOWS example:

- a. Create a text file named :

```
c:\IBM\itic\bin\execute_ci_type_map.bat
```
- b. Edit the file with a text editor (for example notepad) and enter:

```
c:\IBM\itic\bin\commandLine taddm2ccmdb_ci_type maximo maximo  
administrator collation maximo maximo
```
- c. Create a text file named :

```
c:\IBM\itic\bin\execute_actual_ci_map.bat
```
- d. Edit the file with a text editor (for example notepad) and enter:

```
c:\IBM\itic\bin\commandLine taddm2ccmdb_actual_ci maximo  
maximo administrator collation maximo maximo
```

7.4.2 Importing CI types

You need to import Common Data Model classifications into CCMDB before importing configuration items from TADDM database. This is a one time only process after configuring the environment.

Attention: In Integration Composer 7.2, mapping execution is only possible from the command line. The graphical user interface to run mappings, which was found in previous versions, is no longer available.

You can execute CI Types mapping with the following commands:

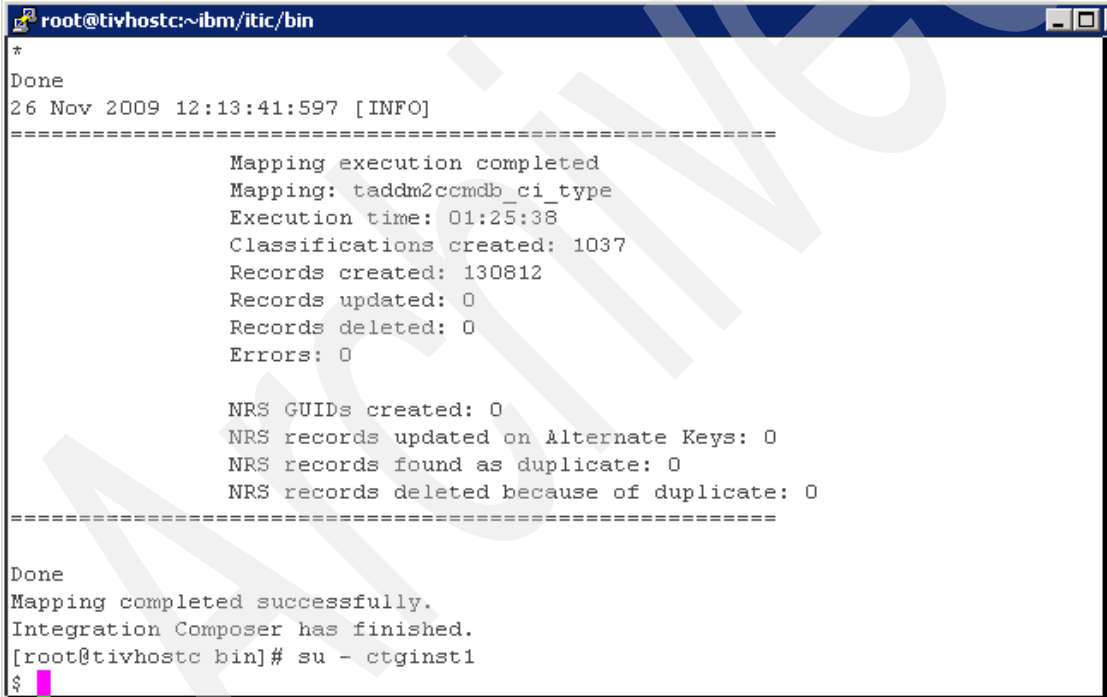
For UNIX:

`/opt/IBM/itic/bin/execute_ci_type_map.sh`

For Windows:

`c:\IBM\itic\bin\execute_ci_type_map.bat`

See Figure 7-20.



```
root@tivhostc:~ibm/itic/bin
*
Done
26 Nov 2009 12:13:41:597 [INFO]
=====
Mapping execution completed
Mapping: taddm2ccmdb_ci_type
Execution time: 01:25:38
Classifications created: 1037
Records created: 130812
Records updated: 0
Records deleted: 0
Errors: 0

NRS GUIDs created: 0
NRS records updated on Alternate Keys: 0
NRS records found as duplicate: 0
NRS records deleted because of duplicate: 0
=====
Done
Mapping completed successfully.
Integration Composer has finished.
[root@tivhostc bin]# su - ctginst1
$
```

Figure 7-20 Running CI Type mapping

Attention: Running CI Type mapping can take a long time (typically more than one hour) to complete.

7.4.3 Importing configuration items

Start with activating the CI types:

1. Log in to CCMDB
2. **Go To → Administration → CI Types.**
3. Activate the first top level configuration item type (Figure 7-21).

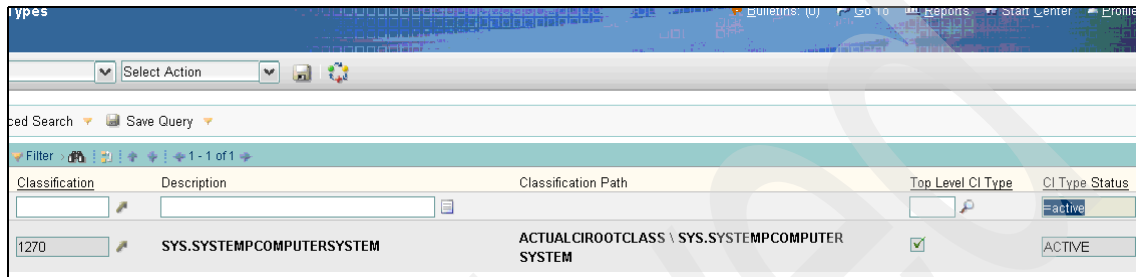


Figure 7-21 Activate Top Level CI Type

Important: Activating Top Level CI type for the first time will take a long time. Do not close your browser until the operation completes.

DB2 bind Java

If your maximo database resides on DB2, you need to bind Java classes. Otherwise you will encounter an error message such as this one when you try to import:

```
com.ibm.db2.jcc.c.SqlException: DB2 SQL error: SQLCODE: -805, SQLSTATE:
51002,
SQLERRMC: NULLID.SYSLH204 0X5359534C564C3031
```

Run the following command on the database server as a root user. Do not execute it under *ctginst1* or another database user.

```
cd /ibm/opt/IBM/db2/V9.5/bin
java -cp
/ibm/home/ctginst1/sqllib/java/db2java.zip:/ibm/home/ctginst1/sqllib/ja
va/db2jcc.jar:/ibm/home/ctginst1/sqllib/function:/ibm/home/ctginst1/sql
lib/java/db2jcc_license_cu.jar:. com.ibm.db2.jcc.DB2Binder -url
jdbc:db2://tivhostc.ic.tr.ibm.com:50005/maxdb71 -user maximo -password
passwOrd -size 70
```

Setting depth level

Depth level 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.

To set the depth level to import configuration items, proceed as follows:

1. Edit `ccmdb.properties` file:

```
vi /opt/IBM/itic/data/properties/provider/ccmdb.properties.
```

2. Specify the default global depth level to use when importing TADDM CIs by setting the `ccmdb.classification.default.depth` property. The default depth level is 3.

This depth setting will be in effect for all “active” top-level classifications.

Attention: Activating a classification will cause not only that classification to be imported but also all of its sub-classes. Those sub-classes will be imported at this global depth setting unless specifically overridden by the `ccmdb.classification.depth` property.

3. The `ccmdb.classification.depth` property provides a way to specify additional classification depths.

For example, if you want all `SYS.COMPUTERSYSTEM` CIs imported at the default depth of 3 but all `APP.WEB.APACHE.APACHESERVER` CIs to be imported at a depth of 5, first you activate these two classifications and then uncomment the `APP.WEB.APACHE.APACHESERVER` line and change the depth to 5:

```
ccmdb.classification.depth.APP.WEB.APACHE.APACHESERVER=5
```

Only 'active' top-level classifications can be imported at a depth that is greater than 1.

An “active” non-top-level classification will be imported at a depth of 1 and must only be made active if this class will not be imported as part of an imported relationship.

The “`ccmdb.classification.depth`.” property can also be used to exclude importing certain classifications. You can set the depth to -1 for non-active classifications to indicate that those instances must not be imported.

Tip: If you already have instances of this excluded class in the database, they will not be deleted. Setting the depth to -1 will avoid importing any new instances of that class but will not remove already existing instances.

Here are the top level classifications:

ADMIN.CONTACTPURPOSE
APP.APPLICATION
APP.APPSERVER
APP.APPSERVERCLUSTER
APP.CITRIX.CITRIXFARM
APP.DB.DATABASESERVER
APP.DB.DB2.DB2DATABASE
APP.DB.DB2.DB2SERVER
APP.DB.HIRDB.HIRDBRDAREA
APP.DB.HIRDB.HIRDBSERVER
APP.DB.HIRDB.HIRDBSYSTEM
APP.DB.HIRDB.HIRDBUNIT
APP.DB.MSSQL.SQLSERVER
APP.DB.MSSQL.SQLSERVERDATABASE
APP.DB.ORACLE.ORACLEDATABASE
APP.DB.ORACLE.ORACLEINSTANCE
APP.DB.SYBASE.SYBASEDATABASE
APP.DB.SYBASE.SYBASESERVER
APP.DEFINITIVESOFTWARELIBRARY
APP.J2EE.J2EESERVER
APP.J2EE.JBOSS.JBOSSSERVER
APP.J2EE.ORACLEAPP.ORACLEAPPJ2EESERVER
APP.J2EE.WEBLOGIC.WEBLOGICSERVER
APP.J2EE.WEBSPHERE.WEBSPHERESERVER
APP.LOTUS.DOMINOSERVER
APP.MESSAGING.MQ.MQQUEUEMANAGER
APP.PACKAGEDAPP.MYSAP.BASIS.MYSAPABAPAPPLICATIONSERVER
APP.PACKAGEDAPP.MYSAP.BASIS.MYSAPCLUSTER
APP.PACKAGEDAPP.MYSAP.BASIS.MYSAPJ2EEENGINEINSTANCE
APP.PACKAGEDAPP.MYSAP.BASIS.MYSAPJAVACENTRALSYSYSTEM
APP.PACKAGEDAPP.MYSAP.DB.MYSAPDB2INSTANCE
APP.PACKAGEDAPP.MYSAP.DB.MYSAPORACLEINSTANCE
APP.PACKAGEDAPP.MYSAP.DB.MYSAPSQLSERVER
APP.PACKAGEDAPP.MYSAP.SAPSYSTEM
APP.SMS.SMSRESOURCE
APP.SOFTWAREIMAGE
APP.SOFTWAREMODIFICATION
APP.SOFTWAREPRODUCT
APP.TWSAGENT
APP.WEB.APACHE.APACHEPROXYSERVER
APP.WEB.APACHE.APACHESERVER
APP.WEB.IIS.IISWEBSERVICE
APP.WEB.IPLANET.IPLANETSERVER

APP.WEB.WEBPROXYSERVER
METRIC.OPERATINGSYSTEMMETRIC
NET.VLAN
PHYS.PHYSICALLOCATION
PROCESS.BUSINESSPROCESS
PROCESS.BUSINESSSERVICE
PROCESS.COMPOSITE.APPLICATIONCOMPOSITE
PROCESS.COMPOSITE.BUSINESSPROCESSCOMPOSITE
PROCESS.COMPOSITE.BUSINESSSERVICECOMPOSITE
PROCESS.COMPOSITE.DESKTOPCOMPOSITE
PROCESS.COMPOSITE.HARDWARECOMPOSITE
PROCESS.COMPOSITE.INFRASTRUCTUREAPPLICATIONCOMPOSITE
PROCESS.COMPOSITE.MAINFRAMECOMPOSITE
PROCESS.COMPOSITE.ROUTERCOMPOSITE
PROCESS.COMPOSITE.SERVERCOMPOSITE
PROCESS.COMPOSITE.SWITCHCOMPOSITE
PROCESS.DOCUMENT
PROCESS.IGS.MONITOR
PROCESS.IGS.PROBESCRIP
PROCESS.ILIL.SLA
PROCESS.ILIL.SLO
PROCESS.PARTNER
SOA.WEBSERVICE
STORAGE.DATASERVICECAPABILITIES
STORAGE.DATASERVICELEVELOBJECTIVEGROUP
STORAGE.FCSWITCH
STORAGE.HOSTBUSADAPTOR
STORAGE.STORAGEPOOL
STORAGE.STORAGESUBSYSTEM
STORAGE.TAPELIBRARY
STORAGE.TAPEMEDIACHANGER
STORAGE.ZONE
SYS.ACTIVEDIRECTORY
SYS.AIX.AIXUNITARYCOMPUTERSYSTEM
SYS.BUSINESSSYSTEM
SYS.COMPUTERSYSTEM
SYS.COMPUTERSYSTEMCLUSTER
SYS.CONTROLSOFTWARE
SYS.DARWIN.DARWINUNITARYCOMPUTERSYSTEM
SYS.DOS.DOSUNITARYCOMPUTERSYSTEM
SYS.FREEBSD.FREEBSDUNITARYCOMPUTERSYSTEM
SYS.HMC
SYS.HPC.CM.CONFIGURATIONMANAGEMENTNODE
SYS.HPUX.HPUXUNITARYCOMPUTERSYSTEM
SYS.IPSO.IPSOUNITARYCOMPUTERSYSTEM

```
SYS.ITSYSTEM
SYS.IVM
SYS.LDAPSERVICE
SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM
SYS.NETWARE.NETWAREUNITARYCOMPUTERSYSTEM
SYS.OPENVMS.OPENVMSUNITARYCOMPUTERSYSTEM
SYS.RUNTIMEPROCESS
SYS.SERVICE
SYS.SUN.SUNSPARCUNITARYCOMPUTERSYSTEM
SYS.SYSTEMPCOMPUTERSYSTEM
SYS.UNITARYCOMPUTERSYSTEM
SYS.VIOS
SYS.VMWARE.VMWAREUNITARYCOMPUTERSYSTEM
SYS.WINDOWS.WINDOWSCOMPUTERSYSTEM
SYS.WPARCOMPUTERSYSTEM
SYS.ZOS.ADDRESSSPACE
SYS.ZOS.CICSREGION
SYS.ZOS.DB2SUBSYSTEM
SYS.ZOS.IMSSUBSYSTEM
SYS.ZOS.LPAR
SYS.ZOS.SYSPLEXGROUP
SYS.ZOS.ZCOUPLINGFACILITY
SYS.ZOS.ZSERIESCOMPUTERSYSTEM
```

Running the actual CI mapping

Refer to step 3 on page 406 to create execution scripts.

Execute CI Types mapping with the following command:

UNIX:

```
/opt/Ibm/itic/bin/execute_actual_ci_map.sh
```

Windows:

```
c:\IBM\itic\bin\execute_actual_ci_map.bat
```

Attention: In Integration Composer 7.2, mapping execution is only possible from the command line. The graphical user interface to run mappings, which was found in previous versions, is no longer available.

A typical actual CI mapping is shown in Figure 7-22.

```

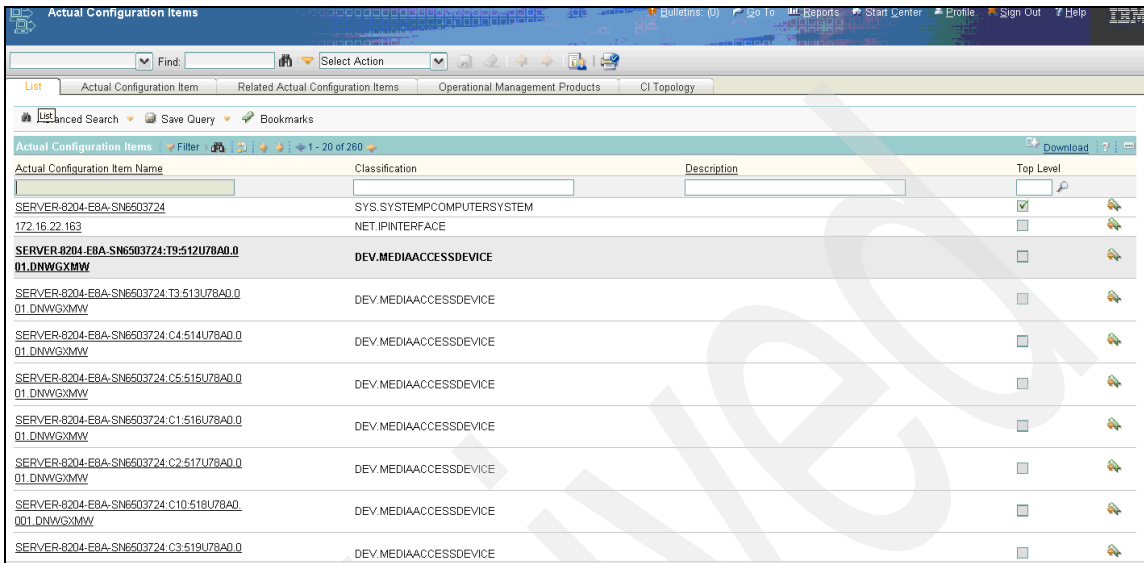
root@tivhostc:~ibm/itc/bin
Actual CI: 236 D1EA06ED701F318682D32303B7D8DDE1
Actual CI: 237 BCDE4266F9939E494AA8B578462EA97
Actual CI: 238 8FBAD08D168B3FE3AE6C661484165FE9
Actual CI: 239 7359C4193E843663B17C7E2282E434F0
Actual CI: 240 2C8C0DB749C63956A1E21E815F041036
Actual CI: 241 83A1DB5ABF5A354F9F22B1139F5CAA52
Actual CI: 242 42407C25689237B782244A25C8A6E6B0
Actual CI: 243 741C704DD39A369CBB4A2F4573991F46
Actual CI: 244 551F231E80AB3B9E93BBF3F5F9E7AEAC
Actual CI: 245 F47FD135F84939C288EAF1737645C75
Actual CI: 138 29BA64C100D532E8E20B07E7BCCCC74
Actual CI: 261 13FB407E79A63BF0869F4FC9A63D6C8D
Actual CI: 247 60D55FCF31323A2DBB597A738925D476
Actual CI: 248 636F670D9B2E3D5B951163AA90F3F9A0
Actual CI: 249 7DC982CAB1B8378CB802C59B0BF12A82
Actual CI: 250 A868C5DCFBFF38D5AD7B607C53A13D38
Actual CI: 251 9099FF4F1CEA3F22863AE600E23E0139
Actual CI: 252 410767902C1E345C9E5758CC577361FD
Actual CI: 253 11546C5E43A035D2B598A70C2E001112
Actual CI: 254 9A6D78225566376CB98AD93C91EF350D
Actual CI: 255 F553E5901C0438EAB772105649BA217A
Actual CI: 256 AF8B6D3B9A7636BC81F5EBA43627C344
Actual CI: 257 B6045158211932C4956DD7CE43B3FDE5
Actual CI: 258 E45DD98C97573A018588F3757A55AE5E
Actual CI: 259 40C06946CA51337CAE9B496D887A72FA
Actual CI: 260 EAF882F8BCAF33B68401D47E23D87805
Actual CI: 139 8EE0A77E77953C94A37FC1DC07FAFC2D
26 Nov 2009 16:47:05:433 [INFO]
=====
Mapping execution completed
Mapping: taddm2ccmdb actual_ci
Execution time: 00:00:45
Actual CIs created: 3
Records created: 4104
Records updated: 0
Records deleted: 0
Errors: 0

NRS GUIDs created: 0
NRS records updated on Alternate Keys: 0
NRS records found as duplicate: 0
NRS records deleted because of duplicate: 0
=====
Done
Mapping completed successfully.
Integration Composer has finished.
[root@tivhostc bin]#

```

Figure 7-22 Running actual CI mapping

- 4. Log in to Tivoli's process automation engine.
- 5. **Go To** → **IT Infrastructure** → **Actual CI** (see Figure 7-23).



Actual Configuration Item Name	Classification	Description	Top Level
SERVER-8204-E8A-SN6503724	SYS.SYSTEMPCOMPUTERSYSTEM		☒
172.16.22.163	NET.IPINTERFACE		☐
SERVER-8204-E8A-SN6503724-T9-512U78A0.0 01.DNWGXMMW	DEV.MEDIAACCESSDEVICE		☐
SERVER-8204-E8A-SN6503724-T3-513U78A0.0 01.DNWGXMMW	DEV.MEDIAACCESSDEVICE		☐
SERVER-8204-E8A-SN6503724-C4-514U78A0.0 01.DNWGXMMW	DEV.MEDIAACCESSDEVICE		☐
SERVER-8204-E8A-SN6503724-C5-515U78A0.0 01.DNWGXMMW	DEV.MEDIAACCESSDEVICE		☐
SERVER-8204-E8A-SN6503724-C1-516U78A0.0 01.DNWGXMMW	DEV.MEDIAACCESSDEVICE		☐
SERVER-8204-E8A-SN6503724-C2-517U78A0.0 01.DNWGXMMW	DEV.MEDIAACCESSDEVICE		☐
SERVER-8204-E8A-SN6503724-C10-518U78A0.0 001.DNWGXMMW	DEV.MEDIAACCESSDEVICE		☐
SERVER-8204-E8A-SN6503724-C3-519U78A0.0	DEV.MEDIAACCESSDEVICE		☐

Figure 7-23 Actual configuration items application

7.4.4 Logging

If you want ITIC to log the classifications and depths used for importing, modify the `/opt/IBM/itic/data/properties/logging.properties` file and set `taddmactualci` to `INFO`:

```
log4j.logger.fusion.provider.taddmactualci=INFO
```

When Integration Composer executes mappings, it provides information about mapping executions and data transactions as well as errors in log files. You can view log files in the following location:

```
\opt\IBM\itic\log
```

7.5 References

You can find the *Integration Composer Reference Guide* at the following website:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.cmdb.doc/reference/72_int_comp_sag.pdf

CI promotion

In this chapter we discuss the process called promotion and explain how this process can help you create authorized CIs in the CCMDB database.

In previous chapters, we have described the processes related to discovering CIs using the TADDM application and the creation of actual CIs using Integration Composer.

Important: This chapter assumes that you have already imported both CI Types and actual CI data into the CCMDB application tables using the ITIC adapter.

We cover the following topics:

- ▶ 8.1, “Various states of CIs” on page 418
- ▶ 8.2, “Promoting actual CIs to authorized CIs using authorized CI hierarchies” on page 419
- ▶ 8.3, “Summary” on page 441

8.1 Various states of CIs

See Figure 8-1 to understand the various states of CIs in CCMDB.

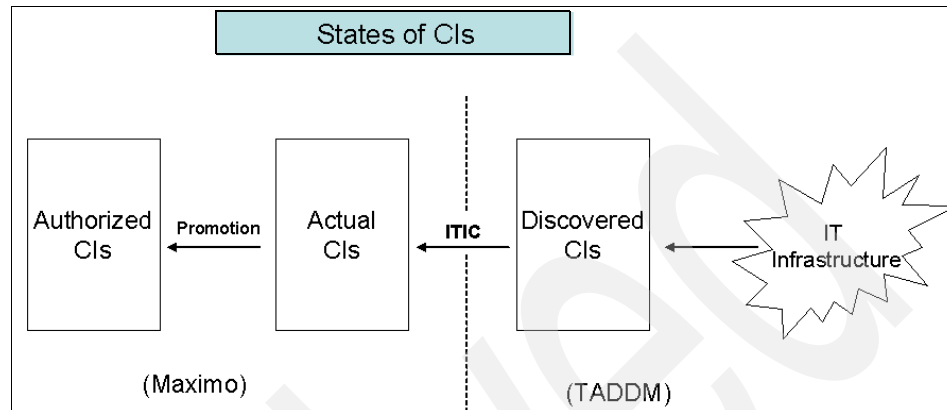


Figure 8-1 Configuration item states

The discovered CIs and actual CIs in your environment can have many attributes and relationships. In certain cases, you might not want to manage and control every attribute and relation for every CI that displays in the actual space, and you might also want to add new attributes and relations. To do so, you need to have a space where you can have your final authorized space containing authorized CIs.

An authorized configuration item (CI) is a configuration item (CI) 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, or by creating it manually without using an actual CI.

The version of the configuration item that you manage through the Change Management and Configuration Management processes is always the *authorized version*. The authorized CI is usually created from an actual CI by a process called promotion. After we have actual CIs in the database, they are then ready to be promoted to another state as authorized CIs. You can choose which child configuration items and which attributes are included when an actual CI is promoted to create an authorized CI.

Tip: Promotion helps you to bring over selected actual CIs into the authorized space. You can modify these CIs in the authorized space and can also add new CIs directly into the authorized space.

There are two ways to promote actual CIs to authorized CIs: using authorized hierarchies or not. These two options are described in the following sections.

8.2 Promoting actual CIs to authorized CIs using authorized CI hierarchies

Actual CIs are created in a hierarchy that follows the common data model. You can choose to import only certain types of CIs. You also can choose to limit the number of levels of child CIs that are imported. You can further limit the CIs and the attributes that you include as authorized CIs.

In order to control which CI types and attributes are included as authorized CIs, you must build a hierarchy of authorized CIs that mirrors the hierarchy of actual CIs, but contains only those CIs and attributes that you want to manage. Manage CI Hierarchies has been removed in 7.2.1 in favor of using the Deployer's Workbench. see 8.3, "Summary" on page 441 for creating promotion hierarchies.

You can think of the hierarchy of authorized CIs that you create as templates for your authorized CIs. You can have more than one authorized CI space for a particular type of CI. For example, for Computer System, you can have one hierarchy for your mail servers and another for database servers. When you promote one or more actual CIs of a particular type, you can choose one of these hierarchies, or templates, to be used in the promotion process. The authorized CIs created in this way will have only those attributes that are present both in the actual CI and in the authorized CI template that you chose.

You can define default values for attributes in your authorized CI space. Then any actual CI that is promoted and does not have a value for that attribute will have that value in the authorized Version. To define a default value, while you are defining a CI in the hierarchy, click the icon and enter the default value in the dialog box.

You might want your authorized CIs of a certain type to have one or more attributes that are not included in the actual CI. You can add these attributes to your authorized CI space. Because they do not have values in the actual CI, they will not have values in the authorized CI, unless you specify a default value when you create the template. You can add values for these extended attributes to the authorized CI record after it is created.

After you have created the hierarchies for the types of authorized CIs that you want to manage, you can create authorized CIs by promoting actual CIs. Open the actual CIs application and follow the help provided to promote actual CIs to authorized CIs.

The flowchart in Figure 8-2 illustrates how to promote the actual CIs using authorized hierarchies.

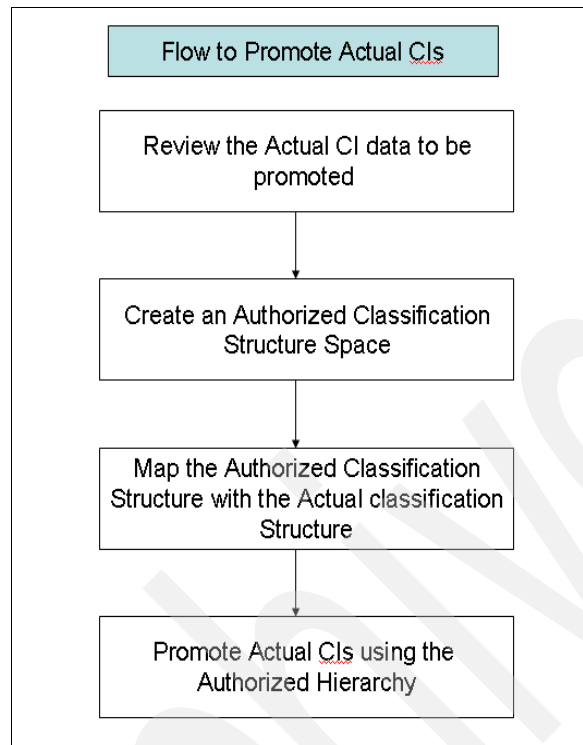


Figure 8-2 Flow to promote actual CIs to authorized CIs

8.2.1 Determining authorized CI classifications

CI promotion is the process of promoting actual CIs to authorized CIs. You only want to promote a subset of your actual CI data. You need to determine what data you want to promote, by looking at your Service Management process implementations. This includes looking at what you want to audit, control, and report.

Before you can promote CIs, you need to have authorized CI classifications defined for your authorized CI space. You first need to determine what CI classifications you want to include in your authorized CI space. This must be done in the data design phase of your deployment of CCMDb. The Data Design phase goes along with the process design phase.

Use these questions to help determine which actual CI classifications are to be included in your authorized CI space:

1. What data do your processes rely on? Here are a few examples:
 - a. If you are implementing Change processes, which CIs will be targeted in these processes?
 - b. Which CIs need to be under Change control?
 - c. Which CIs do you want to audit?
 - d. Are there additional CIs (for example, business applications) that are not needed as targets in processes, but are needed to facilitate the processes or manage them?
2. What discovery sensors and DLAs are used in TADDM? This will tell you more about what is included in your discovered CIs. Then you can look through this data to determine the subset that is needed for authorized CIs. You can look at what actual CI types were imported from TADDM. You can use this SQL command to get all CI classifications names that have at least one actual CI instance:

```
select distinct c.classificationid from actci a, classtructure c,  
where (a.CLASSSTRUCTUREID=c.CLASSSTRUCTUREID)
```

8.2.2 Determining authorized CI attributes

Each CI class in the CDM has a set of attributes. Many of these attributes are not needed in your authorized CI space and therefore, must not be included in your list of authorized CI classification attributes.

Follow these steps to determine what attributes you need in your authorized CI space:

1. For each CI classification you chose to include in your authorized CI space, get a list of all the attributes available by looking at the actual CI classification that it was duplicated from.
2. Remove any attributes that are not important in your discovered CI space. Many attributes are not even populated in TADDM, by discovery sensors or DLAs. Many that do get populated will never be used in TADDM, because the sensor or DLA populated them automatically without anyone specifying they need that particular attribute information. None of those attributes must be in your authorized CI space, because they are not important in the discovered CI space.

3. List out the remaining attributes and think about which you need in your CCMDB processes. Here is a list of questions to ask:
 - a. What attributes do you need in your processes? For example, what attributes do you want to put under Change control? Which attributes do you want to audit? Do you need to know when the actual CI attribute has changed in the discovered CI space, and compare it the authorized CI attribute?
 - b. Which attributes do you want to report on?
 - c. Which attributes are needed to supplement your CCMDB processes? For example, if you have a Change process that has an action in a workflow that has to run an operation on an OMP, what parameters are needed to call that operation? What is the IP address and port? These attributes will also be needed in your authorized CI space.
4. Remove attributes that are not needed to answer any of these questions.
5. Now you will have a list of attributes that must be created on your authorized CI classification. If you've duplicated the attributes from your actual CI classification, just delete any attributes that are not in your list.

8.2.3 Determining authorized CI relationships

You have to create relationships for your authorized CI classifications to be used in your authorized CI space. Note that relationships are relationship types and not instances.

For example, if you have decided that ComputerSystem and OperatingSystem will be in your authorized CI space, you most likely want a relationship between them. In the actual CI hierarchy, there is the relationship "OperatingSystem installedOn ComputerSystem." That is a relationship type (or just a relationship), because it is referring to CI Classifications and not actual CI instances.

To determine what relationships you want, first look at all the relationships that exist between the actual CI classifications. Then decide which relationships are important for your authorized CI space. Relationships are important for use cases that require data traversing.

Here are a few use cases to think about when determining which relationships are needed:

- ▶ CI Promotion
- ▶ Impact analysis in your Change process
- ▶ Full CI comparison (part of reconciliation, which is part of the CI audit process); full CI comparison starts with one CI and then uses the relationships to be able to compare all CIs and that are related to it and compare the relationships
- ▶ CI Advanced Attribute Search (Action “Attribute Search” under “Advanced Search” in the CI Application)
- ▶ Custom workflows needed for your process implementation

8.2.4 Putting authorized CI classifications into hierarchies

After you know which authorized CI classifications you want in your authorized space, you need to create hierarchy diagrams for them, so you will know the classification structure that will be used when you create them.

The easiest way to do this is described in a spreadsheet file that ships with this document (see appendix for the contents of *DesignCIHierarchyByExample.xls*). Consult that file for a step by-step walkthrough of the process of creating a separate authorized hierarchy.

8.2.5 Promoting CIs from actual CIs to authorized CIs

While it is possible to create authorized CIs that are not linked to actual CIs, most authorized CIs are created by copying an actual CI into a new authorized CI, and then linking the authorized CI back to the actual CI. We call this copying process promotion.

The action of promoting is called “Create authorized Configuration Item” in the actual CI application.

You start a promotion from an instance of a top-level actual CI. The promotion creates a new authorized CI and (optionally) copies the values of the attributes you selected in the previous phase into the authorized CI. The promotion process also follows the selected relationships to create authorized CIs for the CIs that the top-level CI is related to.

8.2.6 Promotion rules

When you promote CIs, the following rules apply. In addition, only containment relationships will be promoted. (The “containment” flag must be set on those relationships.)

Starting from an actual CI, the CI promotion follows relationships according to these rules:

- ▶ You must start promotion from an actual CI whose classification is has its “top-level” flag set.
- ▶ Each relationship must be defined in the relationship rules table.
- ▶ The authorized CI classification must exist, and the classification must have a “Use With CI” record.
- ▶ You must have a mapping between the authorized CI classification and the actual CI classification for both the source and target CIs (when using separate authorized CI and actual CI classifications). This mapping is defined in an authorized CI template using the “Manage CI Hierarchies” dialog.
- ▶ One of the following conditions is true:
 - The current CI is the source of a relationship, and the target CI's classification is assigned to an authorized CI (“Use With CI”), OR
 - The relationship's source CI classification is assigned to an authorized CI, the current CI is the target, and the “Is Target Parent” flag on the authorized relationship is true.
 - If the cardinality of the authorized relationship says that there can only be one CI as the source or target, you have not already promoted another CI using that relationship. (Beginning with CCMDb v7.1.1.2, this check is skipped because we assume it was already done when the relationship instance was created).

If you want to include more relationships or CIs in your promotion, you have the following options:

- ▶ Flag additional actual CI classifications as top-level.
- ▶ Change the logical direction of certain relationships by changing the “Is Target Parent” flag.
- ▶ Flag additional relationships as containment relationships.

If you check the “Copy Attributes” checkbox, the promotion will copy the attribute values from your actual CIs to your authorized CIs. Otherwise, it will use the default values assigned to the attributes in your authorized CI classifications.

8.2.7 Best practices for using business applications

Business applications are a special CI type. A business application is a collection of components that are typically deployed and assigned a version number as a unit. You can manually create business applications in the TADDM GUI, or you can use application descriptors to assign discovered components to business applications dynamically.

In most cases, a business application contains a Functional Group and the Functional Group contains the components that are treated as the members of the application (and the Functional Group).

The CI Type for business application is APP.APPLICATION. There are four CI types for Functional Groups in CCMDB 7.2.1:

- ▶ APP.COMPUTERSYSTEMFUNCTIONALGROUP
- ▶ APP.FUNCTIONALGROUP
- ▶ APP.APPSERVERFUNCTIONALGROUP
- ▶ APP.SOFTWAREMODULEFUNCTIONALGROUP

The Business Application CI Type is special in CCMDB because it is not an IT asset that you manage. It groups IT assets together. Therefore it is also special in how it exists in the data hierarchy. It is not a child to any other CI type. It can be imagined as another hierarchy to the side of the main hierarchy. Its members are generally components that are children to other top level CI types, so it is not above the top level CI types either. Therefore, if you want to import Business Applications into Tivoli's process automation engine, you always have to activate those CI Types.

8.2.8 Step by step procedure to promote CIs

Take the following steps to promote an actual CI to an authorized CI.

Step 1: Review the actual CI data classifications to be promoted

Proceed as follows:

1. At this point, determine the amount of details that you want to manage in the authorized space. There are two major decisions to be made: depth and width. We need to determine what classifications will be in the defined authorized space that can hold authorized CIs. We advise starting the investigation at the top-level classification and working downward.
2. Select a top-level class as we have done in Figure 8-3. We selected SYS.COMPUTERSYSTEM. Notice that this class has 14 children in the actual space. For example, we have chosen two children classifications that we can bring over to our authorized space.

The screenshot shows the 'Classifications' interface in the IBM Tivoli Change and Configuration Management Database (ITCM). The interface is divided into several sections:

- Classification Details:** Shows the selected classification 'SYS.COMPUTERSYSTEM' and its path 'TOPICCLASS \ SYS.COMPUTERSYSTEM'. It also displays the parent classification 'TOPICCLASS' and a checkbox for 'Generate Description?'. On the right, there are fields for 'Organization', 'Site', and a checkbox for 'Use Classification?'.
- Use With:** A section with a 'Filter' button and a list of objects. The 'ACTCI' object is selected, and the 'Top Level?' checkbox is checked.
- Children:** A table listing the children of the selected classification. The table has columns for 'Classification', 'Description', 'Generate Description?', 'Use Classification?', 'Organization', and 'Site'. The children listed are: 'NET.IPINTERFACE', 'SYS.FILESYSYSTEM', 'PHYS.PHYSPKG.PHYSICALPACKAGE', 'SYS.OPERATINGSYSTEM', and 'DEV.PROTOCOLCONTROLLER'. The 'NET.IPINTERFACE' and 'SYS.OPERATINGSYSTEM' rows are highlighted.
- Attributes:** A table listing the attributes of the selected classification. The table has columns for 'Attribute', 'Description', 'Section', 'Domain', 'Data Type', and 'Unit of Measure'. The attributes listed are: 'COMPUTERSYSTEM_ARCHITECTURE', 'COMPUTERSYSTEM_ASSETID', 'COMPUTERSYSTEM_ASSETTAG', 'COMPUTERSYSTEM_AUTOSTART', and 'COMPUTERSYSTEM_CDPDEVICEID'. The 'COMPUTERSYSTEM_ARCHITECTURE' and 'COMPUTERSYSTEM_ASSETID' rows are highlighted.

Figure 8-3 Choosing classifications for authorized space

3. Look at the child classes. NET.IPINTERFACE does not have any children, so this leg of the hierarchy is complete (Figure 8-4).

The screenshot shows the 'Classifications' application window. The 'Classifications' tab is active. The 'Classification' field is set to 'NET.IPINTERFACE'. The 'Classification Path' is 'TOPCLASS \ SYS.COMPUTERSYSTEM \ NET'. The 'Parent Classification' is 'TOPCLASS \ SYS.COMPUTERSYSTEM'. The 'Generate Description?' checkbox is unchecked. The 'Use With' section shows a table with one row: 'ACTCI' with the description 'Use with Actual Cts'. The 'Children' section shows a table with the header 'Classification', 'Description', 'Generate Description?', 'Use Classification?', and 'Organization', but it contains the text '...No rows to display...'. The 'Attributes' section shows a table with the header 'Attribute', 'Description', 'Section', 'Domain', and 'Data Type'. The attributes listed are 'IPINTERFACE_MANAGEDSYSTEMNAME', 'IPINTERFACE_PORTLIST', 'IPINTERFACE_STATUS', 'MODELOBJECT_ADMINSTATE', and 'MODELOBJECT_BIDIFLAG'.

Attribute	Description	Section	Domain	Data Type
IPINTERFACE_MANAGEDSYSTEMNAME				ALN
IPINTERFACE_PORTLIST				ALN
IPINTERFACE_STATUS				NUMERIC
MODELOBJECT_ADMINSTATE				NUMERIC
MODELOBJECT_BIDIFLAG				NUMERIC

Figure 8-4 Viewing child classes

4. We look at another child class of SYS.COMPUTERSYSTEM, SYS.OPERATINGSYSTEM (Figure 8-5). This class has three children. Let us say that we want to manage only SYS.SOFTWARECOMPONENT in our authorized space (Figure 8-5).

The screenshot shows the 'Classifications' window in the IBM Tivoli CCM interface. The main area displays the 'SYS.OPERATINGSYSTEM' class, which is a child of 'TOPICCLASS \SYS.COMPUTERSYSTEM'. The 'Use With' section shows two entries: 'ACTCI' (Use with Actual CIs) and 'CI' (Use with Configuration Items). The 'Children' section lists three child classes: 'SYS.DNSRESOLVEENTRY', 'SYS.SOFTWARECOMPONENT', and 'APP.SOFTWAREINSTALLATION'. The 'Attributes' section lists five attributes: 'MODELOBJECT_ADMINSTATE', 'MODELOBJECT_BIDFLAG', 'MODELOBJECT_BIDFORMAT', 'MODELOBJECT_CDMSOURCE', and 'MODELOBJECT_CONTEXTIP'.

Classification	Description	Generate Description?	Use Classification?	Organization
SYS.DNSRESOLVEENTRY	SYS.DNSRESOLVEENTRY	☐	☐	
SYS.SOFTWARECOMPONENT	SYS.SOFTWARECOMPONENT	☐	☐	
APP.SOFTWAREINSTALLATION	APP.SOFTWAREINSTALLATION	☐	☐	

Attribute	Description	Section	Domain	Data Type
MODELOBJECT_ADMINSTATE				NUMERIC
MODELOBJECT_BIDFLAG				NUMERIC
MODELOBJECT_BIDFORMAT				ALN
MODELOBJECT_CDMSOURCE				ALN
MODELOBJECT_CONTEXTIP				ALN

Figure 8-5 SYS.OPERATINGSYSTEM class

5. If we look further at the SYS.SOFTWARECOMPONENT class, we find that there is no child class defined under this class, so this leg of the hierarchy is complete (Figure 8-6).

The screenshot shows the 'Classifications' application interface. At the top, there's a navigation bar with 'Go To', 'Reports', 'Start Center', and 'Profile'. Below it is a search bar with 'Find:' and a 'Select Action' button. The main content area has a 'List' tab and a 'Classifications' tab. Under 'Classifications', there's a form for editing a classification. The 'Classification' field is set to 'SYS.SOFTWARECOMPONENT'. The 'Classification Path' is 'TOPICCLASS \ SYS.COMPUTERSYSTEM \ SYS'. The 'Parent Classification' is 'TOPICCLASS \ SYS.COMPUTERSYSTEM \ SYS'. The 'Generate Description?' checkbox is unchecked. To the right, there are fields for 'Organization', 'Site', and 'Use Classification?'. Below this is a 'Use With' section with a 'Filter' button and a table. The table has columns 'Use With Object', 'Description', and 'Top Level?'. There's one row with 'ACTCI' in the 'Use With Object' column and 'Use with Actual CIs' in the 'Description' column. Below that is a 'Children' section with a 'Filter' button and a table. The table has columns 'Classification', 'Description', 'Generate Description?', 'Use Classification?', and 'Organization'. It shows '...No rows to display...'. At the bottom is an 'Attributes' section with a 'Filter' button and a table. The table has columns 'Attribute', 'Description', 'Section', 'Domain', 'Data Type', and 'Unit'. It shows five rows: 'MODELOBJECT_ADMINSTATE' (NUMERIC), 'MODELOBJECT_BIDFLAG' (NUMERIC), 'MODELOBJECT_BIDFORMAT' (ALN), 'MODELOBJECT_CDMSOURCE' (ALN), and 'MODELOBJECT_CONTEXTIP' (ALN).

Figure 8-6 SYS.SOFTWARECOMPONENT class

Tip: In real scenarios, gather all the information related to classifications according to your needs. A real authorized structure might not look like the foregoing examples.

Figure 8-7 shows the structure of our final authorized space.

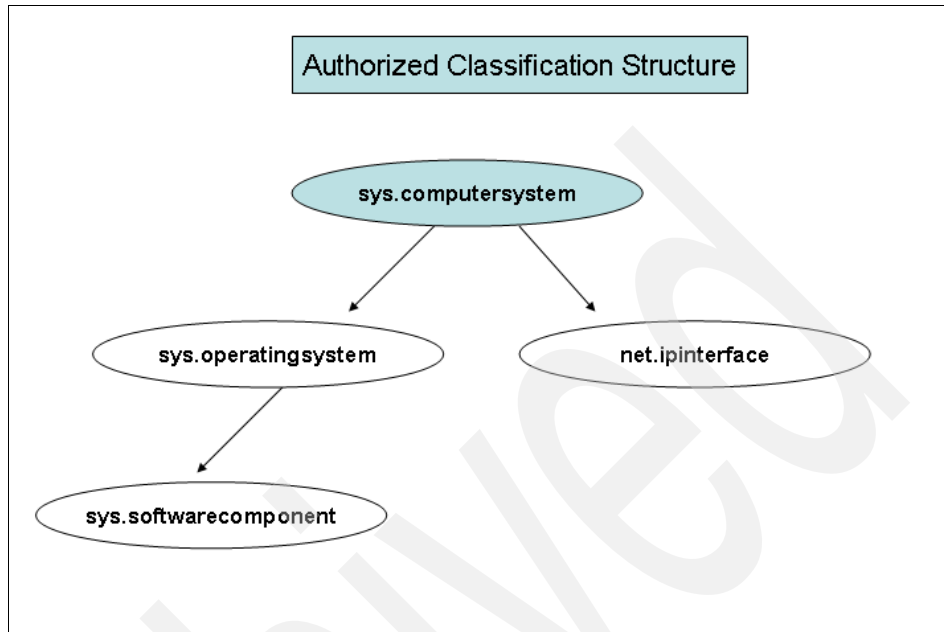


Figure 8-7 Sample authorized classification structure

This structure might differ in real scenarios, depending upon your needs. In Figure 8-7, we have only one top-level class (sys.computersystem), but you might have multiple top-level CI classifications.

Step 2: Create an authorized classification structure space

You have already created TOPCICLASS during ITIC configuration. Now you need to have a companion classification (authorized), similar to TOPCICLASS, for example, AUTH.TOPCICLASS. You need to create this manually. Similarly, create other children authorized classes and also decide on a naming convention for these authorized classifications. There is no restriction on the naming convention. Here are a few examples:

- ▶ Actual: TOPCICLASS \ SYS.COMPUTERSYSTEM \ SYS.OPERATINGSYSTEM
- ▶ Authorized: AUTH.TOPCICLASS \ SYS.COMPUTERSYSTEM \ SYS.OPERATINGSYSTEM

To create a classification, select **Administration** → **Classification**.

Attention: You have to prepare these authorized classes manually. Make sure that every classification in the AUTHORIZED class structure has a “Use with CI” record.

Step 3: Map the entire authorized classification structure to the actual class structure

After you have created an authorized classification structure, you can think of it as a template for actual CIs. This means you can promote only those CIs that fit into your authorized classification structure. It is important to understand that you can only promote top-level CIs using the actual CI application. The related CIs to that actual CI will get promoted automatically, provided that they belong to the authorized classification structure. Anything outside the authorized space will not get promoted.

Important: In previous releases of CCMDB, you used the Manage CI Hierarchies dialog, part of the Classifications application, to configure a set of classifications for authorized CIs. These classifications were used as templates to promote actual CIs and create authorized CIs.

In CCMDB 7.2.1, by default, access to this dialog is disabled because the best practice for defining an authorized CI space is the Deployer's Workbench, as described in 6.2.3, "Defining a promotion scope" on page 372.

In CCMDB 7.2.1, if you want to use the Manage CI Hierarchies dialog, you need to issue the following command against your database:

```
update maxvars set varvalue='1' where varname='SHOWMANAGECIHIERAR'
```

If you have authorized CI classifications that you created using the Manage CI Hierarchies function, you can continue to use them. You must decide how you will continue to maintain and update your CI space. You can choose either of these options.

The following example assumes that you want to use the old method. If you prefer to use the Deployer's Workbench, see 6.2.3, "Defining a promotion scope" on page 372.

For more information about this topic, see the CCMDB InfoCenter at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v34r1/index.jsp?topic=/com.ibm.ccmdb.doc/config/c_ci_mch.html

Proceed as follows:

1. First, query your top level authorized classifications using the classification application. In our example, we have only one top-level classification, AUTH.SYS.COMPUTERSYSTEM. It is shown in Figure 8-8. In order to map the authorized classification to the actual one, you have to select the **Manage CI Hierarchies** from the Action menu.

Classifications

Find: Select Action

Classifications

Classification: AUTH.SYS.COMPUTERSYSTEM
 Classification Path: AUTH.TOPICCLASS \ AUTH.SYS.
 Parent Classification: AUTH.TOPICCLASS
 Generate Description? ☐

Space for Sys.ComputerSystem
 d Space for Sys.ComputerSystem

Organization:
 Site:
 Use Classification? ☐

Use With: Filter 1 - 1 of 1

Use With Object	Description	Top L
CI	Use with Configuration Items	☒

Children: Filter 1 - 2 of 2

Classification	Description	Generate Description?	Use Classification?	Organization
AUTH.NET.IPINTERFACE	Authorized NET.IPINTERFACE	☐	☐	
AUTH.SYS.OPERATINGSYSTEM	Authorized SYS.OPERATING SYSTEM	☐	☐	

Attributes: Filter

Attribute	Description	Section	Domain	Data Type	Unit of Measure
...No rows to display...					

Figure 8-8 Classifications

- After selecting **Manage CI Hierarchies**, you will get the window shown in Figure 8-9.

Manage CI Hierarchies

CI Classification: AUTH.TOPICCLASS \ AUTH.SYS.COMPUTERSYSTEM
 Actual CI Classification:
 Current Level: AUTH.TOPICCLASS \ AUTH.SYS.COMPUTERSYSTEM
 Belongs To:

Authorized Space for Sys.ComputerSystem
 Authorized Space for Sys.ComputerSystem

Children: Filter 1 - 2 of 2

CI Classification	Description	Actual CI Classification	Description
AUTH.TOPICCLASS \ AUTH.SYS.COMPUTERSYSTEM	Authorized SYS.OPERATING SYSTEM		
AUTH.TOPICCLASS \ AUTH.SYS.COMPUTERSYSTEM	Authorized NET.IPINTERFACE		

Relationships: Filter

Source Classification	Relationship	Target Classification
...No rows to display...		

Select Relation Rules

Figure 8-9 Manage classifications

- Here you can map the authorized classification to the respective actual classification. First, map the top-level classification and then the children classifications. In our example, the top-level authorized classification AUTH.TOPICCLASS \ AUTH.SYS.COMPUTERSYSTEM must be mapped to TOPICCLASS.SYS.COMPUTERSYSTEM (actual CI classification).

The children classification must also be mapped to their respective actual classes., as shown in Figure 8-10.

The screenshot shows the 'Manage CI Hierarchies' window. It has several sections:

- Top Section:** Fields for 'CI Classification' (AUTH.TOPICCLASS \ AUTH.SYS.COMPUTERS), 'Actual CI Classification' (TOPICCLASS \ SYS.COMPUTERSYSTEM), 'Current Level' (AUTH.TOPICCLASS \ AUTH.SYS.COMPUTERS), and 'Belongs To'. To the right are two buttons labeled 'Authorized Space for Sys.ComputerSystem'.
- Children Section:** A table with columns 'CI Classification', 'Description', 'Actual CI Classification', and 'Description'. It shows two rows:

CI Classification	Description	Actual CI Classification	Description
AUTH.TOPICCLASS \ AUTH.SYS.COMPUTERS	Authorized SYS.OPERATING SYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM \ SYS	
AUTH.TOPICCLASS \ AUTH.SYS.COMPUTERS	Authorized NET.IPINTERFACE	TOPICCLASS \ SYS.COMPUTERSYSTEM \ NET.	
- Relationships Section:** A table with columns 'Source Classification', 'Relationship', and 'Target Classification'. It currently shows 'No rows to display...'. Below this table are buttons for 'Select Relation Rules' and 'New'.

Figure 8-10 Manage CI hierarchies

- After mapping the top-level Classification and child classes, also map the relationships between these classifications. You can use the section named "Relationships" on the same window. If you click the **Select Relations Rules** button, then it can automatically establish the relations based on the relations between actual classifications. But you can also modify/create your own relationships that you want to manage in your authorized space. To create new relations, click the **New Row** button under the Relationships section.

Figure 8-11 shows the relationships that can be managed in an authorized space.

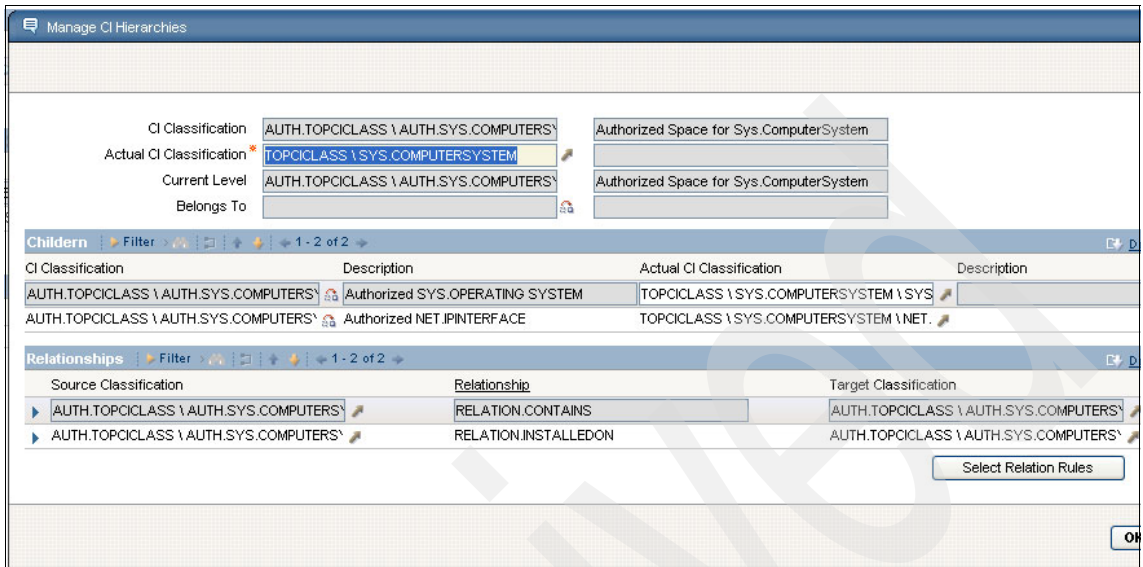


Figure 8-11 Relationships managed in an authorized space

5. Repeat the foregoing steps until the entire authorized hierarchy is defined.

Step 4: Promoting actual CIs

There are two ways to promote the actual CIs

- ▶ Promoting one CI at a time
- ▶ Promoting more than one CI at time

Promoting one CI at a time

Attention: It is important to understand that you can only promote a CI with a top-level classification, so the first step is to query all CIs with your identified top-level.

In the prior example, our top-level actual class is SYS.COMPUTERSYSTEM. We can promote any CI defined with this actual class, as follows:

1. You can query all CIs under the actual configuration items application and pick the desired actual CI. For example, we queried actual CIs for class = SYS.COMPUTESYSTEM beginning with the letter G%. Refer to the example shown in Figure 8-12 on page 436.

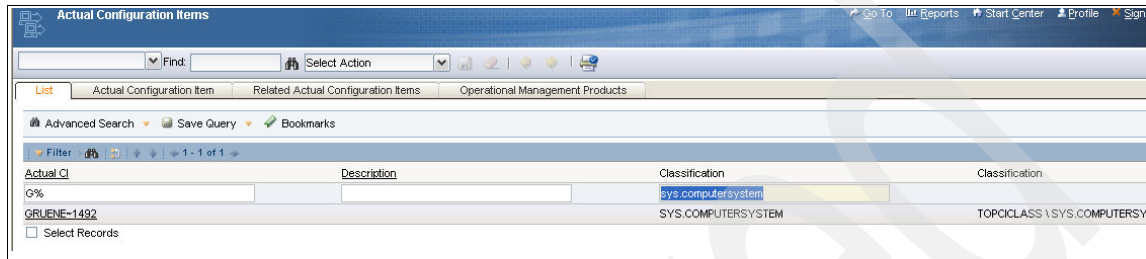


Figure 8-12 Actual CI

2. Click the **Actual Configuration Item** tab. This view shows the details about the respective actual CI, as shown in Figure 8-13.

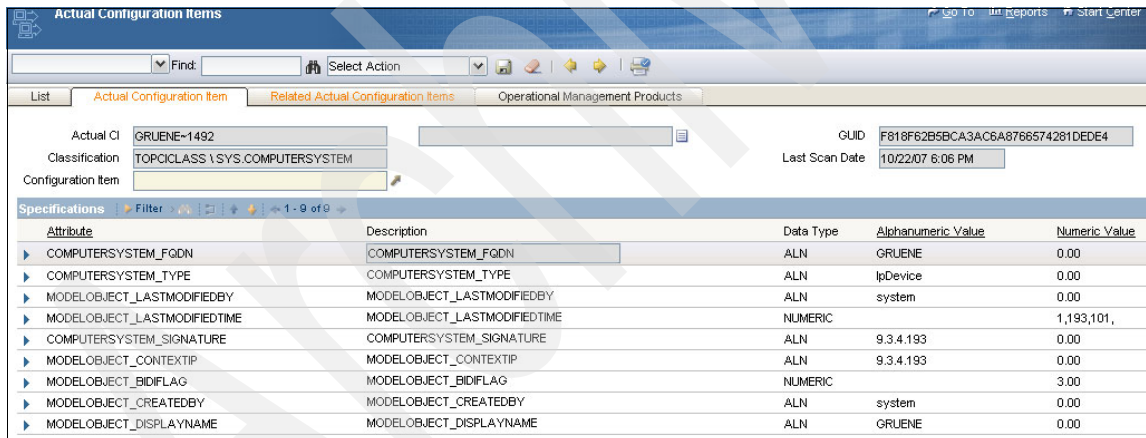


Figure 8-13 Details of actual CI

3. Choose **Select Action** → **Create Authorized Configuration Item** to promote the CI, as shown in Figure 8-14.

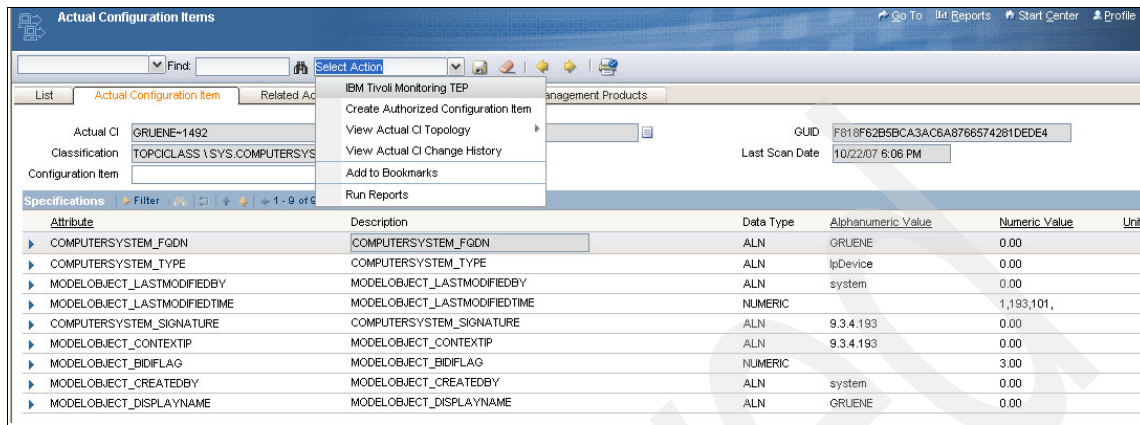


Figure 8-14 Selecting action to promote

4. This shows a dialog box, as shown in Figure 8-15. In this case, as you are promoting CIs with the authorized classification, the CI class must be the same as the defined authorized structure.

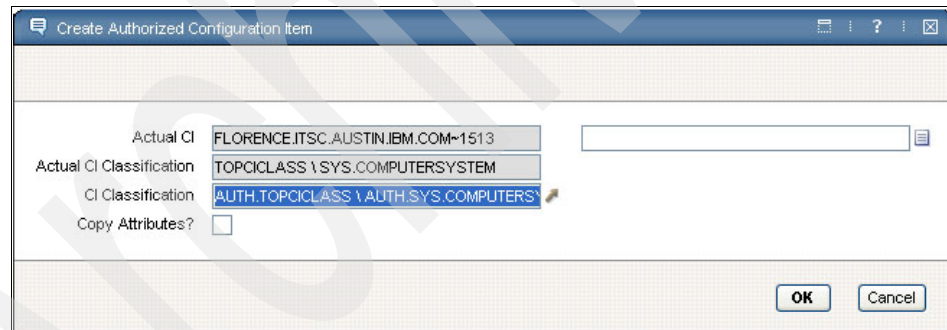


Figure 8-15 Promotion details

Tip: If you want to promote an actual CI with all its attributes values, then check the **Copy Attributes** check box. This brings all attributes associated with the CI to the authorized space. These attributes values can be modified or deleted, and new attributes can also be added.

5. Click **OK** to start the promotion. This might take a while. After a successful promotion, the actual CI can be seen in the authorized CIs space. To view the promoted CI, go to the **Configuration Item Application** and query the CI.

Figure 8-16 shows the Promoted CI in the configuration items application.

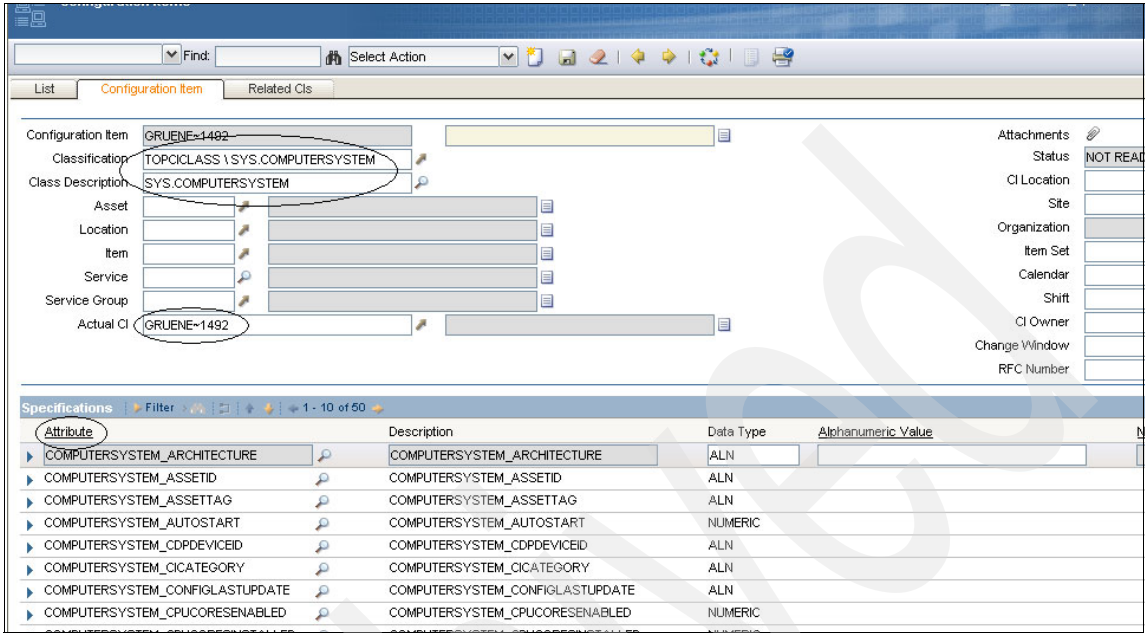


Figure 8-16 Viewing the promoted CI

You can review the desired results with classification details, associated actual CIs, and attributes.

- To check the related CIs and the relations, select the **Related CIs** tab. This shows that all related CIs and relations have been promoted successfully, as shown in Figure 8-17.

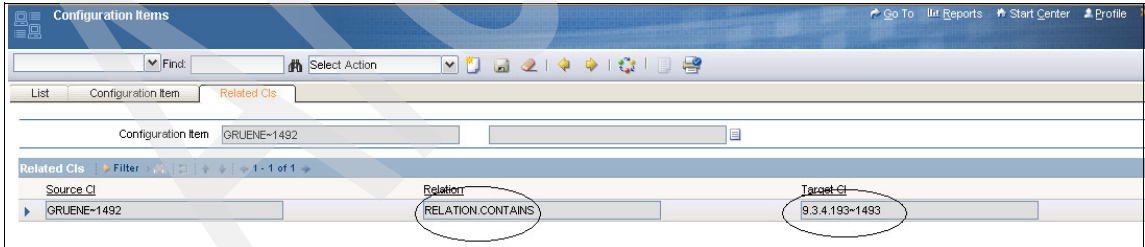


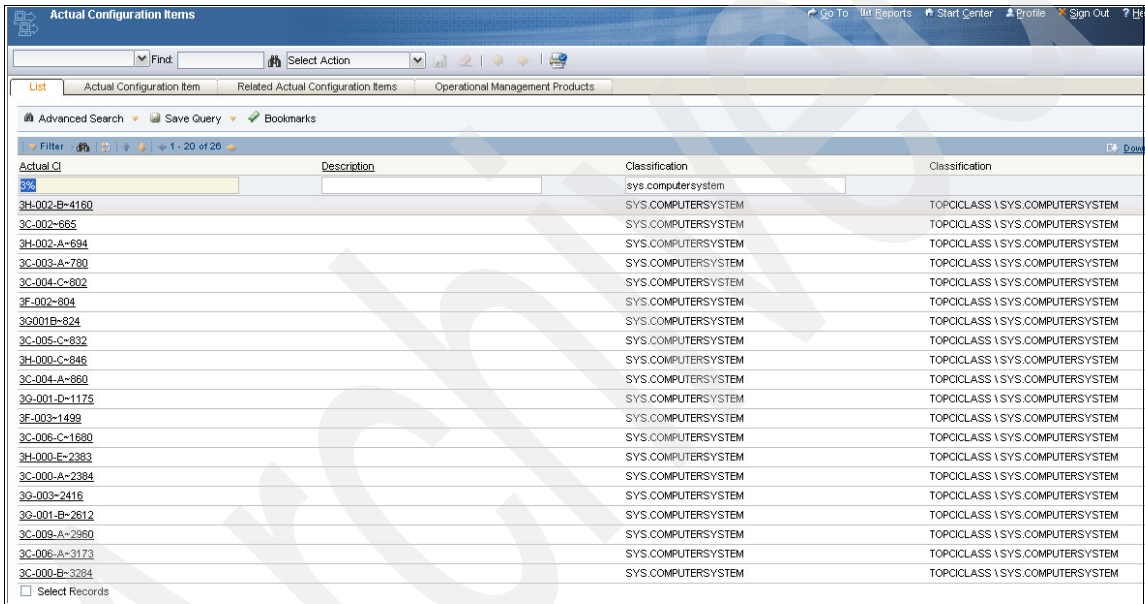
Figure 8-17 Viewing Related CIs

Promoting more than one CI at time

Attention: It is important to understand that you can only promote CIs with a top-level classification, so the first step is to query all CIs with your identified top-level classification.

To promote more than one CI at a time, perform the following steps:

1. Go to the **Actual Configuration Application**, and query the desired actual CIs using filter criteria. For example, we have queried class=SYS.COMPUTERSYSTEM and the actual CIs that begin with “3”, as shown in Figure 8-18.



The screenshot shows the 'Actual Configuration Items' application interface. At the top, there is a navigation bar with links like 'Go To', 'My Reports', 'Start Center', 'Profile', and 'Sign Out'. Below this is a search bar with a 'Find' field and a 'Select Action' dropdown. The main content area has tabs for 'List', 'Actual Configuration Item', 'Related Actual Configuration Items', and 'Operational Management Products'. The 'List' tab is active, showing a table of configuration items. The table has columns for 'Actual CI', 'Description', 'Classification', and 'Classification'. The 'Actual CI' column contains various IDs starting with '3H', '3C', '3F', '3G', '3H', '3C', '3H', '3C', '3H', '3C', '3H', '3C', '3H', '3C', '3H', '3C'. The 'Description' column is empty. The 'Classification' column contains 'SYS.COMPUTERSYSTEM' for all items. The 'Classification' column contains 'TOPICCLASS \ SYS.COMPUTERSYSTEM' for all items. At the bottom left, there is a checkbox labeled 'Select Records'.

Actual CI	Description	Classification	Classification
3H-002-B-4160		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3C-002-665		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3H-002-A-694		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3C-003-A-780		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3C-004-C-802		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3F-002-804		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3G001B-824		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3C-005-C-632		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3H-000-C-846		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3C-004-A-860		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3C-001-D-1175		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3F-003-1499		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3C-006-C-1680		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3H-000-E-2383		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3C-000-A-2384		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3C-003-2416		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3G-001-B-2612		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3C-009-A-2960		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3C-006-A-3173		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM
3C-000-B-3284		SYS.COMPUTERSYSTEM	TOPICCLASS \ SYS.COMPUTERSYSTEM

Figure 8-18 Results of CI query

- For mass promotion of CIs, you must enable the selected check boxes. To do so, check the **Select Records** check box at the bottom of the list, and select the **Create Authorized Configuration Items** action, as shown in Figure 8-19.

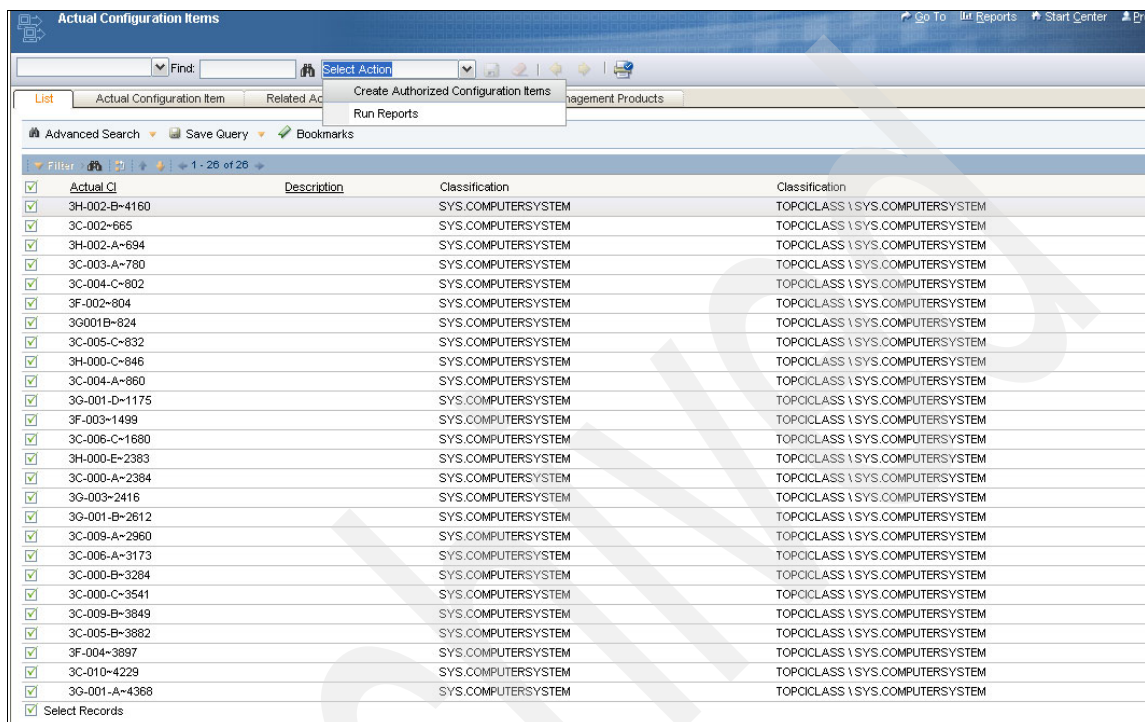


Figure 8-19 Create authorized CIs menu

3. After selecting the **Authorized Configuration Items** option, you see a dialog box as shown in Figure 8-20. Specify the authorized CI classification name. The Copy Attribute option is optional, as explained previously.

Create Authorized Configuration Items

Use this dialog to create Authorized Configuration Items from the selected Actual Configuration Item(s) and all related Actual C selected Configuration Item hierarchy. If an Authorized Configuration Item already exists, any new related Configuration Items of the Actual Configuration Items to the new Authorized Configuration Items, check the Copy Attributes box. If this box is not c be created with the default attribute values for their classification. If you want to copy the attributes of the Actual Configuration check the Update Existing Configuration Items box.

Configuration Item Classification RB.WINDOWSCOMPUTERSYSTEM

Copy Attributes? ☒

Update Existing CIs? ☐

Creation Runtime Options

Run in Background Mode? ☐ E-Mail Address Notification

Figure 8-20 Create authorized CI dialog

4. Click **OK** to start the promotion process. This might take a while.

The results can be reviewed in the configuration items application, as explained previously.

8.3 Summary

This chapter has provided a brief overview of the process for promoting actual CIs to authorized CIs. This is an integral process for populating the database used by the Change and Configuration Management processes. See Chapter 6, “CI space development” on page 359 for more information about Deployer’s Workbench and how to improve your CI space definition.

Implementing federation

In this chapter we discuss federation, which is a database layer integration. To illustrate the technology, we created a specific use case in order to show how to display external data from other data sources on the CCMDB applications.

We cover the following topics:

- ▶ 9.1, "Introduction" on page 444
- ▶ 9.2, "Federation scenario" on page 446
- ▶ 9.3, "Setting up federation at the database layer" on page 449
- ▶ 9.4, "Creating a Maximo Business Object (MBO)" on page 470
- ▶ 9.5, "Generating the object in the CCMDB database" on page 474
- ▶ 9.6, "Defining a relationship" on page 478
- ▶ 9.7, "Creating a new application" on page 483
- ▶ 9.8, "Using the new application" on page 490
- ▶ 9.9, "Summary" on page 492

9.1 Introduction

In this scenario, we extended the actual configuration items application in order to enrich the screen of this application with additional data from the federated data source relevant to the CIs. It is important to note that after you have exposed the external data to the CCMDB process runtime environment, you can use the data with any application of the platform.

There are slight differences in how you have to set up the federation depending on your CCMDB runtime environment and the target data source that you want to federate. The following questions are relevant to guide you to what you need to do in order to set up the federation scenario:

- ▶ Which database platform (DB2, Oracle, or SQL Server) did you use to implement the CCMDB process runtime database?
- ▶ If using DB2, what is the target platform that you want to integrate to?

If your CCMDB process runtime database (also referred to as the Maximo database) is DB2 and your target data source is either DB2 or IBM Informix®, then you do not have to install anything in addition to what was installed with CCMDB. DB2 is capable of federating other DB2 and Informix data sources without any add-ons.

If your CCMDB process runtime database is DB2 and your target data source *is not* DB2 or Informix, then you have to install the *WebSphere Federation Server V9.1* component on top of your existing DB2 implementation. WebSphere Federation Server V9.1 is part of the CCMDB V7.1.2 package and is actually an extension to DB2 specifically for federation purposes.

Attention: Sometimes WebSphere Federation Server is referred to as DB2 Information Integrator or WebSphere Information Integrator. Though there have been changes in the product's name, the technology referred to is the same. The current version of the product is called IBM InfoSphere™ Federated Server 9.7, but it is not supported yet on this CCMDB release.

9.1.1 Scope

For the purpose of this book, we are explaining in detail what you need to do if you want to federate from DB2 to another DB2 data source. We *do not* show how to install WebSphere Federation Server V9.1 or another vendor technology. For more details, see “Vendor related technology” on page 445.

After describing the scenario, we guide you through the configuration steps that you have to take in the database layer and in the process environment in order to enrich your CCMDB applications with external federated data.

9.1.2 Vendor related technology

Besides DB2 database, the CCMDB 7.2.1 release supports Oracle and Microsoft SQL Server databases as of the previous release, but IBM does not provide them in the CCMDB 7.2.1 package, and does not provide or support the solutions to federate from these databases.

If your CCMDB process runtime database is Oracle, you are required to have a federation solution provided by Oracle. The solution is called *Oracle Database Gateways* and works pretty much like the *IBM WebSphere Federated Server* solution provided in the package. The Oracle solution allows you to access non-Oracle systems from an Oracle based (CCMDB) environment. The base Oracle environment provides the ability to federate other Oracle environments.

For the Microsoft SQL Server customers, at the time of this publication, there are no solutions that can be used to fulfill our business scenario.

9.2 Federation scenario

In this section, we describe the overall setup of the scenario that we use for our federation example and the steps that need to be taken.

Figure 9-1 gives you an overview of the scenario architecture that we are using in our lab environment.

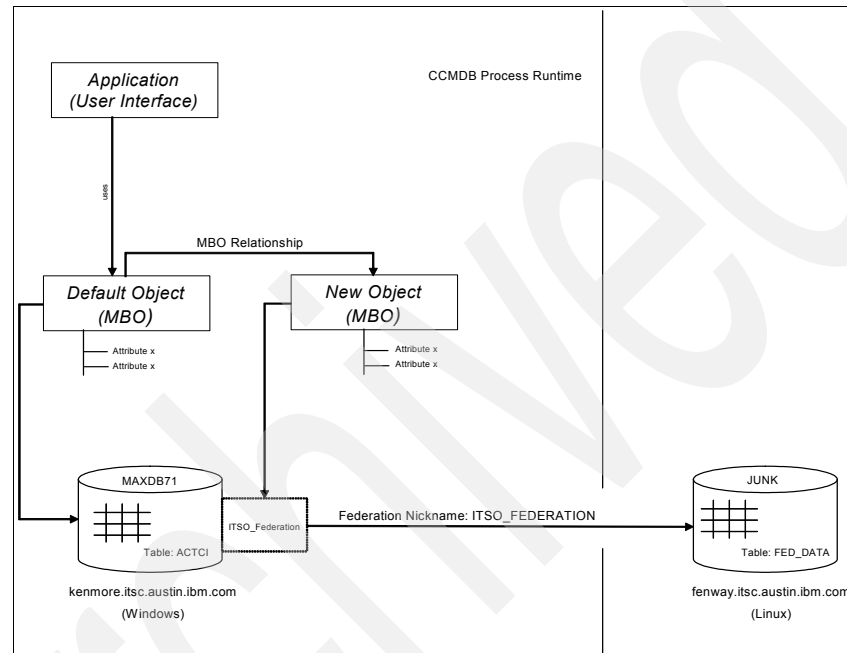


Figure 9-1 Lab environment for federation

Our CCMDDB process runtime database (MAXDB71) is hosted on kenmore.itsc.ibm.com. This is a Windows-based system and is referred to as the federation server. The MAXDB71 database is keeping all the database tables that are used in the CCMDDB environment, including the ACTCI table, that is the main table used by the actual CI application.

We created another DB2 database, JUNK, on a Linux system with the host name of fenway.itsc.austin.ibm.com. The specific table that we created for our federation example inside the JUNK database is FED_DATA.

In order to utilize data inside the FED_DATA table, a couple of steps that we describe in detail in this chapter have to be taken on the pure database layer. They need to be taken in order to make the FED_DATA table appear to the MAXDB71 database as through it is a local database. You can regard this as a kind of virtualization scenario.

After you set up the federation at the database layer and have the federated database appear as though it is local to the kenmore system, you can utilize the federated data source to define a new Maximo Business Object (MBO).

MBOs: We do not explain the basic concepts of Maximo Business Objects (MBOs) in this chapter. Instead, see the product documentation. An MBO is a Java object with business logic that encapsulates a database table in the CCMDB process database.

The new Maximo Business Object that you define imports the definitions of the FED_DATA table.

In order to make use of the federated data that is encapsulated by the new MBO, a relationship between the new MBO and the primary MBO that is used for the application that you want to extend has to be defined. In our example, we have to define a relationship between the MBO for the actual CI application and the MBO that represents our FED_DATA table.

If this relationship is defined, an existing application can be enhanced or duplicated and then modified. We took the second option in our example. Modifying or extending an application requires you to use the *Application Designer* tooling within the CCMDB environment to modify the applications to your needs. In our example, we duplicated the actual CI application and added some additional fields to represent attributes of the federated data source.

To summarize, in order to set up a federation, you have to go through configuration steps at the pure database layer, the CCMDB runtime database, and the application layout.

The following listing highlights the steps we use to set up our federation scenario. This excludes the step of actually creating the remote database, because we assume that the remote data source already exists in your environment. We describe each step in more detail in the upcoming sections of this chapter:

1. Set Federation at the DB2 Database Layer:
 - a. Catalog the remote node (fenway) and database (JUNK).
 - b. Create a wrapper in order to communicate with the remote DB2 data source.
 - c. Register the remote server as a federated data source.
 - d. Create a user mapping between the user of the federated server and the user of the remote data source in order to transparently get access without having the necessity of authenticating manually.
 - e. Create a nickname for the remote data source. A nickname is a local name for a remote database table.
2. Create a new Maximo Business Object (MBO) for the newly created remote data source you have defined in step 1.
3. Generate the object (MBO) in the CCMDB database by running the **configdb** command.
4. Define a relationship between an existing MBO that is the primary object for an existing application and the new MBO that you created in steps 2 and 3. We use the ACTCI object that is the primary MBO for the actual CI application.
5. Duplicate the actual CI application and modify it in order to present attributes that point to your federated data. Each attribute is pointing to a column of the federated data source.
6. Use the application you created in step 6 and modify the data in the remote data source in order to check if the application picks up the remote data dynamically.

9.3 Setting up federation at the database layer

In this section, we guide you through the steps to set up the federation between our database on the federation server (MAXDB71) and the database on the remote system (JUNK). To be more precise, we set up the federation to the table FED_DATA inside the database JUNK. As mentioned, both databases are inside a DB2 system, so we do not have to install the WebSphere Federation Server in addition to the base DB2 system.

Before we get started, we want to give you an insight into the FED_DATA table of the JUNK database that we manually created using the DB2 Control Center (Figure 9-2).

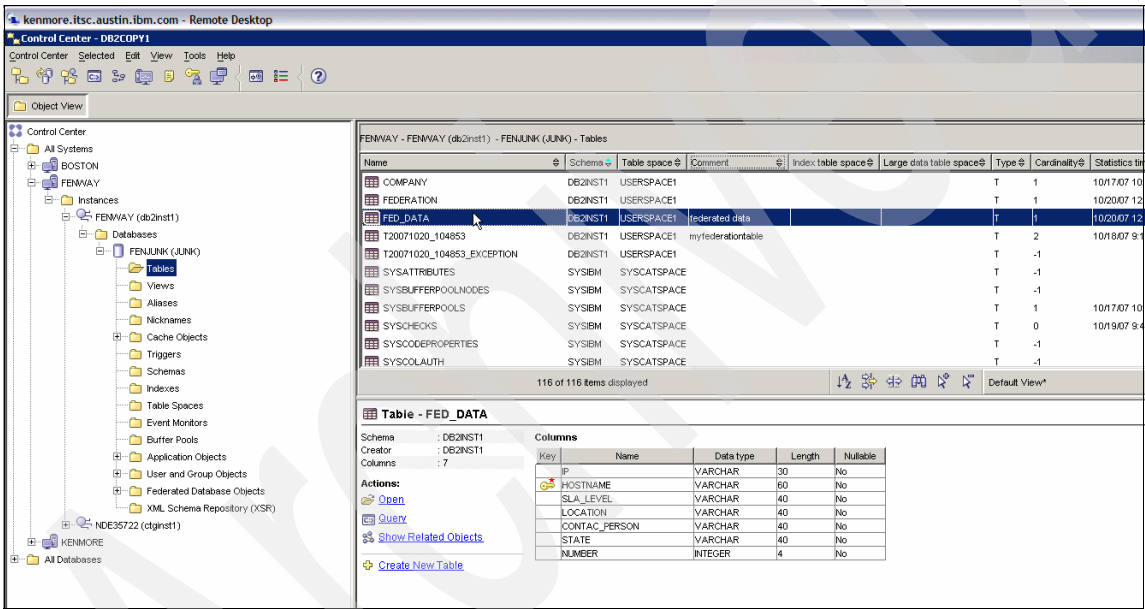


Figure 9-2 FED_DATA table

As you can see, the JUNK database is hosted on fenway.itsc.austin.ibm.com. The FED_DATA table has seven columns defined, most of them using a data type of VARCHAR. We use most of these columns later when we enhance the actual CI application with additional attributes coming from this table structure.

There is one *prerequisite step* that you have to go through to enable your CCMDB database system for federation. This enablement is applied on an instance level, so all databases of the instance that is enabled for federation can be used for federating to remote data sources.

In order to enable federation for the database instance to which your MAXDB71 database belongs, proceed as follows:

1. Open the DB2 Control Center and search for the instance (by default, the instance name is CTGINST1).
2. Select the instance, right-click it, and select **Configure Parameters** (Figure 9-3).

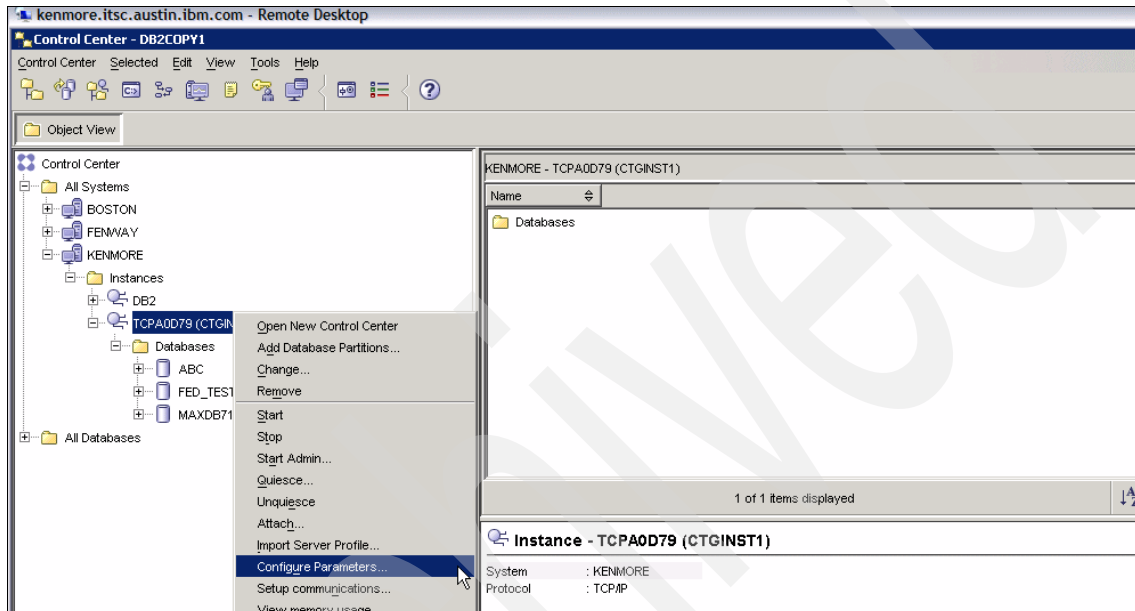


Figure 9-3 Configure parameters option

3. Search for the keyword **FEDERATED** and set the value to **Yes**. Click **OK** in order to save your modifications (Figure 9-4).

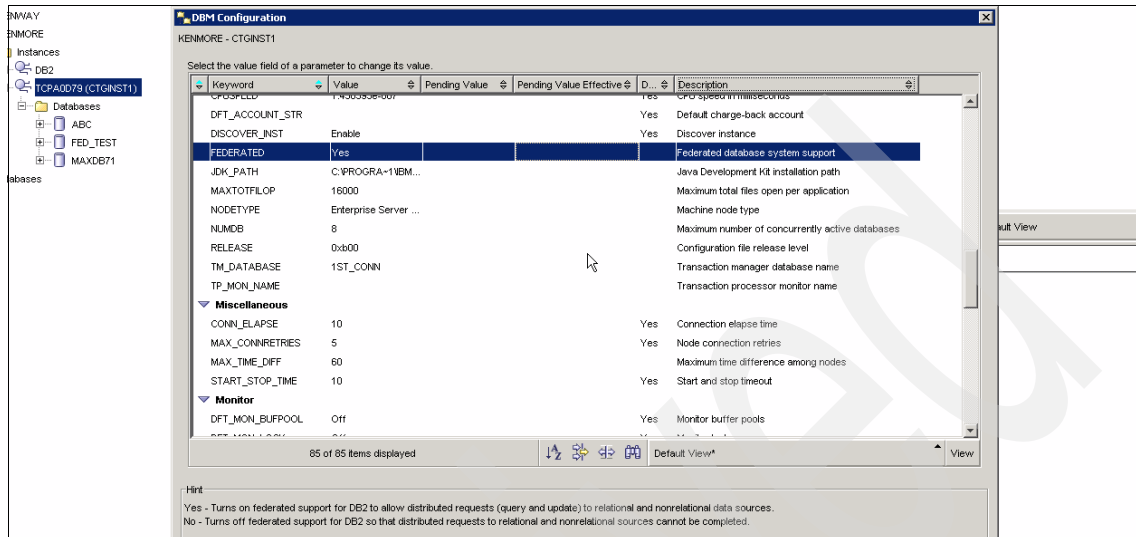


Figure 9-4 Setting the Federated option

4. Now that you have enabled your instance for federation, you can set up your specific MAXDB71 database to federate the FED_DATA table inside the remote JUNK database.

9.3.1 Catalog node and database

The first step in federation setup after enabling the instance for federation is to add a node entry to the federated server (kenmore). The federated server uses a node entry to determine the proper access method to connect to a remote data source. Cataloging the remote database describes the DB2 database to federate.

Proceed as follows:

1. In the DB2 Control Center, select your local database on the federation server. By default this is the MAXDB71 database. Right-click it and select **Configuration Assistant**, as shown in Figure 9-5.

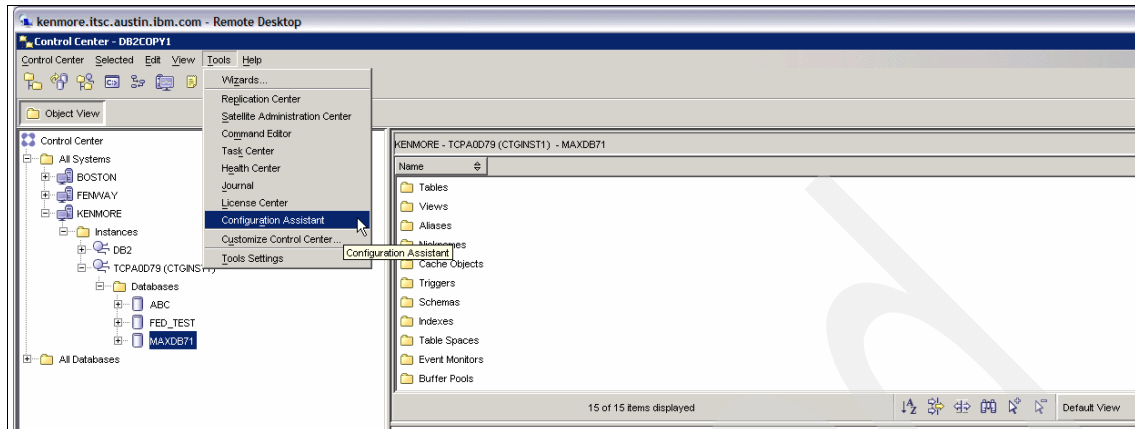


Figure 9-5 Opening the configuration assistant

This opens up the Configuration Assistant Dialog.

2. Right-click in the white space of this dialog and select **Add Database using Wizard...**, as shown in Figure 9-6.

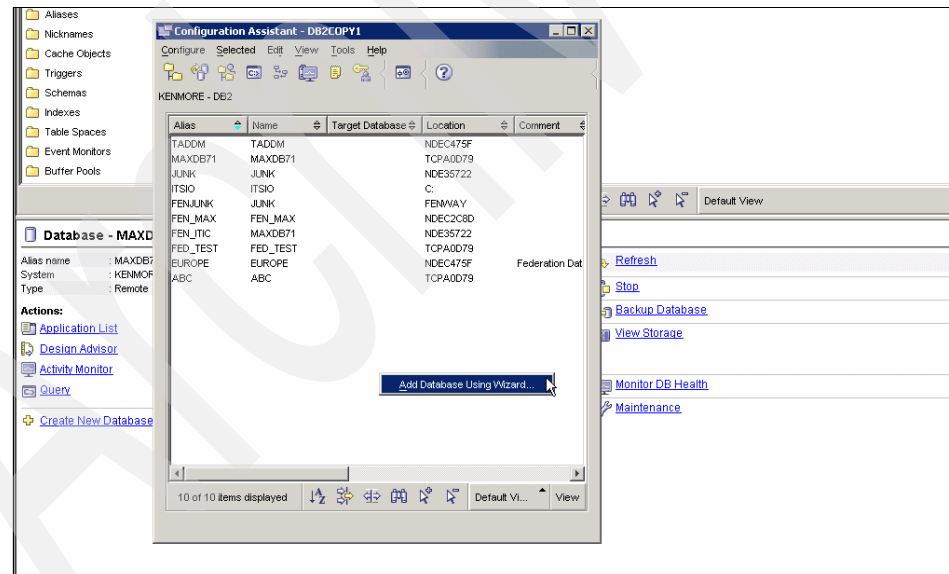


Figure 9-6 Selecting the Add Database Wizard

This brings up the Add Database Wizard.

3. Select the **Search the network** radio button, as shown in Figure 9-7.

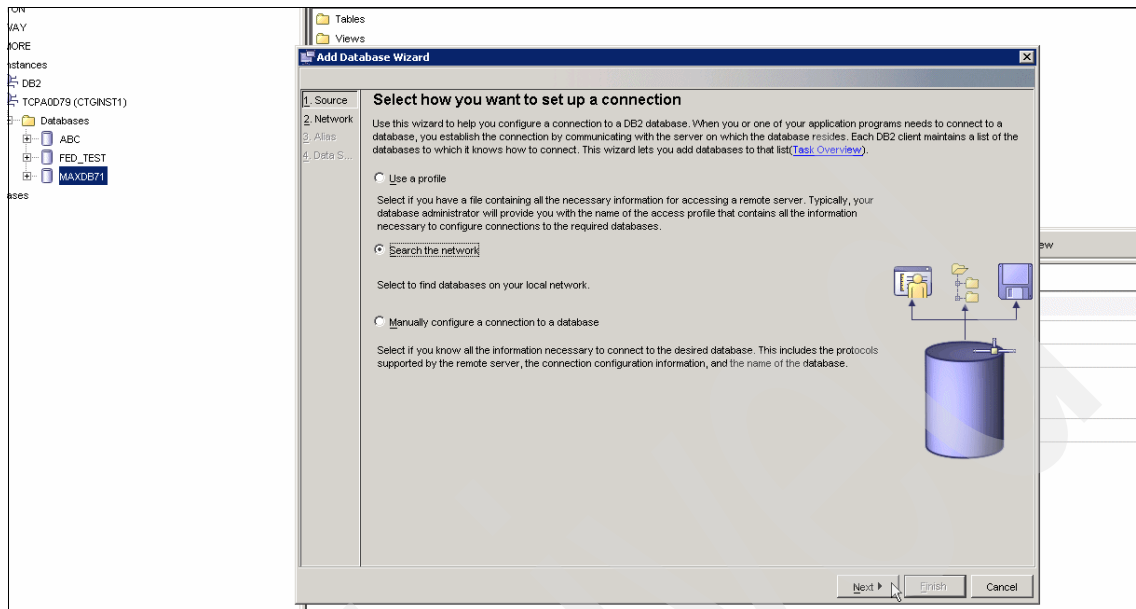


Figure 9-7 Select Search the network

4. Click **Next**, and the dialog box shown in Figure 9-8 displays.

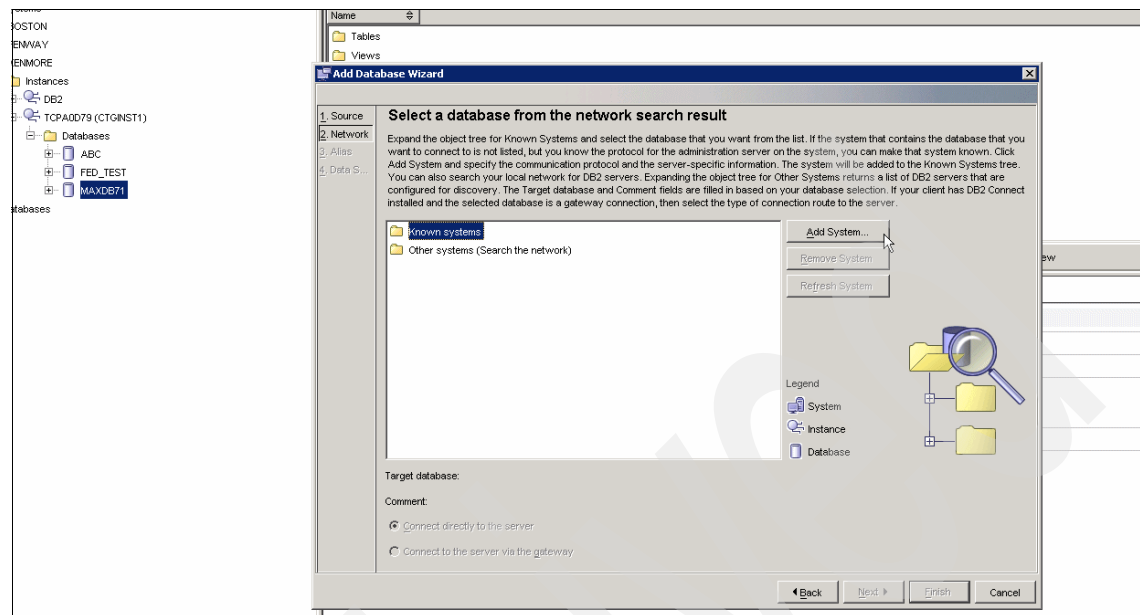


Figure 9-8 Select Add System

5. Click the **Known systems** folder and click the **Add System...** button. The dialog box shown in Figure 9-9 displays.

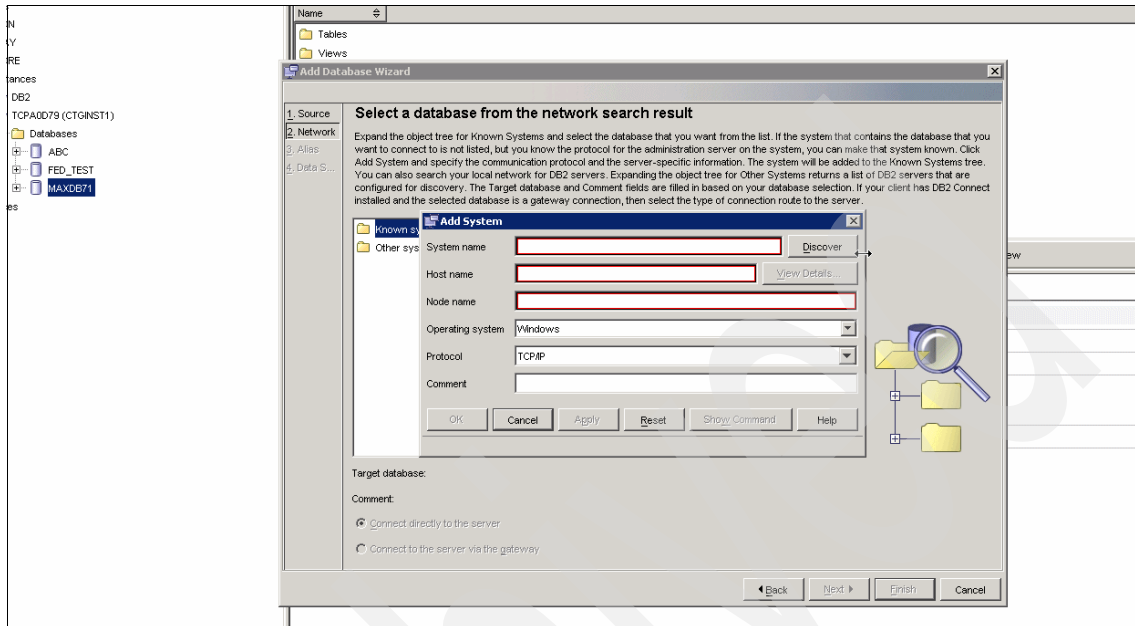


Figure 9-9 Add system dialog

6. Click the **Discover** button and a new dialog comes up with all the systems that have been discovered in your environment, as shown in Figure 9-10.

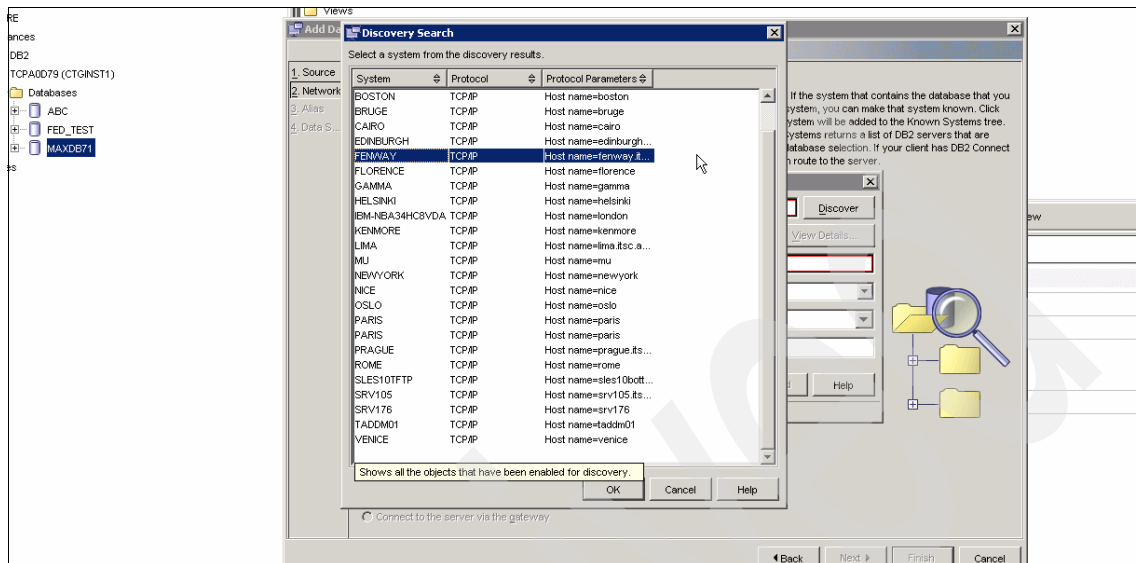


Figure 9-10 List of discovered systems

7. Select the system that you want to catalog; in our case, it is fenway. Click **OK** and then **Next**. The dialog shown in Figure 9-11 is displayed.

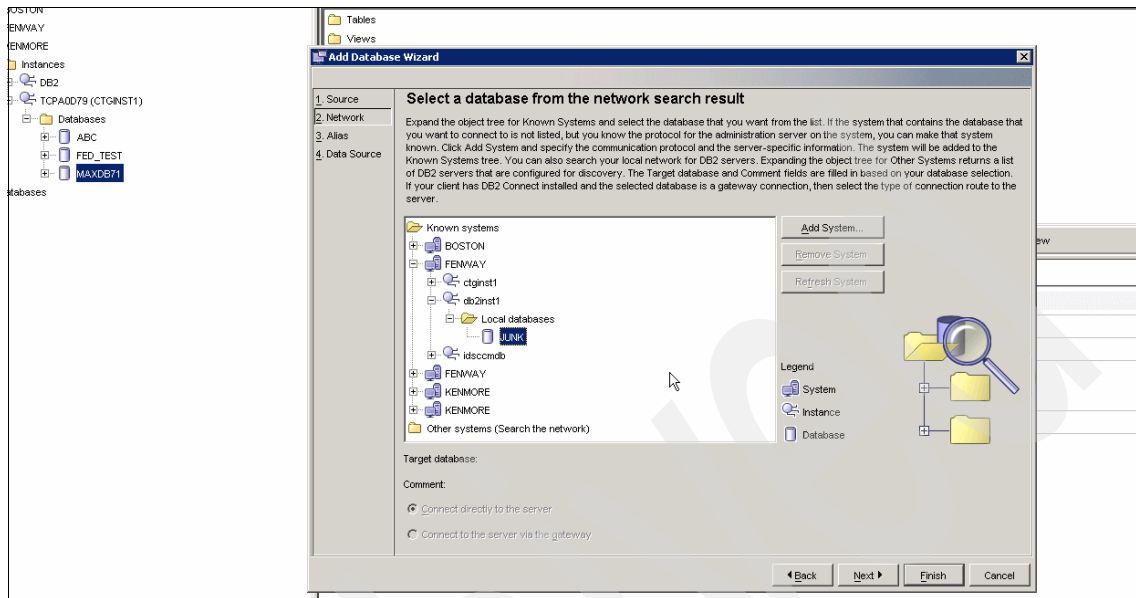


Figure 9-11 Selecting a database from the discovered system

8. Expand the system, expand the instance of interest, and select the database that you want to use as a remote database for your federation scenario. In our environment, we select the JUNK database.

9. After you make your selection, click **Next**. The dialog shown in Figure 9-12 is displayed.

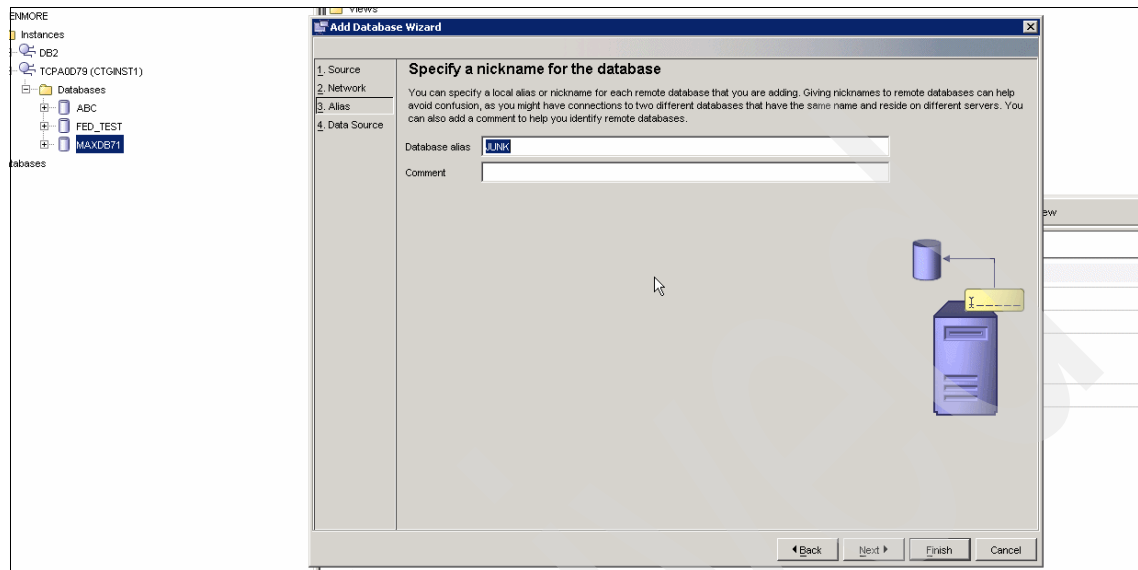


Figure 9-12 Specifying an alias for the remote database

10. Specify an *alias* for the remote database. We keep the default, which is the same name as the local database on the remote system. You need to change it in case you have already defined an alias with the same name or if you want to give the alias a name that is more meaningful to you. Click **Next**. The dialog shown in Figure 9-13 is displayed.

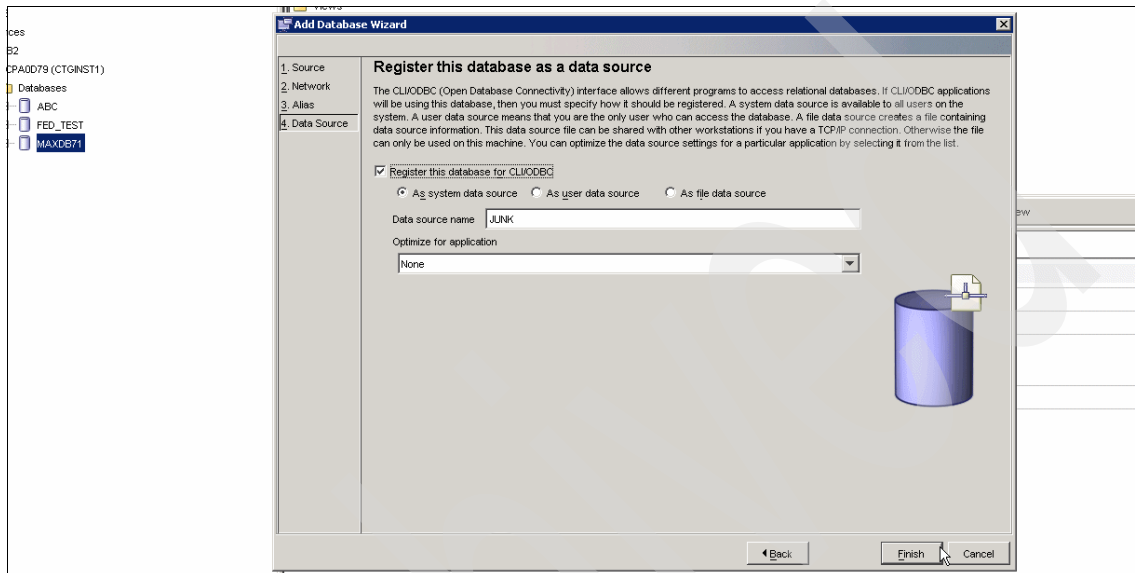


Figure 9-13 Register the database as a data source

11. Click **Finish**; registering this database for CLI/ODBC is optional.

You have now concluded the steps of cataloging the remote node and database.

9.3.2 Creating a wrapper

The next step in the federation setup is to create a wrapper in the federation server to access the remote data source. Wrappers are used by the federation server in order to communicate with and retrieve data from remote data sources. They take over the low level work for communicating with and accessing the remote data source. In our environment, we create a wrapper that can federate data from other DB2 Universal Databases (UDBs). If you are planning to connect to a DB2 database on the mainframe, you have to create another wrapper.

Tip: After you have created a wrapper for a DB2 UDB data source and want to federate to multiple DB2 UDB remote sources, you do not have to create a wrapper for each of them. A wrapper has to be created just once per remote data source type.

Proceed as follows:

1. In the DB2 Control Center, expand the MAXDB71 database definitions, select the entry for **Federated Database Objects**, right-click it, and select **Create Wrapper**, as shown in Figure 9-14.

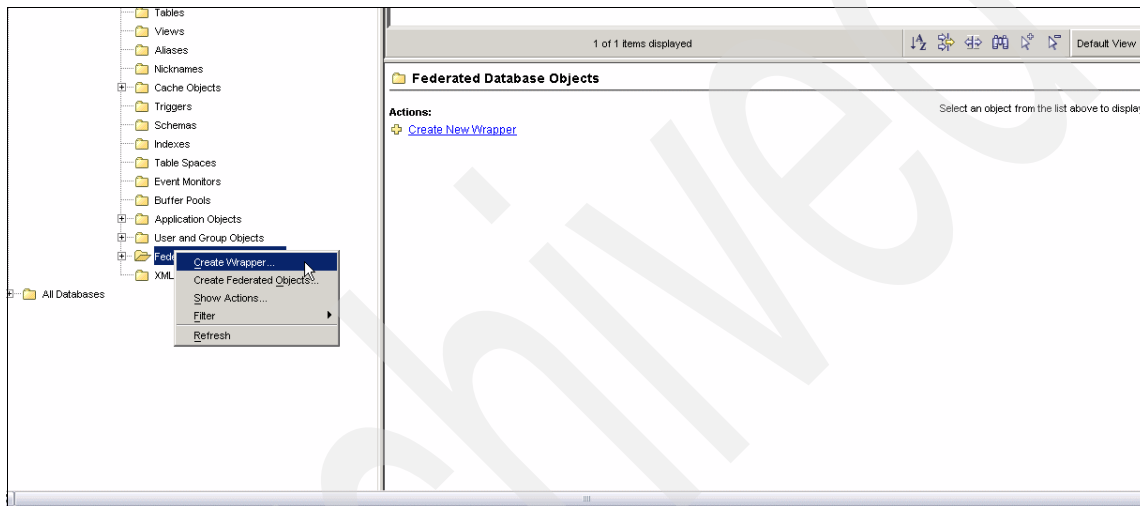


Figure 9-14 Create Wrapper option in DB2 Control Center

2. In the Data Source drop-down menu, select **DB2** and click **OK**, as shown in Figure 9-15.

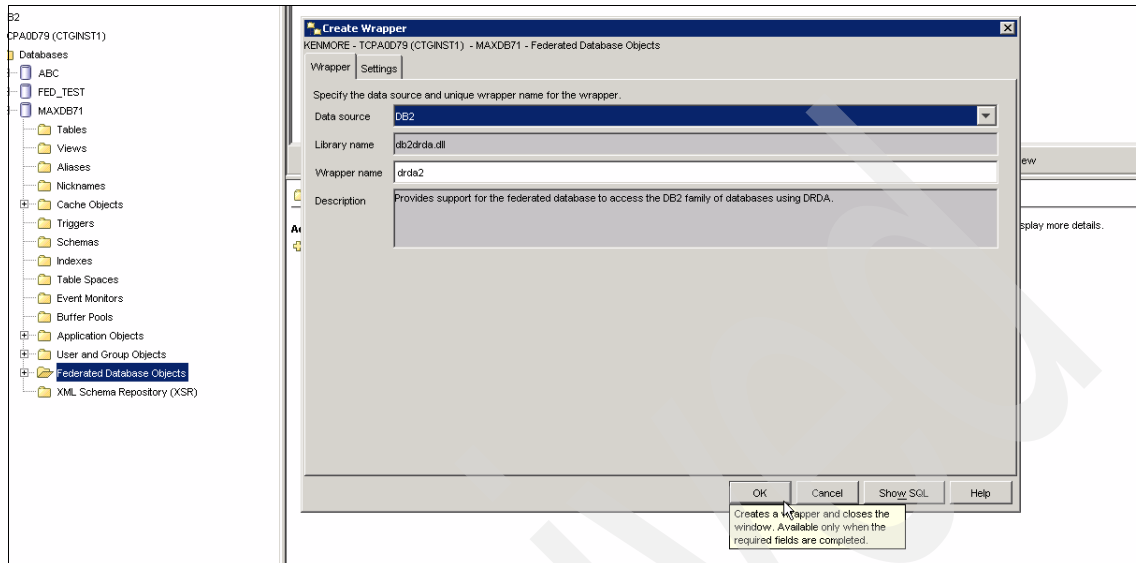


Figure 9-15 Create Wrapper dialog

3. This completes the creation of the wrapper. You see the wrapper defined as DRDA under the Federated Database Objects folder, as shown in Figure 9-16.

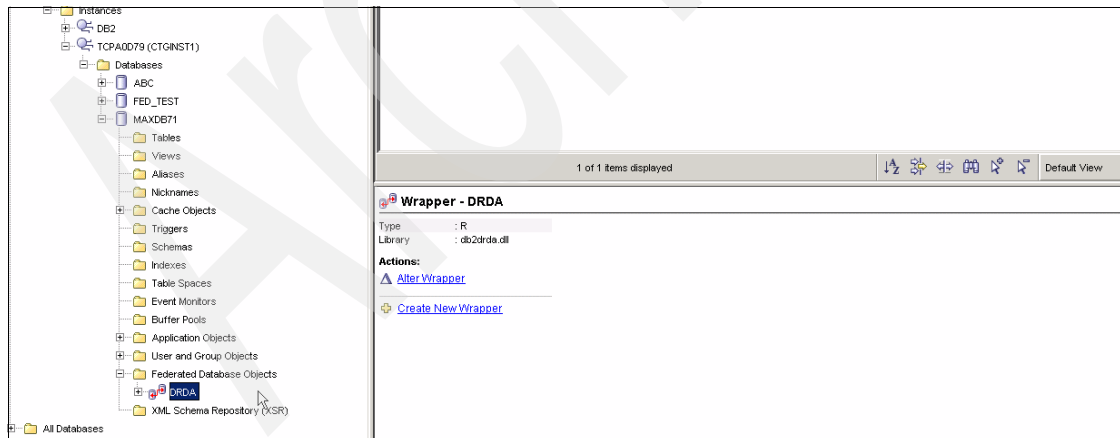


Figure 9-16 Viewing created wrapper

9.3.3 Registering the server

The next step is create a server definition. The register server operation defines a data source specifically to a federated database.

Proceed as follows:

1. In the DB2 Control Center, expand the DRDA wrapper entry, right-click it, and select **Create**, as shown in Figure 9-17.

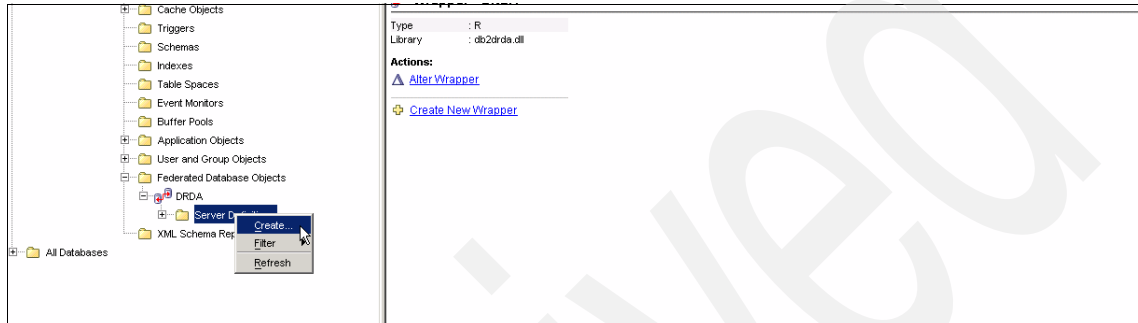


Figure 9-17 Menu to create a server definition

2. The **Create Server Definitions** dialog is shown in Figure 9-18. Click **Add....**

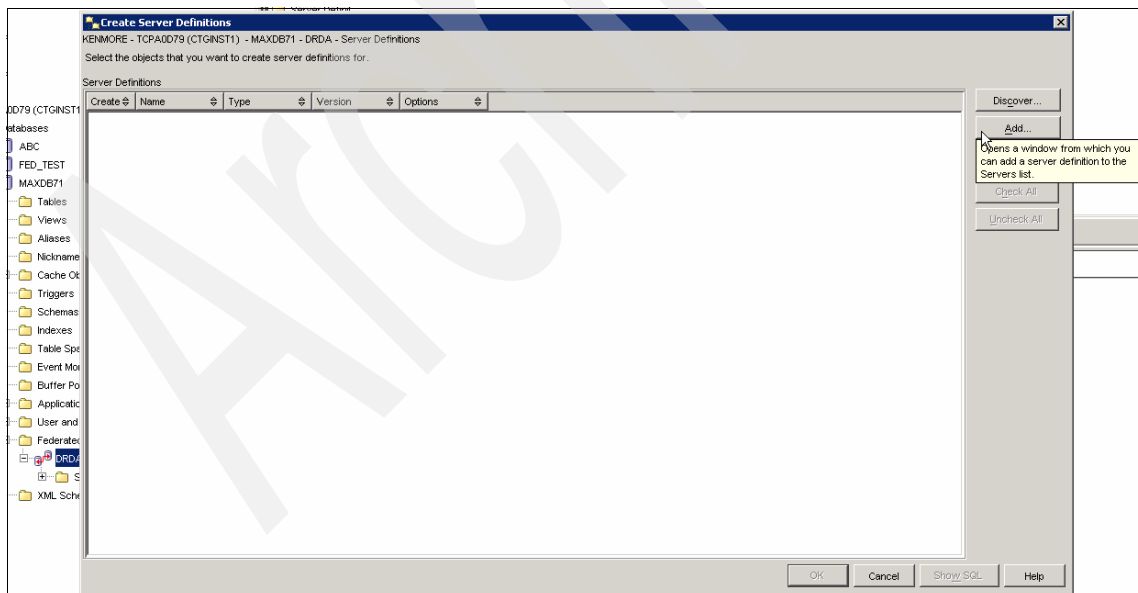


Figure 9-18 Create Server Definitions dialog

The Create Server Definition window is shown. There are two tabs, Server Definition and Settings, that you need to consider.

3. In the **Server Definition** tab window, provide the appropriate data entries according to your environment (Figure 9-19). In our environment, we use fenway as the host name for the remote system, DB2/UDB for the type, 9.1 as the version, and the appropriate credentials for your environment.

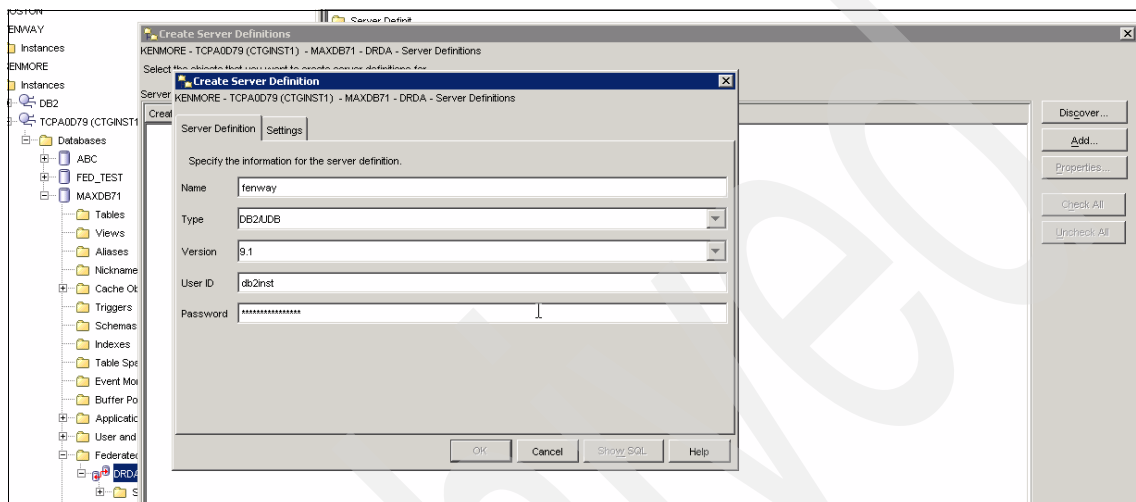


Figure 9-19 Server Definition dialog

4. Click the **Settings** tab, select the parameter **DBNAME** in the option column of the dialog, and enter the name of your remote database in the **Value** column. In our case, we specify JUNK as the database name.
5. Click **OK** and you find a new server definition entry in the DRDA wrapper folder, as shown in Figure 9-20.



Figure 9-20 New server definition displays in the DB2 Control Center

This concludes the definition of the server definition. The next step is to create a user mapping.

9.3.4 Creating a user mapping

The federation server needs to know how to authenticate to the federated database. In order to prevent someone from having to manually enter credentials for authenticating each time a connection is established, credentials are defined for how to connect to the remote data source. You need to define an association, a user mapping, between the user ID on the federation server and the corresponding remote data source user ID and password.

Proceed as follows:

1. In the DB2 Control Center, expand the server definition entry that you just created in the previous step, select **User Mappings**, right-click it, and select **Create...**, as shown in Figure 9-21.



Figure 9-21 Selecting option to create User mappings

The Create User Mappings dialog window is shown.

2. From the Available local User IDs frame, select the local user of your federation server database. In our environment, we select the user MAXIMO.

Tip: Because we are reproducing this example for documentation purposes, you do not see the user MAXIMO in the list of available users anymore for this specific server definition. You can, however, see it in the window behind the Create User Mappings dialog.

By default, the user for the local MAXDB71 database is MAXIMO.

3. Select the user and move it over to the Selected user IDs frame in the Create User Mappings dialog window.

4. In the same dialog, click the **Settings** tab and specify the values for the REMOTE_AUTHID and REMOTE_PASSWORD options, as shown in Figure 9-22.

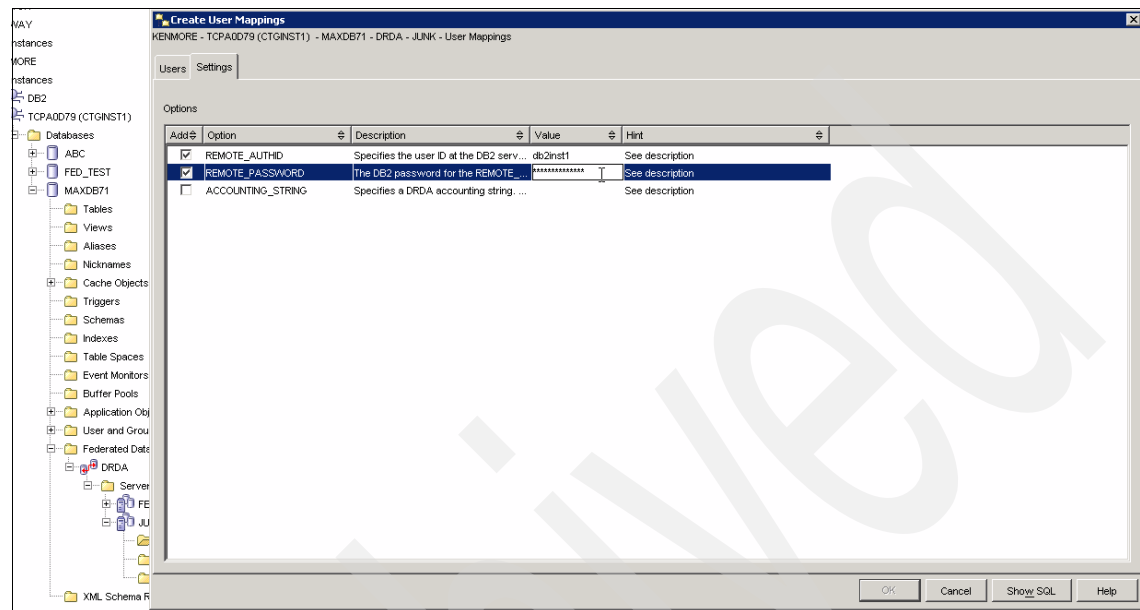


Figure 9-22 User Mapping settings

- Click **OK** and you will find a new entry in the User Mappings section of the Server Definitions folder, as shown in Figure 9-23.

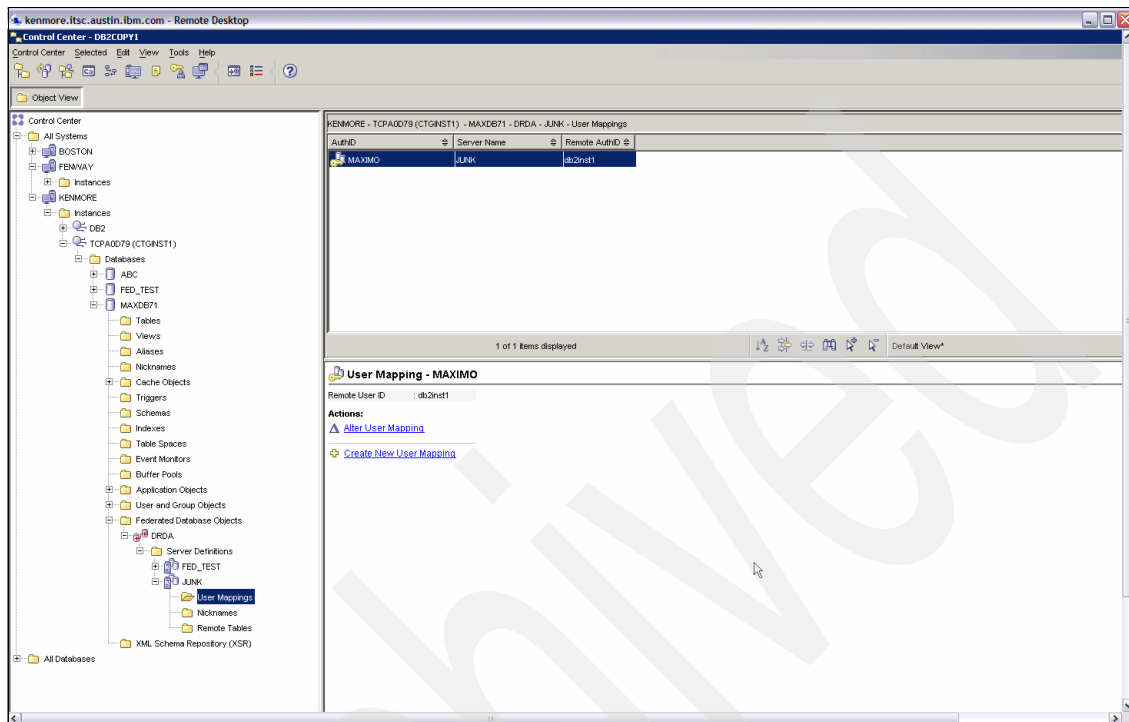


Figure 9-23 New User Mapping entry

This concludes the User Mapping definitions process.

9.3.5 Creating a nickname

The final step in setting up federation in a database or data source layer (depending on whether you are federating a relational database or another kind of data source) is to define a nickname for the remote database table.

A *nickname* is a local name for a remote database table. Users and applications use this name to access the remote database table. You will see that we use this nickname later on when we define the Maximo Business Object (MBO).

Proceed as follows:

1. In the DB2 Control Center, from the Server Definition that you created, select **Nicknames**, right-click it, and select **Create...**, as shown in Figure 9-24 on page 468.

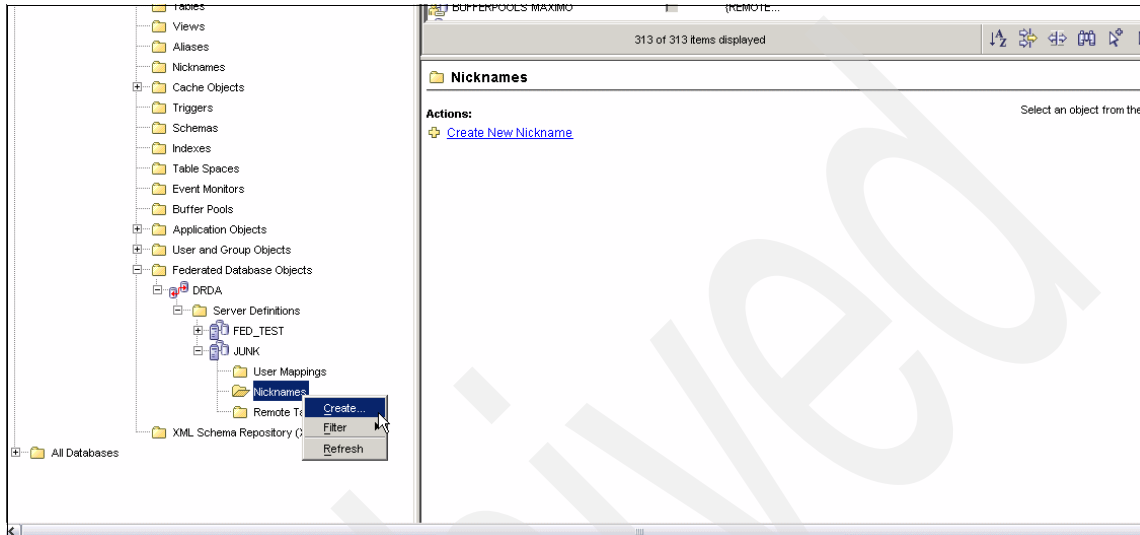


Figure 9-24 Option to create nicknames

The **Create Nicknames** dialog window is displayed.

2. Click **Add....** This displays the **Create Nickname** dialog shown in Figure 9-25.

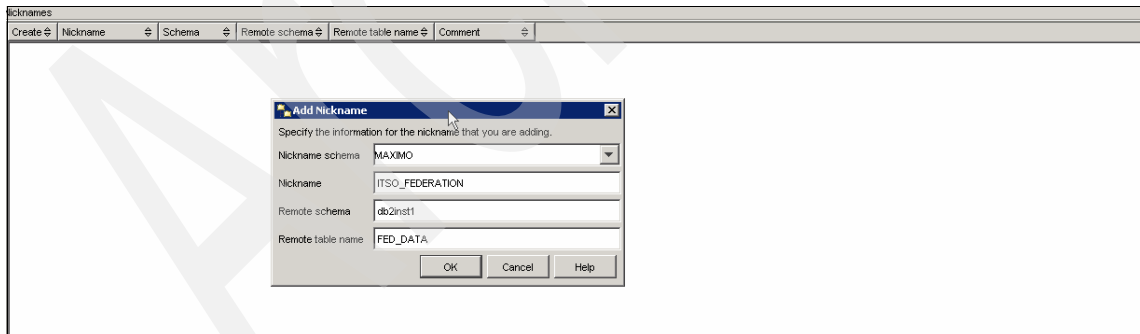


Figure 9-25 Create Nicknames dialog

3. Provide values for the Nickname schema, a nickname itself, the value for the Remote schema, and the value for the Remote table name.

These values depend on your environment. You can see that we specify the nickname as `ITSO_FEDERATION`. You have to remember this nickname when creating the Maximo Business Object in the next step of the overall federation setup.

You now have set up everything to successfully federate the remote data source. In Figure 9-26, we show the successful `ITSO_FEDERATION` nickname definition. Given that you have successfully defined your nickname, you will see and have access to the remote database table. Remember, in our case the remote database table was `FED_DATA` in the remote database `JUNK`.

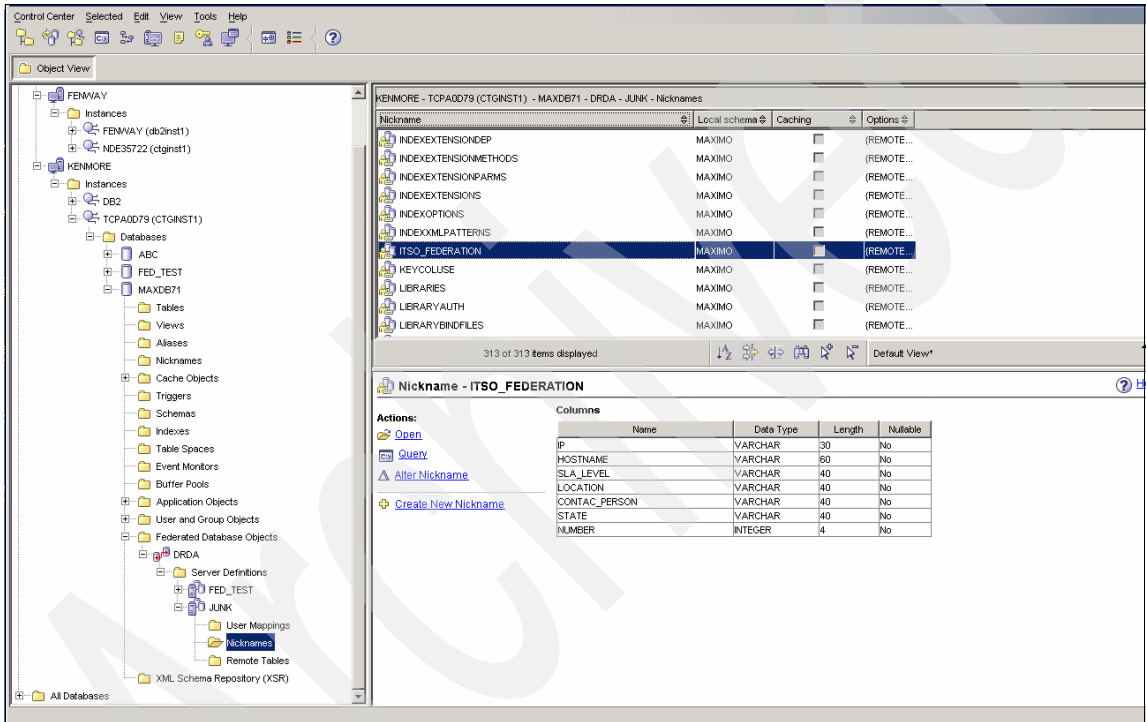


Figure 9-26 Nickname has been created

After you set up the federation at the data source layer, you are now ready to make use of this setup in the CCMDDB environment itself.

9.4 Creating a Maximo Business Object (MBO)

We are now starting to extend our existing CCMDB applications in the process environment. In fact, we duplicate an existing application, the actual configuration items application, and enrich the application panels with additional attributes from the federated table.

First, we need to create a Maximo Business Object (MBO) that represents the federated database table inside the process environment. An MBO is a Java object with business logic that encapsulates a database table. You do not have to care about writing Java code in order to create an MBO. When you create an MBO, a default Java class is specified for you. The Java code is actually the layer between the database tables and the application itself. If you want to have some specialized logic when retrieving the data from the database before presenting them in the user interface, you have to extend the default Java classes and write your own code. In our example, we work with the default class when we create our own MBO.

Proceed as follows:

1. In the CCMDB Web User Interface, select **System Configuration** → **Platform Configuration** → **Database Configuration** in order to launch the Database Configuration application, as shown in Figure 9-27.

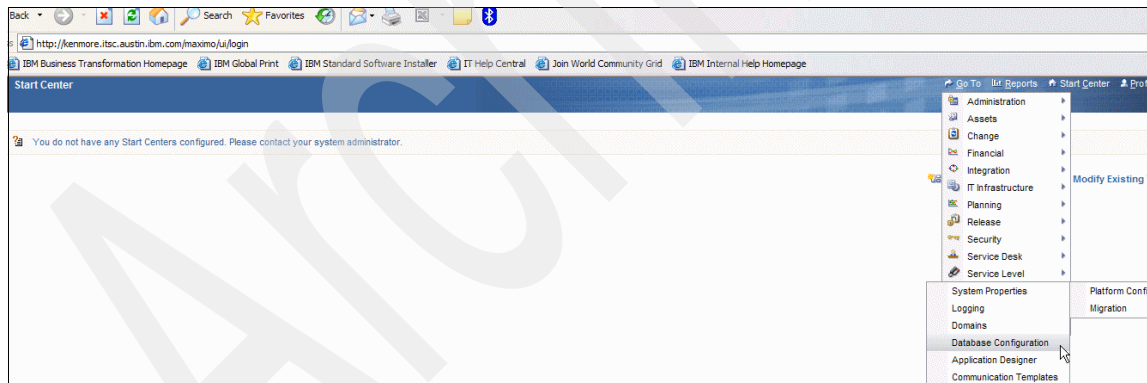


Figure 9-27 Database Configuration window

- Next, in the menu bar, click the icon to select a **New Object**, as shown in Figure 9-28.

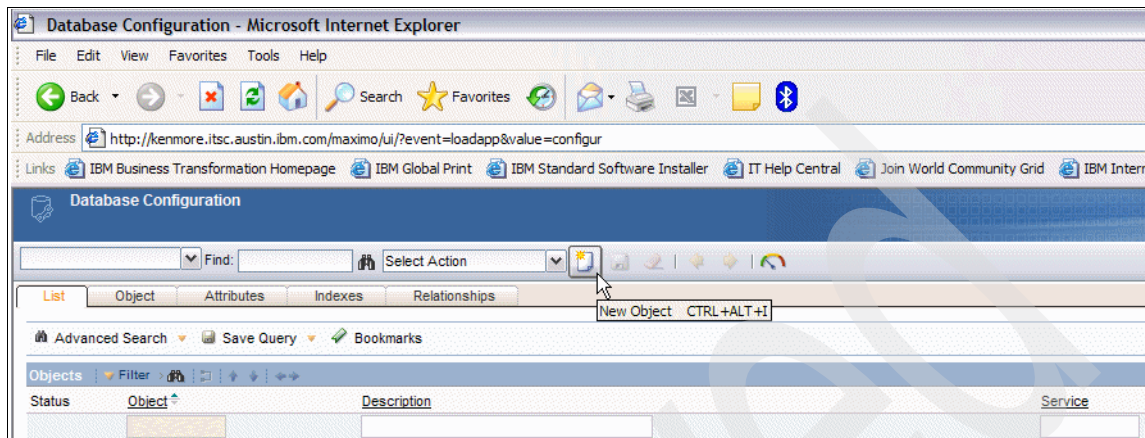


Figure 9-28 Select New Object

This will present a window where you have to specify the nickname that you created in setting up the federation on the database layer.

- In our environment, we use ITSO_FEDERATION as the nickname, as shown in Figure 9-29.

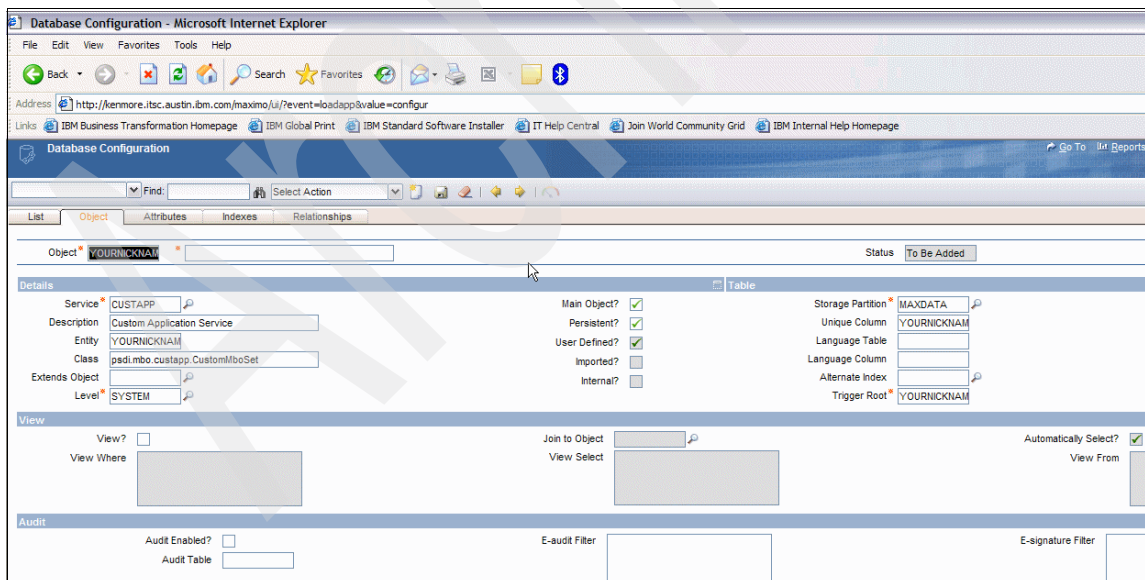


Figure 9-29 Specifying the nickname

4. In the upper left area of the window, fill in the nickname that you have defined in the federation setup in the Object attribute and tab out of the Object field. We use ITSO_FEDERATION in our example because this is the nickname that we use to connect to the federated database table FED_DATA. After you tab out of the Object field, the User Defined and Imported check boxes will be automatically checked. This indicates that the existing federated table is used and that there is no need to create a new table.
5. Also fill in a description of your new MBO in the attribute field right next to the field where you specify the nickname. We use ITSO Federation as our description for the new MBO.
6. Uncheck the **Add Rowstamp** check box in order to make sure that the MBO is read-only. One of the key reasons we favor the federation approach over the approach of importing data is that it leaves the ownership of the data within the group that is responsible for the remote data source. Therefore, set the access to the federated data source to read-only.
7. Select the **Main Object** check box and add the Number column of the FED_DATA table into the Unique Column field of the MBO definition window.

The screenshot shows a web browser window titled "Database Configuration - Microsoft Internet Explorer". The address bar shows the URL: <http://kenmore.itsc.austin.ibm.com/maximo/ui/pevent-loadapp&value=-configur>. The page is titled "Database Configuration" and has a navigation bar with tabs: List, Object, Attributes, Indexes, Relationships. The "Object" tab is selected, and the "Object" field is set to "ITSO_FEDERAT". The "Details" section shows the following fields:

- Service: CUSTAPP
- Description: Custom Application Service
- Entity: ITSO_FEDERAT
- Class: psdi.mbo.custapp.CustomMboSet
- Extends Object:
- Level: SYSTEM

The "Table" section shows the following fields:

- Main Object? ☒
- Persistent? ☒
- User Defined? ☒
- Imported? ☒
- Internal? ☐
- Storage Partition:
- Unique Column: NUMBER
- Language Table:
- Language Column:
- Alternate Index:
- Trigger Root: ITSO_FEDERAT

The "View" section shows the following fields:

- View? ☐
- View Where:
- Join to Object:
- View Select:

The "Audit" section shows the following fields:

- Audit Enabled? ☐
- Audit Table:
- E-audit Filter:

Figure 9-30 MBO Definition window

A default Java class called psdi.mbo.custapp.CustomMboSet is specified in the class attribute field. It is a default class that is specified automatically. You do not have to write any code in order to define the MBO.

8. Click the **Attributes** tab, where you can find all of your column definitions of your federated database. You can see that the Data Type of VARCHAR changed to ALN (alphanumeric) in the MBO definition.

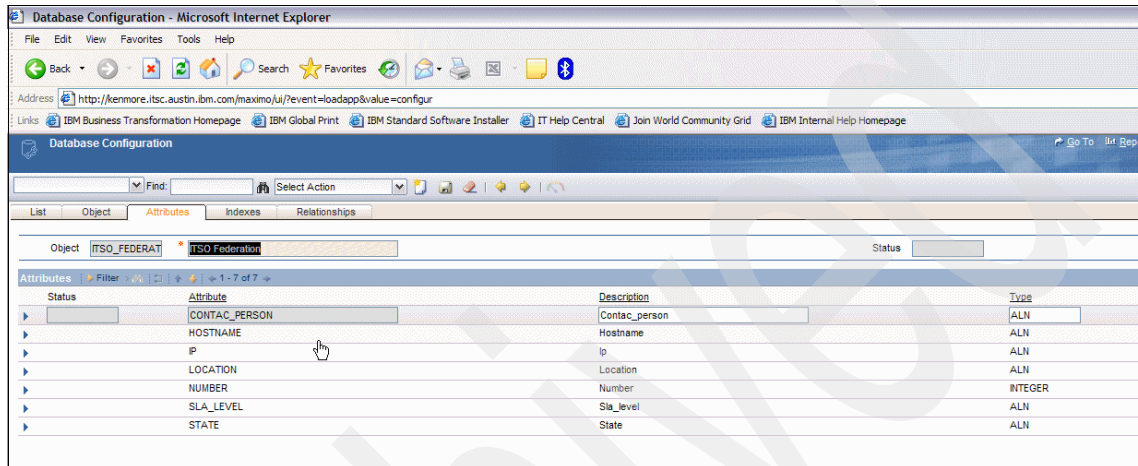


Figure 9-31 Attributes tab

9. Save your record by clicking the small diskette icon in the menu bar. The object status will become "To be Added", that is, you are not yet ready to use the new MBO. You have created the definition, but the CCMDB database needs to be updated first. We show how you can do that in 9.5, "Generating the object in the CCMDB database" on page 474.

Important: If you change anything in the remote database, for example, add a column or change the column length, you have to create a new nickname and a new MBO.

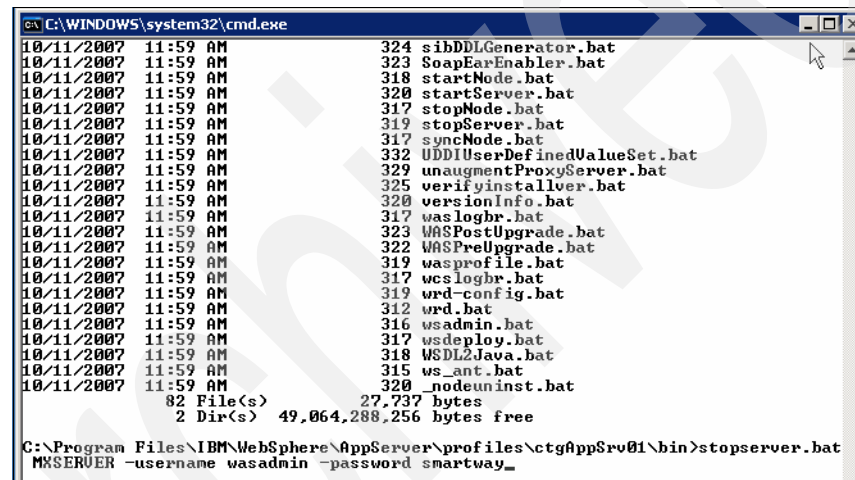
9.5 Generating the object in the CCMDB database

Now that you have defined the object, you must generate it in the database. This can be accomplished by running the **configdb** program that is located on the CCMDB process runtime database system.

Proceed as follows:

1. In order to run the **configdb** program, stop the application server first.
2. Also, log out from the CCMDB Web User Interface before you stop the application server.

You can stop the application server either using the command line or by using the WebSphere Admin Console, as shown in Figure 9-32.



```
C:\WINDOWS\system32\cmd.exe
10/11/2007 11:59 AM 324 sibDDLGenerator.bat
10/11/2007 11:59 AM 323 SoapEarEnabler.bat
10/11/2007 11:59 AM 318 startNode.bat
10/11/2007 11:59 AM 320 startServer.bat
10/11/2007 11:59 AM 317 stopNode.bat
10/11/2007 11:59 AM 319 stopServer.bat
10/11/2007 11:59 AM 317 syncNode.bat
10/11/2007 11:59 AM 332 UDDIUserDefinedValueSet.bat
10/11/2007 11:59 AM 329 unaugmentProxyServer.bat
10/11/2007 11:59 AM 325 verifyinstallver.bat
10/11/2007 11:59 AM 320 versionInfo.bat
10/11/2007 11:59 AM 317 waslogbr.bat
10/11/2007 11:59 AM 323 WSPPostUpgrade.bat
10/11/2007 11:59 AM 322 WSPPreUpgrade.bat
10/11/2007 11:59 AM 319 wasprofile.bat
10/11/2007 11:59 AM 317 wcslogbr.bat
10/11/2007 11:59 AM 319 wrd-config.bat
10/11/2007 11:59 AM 312 wrd.bat
10/11/2007 11:59 AM 316 wsadmin.bat
10/11/2007 11:59 AM 317 wsdeploy.bat
10/11/2007 11:59 AM 318 WSDL2Java.bat
10/11/2007 11:59 AM 315 ws_ant.bat
10/11/2007 11:59 AM 320 _nodeuninst.bat
82 File(s) 27,737 bytes
2 Dir(s) 49,064,288,256 bytes free

C:\Program Files\IBM\WebSphere\AppServer\profiles\ctgAppSrv01\bin>stopserver.bat
MMSERVER -username wasadmin -password smartway_
```

Figure 9-32 Stopping the application server

If you kept all the defaults at installation time, you will find the same directory structure on your application server system, which is kenmore in our case.

3. Use the **stopserver.bat** program shown in Figure 9-32 in order to stop the application server from the command line.
4. If you prefer to use the WebSphere administration console to stop the server, log into the console. In our environment, we use the following URL to log in:

<http://kenmore.itsc.austin.ibm.com:9060/ibm/console>

5. You have to authenticate; we use the *wasadmin* account in our environment, as shown in Figure 9-33.

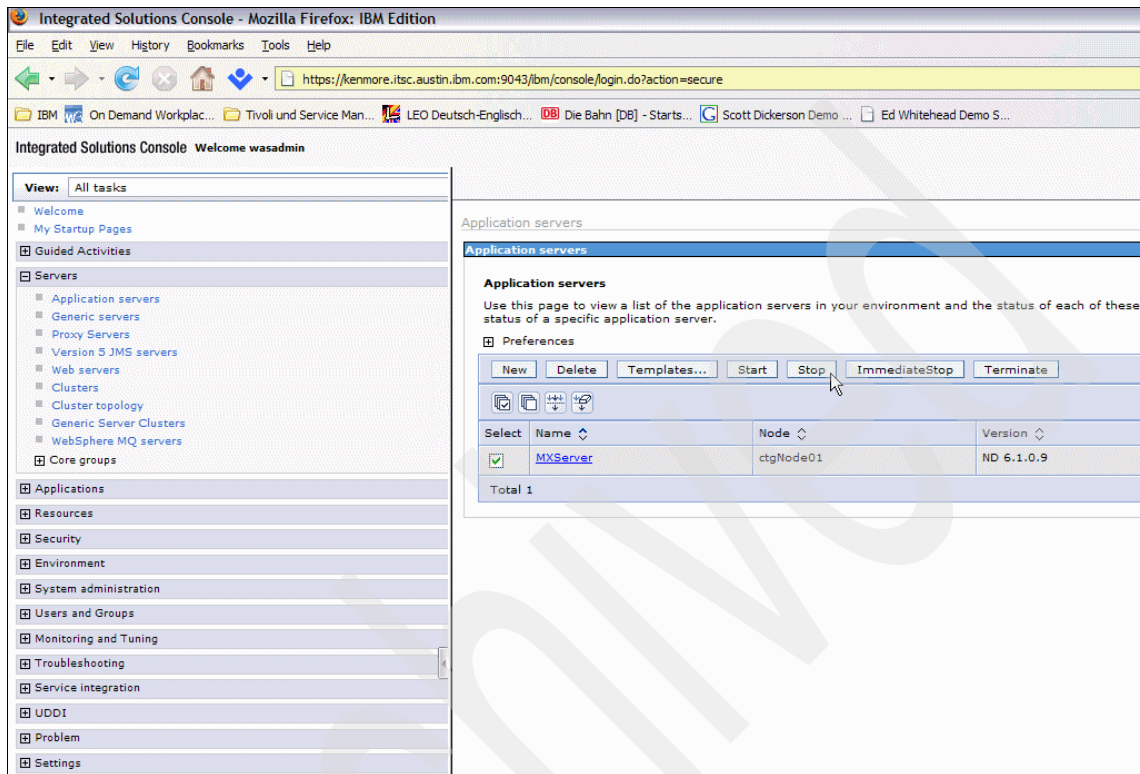


Figure 9-33 WebSphere administration console

6. In the tree view in the left frame, expand **Servers** and click **Application servers**, and you will see all application servers defined in your implementation. In our environment, we have the MXServer, which is the CCMDB J2EE application server. Select it, and click the **Stop** button in order to stop the server.

Important: Before you move onto the next step and run the `configdb` command, wait for a couple of minutes.

The next step is to run the **configdb** command, as shown in Figure 9-34.

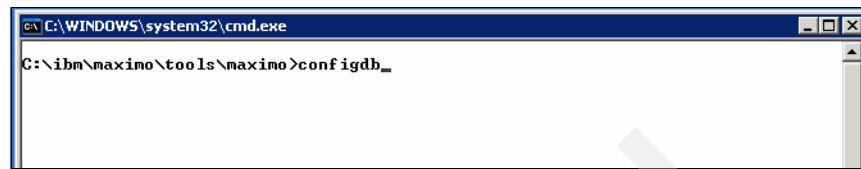


Figure 9-34 Run the configdb command

You can run the **configdb** command from the directory without any arguments.

Running the command produces messages on the current terminal. You can also check for more details in the log file, which is located in `c:\ibm\maximo\tools\maximo\log`. There you can find logs for your recent executions of the **configdb** command, as shown in Figure 9-35. You can sort it by date.

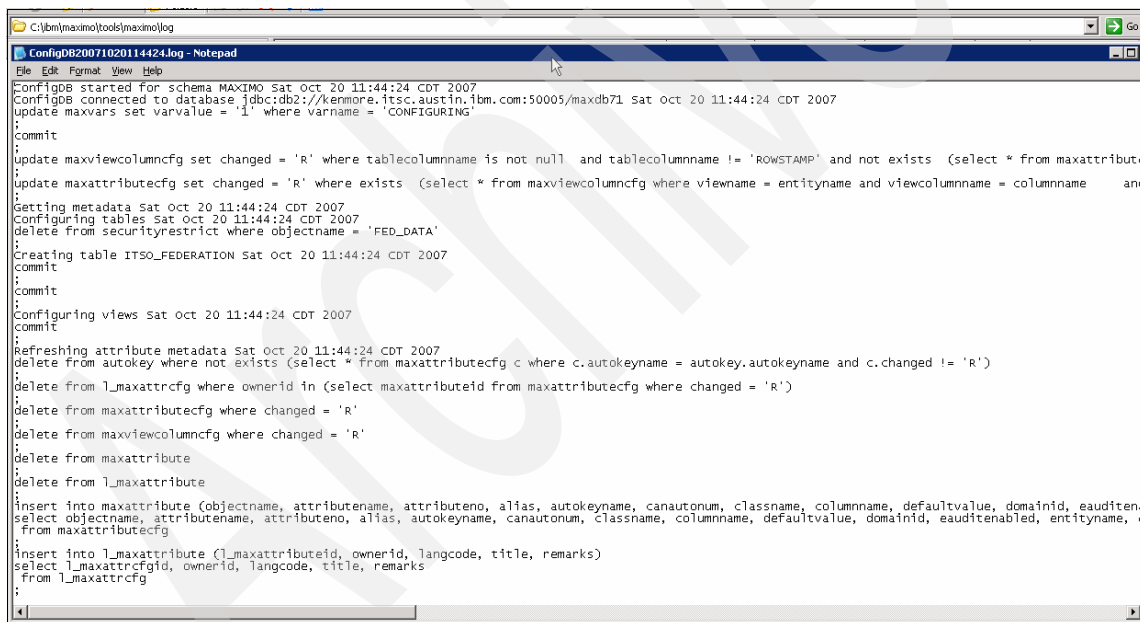


Figure 9-35 Results of configdb command

You can see from the log shown in Figure 9-35 that updates to the local MAXDB71 database are created in order to point to the remote FED_DATA table using the nickname ITSO_FEDERATION.

7. If you do not see any error messages while running the **configdb** command, it is usually not necessary to analyze the log. In case you are running into errors, the log is your primary source for help.

Important: Before you move onto the next step and restart the application server, wait for a couple of minutes.

8. The last step you need to take in this series of configuration steps is to restart the application server, which can be done from the WebSphere Admin Console, as shown in Figure 9-36.

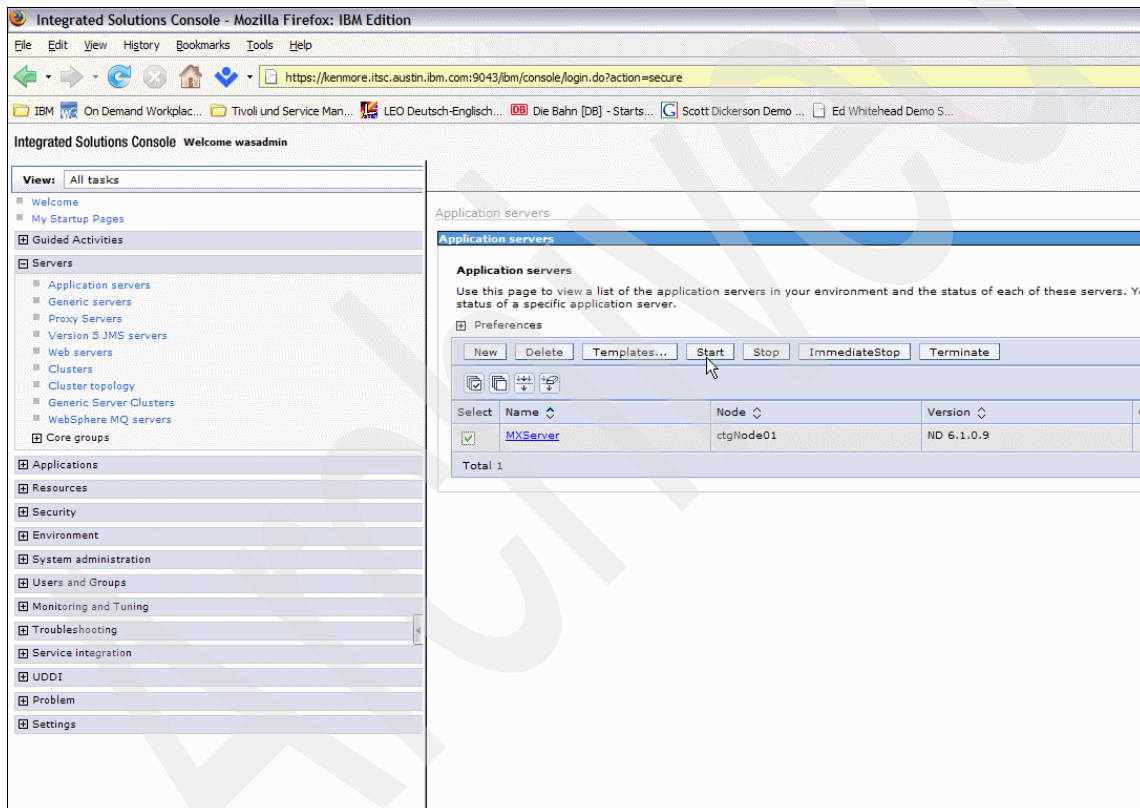


Figure 9-36 Restart the application server through the WebSphere administration console

9. If you prefer to use the command line, replace the **stopserver.bat** command with the **startserver.bat** command using the same arguments shown in Figure 9-36.

9.6 Defining a relationship

In order to display additional data provided by a federated database table in an existing application such as the actual configuration item application, you must establish a relationship between a preexisting MBO and the MBO that represents the federated table.

You need to relate the data from the federated table to a specific record you select in the CCMDB application. In our environment, we want to enrich the actual CI application with additional data from our federated table to provide data, for example, that is not discoverable or that must be kept and maintained separately.

You can use the new MBO pointing to the federated table in isolation, but it is best to always relate it to a standard MBO in order to have an anchor point that it relates to because the purpose is to extend exiting local CI data with additional data from the federated source.

Figure 9-37 shows an overview of our relationship setup. In this section, we explain how to perform this setup.

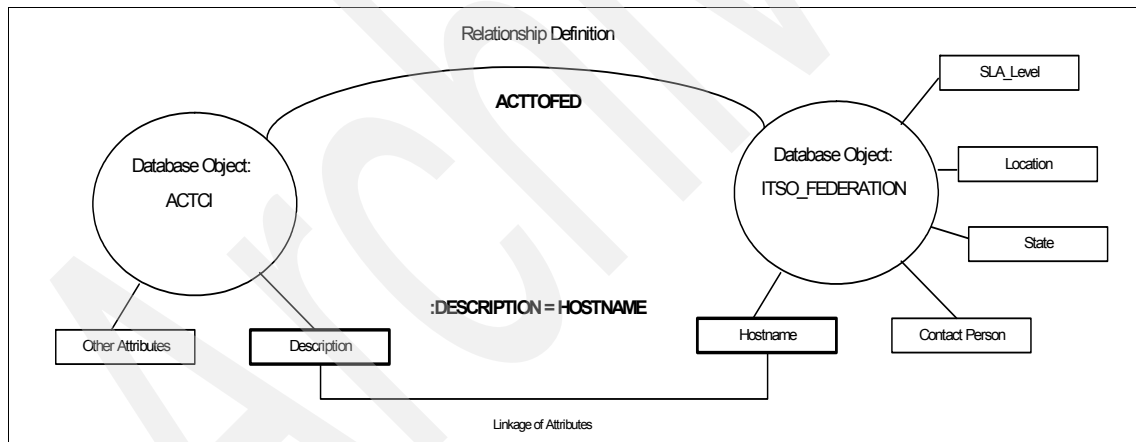


Figure 9-37 Relationship definition overview

As we mentioned already, each application, such as the actual configuration items application, is based on a primary MBO that refers to the database table and attributes being used to work with the appropriate data.

In our example, we use the ACTCI MBO, which is the primary MBO for the actual configuration items application. Each MBO has defined a number of attributes.

Proceed as follows:

1. Use the Database Configuration application and search, for example, for the ACTCI MBO in order to see which attributes are defined for this object or choose the object that you want to link to your newly created MBO.

The MBO that represents our federated table is ITSO_FEDERATION. There are various attributes defined for this object as well that are actually pointers to the columns in the remote database.

In order to link the two objects (MBOs), you have to find at least one qualifier or attribute on each object to relate these objects.

In our example, we use the Description attribute of the ACTCI object to link to the Hostname attribute of the ITSO_FEDERATION object.

2. If you need to discover which application object and attribute a specific field is using in the database, select the field and press Alt+F1. This displays the database definition for this field. Figure 9-38 shows this procedure for the Description field in the actual configuration items application.

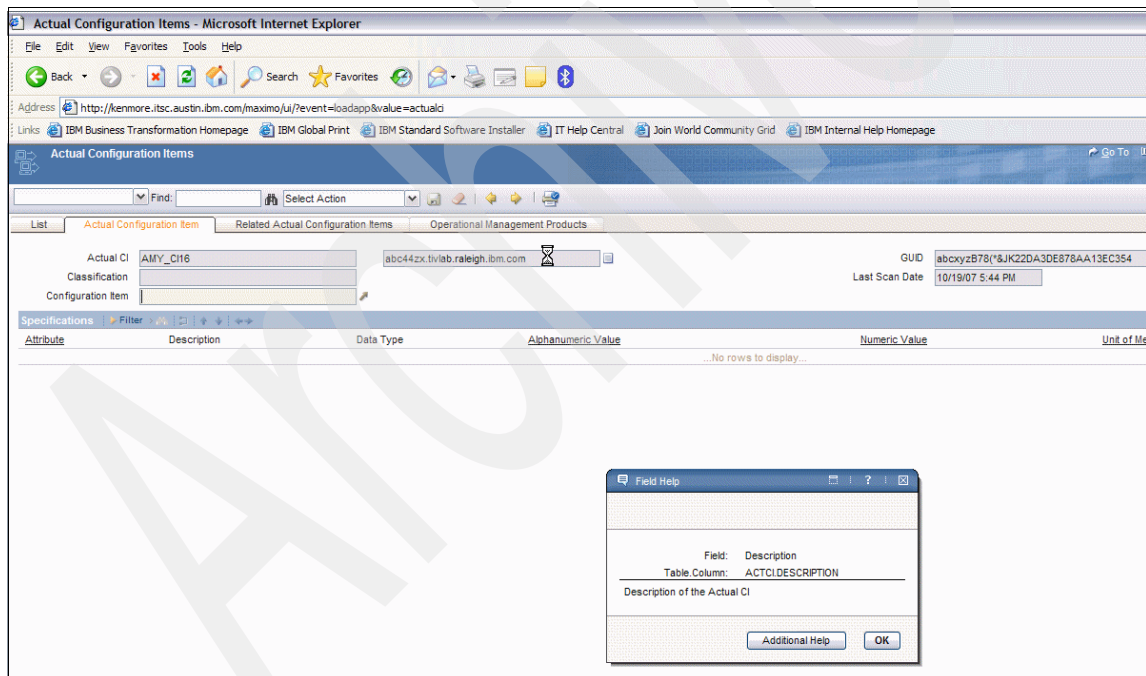


Figure 9-38 Database definition for the Description field

The pop-up dialog shows that the field reflecting the host name of the actual CI is actually point to the ACTCI.DESCRPTION attribute in the database. This is the attribute that we link to the HOSTNAME column of our federated database table.

3. In order to define the relationship, go to the Database Configuration application and search for the ACTCI object. This is the object that you have to use to define the relationship, because it is the primary object of the application that we want to extend (see Figure 9-39).

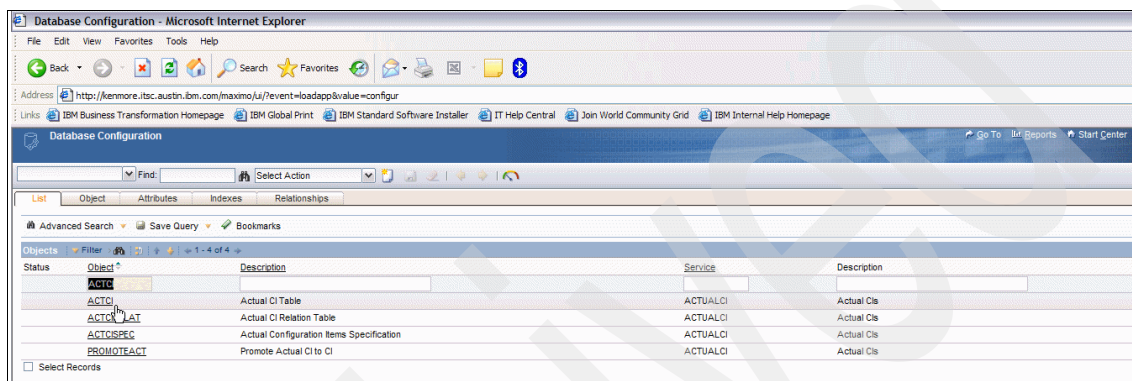


Figure 9-39 Searching the Database Configuration for the ACTCI object

4. Select the **Relationships** tab and you can see all existing relationship definitions (Figure 9-40).

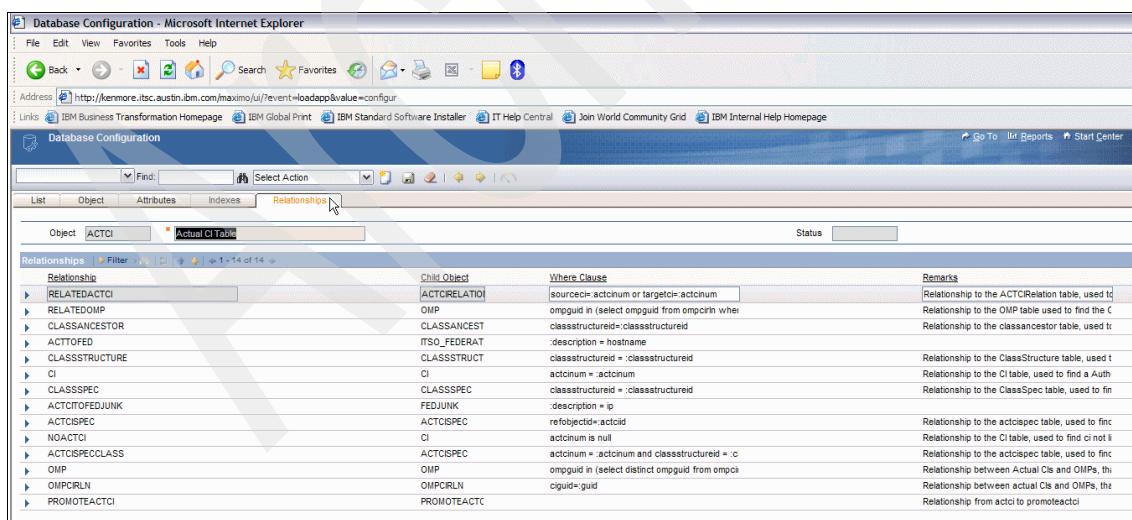


Figure 9-40 Relationships tab

- Click the **New Row** button in order to create a new relationship definition (Figure 9-41 on page 481).

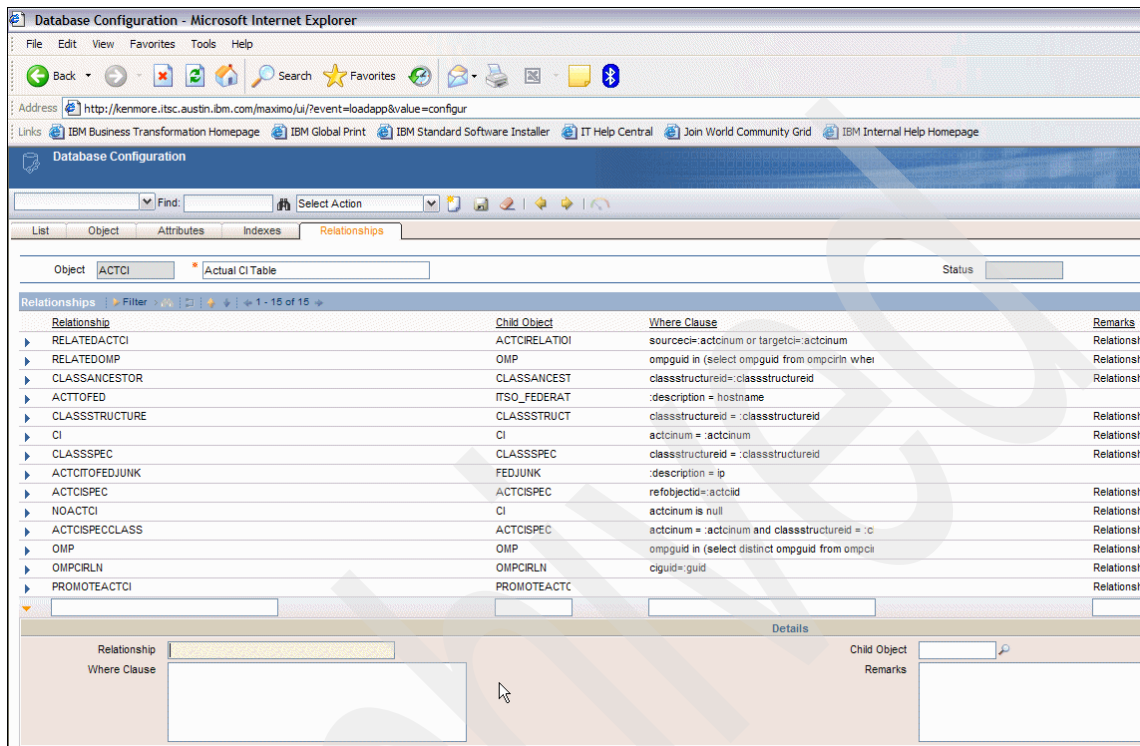


Figure 9-41 Creating a new row in the relationships definition

- Now create the relationship definition by linking the DESCRIPTION attribute of the ACTCI MBO to the HOSTNAME attribute of the ITSO_FEDERATION MBO (Figure 9-42).

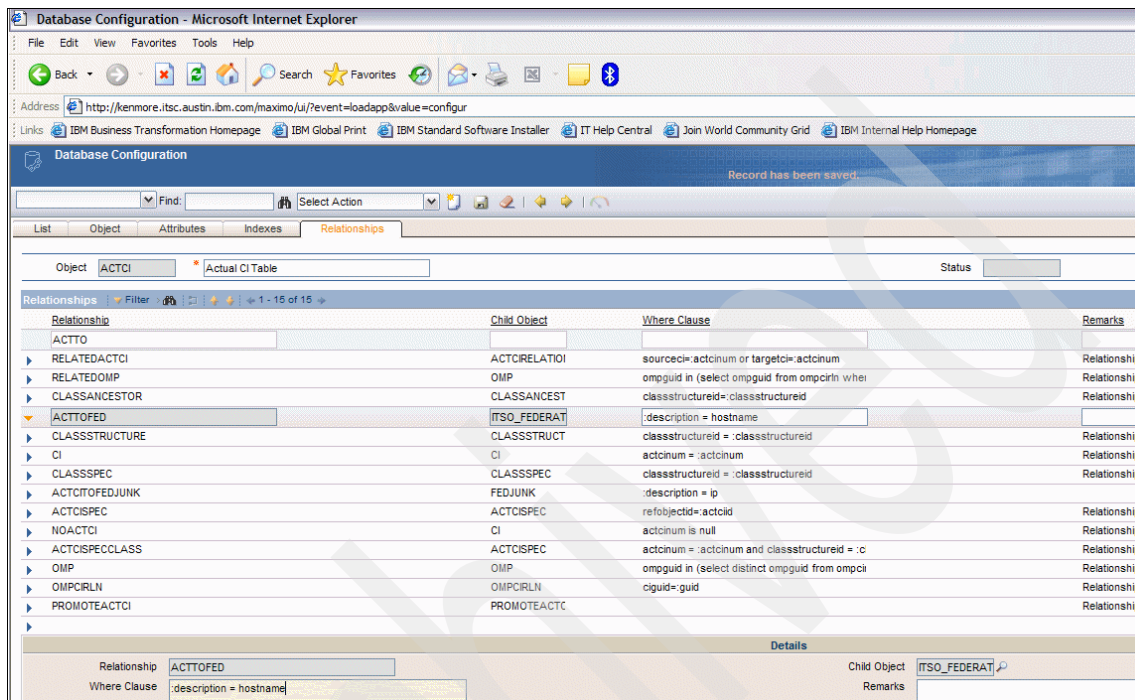


Figure 9-42 Creating the relationship

- Select **ITSO_FEDERATION** as the Child Object, and relate the description field to the host name attribute in the Where Clause text box. Adding the colon sign in front of the attribute (DESCRIPTION in our example) inside the Where Clause field relates the value of the description attribute in the current window when actually using the actual CI application to the host name attribute of the child object. This links the data of the two objects at runtime using a common denominator.

Do not forget to save your relationship definition by clicking the small diskette icon in the menu bar.

9.7 Creating a new application

You can now make use of the object definitions in the CCMDB application layer. Applications are what you link to in the Go To menu of the CCMDB Web User Interface.

In our example, we create a new application named ITSO actual configuration items by duplicating and modifying the standard actual configuration items application.

In order to create, duplicate, or modify an application, you have to use the Application Designer application:

1. In the CCMDB Web User Interface Go To menu, select **System Configuration → Platform Configuration → Application Designer** in order to launch the Application Designer application (Figure 9-43).

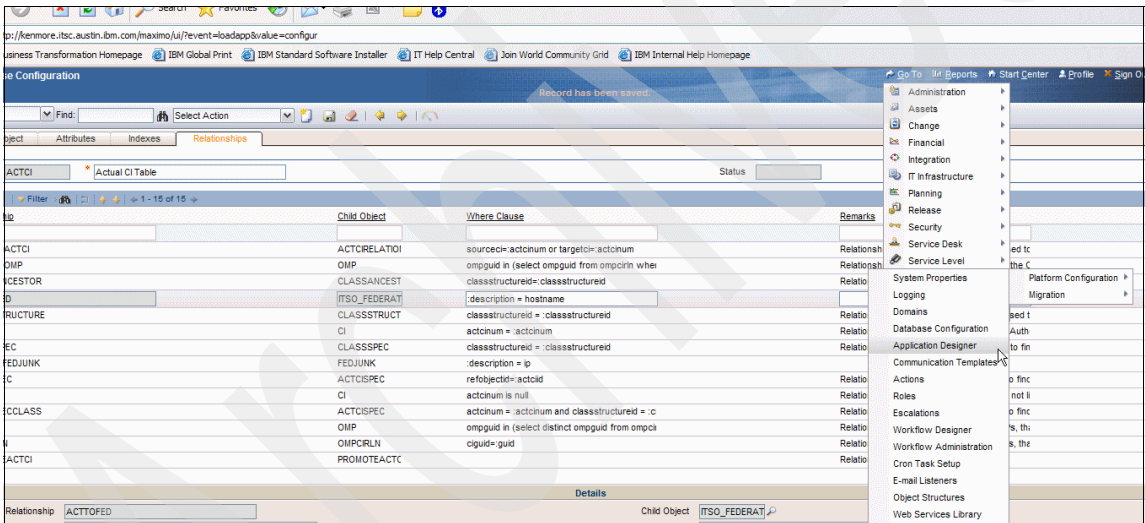


Figure 9-43 Launching the Application Designer

2. Search for the actual configuration items application (Figure 9-44).

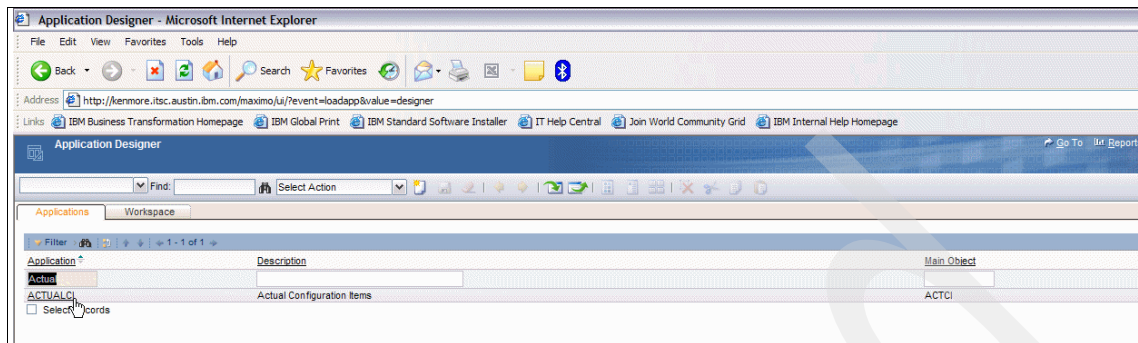


Figure 9-44 Searching for actual CI application

3. Click the link to select the **ACTUALCI** application and click the **Select Action** drop-down menu. Select **Duplicate Application Definition** from the drop-down menu (Figure 9-45).

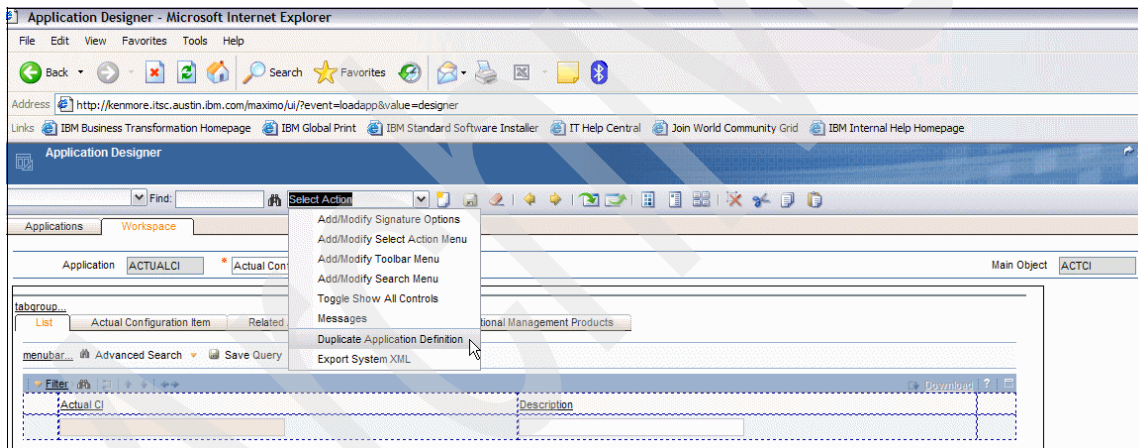


Figure 9-45 Selecting Duplicate Application Definition

This displays the **Duplicate Application** dialog (Figure 9-46).

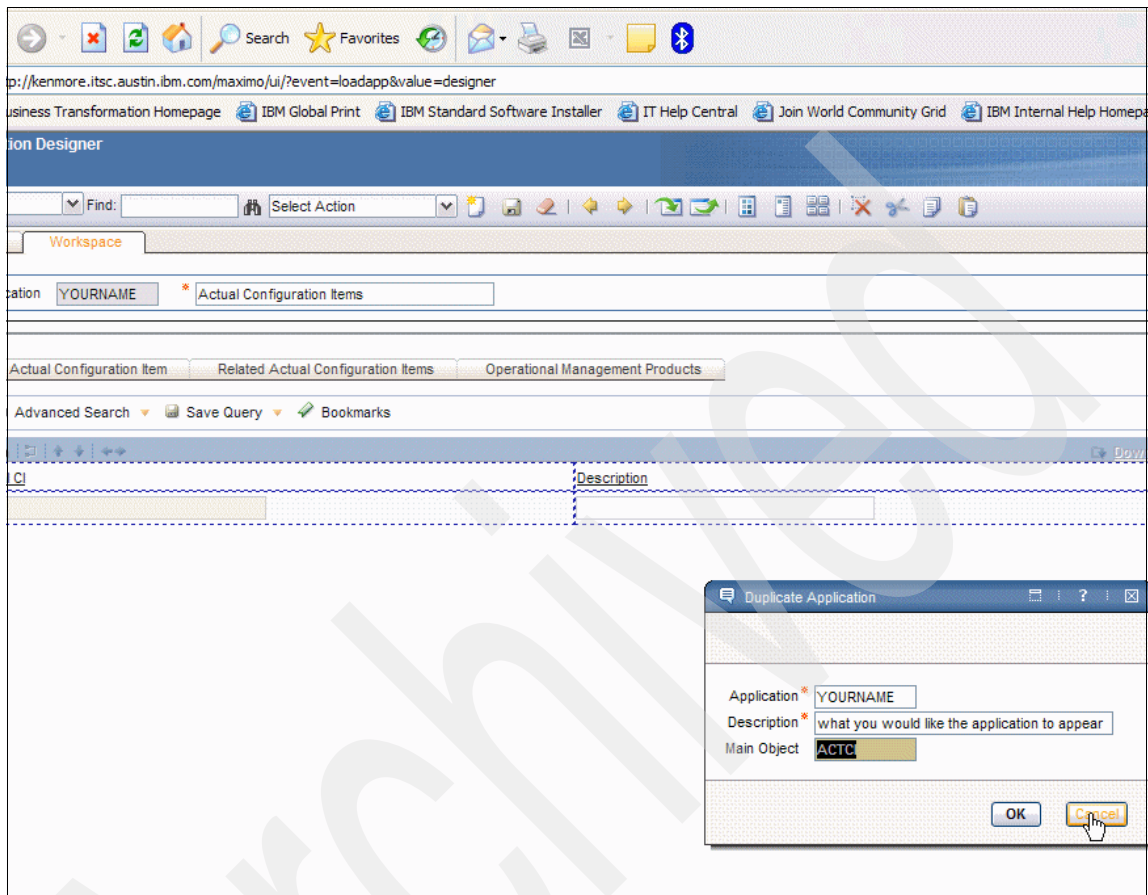


Figure 9-46 Duplicate Application dialog

4. Give your application a name and description. The description is what actually displays in the Go To menu. The Main Object field is already pre-populated because you duplicate an existing application.

In our example, we choose ITSO as the application name and ITSO actual configuration items as the description.

- Click **OK** and you see a window where you can start the modification of the user interface of your new application (Figure 9-47).

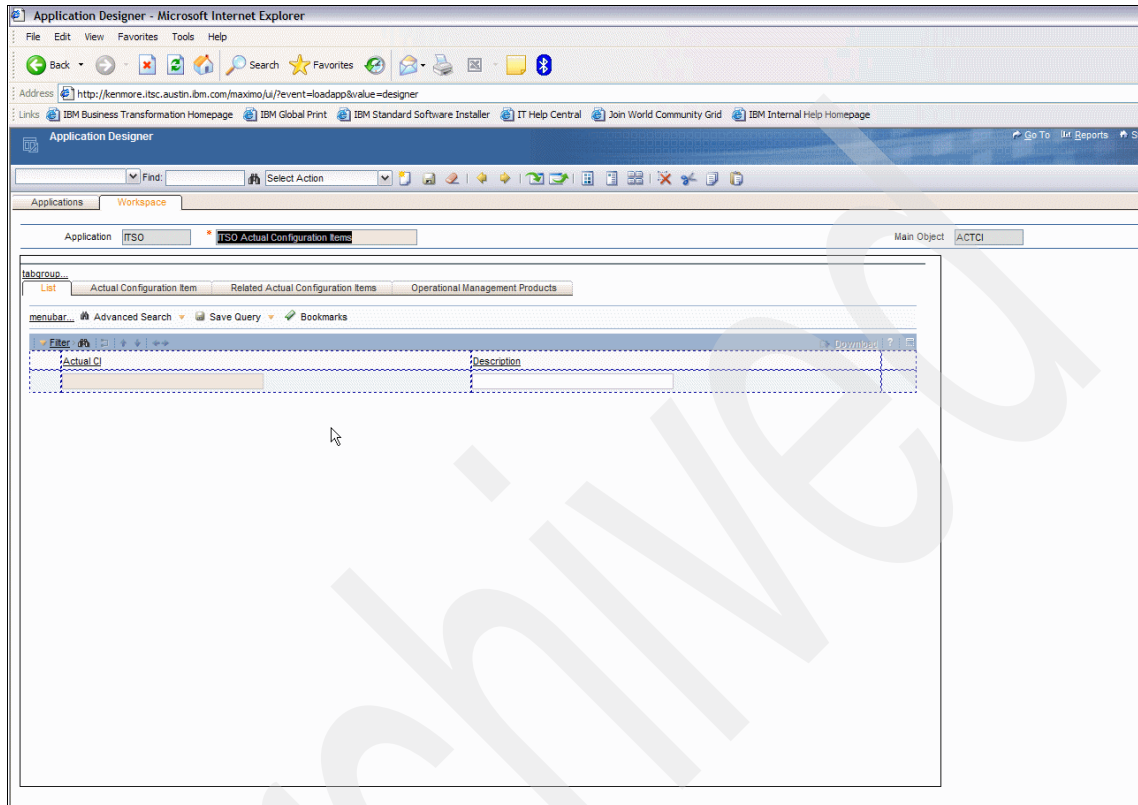


Figure 9-47 Modifying the user interface of the application

- Select the **Actual Configuration Item** tab in order to see the main window of the application. This is the window that we modify by adding additional attributes to present our federated data.
- Activate the Control toolbox by selecting the **Control Palette** icon from the menu bar. This is the icon right next to the icon with the green arrow pointing to the right. It displays the Control Palette.
- From the Control palette, drag and drop the Textbox control into the application section where you want to position your new field. This will add a new field to the window labelled Textbox... with an invalid binding to the database. This is because there is no association to an attribute in the database at this point in time.

9. Right-click the **Textbox** label, this displays a properties dialog (Figure 9-48).

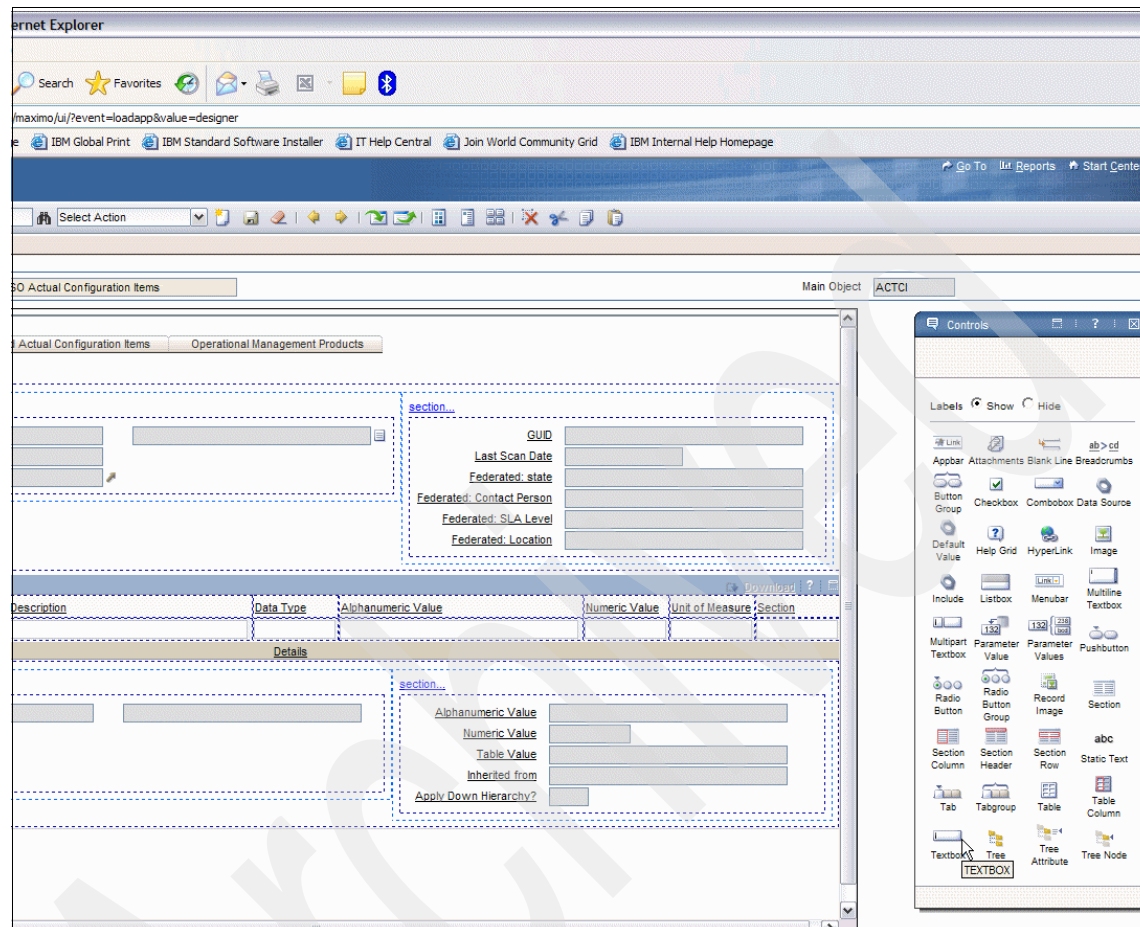


Figure 9-48 Properties dialog selection

10. Select **Properties** to display the dialog where you can define specifications for this field, for example, to resolve the binding to the database (Figure 9-49).

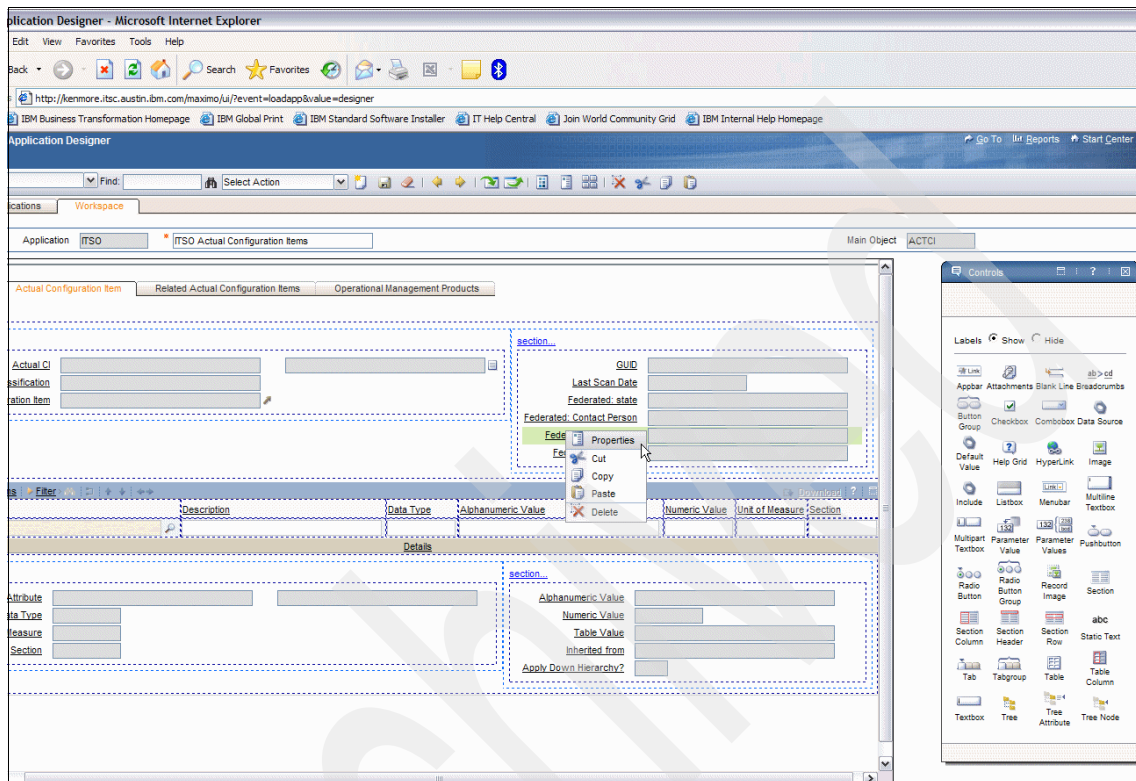


Figure 9-49 Properties dialog

In the Label specification, specify a name for the field as you want to present it to the users of the application. In Figure 9-49, we label the field Federated: SLA Level.

The most important specification in this dialog is to specify the binding to the database using the Attribute specification field. In our example, we have to specify the relationship that we defined in the previous step followed by the name of the attribute of the federated database. This allows the system to calculate if the relationship is true and, if it is true, show the attribute of the federated data source that we specify as SLA LEVEL in our example.

ACTOFED is the name of the relationship while SLA LEVEL is the attribute name pointing to the column of the federated database. Concatenate both by adding a period (.) between them and tab out the field.

11. Close the Textbox properties dialog and close the Control Palette toolbox if you added all the fields that you want to see in the window of your application. We added four attributes in total using the same approach as previously described (Figure 9-50).

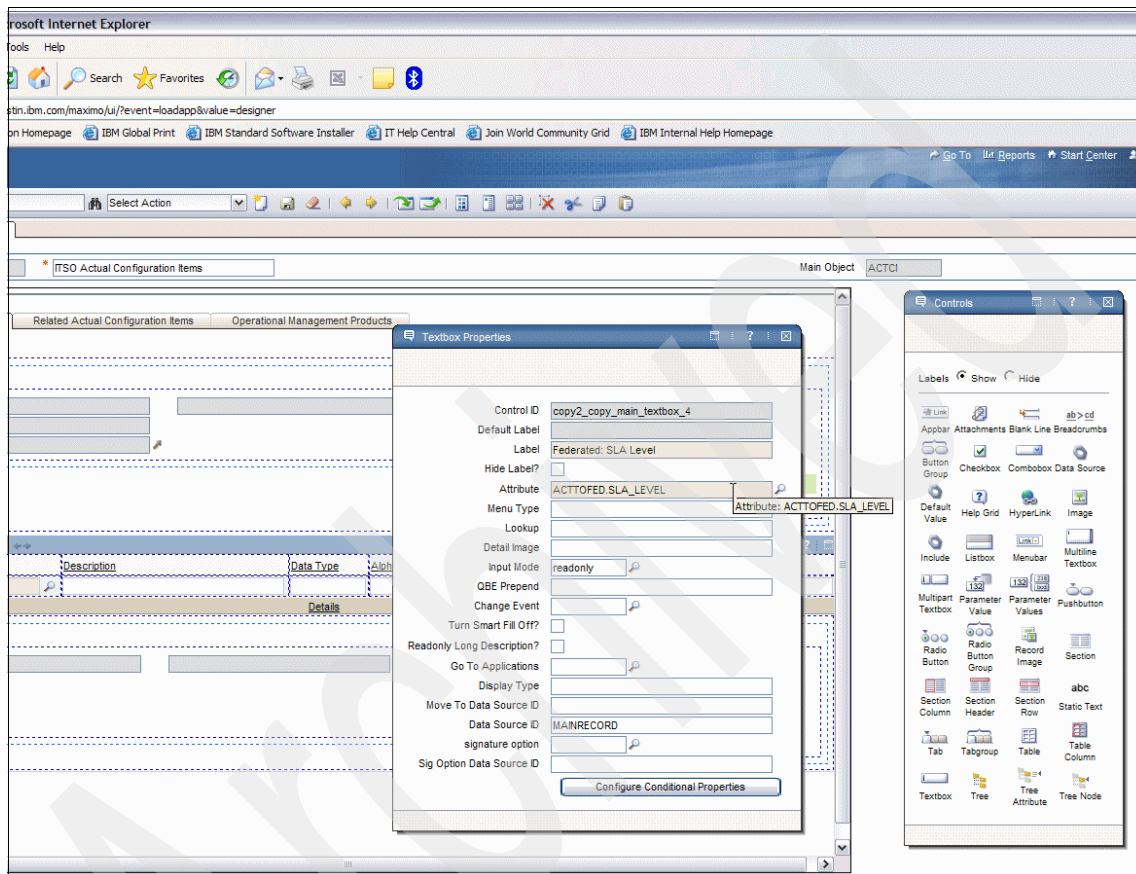


Figure 9-50 Added attributes

You see that we have added the following attributes:

- ▶ Federated: State
- ▶ Federated: Contact Person
- ▶ Federated SLA Level
- ▶ Federated: Location

In the Textbox properties definition dialog, you have to specify the ACTTOFED relationship concatenated with the appropriate attribute pointer in order to define the binding to the database.

We do not use the *hostname* attribute itself outside the relationship definition, because there is already a field (description) that holds the host name of the actual CI.

Important: If you still see an invalid binding in your textbox definition, there is something wrong with your relationship definition, which you need to check.

You are now ready to use your new application. Your configuration setup is completed.

9.8 Using the new application

We launch our application by using the CCMDB Web User Interface Go To menu:

1. Select **IT Infrastructure** → **ITSO Actual Configuration Items** application to launch the ITSO application (Figure 9-51).

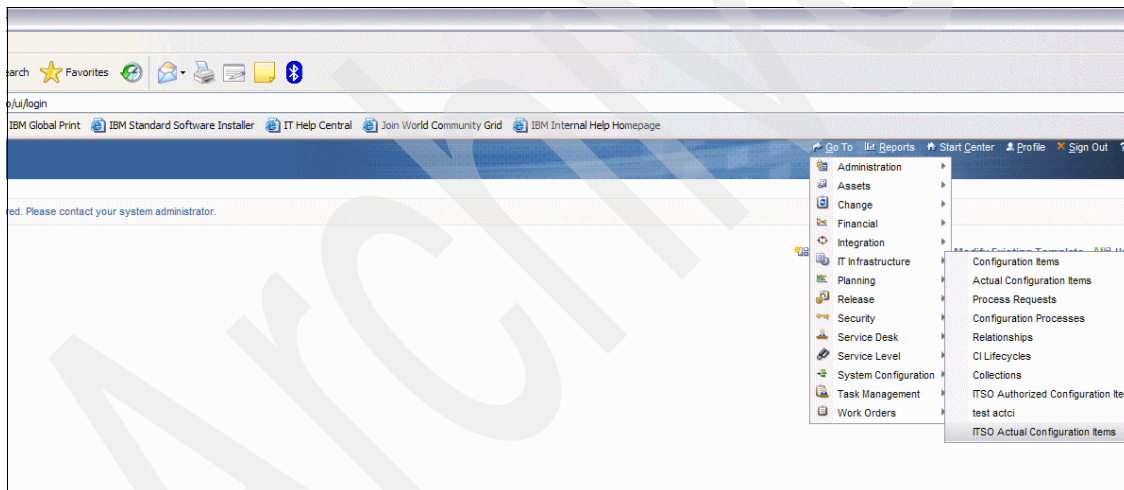


Figure 9-51 Starting the application

Because our new application is a copy of the actual configuration items application, the primary object is the ACTCI object, in order to see all the actual CI data in our system (Figure 9-52).

Attention: The system that we use for our federation example does not contain detailed data. We just used a minimal set of attributes in this environment.

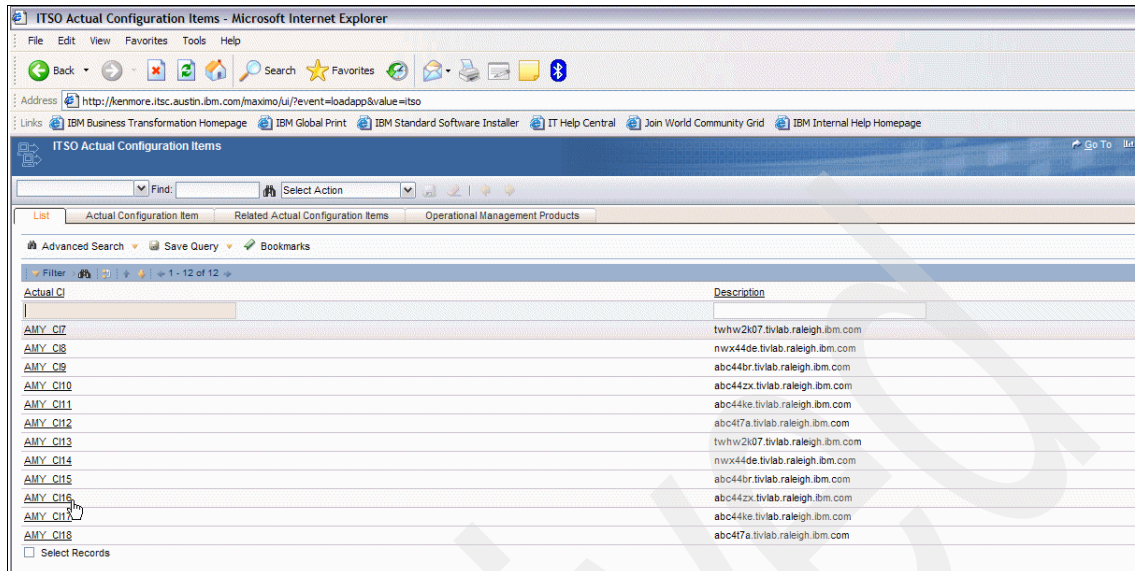


Figure 9-52 Viewing list of actual CI data

2. We select the link for the actual CI labeled AMY_CI16 because we have an entry in our federated database table FED_DATA that has the same entry for the host name to resolve the relationship.

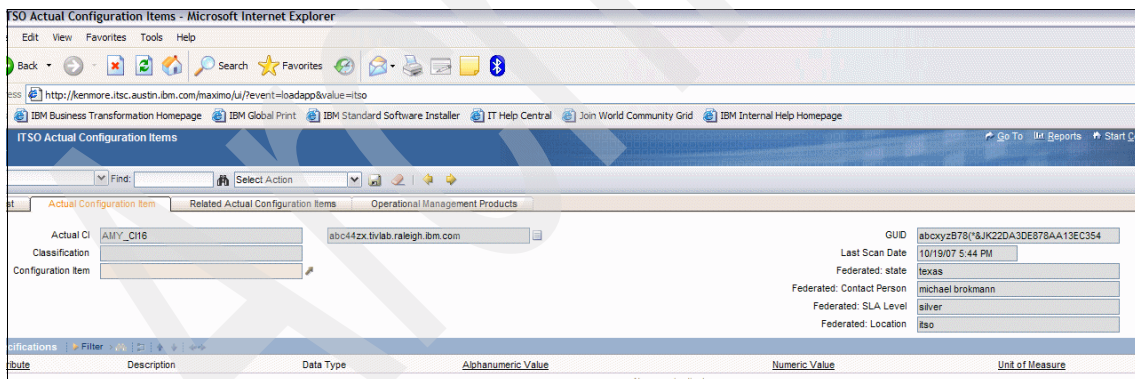


Figure 9-53 Viewing details contained in a federated data source

Based on the CI record selection, the relationship gets resolved and data from our federated database table shows up in the fields that we added to our new application.

This data is resolved at runtime without needing to import it physically into the CCMDDB database.

3. In order to verify this situation, we change some of our entries locally in the FED_DATA table using the DB2 Control Center. We changed the entries for the SLA Level and the Contact Person (Figure 9-54 on page 492).

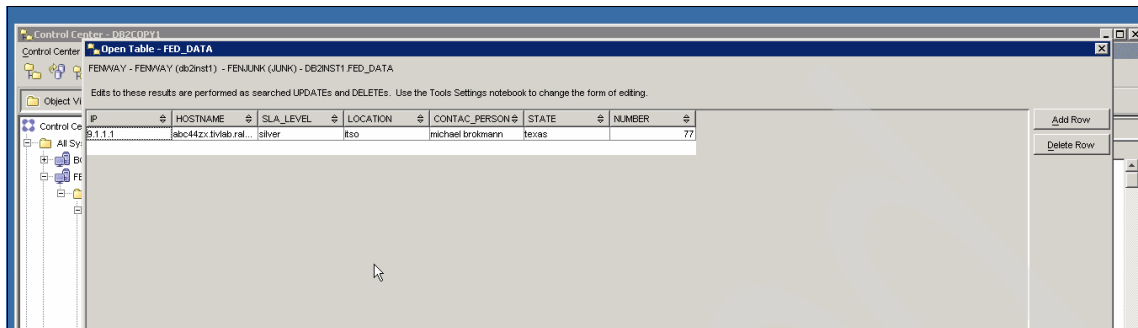


Figure 9-54 Changing data in DB2 Control Center

After changing the data in the federated database, the modified data shows up in the application user interface.

Read-only fields: The additional fields have a grey color because they have been defined as read-only.

This concludes our example of demonstrating how to set up a federated data source and expose the data to a CCMDB application in the process runtime environment.

9.9 Summary

This chapter provides a walkthrough of the steps required to utilize federated data sources and modify the application user interface to display that data. In many environments, federation will be an important capability. Modifying the user interface to take into account various new data items is quite simple and straightforward.



Part 4

Miscellaneous CCMDB topics

In this part of the book, we cover miscellaneous CCMDB topics, such as the new Launch-in-Context technology, performance tuning, and virtualization support.

Virtualization support in CCMDB

CCMDB enables you to effectively manage hosts and virtual machines to deliver efficient and effective services aligned to your business goals.

In this chapter, before delving into the details of how CCMDB works with virtualized CIs in Change workflows using Information Technology Infrastructure Library (ITIL) V3, we provide an overview of virtualization.

We cover the following topics:

- ▶ 10.1, “Virtualization overview” on page 496
- ▶ 10.2, “Virtualization scenarios” on page 497
- ▶ 10.3, “Information Technology Infrastructure Library (ITIL) V3 workflow” on page 504

10.1 Virtualization overview

All physical resources are shared by virtual machines, generating a multiplier effect of resources. You can think of virtualization as a multiplier of resources. It makes the components of the server available for multiple virtual servers rather than just one physical server as usual.

Virtualization lets you share the resources in an accurate and intelligent way among the servers. CPU resources can be shared among multiple virtual machines, ensuring an increase in server utilization, giving each virtual machine access to host processors available. Virtualization uses intelligent process scheduling and load balancing among the available processors. You can set the minimum and maximum number of CPUs that each virtual machine can use, guaranteeing a percentage of CPU resources, with or without restraint.

Virtualization also allows over-allocated safely memory for servers, meaning that the total of the allocated memory for all running virtual machines can exceed physical memory installed on the host. This lets you make the most of your investments in memory for servers without additional costs.

The storage network components can be shared across multiple servers while maintaining the fault tolerance of hardware. Instead of devoting two host bus adapters (HBAs) and Fibre Channel switches a portion of storage to each server, as is necessary in conventional server provisioning, virtualization shares these storage HBAs and switches between multiple virtual machines, maintaining fault tolerance and at the same time reducing the cost of access to storage per server.

Network resources are multiplied in a similar way. If you invested in NICs combined for hardware fault tolerance, each virtual machine running on the host can share these high availability features. Virtual machines can use virtual adapters and VLANs to reduce numbers of NICs and cables needed.

Key features of virtualization: The following new features are provided:

- ▶ **Partitioning:** Simultaneous execution of multiple virtual machines on a single physical server is provided.
- ▶ **Isolation:** Each virtual machine is isolated from the others on the same server.
- ▶ **Encapsulation:** Virtual machines encapsulate entire systems (hardware configuration, operating system, and applications) on files.
- ▶ **Hardware independence:** It is possible to run a virtual machine on any server without modification.

10.2 Virtualization scenarios

The Virtualization function for CCMDB 7.2.1 provides automatic enforcement of Change policies based on the priority and risk of the Changes.

The server business impact is analyzed to determine the overall risk of the Change, and that value is automatically determined. Risk is used to determine the required authorization level and for authorization routing. From this, the Change process appropriate level can be implemented to maximize efficiency and security. This is especially important for support of the dynamic environment of virtual machines, from which the name of this function originates.

We go through four proposed user scenarios to illustrate this function:

► First proposed scenario:

There is a Change Requestor that needs to have the Change implementation team move our local Development VM to another VM host. Because this is our personal DEVVM, the business impact is low, so the Change process is detected to be a Standard Automated Change and the process short-circuits.

► Second proposed scenario:

There is a Change Requestor that needs to have the Change implementation team move our Production VM to another VM host. Because a critical business application runs on this VM, it has a higher risk, so the normal Change process is employed.

► Third proposed scenario:

There is a Change Requestor that needs to have the antivirus updated on our local laptop. The business impact of this is low, and the urgency is low. This is determined to be a standard Change and shortens the process.

► Fourth proposed scenario:

There is a new worm spreading throughout the industry and only the most recent version of our antivirus software will detect it. For this, the impact and urgency is critical. Because the urgency is critical, this is determined to be an Emergency Change, so the scheduling step is skipped.

10.2.1 First proposed scenario

There is a Change Requestor that needs to have the Change implementation team move our local Development VM to another VM host. Because this is our personal DEVVM, the business impact is low, so the Change process is detected to be a Standard Automated Change and the process short-circuits. We have created a new Change using the information in Table 10-1 and Figure 10-1.

Table 10-1 First proposed scenario

Field	Information
Summary	This is a test to validate the first case regarding the CCMDB 7.2.1 Redbooks publication.
Classification	PMCHG \ PMCHGSFW \ PMCHG_VMMOVE
Configuration Item	DEVVM
Urgency	4
Impact	4
Change Type	Normal

The screenshot displays the 'Change Details' form. The 'Summary' field contains the text 'This is a test to validate the first case regarding'. The 'Priority' is set to 4, 'Urgency' to 4, and 'Impact' to 4. The 'Change Type' is 'Standard'. The 'Classification' is 'PMCHG \ PMCHGSFW \ PMCHG_VMMOVE' and the 'Class Description' is 'Move a Virtual Machine to a different server'. The 'Primary Targets' section shows 'Configuration Item: DEVVM', 'Configuration Item Name: DEVVM', and 'Target Description: This is a developer VM'. The 'Asset', 'Location', and 'Asset/Location Priority' fields are also visible.

Figure 10-1 Change opened according to information listed on 1st proposed scenario

We submit this Change using the Information Technology Infrastructure Library (ITIL) V3 workflow. The workflow used by ITIL V3 is PMCHGCALPR.

Notice now that the Progress Map has Changed and the Change Type was automatically set to Standard for this Change and it was assigned a Priority of 4, Low.

Because the automation has determined that this is a standard automated Change and the process is short-circuited, authorization is automatic and the Change now has a status of Complete (Figure 10-2).



Figure 10-2 Progress Map on 1st proposed scenario after be applied ITIL V3 workflow

10.2.2 Second proposed scenario

There is a Change Requestor that needs to have the Change implementation team move our Production VM to another VM host. Because a critical business application runs on this VM, it has a higher risk, so the normal Change process is employed.

We have created a new Change using the information in Table 10-2 and Figure 10-3.

Table 10-2 Second proposed scenario

Field	Information
Summary	This is a test to validate the second scenario regarding the CCMDB 7.2.1 Redbooks publication.
Classification	PMCHG \ PMCHGSFW \ PMCHG_VMMOVE
Configuration Item	PRODVM
Urgency	4
Impact	4
Change Type	Normal

Change Details

Summary: Priority: Change Type:
Reason for Change: Priority Justification: Risk:
Effect Of Not Implementing: Urgency: Change Category:
Impact: Verification:

Classification: >> Class Description:

Specifications Filter >

Attribute	Description	Data Type	Alphabetic Value	Numeric Value	Unit of Measure	Table Value
...No rows to display...						

Primary Targets

Configuration Item: >> Asset: >>
Configuration Item Name: Location: >>
CI Business Impact: Asset Location Priority:
Target Description:

Figure 10-3 Change opened according to information listed on 2nd proposed scenario

We submit this Change using the Information Technology Infrastructure Library (ITIL) V3 workflow.

The Normal Change process was applied to this Change. Even though a Priority of 4 was assigned, a Risk of 1 was determined (Figure 10-4).

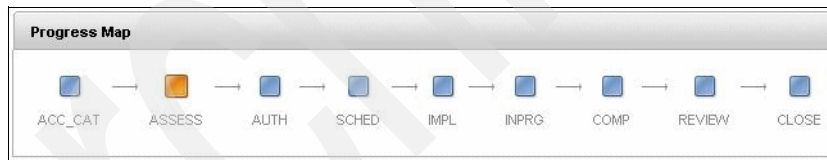


Figure 10-4 Progress Map on second proposed scenario after be applied ITIL V3 workflow

10.2.3 Third proposed scenario

There is a Change Requestor that needs to have the antivirus updated on our local laptop. The business impact of this is low, and the urgency is low. This is determined to be a standard Change and shortens the process.

We have created a new Change using the information in Table 10-3 and Figure 10-5.

Table 10-3 Third proposed scenario

Field	Information
Summary	This is a test to validate the third case regarding the CCMDB 7.2.1 Redbooks publication.
Classification	PMCHG/PMCHGSFW/PMCHG_AVUPDATE
Configuration Item	LOCALLAPTOP
Urgency	4
Impact	4
Change Type	Normal

Change Details

Summary: This is a test to validate the third case regarding the CCMDB 7.2.1 Redbooks publication.

Reason for Change:

Effect Of Not Implementing:

Priority: 4

Priority Justification:

Urgency: 4

Impact: 4

Change Type: Standard

Risk: 5

Change Category:

Verification:

Classification: PMCHG\PMCHGSFW\PMCHG_AVUPDATE

Class Description: Update Antivirus Version

Specifications

Attribute	Description	Data Type	Alphanumeric Value	Numeric Value	Unit of Measure	Table Value
No rows to display...						

Primary Targets

Configuration Item: LOCALLAPTOP

Configuration Item Name: LOCALLAPTOP

Business Impact: 5

Target Description:

Asset:

Location:

Asset Location Priority:

Figure 10-5 Change opened according to information listed on 3rd proposed scenario

We submit this Change using the Information Technology Infrastructure Library (ITIL) V3 workflow.

Note that the Progress Map again is unique for this Change. It was determined to be a standard Change, but with a Risk of 5 and a Priority of 4, this Change is assigned to the Standard Change processing (Figure 10-6)..

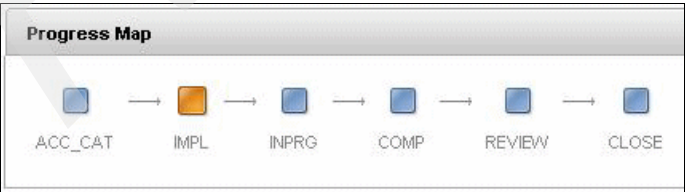


Figure 10-6 Progress Map on 3rd proposed scenario after be applied ITIL V3 workflow

10.2.4 Fourth proposed scenario

There is a new worm spreading throughout the industry and only most recent version of our antivirus software will detect it. For this, the impact and urgency is critical. Because the urgency is critical, this is determined to be an Emergency Change, so the scheduling step is skipped.

We have created a new Change using the information in Table 10-4 and Figure 10-7.

Table 10-4 Fourth proposed scenario

Field	Information
Summary	This is a test to validate the fourth case regarding the CCMDB 7.2.1 Redbooks publication.
Classification	PMCHG/PMCHGSFW/PMCHG_AVUPDATE
Configuration Item	LOCALLAPTOP
Urgency	1
Impact	1
Change Type	Normal

The screenshot displays the 'Change Details' form in the IBM Tivoli Change and Configuration Management Database (CCMDB) interface. The form is organized into several sections:

- Summary:** Contains a text field with the value 'This is a test to validate the 4th case regarding the CCMDB 7.2.1 Redbooks publication.' and a search icon.
- Priority:** A dropdown menu set to '1'.
- Change Type:** A dropdown menu set to 'Emergency'.
- Reason for Change:** An empty text field.
- Priority Justification:** An empty text field.
- Risk:** A dropdown menu set to '5'.
- Effect Of Not Implementing:** An empty text field.
- Urgency:** A dropdown menu set to '1'.
- Change Category:** An empty text field.
- Impact:** A dropdown menu set to '1'.
- Verification:** An empty text field.
- Classification:** A dropdown menu showing 'PMCHG/PMCHGSFW/PMCHG_AVUPDATE'.
- Class Description:** A dropdown menu showing 'Update Antivirus Version'.
- Specifications:** A table with columns: Attribute, Description, Data Type, Alphanumeric Value, Numeric Value, Unit of Measure, and Table Value. The table is currently empty, showing 'No rows to display...'. A 'New Row' button is located at the bottom right of the table.
- Primary Targets:** A section containing several fields:
 - Configuration Item:** A dropdown menu showing 'LOCALLAPTOP'.
 - Configuration Item Name:** A text field showing 'LOCALLAPTOP'.
 - CI Business Impact:** A dropdown menu set to '5'.
 - Target Description:** An empty text field.
 - Asset:** A dropdown menu.
 - Location:** A dropdown menu.
 - Asset/Location Priority:** A dropdown menu.

Figure 10-7 Change opened according to information listed on 4th proposed scenario

We submit this Change using the Information Technology Infrastructure Library (ITIL) V3 workflow.

Note that the Progress Map again is unique for this Change.

Because of the criticality of this Change, it was determined to be an Emergency Change and assigned a Priority of 1 (Urgent), and will flow through another authorization process (Figure 10-8).

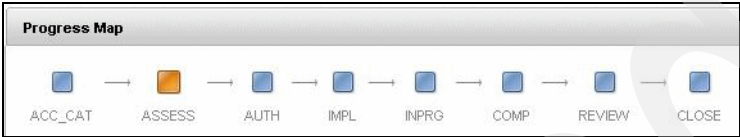


Figure 10-8 Progress Map on fourth proposed scenario after be applied ITIL V3 workflow

After submitting the proposed scenarios, the Change will be listed on the IBM Tivoli Change and Configuration Management Database (CCMDB) console (Figure 10-9).

Change	Summary	Status	Risk	Priority
	CCMDB 7.2.1 Redbook			
1085	This is a test to validate the 1st case regarding CCMDB 7.2.1 Redbook	COMP		4
1088	This is a test to validate the 2nd case regarding CCMDB 7.2.1 Redbook	ASSESS	1	4
1090	This is a test to validate the 3rd case regarding CCMDB 7.2.1 Redbook	IMPL	5	4
1092	This is a test to validate the 4th case regarding CCMDB 7.2.1 Redbook	ASSESS	5	1
☐ Select Records				

Figure 10-9 Change number, Summary, Change status, Risk and Priority on CCMDB console

10.3 Information Technology Infrastructure Library (ITIL) V3 workflow

According to values input in the fields Urgency and Impact, the ITIL V3 workflow automatically calculates the Change Priority from the information in Table 10-5.

Table 10-5 Priority Matrix

Urgency	Impact	Priority
1	1	1
1	2	1
2	1	1
2	2	1
1	3	2
1	4	2
2	3	2
3	1	2
3	2	2
1	5	3
5	1	3
4	2	3
3	3	3
2	4	3
5	2	4
4	3	4
5	3	4
3	4	4
4	4	4
2	5	4
3	5	4
4	5	4
Other	Other	5

Another field input automatically will be Risk (workflow PMCHGCALRS), based on the field values Impact and Failure Probability, according to the information listed in Table 10-6.

Table 10-6 Risk Matrix

Impact	Failure Probability	Risk
1	1	1
2	1	1
1	2	2
1	3	2
2	2	2
2	3	2
3	1	3
4	1	3
3	2	4
3	3	4
4	2	4
4	3	4

Launch In Context

In this chapter, we explain how CCMDB V7.2.1 enables a new Launch In Context (LIC) technology, which addresses certain requirements that customers face today with the existing technology. The current LIC technology is still available in CCMDB V7.2.1 along with the new technology.

We cover the following topics:

- ▶ 11.1, “Purpose of new Launch In Context” on page 508
- ▶ 11.2, “Components of the new LIC technology” on page 509
- ▶ 11.3, “Naming and Reconciliation Service” on page 510
- ▶ 11.4, “Deployment and configuration of DIS/CMS” on page 512
- ▶ 11.5, “CMS Registry Loader externals” on page 517

11.1 Purpose of new Launch In Context

Launch In Context (LIC) allows integration of various Tivoli products, increasing operational productivity and reducing the cost of ownership of these products. Although LIC is a powerful feature with the current implementation, the products must create product to product agreement for launching URLs, so with every change or new releases, LIC must be synchronized and revisited. In addition, the launches must be manually configured.

The new LIC technology provides the following benefits:

- ▶ Better utilization of the existing products: You can use multiple Tivoli products in their environment, each providing information. Using the new technology, it gives a comprehensive view for decision making that can be used to solve issues, thus providing more value for investments.
- ▶ Fast time to value: You do not have to configure launch points manually. Also, they do not have to go to each product to view the information related to a component in the environment. The new LIC technology provides a single repository from where each product can fetch all information quickly without any configurations.
- ▶ Navigation across products using launch points: It enables a common cross product registration mechanism for launch points thus providing integration point for all CI related products. This enhances the operational experience across the Tivoli products, each having information related to the CI.

Many integration scenarios contain the need for Launch In Context; for example:

- ▶ Business Service Impact analysis by Incident and Availability Management
- ▶ Change Lifecycle Management
- ▶ IBM Tivoli Monitoring with Systems Director Monitoring
- ▶ IBM Tivoli Network Manager for IP Edition with Systems Director Monitoring

To explain this idea, let us take the example of a Link Failure at Edge of Network:

1. Service Operator analyzes the event and evaluates service impact in Tivoli Business Service Manager.
2. Service Operator validates the problem in IBM Tivoli Monitoring.
3. Service Operator opens a ticket and assigns to Network Operator.
4. Network Operator views network event in Omnibus.
5. Network Operator views network topology in IBM Tivoli Network Manager.
6. Network Operator assigns the ticket to Network Subject Matter Expert (SME).
7. SME views network data in IBM Tivoli Network Manager for IP Edition.

As shown in Figure 11-1, by using the new LIC technology, you can seamlessly navigate across various products, instead of going to each launch entry of the product and navigate.

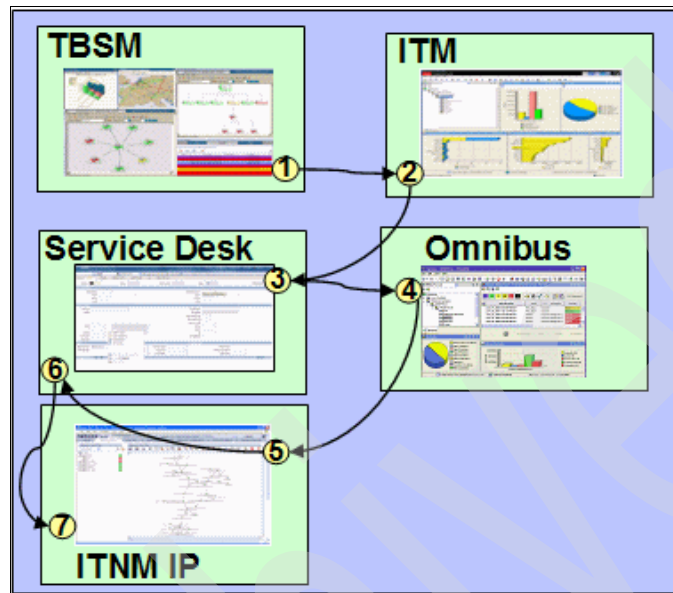


Figure 11-1 Navigation across products

LIC technology: IBM Director, TADDM, and CCMDB are initial adopters of the new LIC technology. The products explained in the demonstration are yet to adopt this technology. The example is given only to show users how the LIC can work with the new technology.

11.2 Components of the new LIC technology

The following components are involved in the new LIC technology:

- ▶ Data Integration Services (DIS): The component that handles multiple naming contexts for resources through enhanced reconciliation with a centralized database of resources and all their naming contexts and naming attributes
- ▶ Context Menu Services (CMS): The component that manages a centralized registry of Launch In Context entries and can build the context menus for resources using the naming information in DIS

- Operational Management Products (OMP): The components that register their naming context for resources they manage with DIS and use the CMS to build context menus for those resources in the OMP user interfaces

Figure 11-2 illustrates these key components used in the new LIC technology.

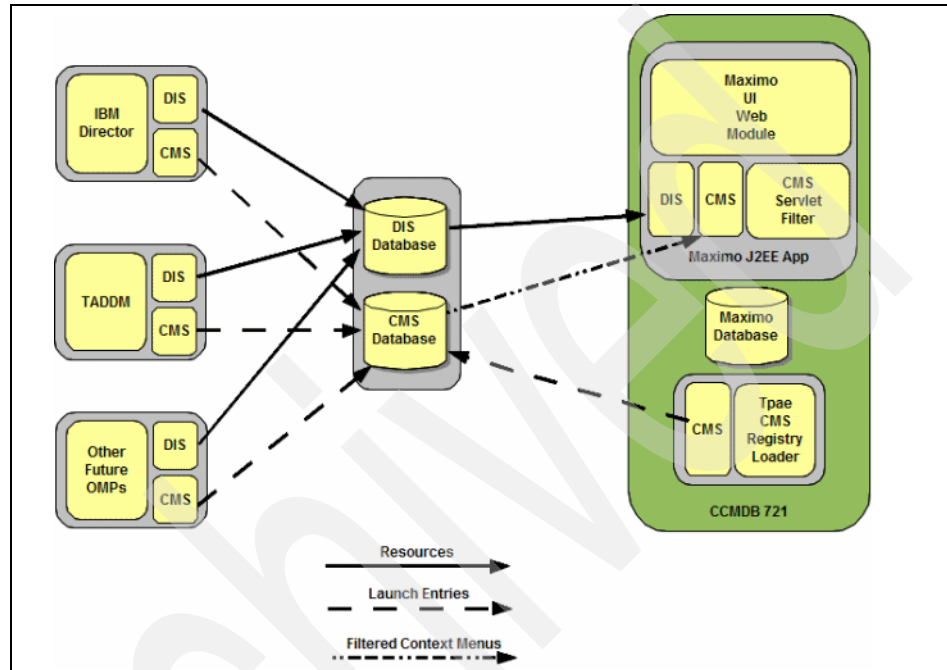


Figure 11-2 LIC components

Launch In Context: IBM Systems Director and TADDM are called out as specific participants with CCMDDB in Launch In Context, but by instrumenting to use CMS and DIS, CCMDDB is positioned to support launch from and to any future adopter of DIS and CMS.

11.3 Naming and Reconciliation Service

Naming and Reconciliation Service helps OMPs get and set attributes that CMS uses. It provides a common “CDM aware” data source for all OMPs and CMS to use. It also provides basis for common language across products.

11.3.1 Reconciliation functions

An OMP stores the attributes it knows about a resource by DIS. CMS uses DIS to look up the attributes, that is, Resource Context Filters (Figure 11-3).

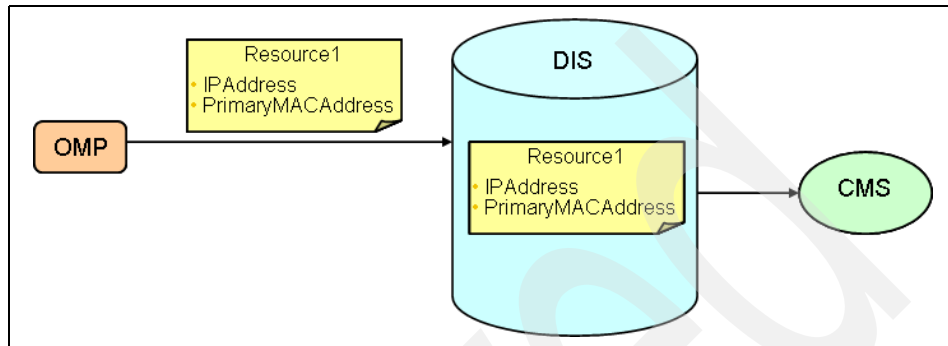


Figure 11-3 Naming

Reconciliation is a fundamental and critical function of DIS. DIS can “reconcile” resource information provided by multiple OMPs. DIS identifies a resource as being the same when reported by two unique OMPs. Reconciliation then merges the attributes provided by multiple OMPs (Figure 11-4).

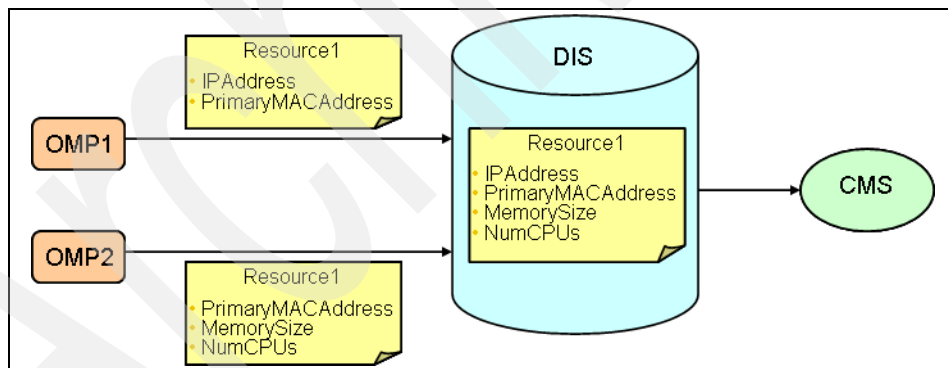


Figure 11-4 Reconciliation

11.3.2 Context Menu Services

The use of CMS decouples OMPs from each other for Launch In Context. As additional OMPs are instrumented to use CMS to build their context menus, these OMPs are able to launch into the CCMDB CI applications with no updates required to CCMDB and with no product-to-product interlock between the launching OMP and the OMP into which the launch into is performed (CCMDB).

The use of CMS allows the launching products to determine launch points on the fly (at run-time). When user selects the launch point, the other product UI starts up with context (or previously started UI gets new context) which is dynamic without hard wiring. The launched product can provide integrations without waiting for synchronization with launching products release schedule.

The use of CMS in CCMDB V7.2.1 provides the following capabilities:

- ▶ Ability to launch into the CCMDB CI Applications from any OMP that adopts DIS and CMS
- ▶ Ability to launch from the CCMDB CI Applications into any OMP that adopts DIS and CMS

11.4 Deployment and configuration of DIS/CMS

In this section we discuss deployment and configuration of DIS/CMS:

11.4.1 Deployment components

The following list describes the components for this integration:

- ▶ The *TADDM DIS/CMS Integration Package* is a TADDM Interim Fix Package that can be used to do the following functions:
 - Create the database tables required for DIS and CMS
 - Import and register resources from TADDM into DIS
 - Register CMS Launch Entries for launching into TADDM from other OMPs such as CCMDB
- ▶ The *Base Services Installer* provides functions for configuring the Tivoli's process automation engine environment to integrate with DIS and CMS and optionally for creating the DIS and CMS tables in the Maximo Database.
- ▶ The *CI Applications CMS Integration PSI Package* configures the CCMDB CI Applications to support:
 - Launching out from the CI Applications to other OMPs (such as TADDM) that have registered launch entries with CMS
 - Launching into the CI Applications from other OMPs

11.4.2 Options for creating the CMS and DIS tables

The DIS and CMS tables are shared resources that multiple Tivoli products can integrate with. Based on the Tivoli products that you have deployed as well as the installation sequence of those products, you have three options to deploy the DIS and CMS:

1. The DIS and CMS tables have already been set up. In this case, configure the Base Services Installer and the TADDM DIS/CMS Integration Package to use the already-created tables.
2. Use the Base Services Installer to create the DIS and CMS tables in the Maximo Database along with the other CCMDB and Tivoli's process automation engine tables. In this option, configure the TADDM DIS/CMS Integration Package to load resources and launch entries into those tables in the Maximo Database.
3. Use scripts in the TADDM DIS/CMS Integration Package to create the tables. In this option, use the Base Services Installer to configure the Tivoli's process automation engine environment to use the tables set up by TADDM.

CCMDB and TADDM deployment considerations:

- ▶ DIS and CMS tables must be in the same database. TADDM does not support an environment where the DIS and CMS tables are created in separate databases. CCMDB support for CMS-based Launch In Context depends on TADDM registering resources in DIS. Therefore, you must not use the Base Services Installer functions that allow Tivoli's process automation engine environment to be configured to connect to DIS and CMS in various databases. On the panel titled "Data Integration Service Database Information", you must always select the check box titled "Use the same database for both data integration and context menu services".
- ▶ CCMDB and TADDM do not support CMS Launch In Context using SQL Server. TADDM does not support SQL Server. CCMDB support for CMS-based Launch In Context depends on TADDM registering resources in DIS. Therefore, you must use DB2 or Oracle as the databases for the DIS and CMS tables. You cannot select the Base Services Installer option titled "Deploy data integration and context menu services into the same database that you created for the product" on the panel titled "Data Integration Service and Context Menu Service Deployment Options" if you have configured the Maximo Database to use SQL Server.

11.4.3 Deployment scenarios

Users performing a base install of CCMDB V7.2.1 or upgrading an existing CCMDB version to CCMDB V7.2.1 have various options related to DIS/CMS deployment:

- ▶ Set up DIS and CMS tables in the database used by CCMDB and configure Tivoli's process automation engine to use those tables.
- ▶ Configure Tivoli's process automation engine to use a previously deployed DIS and CMS.
- ▶ Defer or skip DIS and CMS setup and configuration.
- ▶ Set up DIS/CMS and/or configure Tivoli's process automation engine for DIS and CMS after installing or upgrading to V7.2.1. After installing or upgrading CCMDB V7.2.1, the user might need the ability to set up Tivoli's process automation engine for DIS and CMS. Again, due to the TADDM dependency, a customer might need to initially defer the DIS and CMS setup. This capability allows them to add this integration subsequent to point in time where CCMDB V7.2.1 is upgraded or base installed.

Scenario: Set up DIS and CMS tables in the database used by CCMDB and configure Tivoli's process automation engine to use those tables

This option will be used if the DIS and CMS databases have not previously been deployed by another Tivoli product or OMP. In this scenario, the DIS and CMS Tables are stored in the Maximo Database. For this, first install or upgrade CCMDB before running the TADDM DIS/CMS Integration Package to load launch entries into CMS and resources into DIS.

1. Run the Base Services Installer that is available with CCMDB V7.2.1.
2. In the Base Services Installer UI on the panel titled "Data Integration Service and Context Menu Service Deployment Options", select the option to **Deploy data integration and context menu services into the same database that you created for the product.**
3. From the panel titled "Run Configuration Step", select the check box titled **Deploy application files manually later.**
4. After the Base Services Installer completes, use the Process Solution Installer (PSI) GUI to install the CI Applications CMS Integration PSI Package. This package is located in the pmp sub-directory of the product installation directory and is named cci_cms_lic_7.2.1.0.zip. From the panel titled "Package Options" ensure that the check boxes titled **Defer Maximo Application Redeployment** and **Defer the Update of the Maximo Database** are *not selected*.

5. Run the TADDM DIS/CMS Integration Package to populate the CMS Registry database with TADDM Launch Into Entries and to register actual configuration item resources into DIS.
6. Import CI types and actual CIs into CCMDB using ITIC (corresponding to the resources registered into DIS by the TADDM DIS/CMS Integration Package in Step 5).

Scenario: Configure Tivoli's process automation engine to use a previously deployed DIS and CMS

If CCMDB is being added on or upgraded in an environment where another product (or the customer) has set up the DIS and CMS databases, then this option will be used. Connection information about the existing DIS and CMS databases will be collected and Tivoli's process automation engine specific actions (such as registration of Launch Into entries for CCMDB) will be initiated against the current DIS and CMS.

In this scenario the DIS and CMS Tables are not created in the Maximo Database. The Base Services Installer configures Tivoli's process automation engine environment to use a database already set up by another product to hold the DIS and CMS tables. Two typical scenarios clients have where DIS and CMS is already deployed are:

- ▶ CCMDB Installed After TADDM Creates DIS and CMS tables
- ▶ CCMDB Installed After Another IBM Product Creates DIS and CMS tables

Scenario: CCMDB installed after TADDM creates DIS and CMS tables

This is a typical scenario where the Base Services Installer configures Tivoli's process automation engine environment to use a database already set up by TADDM to hold the DIS and CMS tables.

1. Run the TADDM DIS/CMS Integration Package to create the DIS and CMS tables (these must be in the same database and only support DB2/Oracle).
2. Run the TADDM DIS/CMS Integration Package to populate the CMS Registry database with TADDM Launch Into Entries and to register actual configuration item resources into DIS.
3. Run the Base Services Installer that comes with the CCMDB 721 product.
4. In the Base Services Installer UI on the panel titled "Data Integration Service and Context Menu Service Deployment Options", select the option to **Configure data integration and context menu services for this product using a previously deployed data integration and context menu service instance.**

5. On the subsequent panel titled “Data Integration Service Database Information”, select the check box titled **Use the same database for both data integration and context menu services**. On that panel, specify the database connection properties for the DIS/CMS database that you created using the TADDM DIS/CMS Integration Package.
6. From the panel titled “Run Configuration Step”, select the check box titled **Deploy application files manually later**.
7. After the Base Services Installer completes, use the PSI (IBM Process Solution Installer) GUI to install the *CI Applications CMS Integration* PSI package. This package is located in the pmp sub-directory of the product installation directory and is named cci_cms_lic_7.2.1.0.zip. From the panel titled “Package Options” ensure that the check boxes titled **Defer Maximo Application Redeployment** and **Defer the Update of the Maximo Database** are not selected.
8. Import CI types and actual CIs into CCMDB using ITIC (corresponding to the resources registered into DIS by the TADDM DIS/CMS Integration Package in Step 2).

Scenario: CCMDB installed after another IBM product creates DIS and CMS tables

This scenario is similar to the *CCMDB Installed After TADDM Creates DIS and CMS Tables* scenario, with the exception that another product (other than TADDM) has already created the DIS and CMS Tables when CCMDB is installed or upgraded. The deployment steps from the previous Scenario are reused with the following exceptions:

Step 1 is not required because the DIS and CMS Tables have already been created.

In Step 5, the DB connection properties specified must be for those associated with the DIS and CMS database created previously by another IBM product.

Scenario: Defer or skip DIS and CMS setup and configuration

The default option is to set up CCMDB to utilize DIS and CMS. However there will be an option to install/upgrade CCMDB without this support. CCMDB has a dependency on TADDM to register resources with NRS. This is required to enable the Launch In Context using CMS. The customer installing or upgrading CCMDB might not have already upgraded or installed TADDM at the required V7.2.1 level.

Based on your requirements, you might need to install or upgrade CCMDB while electing to defer the configuration of DIS or CMS until a later point in time. For example, you might want to coordinate the deployment of DIS and CMS in your organization across a variety of IBM products on another deployment schedule than the schedule for rolling out configuration and Change management in CCMDB. You can do this by electing to defer DIS and CMS configuration.

1. Run the Base Services Installer.
2. In the Base Services Installer UI on the panel titled “Data Integration Service and Context Menu Service Deployment Options”, select the option to **Defer or skip data integration and context menu services setup and configuration**.
3. From the panel titled “Run Configuration Step”, do not select the check box titled **Deploy application files manually later**.
4. At a later point in time, when you want to deploy data integration and context menu services within CCMDB, rerun the Base Services Installer.
5. The Base Services Installer will detect that data integration and context menu services have not been configured and will display the panel titled “Data Integration Service and Context Menu Service Deployment Options”.
6. From this panel, you might continue using one of the previously described deployment scenarios.

11.5 CMS Registry Loader externals

To support launch into Tivoli’s process automation engine applications from other products using CMS, those Tivoli’s process automation engine applications must be registered as launch points in the CMS Registry. Tivoli’s process automation engine provides a common program for performing this registration. The CMS Registry Loader is a command line interface program for loading launch entry definitions for Tivoli’s process automation engine applications into the CMS Registry.

The CMS Registry Loader provides functions for:

- ▶ Creating and updating launch entries in CMS.
- ▶ Listing currently registered launch entries in CMS.
- ▶ Removing launch entries from CMS.
- ▶ Validating launch entry XML files.

11.5.1 Syntax

The CMS Registry Loader is deployed in the bin directory of the base services installation directory (by default, this is \IBM\SMP\bin) and is implemented as follows:

- ▶ **registryLoader.bat** for Windows-based Administrative Workstations
- ▶ **registryLoader.sh** for Unix-based Administrative Workstations
- ▶ **registryLoader.bat(.sh)**
 - action create|remove|list|validate
 - <action-specific parameter>

11.5.2 Supported actions

Each invocation of the registryLoader script must specify the -action parameter. This parameter identifies the action to be performed. Support actions are defined in the following table.

Table 11-1 Supported actions

Action	Description
create	Creates and updates launch entries in CMS
list	Displays currently registered launch entries
remove	Removes launch entries from CMS
validate	Validates a launch entry XML file

See the following InfoCenter link for details on how to use this command:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/index.jsp?topic=/com.ibm.ccmdb.doc_721/admin/c_cms_cmsintro.html

11.6 LIC using TADDM

Let us consider a LIC scenario from CCMDB into TADDM. The first step involved here is the setup of the TADDM and CCMDB servers. After installing TADDM 7.2.1, you use TADDM Integration Package to register the TADDM launch points.

The TADDM Integration Package provides utilities for:

- ▶ Creating the CMS and DIS Database tables.
- ▶ Register CMS launch entries for launching in to TADDM.
- ▶ Register resources in DIS.

Follow these steps to complete the DIS/CMS setup:

1. Run TADDM DIS/CMS Integration Package to create the DIS and CMS tables.
2. Run the TADDM DIS/CMS Integration Package to populate the CMS Registry database with TADDM Launch Into Entries and to register resources in DIS.
3. Run the BSI with the CCMDB 721 Payload.
4. Select the BSI option to configure DIS/CMS for Tivoli's process automation engine and specify the DB connection properties for the DIS/CMS databases.
5. CMS and DIS binaries are unpacked and CMS servlet filter is configured.
6. Use the PSI GUI to install the CCI LIC package to enable the CCMDB specific support for CMS LIC.
7. The MAXIMO EAR is rebuilt and redeployed.
8. Import CI types and actual CIs into CCMDB using the ITIC.
9. Promote the CIs.

To test the LIC function, **Go To** → **IT Infrastructure** → **Configuration Items**:

1. Select one of the CIs you had promoted earlier and click the **Details Menu** of the actual configuration item (Figure 11-5).

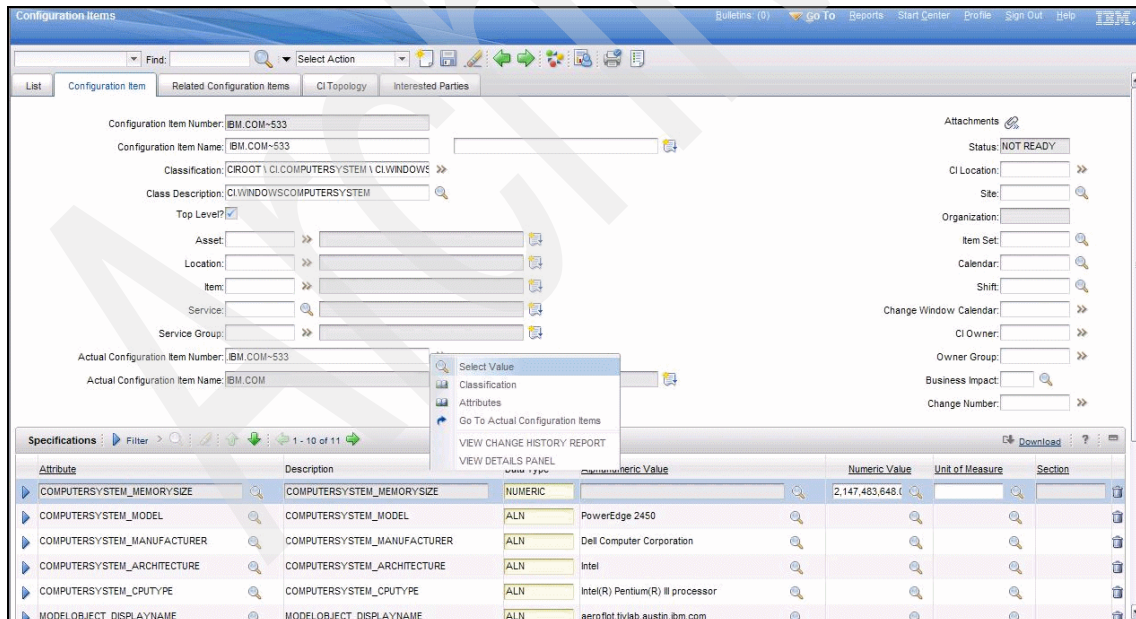


Figure 11-5 Details Menu of the actual configuration item

Here the View Change History Report and View Details panels enable you to launch into TADDM CI. You can also select the LIC items from the Select Action menu of the CI.

2. Click **View Change History Report**. This will take you to the TADDM authentication window as shown in Figure 11-6.

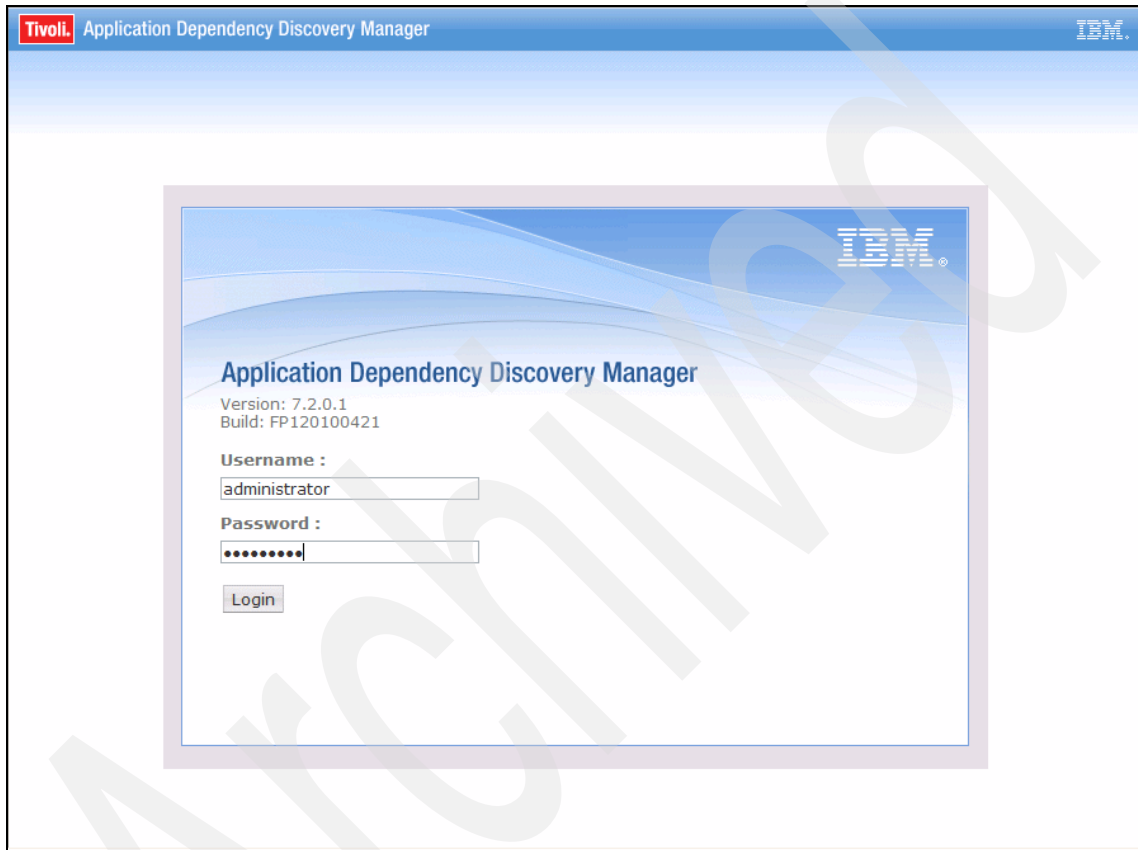


Figure 11-6 TADDM authentication window

3. After providing the required credentials, the Change history is listed as shown in Figure 11-7.

Change History						
Type	Component	Change	Date	Attribute	Old	New
Windows Computer System	ibm.com	Created	Monday, April 19, 2010 1:41:08 AM CDT			

Figure 11-7 Change history

The View Details panel from CCMDB is shown in Figure 11-8.

General	OS	Devices	Storage	IP	Active Services	Interfaces	Software Components	Services	Config Files	Network Connections
Metrics	CI Info	Dependencies	Admin Info	MSS Info	Artifact Info	Extended Attributes				
General										
Name:		ibm								
Fully qualified domain name:		ibm.com								
Object Type:		Windows Computer System								
Last Modified Time:		4/18/10 23:52 CDT								
Manufacturer:		Dell Computer Corporation								
Model:		PowerEdge 2450								
Serial Number:										
Architecture:		Intel								
CPU Speed:		728 MHz								
CPU Type:		Intel(R) Pentium(R) III processor								
Number of Logical CPUs:		2								
Memory Size:		2.00 GB								
LIC URL		http://ibm.com:9430/cdm/servlet/LICServlet?console=web&guid=F1200C61DD743D1BA1CC37DD391EF2B9								

Figure 11-8 View Details panel

CCMDB performance tuning

In this chapter we discuss CCMDB performance tuning.

We cover the following topics:

- ▶ 12.1, “Performance tuning” on page 524
- ▶ 12.2, “The network” on page 525
- ▶ 12.3, “Hardware” on page 528
- ▶ 12.4, “Operating systems” on page 529
- ▶ 12.5, “Services” on page 530
- ▶ 12.6, “IBM Tivoli Change and Configuration Management Database (CCMDB) tuning” on page 533

12.1 Performance tuning

Although each of us knows what we mean when we describe performance problems, it turns out that performance is a rather ambiguous word to describe the situation when an application is not responding the way it is expected to. Performance can mean many things to many people. In the end, it all boils down to the user experience, but that is not the best way to describe the technical issues behind a performance related problem.

A user who only uses purchasing modules might be quite satisfied with the performance, while a user whose job depends on work management modules might be quite frustrated. An administrative user with a simple login who only manages the application might not experience the application the same way a power user might.

This chapter is dedicated to understanding the application architecture, planning for good stable performance, and troubleshooting issues when they arise.

The IBM Tivoli Change and Configuration Management Database (CCMDB) application is built on a common Java based application architecture known as Tivoli's process automation engine (it used to be called Base Services). Tivoli's process automation engine provides core services to applications that are built on its platform so that only those functions that are unique to the application must be coded. This provides for a powerful and stable development environment and a common set of tools for managing a wide range of applications. For more information about Tivoli's process automation engine, see the online support document 1395627 at the following location:

<http://www-01.ibm.com/support/docview.wss?uid=swg21395627>

Change and Configuration Management Database (CCMDB) application is a complex integration of technologies including networking, hardware, operating systems, supporting services, and the application code itself. Successful tuning will involve each of these technologies. Any one of the reliant technologies can be a bottleneck and impact your end-user experience.

Because web based applications rely on the network for communications with the client, you can think of the network as being at the base of the technology stack reliance pyramid. Sitting on the network is hardware such as physical servers and client computers. On the hardware, there are operating systems. On the operating systems there are services applications such as database servers, web application server, reporting services, http servers, and more. Finally, on the web application server sits the application which has its own collection of tuning parameters for each unique environment. Figure 12-1 depicts this technology relationship.

Identifying Problem Areas (End to End – bottom up)

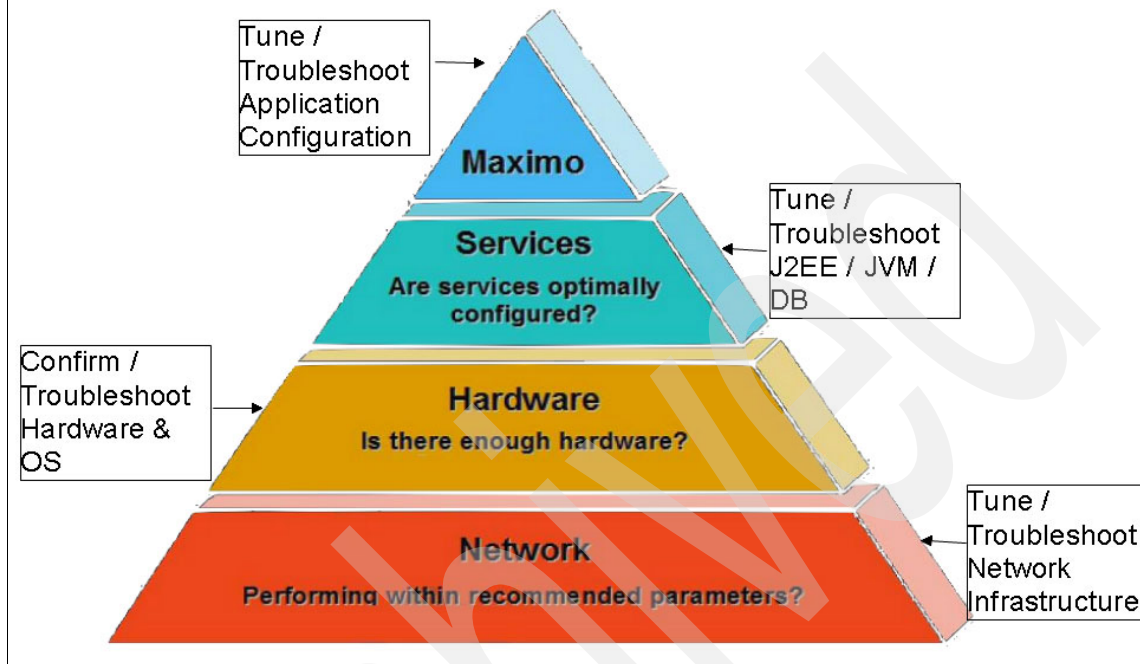


Figure 12-1 Identifying problem areas

12.2 The network

The Change and Configuration Database application relies on Hypertext Transfer Protocol (HTTP) to communicate client requests and server responses. In addition, other forms of network traffic include an “Ajax-like” communication that can be used for field validation and other background tasks.

IBM testing has shown that the application can perform within published specification when the network latency is less than fifty (50) milliseconds and when available bandwidth is at least thirty-two (32) kbps.

If the network has ICMP enabled and any firewall or protection configurations allow it, the “ping” command can be used from a client to the server to get a ballpark latency figure. More exact network monitoring tools can be used for environments where ping will not work or more precise results are desired.

To execute the ping command, fFrom a command prompt on the client, enter:

```
ping <application_server_name>
```

Where application_server_name is the server where the application is deployed.

The results have to show an average of less than 50 milliseconds, and each response has to be relatively consistent (within a few milliseconds of each other). If the average is greater than 50 milliseconds, users can expect slow user interface refresh rates. The higher the latency, the slower the refresh rate will be.

Latency issues can be addressed in a number of ways:

- ▶ Client upgrade of the network
- ▶ Multi-device network accelerators
- ▶ Single device network accelerators
- ▶ Application virtualization and delivery software

12.2.1 Multi-device network accelerators

The most well known, multi-device network accelerators are produced by Riverbed Technology. These devices provide Wide Area Network (WAN) optimization by deploying a device on both sides of the network and employing methods for data reduction, TCP acceleration, and application layer protocol optimizations. While the technology is extremely efficient, it can only be cost effective in environments where large populations of users are connect to a Local Area Network (LAN) that is remote from where the servers are deployed. In this scenario, a device can be deployed on the server side of the network and the Local Area Network gateway of the remote site.

12.2.2 Single-device network accelerators

The most well known, single-device network accelerators are produced by Juniper Networks. These devices provide Wide Area Network (WAN) optimization by deploying a device on the server side of the network and employing Ethernet-optimized multilayer packet processing. This technology improves the quality and size of network traffic and can be a cost effective solution where small populations of users are connected to the network in multiple locations. In this scenario, a device can be deployed on the server side of the network and the Local Area Network gateway of the remote site.

12.2.3 Application Virtualization and Delivery Software

These products, which include Citrix XenApp and Microsoft Terminal Server, overcome network performance issues by minimizing the traffic through a form of remote access to a server operating system. Because the impact of network performance is primarily caused by consistent back and forth communications of HTTP requests and responses, removing that traffic from the wide area network can greatly improve the network response. Application Virtualization and Delivery software uses a highly optimized protocol to provide a connection and remote terminal to the client. Using minimal data, screen changes, keystrokes, mouse movements, and other client side activities are communicated while the client's application actually runs on the server.

In the case of IBM Tivoli Change and Configuration Management Database, the client connects to the Application Virtualization and Delivery server and runs the browser there. The Application Virtualization and Delivery server resides on the same local area network as the Change and Configuration Management Database server. In this configuration, all high demand HTTP traffic travels between the Application Virtualization and Delivery server and Change and Configuration Management Database server, whereas only low demand traffic travels between the Application Virtualization and Delivery server and the client. Figure 12-2 depicts the flow of traffic described here.

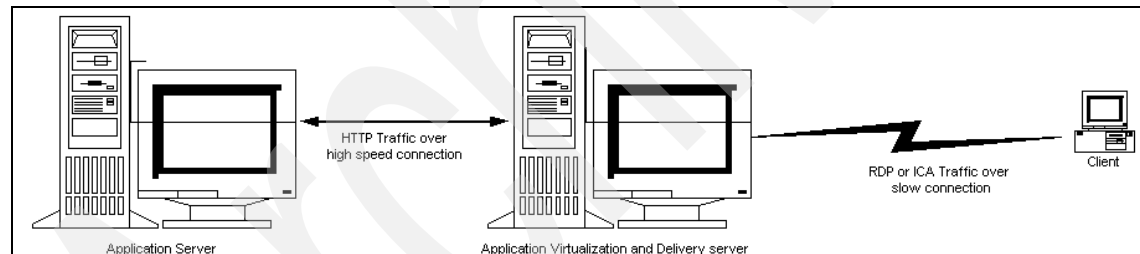


Figure 12-2 Flow of traffic

In this model the Application Virtualization and Delivery server must be robust enough to support the number of users connecting to it and the instances of the application that will be run on it. Typically this is done through the implementation of a "Farm" of servers.

12.3 Hardware

Multiplying the minimum requirements provided in the installation guides by the number of users and/or workload can be done to define hardware requirements. The guides describe a minimum processor speed of 2.0 GHz as well as the requirement of one (1) dedicated processor and two (2) dedicated gigabytes of memory for the application. This description does not include memory or processor requirements for the base operating system or core server functionality. These minimum requirements can support up to a maximum of fifty (50) concurrent typical users, but might support less, depending on the workload of reports, interfaces, cron tasks, and escalations.

An example of a well organized deployment to support up to 100 users might include the following (for the application only):

- ▶ 1 four processor server with 8 GB of memory running a 32 bit operating system for the application.

This will provide enough processors and memory to support the operating system and core services, two (2) instances of the application for the user interface servers and one (1) instance of the application for the interfaces, cron tasks, escalations, and reporting.

- ▶ 1 Processor and 2 GB memory for the operating system
- ▶ 1 Processor and 2 GB memory for the first instance of the application for user interface
- ▶ 1 Processor and 2 GB memory for the second instance of the application for user interface
- ▶ 1 Processor and 2 GB memory for the background processing application

Note that 32 bit operating systems cannot directly access more than 3 GB of memory. There are versions that can implement an addressable extension to enable more memory availability.

Using 64 bit operating systems can be a powerful option but because machine language words are twice as big, memory requirements are double those of 32 bit operating systems. The same configuration noted previously requires 16 GB of memory.

- ▶ 1 Processor and 4 GB memory for the operating system
- ▶ 1 Processor and 4 GB memory for the first instance of the application for user interface
- ▶ 1 Processor and 4 GB memory for the second instance of the application for user interface
- ▶ 1 Processor and 4 GB memory for the background processing application

12.4 Operating systems

Operating systems (OS) are the base of any running software. The operating system is responsible for memory allocation, Input and Output (I/O) operations, and all calls to the hardware. The operating system must be properly tuned for using and allocating memory, swap and temp space, access rights and security, file handles and many more resources. The tuning of operating systems is beyond the scope of this guide because there are five (5) supported operating systems in both 32 bit and 64 bit releases. Still, it is worth noting that proper tuning of the core operating system is key to the performance of the applications running on it.

With the advent of virtual environments comes a great deal of flexibility but, as with any type of technology, there are trade offs. Virtual servers can help Information Technology (IT) departments maximize hardware usage and minimize server footprints. At the same time, a virtual environment can offer the flexibility to move and backup servers across multiple pieces of equipment. This capability does not come without a price. A VM Ware ESX server can be expected to impact performance up to twenty (20) percent and for virtual environments that run on top of other operating systems there might be a thirty (30) percent hit.

If the virtual environment is not properly configured, you can expect even greater impact to performance. Configuration of multiple VM servers on a single hardware server might stretch Input/Output bridges beyond their capacity and incorrect memory configurations allowing shared memory can cause disk swapping that will severely impact performance. If virtual environments are to be used, it is imperative that both the operating system and the virtual server are properly configured to maximize performance.

With the advent of virtual environments comes a great deal of flexibility but, as with any type of technology, there are trade offs. Virtual servers can help Information Technology (IT) departments maximize hardware usage and minimize server footprints. At the same time, a virtual environment can offer the flexibility to move and backup servers across multiple pieces of equipment. This capability does not come without a price. VMWare VI3 / vSphere incurs a 12%-15% overhead compared to an OS running on bare metal whereas virtualization solutions that are hosted on top of other operating systems there might be a thirty (30) percent hit. Furthermore, it is crucial that the application, virtual, and physical environments are appropriately sized and configured for each specific workload. Failure to properly evaluate and configure a virtualization solution can potentially incur a much higher performance hit.

The technology team implementing VMWare needs an in-depth understanding of the concepts at the following VMWare link:

http://www.vmware.com/pdf/vi_performance_tuning.pdf

If virtual environments are to be used, it is imperative that both the operating system and the virtual server are properly configured to maximize performance.

12.5 Services

The multi layer, scalable architecture of the application relies on various service type technologies such as these:

- ▶ HTTP Servers
- ▶ Application Servers (WebSphere and WebLogic)
- ▶ Java JVMs, JREs, and JDKs (IBM and Oracle/Sun)
- ▶ Databases (IBM DB2, Oracle, and Microsoft SQL Server)
- ▶ Reporting Servers (Primarily BIRT but other third party services as well)

Each of these services can impact not only their own aspect of the related technology but the application as a whole. As an example, consider that there might be a poorly written report with inefficient SQL that is scheduled to run regularly throughout the day. The impact of this will not only impact reporting but also database response time and ultimately the user response time.

12.5.1 HTTP Servers

HTTP Servers need to be tuned and configured to handle any load might be placed on them. If the HTTP Server cannot support the number of connections requested or generates logs that exceed disk quotas, the HTTP Server can fail.

12.5.2 Application Servers

Each Application Server has unique settings that can improve its performance and is responsible for the values passed to the JVMs it manages. If these values and resources are not properly tuned, poor client response and throughput can result.

WebSphere

The known configuration values for IBM WebSphere include setting values for the AckTimeout variable and the preferIPv4 variable. In production environments, disable verbose garbage collection, and when deploying applications, clear the temp and tranlog directories to prevent conflicts.

See the following documents for more information about these topics:

- ▶ *Parameters to Improve WebSphere Performance:*
<http://www.ibm.com/support/docview.wss?uid=swg21261874>
- ▶ *Improving WebSphere Performance and Reducing Hung Threads:*
<http://www.ibm.com/support/docview.wss?uid=swg21395626>

WebLogic

The known configurations for Oracle WebLogic include changing the default thread count to 45 and not depending on the WebLogic redirector for load balancing. Configurations to use external plug-ins such as IIS can also cause response slowness when not properly configured. See the following documents for more information about these topics:

- ▶ *BEA WebLogic 7 Tuning for Better Performance:*
<http://www.ibm.com/support/docview.wss?uid=swg21294300>

This document is specific to V7 but also applies to V8 and V9.

12.5.3 Java Virtual Machines (JVMs)

The Java Virtual Machine is the core resource environment for the application to run in. It is critical that this level of technology is tuned properly for the product to run correctly. While most of the parameters that default on a Java Virtual Machine are acceptable, memory and its management are vital settings. The installation guides specify to set the minimum memory to 512MB and the maximum memory to 1536MB.

Keep in mind that these parameters are for 32 bit systems. For environments where there is a 64 bit operating system and a 64 bit application server, set the parameters to a minimum of 1024 MB and maximum of 3072 MB (note that many 64 bit environments run well with a maximum of 2560MB and this can be used when memory is in short supply). Although there are many parameters that can be fine tuned to control garbage collection (GC) processes, environments rarely benefit to any noticeable extent from the use of these parameters. At the very least, tuning garbage collection beyond its default approach ought to be left for last on the tuning agenda while the more valuable efforts are undertaken.

One of the questions that often comes up in with regard to the IBM memory preferred settings is why we require 2 GB of memory at the hardware level but only allocate 1.5 GB of memory at the JVM level. The answer to this question is that the JVM runtime requires overhead, so although only 1.5 GB has been allocated that is specific to the application resources, the JVM itself also requires memory, and the result is closer to 1.8 GB of memory in use.

For more information about Java Memory Management, see the following document:

<http://www.ibm.com/support/docview.wss?uid=swg21326774>

12.5.4 Databases

The database is central to all functionality within the product so any configuration or functionality that impacts the database can impact the end user experience. Each database server platform has unique settings that can improve its performance. The three database platforms supported are:

- ▶ IBM DB2
- ▶ Oracle
- ▶ Microsoft SQL Server

Each of these platforms has manufacturer preferred settings for maximizing performance. In addition, IBM might have specific settings that are known to improve performance of the CCMDDB application. When there is a conflict between a manufacturer preferred setting and an IBM preferred setting, select the IBM preferred setting, because of the unique architecture and design of the CCMDDB application. The IBM settings are the result of extensive testing.

For DB2, it is best that the database be configured using the following parameters:

- ▶ Automatic maintenance (AUTO_MAINT) = ON
- ▶ Automatic table maintenance (AUTO_TBL_MAINT) = ON
- ▶ Automatic reinstates (AUTO_RUNSTATS) = ON
- ▶ Automatic statistics profiling (AUTO_STATS_PROF) = OFF
- ▶ Automatic profile updates (AUTO_PROF_UPD) = OFF
- ▶ Automatic reorganization (AUTO_REORG) = ON

For Oracle, it is best that the cursor_sharing parameter be set to SIMILAR or FORCE. See document 1262959 published at:

<http://www.ibm.com/support/docview.wss?uid=swg21262959>

For SQL Server, it is best that row level locking be enabled. See document 1268567 published at:

<http://www.ibm.com/support/docview.wss?uid=swg21268567>

Indexes

Regardless of database platform, proper indexing is a key performance tuning requirement. With a product that is as powerful and configures as CCMDB, it is expected that each implementation of the product will use it in a targeted way that most benefits the organization. For this reason, the generic tuning and indexing applied to the base application is often not sufficient for a specific use. For each implementation, evaluate how users access the data and add indexes accordingly. In addition, do this any time custom tables, fields, or saved queries are configured,

12.5.5 BIRT reporting

IBM has identified that while the BIRT Reporting engine is integrated into the J2EE application much like Crons, Interface, Escalations, and User Interface, the BIRT engine performs best when it can be separated to a dedicated JVM or JVM cluster. For more information about configuring BIRT, see document 1305031, published at:

<https://www.ibm.com/support/docview.wss?uid=swg21305031>

Access: An IBM ID is required to access this document. If you do not have an IBM ID, you can request one at the authentication page that displays when you click this link.

12.6 IBM Tivoli Change and Configuration Management Database (CCMDB) tuning

Application configurations through properties, implementation choices, and deployment methodologies all have an impact on application performance.

12.6.1 Properties

An effort is made at the time of installation to select the best properties for a given environment but there are a number of things that can interfere with the best choices. Examples of these are:

- ▶ Manual Property modification
- ▶ Environments exported from one database to another (changing the environment)
- ▶ Platform Changes such as application server or database platform

In general, it must be confirmed that the database related properties are properly configured for the specific platform, the instance based properties for clustering such as the DoNotRun property for cron tasks and reporting are configured to match the cluster, and the debug properties for monitoring performance and stability are properly configured for proactive tuning and root cause analysis in the incidence of an event.

Properties for DB2 are:

- ▶ mxe.db.driver=com.ibm.db2.jcc.DB2Driver
- ▶ mxe.db.initialConnections=6
- ▶ mxe.db.maxFreeConnections=10
- ▶ mxe.db.minFreeConnections=4
- ▶ mxe.db.newConnectionCount=3
- ▶ mxe.db.systemdateformat=current timestamp
- ▶ mxe.db.transaction_isolation=TRANSACTION_READ_COMMITTED
- ▶ mxe.db.format.upper=UPPER
- ▶ mxe.db.autocommit=false
- ▶ mxe.db.logSQLTimeLimit=1000
- ▶ mxe.db.fetchResultLogLimit=200
- ▶ mxe.mbocount=1

Properties for Oracle are:

- ▶ mxe.db.driver=oracle.jdbc.driver.OracleDriver
- ▶ mxe.db.initialConnections=6
- ▶ mxe.db.maxFreeConnections=10
- ▶ mxe.db.minFreeConnections=4
- ▶ mxe.db.newConnectionCount=3
- ▶ mxe.db.sqlserverPrefetchRows=200
- ▶ mxe.db.disableservercursor=Y
- ▶ mxe.db.transaction_isolation=TRANSACTION_READ_COMMITTED
- ▶ mxe.db.format.upper=UPPER
- ▶ mxe.db.autocommit=false
- ▶ mxe.db.systemdateformat= sysdate
- ▶ mxe.db.format.nullvalue=NVL
- ▶ mxe.db.logSQLTimeLimit=1000
- ▶ mxe.db.fetchResultLogLimit=200
- ▶ mxe.mbocount=1

Properties for SQL Server 2005 and 2008 are:

- ▶ mxe.db.schemaowner=dbo
- ▶ mxe.db.driver=com.inet.tds.TdsDriver
- ▶ mxe.db.initialConnections=6
- ▶ mxe.db.maxFreeConnections=10
- ▶ mxe.db.minFreeConnections=4

- ▶ mxe.db.newConnectionCount=3
- ▶ mxe.db.sqlserverPrefetchRows=200
- ▶ mxe.db.disableservercursor=Y
- ▶ mxe.db.resultsettype=TYPE_FAST_FORWARD
- ▶ mxe.db.fetchsizeuse=Y
- ▶ mxe.db.fetchsize=40
- ▶ mxe.db.optionuse = Y
- ▶ mxe.db.optionnum=1000
- ▶ mxe.db.transaction_isolation=TRANSACTION_READ_COMMITTED
- ▶ mxe.db.format.upper=UPPER
- ▶ mxe.db.autocommit=false
- ▶ mxe.db.systemdateformat=getdate()
- ▶ mxe.db.format.nullvalue=ISNULL
- ▶ mxe.db.logSQLTimeLimit=1000
- ▶ mxe.db.fetchResultLogLimit=200
- ▶ mxe.mbocount=1

12.6.2 The mxe.crontask.donotrun property

The donotrun property is used to specify which cron tasks are not allowed to run on a specific JVM. This property is set at the instance level so that each instance of a JVM can have another value. There are 3 possible configurations for this property are:

```
mxe.crontask.donotrun=ALL
//mxe.crontask.donotrun=
mxe.crontask.donotrun=crontaskname,instancename,
crontaskname,instancename
```

ALL will prevent any cron task from running on this JVM.

Commented out will allow all cron tasks to run on this JVM.

Cron tasks separated by commas will specify specific crons that will prevent the specified cron tasks from running on this JVM. (Although the line can wrap in an editor, these cron tasks cannot be broken by line feeds.)

In addition to this configuration, it is best that the JVM generic parameter be used to name the JVM instances, as follows:

```
-Dmxe.name=<servername>
```

Using the internal method for naming JVMs can lead to dynamic naming and JVM names shifting across JVMs in the cluster.

12.6.3 Implementation choices

Implementation using any of the following possibilities can impact the user's perception of performance:

- ▶ Inefficient search methodologies
- ▶ Poorly structured saved queries
- ▶ Poorly structured workflow processes or inefficient workflow coding
- ▶ Escalations and cron tasks scheduled to execute too frequently
- ▶ Extensive cron tasks that execute during peak usage periods
- ▶ Extensive reports that execute during peak usage periods
- ▶ Large customized screens
- ▶ Extensive field level validation and security
- ▶ Deployment without process analysis and user training
- ▶ Ineffective clustering and task separation
- ▶ Business processes that fail to consider data ramifications
- ▶ Loading data into a database without the use of MBO validation
- ▶ Poorly coded Java customizations and unclosed MBO sets
- ▶ Failure to consider WAN performance ramifications for remote users

Inefficient search methodologies

The CCMDB product is shipped with wildcard searching enabled for most text-based fields. While this search type provides simple and powerful searching capability, the default behavior of this search method forces the database to execute full table scans instead of using indexes. When databases begin to reach their capacity, this consistent use of table scans will impact their ability to respond, and the application performance will suffer. Generally, deploy the application with the exact search type enabled and train users on how to use wildcards when they need them. See document 1321289 for more information about search methodologies:

<http://www.ibm.com/support/docview.wss?uid=swg21321289>

Poorly structured saved queries

Administrators and users have the ability to store queries that are frequently used in day-to-day operations. If these queries contain wildcards or are poorly designed, the user that executes them might experience long waits for a response and other users on the system might suffer while the database tries to service the request. Check the saved queries in a query analysis tool to determine the cost and decide on efficient SQL.

Poorly structured workflow processes or inefficient workflow coding

Workflow is a customizable product that can be used to develop data flow for business processes. Each node in a workflow diagram is programmable and can call internal or external processes. The efficiency of these nodes can directly impact the performance of the data being managed. Starting a record in a workflow that has complex SQL, many conditions, or poorly formed SQL commands in the first series of nodes can impact the users perception of performance. As the workflow process is initialized, all nodes will execute until it reaches a condition that can return control to the user. If there are many complex nodes the user might wait for extended periods of time for the workflow to initialize. Actions can also have an impact. Extensive operations such as changing a status, applying an SLA, and changing status again can take time while the user waits for control to be returned.

Escalations and cron tasks that are scheduled to execute too frequently or inefficiently

An escalation is a process based on a cron task. Escalations can be used similar to workflow but in a more automated decision based approach. Escalations can be triggered based on data or time frames. If an escalation is complex and scheduled to run too often, it can amount to the escalation running perpetually and both the application and database can suffer performance degradation due to the frequency.

Any inefficient SQL operation can lead to poor database performance. Testing in real situations is often critical. A SQL statement that executes instantly on ten records might run very poorly on a million records. If an escalation or cron task is scheduled with this type of issue, the entire environment can suffer.

Extensive cron tasks that execute during peak usage periods

Cron tasks that perform extensive work in the database can impact performance when run during peak usage times. Schedule a reorder task or a record generation task that operates on thousands of records to execute at off peak periods.

Extensive reports that execute during peak usage periods

Similar to cron tasks, schedule reports that do extensive complex queries such as accounting rollups to run at off peak periods.

Large customized screens

When customizing screens, it is important to consider the end-user audience and the requirement for all fields on a single screen. As users navigate through the various fields on a screen, CCMDB will validate certain data in many cases using a protocol similar to Ajax to communicate with the server. Poor performing networks and too many fields on a screen can result in pauses between when a field can get focus and begin receiving data.

A best practice in developing custom screens is to determine what fields the various work groups in the organization require. If there is a contingent of users that only do dispatching and require only 10 fields to do that, create a screen for the dispatching team. The larger screen can still exist but the users day-to-day operations can be performed using the more efficient screen.

Extensive field level validation and security

CCMDB provides the capability to secure and validate data down to the field level. Before implementing this functionality, undertake an in-depth analysis to understand what exactly needs this functionality. A screen with one hundred fields on it, all of them executing validation and security methods, will likely appear to be very slow and uncomfortable for the end-user.

Deployment without process analysis and user training

As mentioned in the section under screen customization, understanding how users will use the product and what the goals are is key to a successful implementation. Have a well-trained application expert spend time analyzing what the various groups do and developing step-by-step procedures for accomplishing those tasks. Consider turning a high performance vehicle over to someone who has never driven before. In general, they will figure out how to get somewhere with it but the proper use of the vehicle will escape them. An end user who must accomplish a task will find a way to accomplish it, but without proper training, might execute 10 steps that are not necessary.

Ineffective clustering and task separation

The CCMDB application includes many powerful features such as cron task scheduling, escalations, interface framework, and reporting. Although all of these can reside in the same JVM along with User Interface functionality, it is understandable that the more resources used by background functionality, the less resources there are for user activity. For this reason, it is best that functions be separated into dedicated JVMs.

Generally, it is not advisable to go into production with a single JVM. At the very least, production environments must be clustered with a minimum of two JVMs in a load balanced topology to provide UI functionality. This serves the purpose of sharing load, multiplying resource availability, and provides a level of JVM fault tolerance.

While a dual JVM cluster might be the answer for a 10 to 35 concurrent user environment, after concurrent users exceed 35, consider having one or more separate JVMs for background processing. For 200 concurrent users, have at least 4 or more UI dedicated JVMs, 1 cron JVM, 1 interface JVM, and 1 report JVM for a total of 7 JVMs to support the environment.

Business processes that fail to consider data ramifications

Understanding how CCMDB uses data can help in developing business processes that work best in the CCMDB model. Consider that the default behavior of the Workorder application is to work with records that have a status of open. If the business process that is developed consistently completes work orders but does not close them, the pool of records to work with continues to grow and can begin to impact the users response times.

Loading data into a database without the use of MBO validation

Loading data without utilizing the Business Objects can lead to corrupt data, broken links and generally compromised data integrity. Hierarchies can have broken links or circular references and required references can reference incorrect locations. Because the application relies heavily on data for normal function, this bad data can cause performance problems and even memory leaks.

Poorly coded Java customizations and unclosed MBO sets

CCMDB is highly customizable. IBM provides information about how to extend Java classes to better represent a required business process. The coding of Java for CCMDB requires more than just an understanding of Java. Developers must have a clear understanding of the CCMDB and Base Services methodology of coding. Unclosed MBO sets and MBOs can happen in various ways and can cause memory leaks. As always, extending private classes cannot be supported and must be avoided.

Failure to consider WAN performance ramifications for remote users

When deploying CCMDB over the Wide Area Network (WAN) it is important to recognize the physical limitations of connectivity. Generally speaking, IBM has determined that the application can perform within specification when the communication between the client and the application server is less than 50 milliseconds. Latency figures higher than that will cause delays in rendering responses. A simple litmus test of if a performance problem is related to the application architecture or the network is to use a client on the same Local Area Network (LAN) as the server to determine if it performs well there. If performance is good locally but poor remotely, the issue is likely network related. There are a number of solutions that might help improve WAN performance included in the following document link:

<http://www.ibm.com/developerworks/wikis/download/attachments/138318044/BestPracticesForMaximoSystemPerformance7x.pdf>

A primer on Deployer's Workbench

In this appendix, we introduce the Deployer's Workbench. If you not familiar with the Deployer's Workbench, we suggest that you read this chapter first before Chapter 6, "CI space development" on page 359.

For more information about Deployer's Workbench, you can refer to Deployer's Workbench information about the InfoCenter at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/index.jsp?topic=/com.ibm.dw.doc_721/dw721_welcome.htm

Deployer's Workbench

The Deployer's Workbench is a set of plug-ins for use within an Eclipse environment. These plug-ins provide a pluggable foundation for additional tooling for CCMDB to be delivered in the future. The CI space design tooling ships as a set of additional plug-ins for Deployer's Workbench for use within Eclipse.

Because Deployer's Workbench is based on the Eclipse platform, many of the features of Eclipse are available for use within the workbench. We outline a few of the more important concepts and features.

Eclipse uses a concept of a *workspace* as a way to compartmentalize a group of related projects within one area. Typically a user might only work within one workspace at a time while breaking their work down into multiple projects.

The user can change the workspace location if desired, but will need to create a new project, import their old projects as needed.

Eclipse projects are not just simple folders on your file system, they contain various attributes based upon the type of project a user selected when they created the project in the first place. These attributes are defined by the plug-ins within the Eclipse workspace. Deployer's Workbench has its own project type, and when a user needs to create a new project for use with the Deployer's Workbench, they must use that type.

CI space design tool

The authorized CI space is composed of three basic pieces that provide CCMDB with the needed configuration to properly promote a CI from the actual space to the authorized space. These pieces are:

- ▶ **Authorized Classification:** A classification in CCMDB that has specific properties set that allow it to work with a CI.
- ▶ **Promotion Scope (template):** An object that contains a mapping of classifications between the actual and authorized classification spaces. This scope is the roadmap used by CCMDB to promote a CI from the actual to the authorized space.
- ▶ **Relationship Rules:** These rules define the relationships between the CIs in the authorized space. These rules must be setup correctly within the context of a promotion scope and the authorized classification set in order for CCMDB to properly promote a CI.

The CI space design tool provides a visual UI to help the user not only understand what the authorized CI space looks like, but also provides an intuitive drag and drop interface to help the user quickly build and edit their authorized CI space.

One main feature of this tool is the partial sandbox environment for editing your CI space. Essentially, all edits you make to the authorized classification tree, promotion scopes, and relationship rules are only saved to the local copy of the overall model. No Changes are made to your CCMDB server until you are ready to commit those Changes.

The following considerations are caveats to the current implementation:

- ▶ Last Change wins: Because you are only making Changes locally, if another user makes a Change to the CCMDB system while you are editing, their Change might get removed when you commit your Changes from this tool. This behavior is similar in nature to the existing CCMDB UI.
- ▶ Local model synchronization: This tool supports partial synchronization to the server. Before beginning an editing session, ensure that your local copy of the CI space is up to date by importing it from the server.

Partial-sandbox environment

A sandbox environment is defined as an environment that allows the user to make Changes without affecting or communicating with a live system. The current CI space design tooling does not provide a pure sandbox environment to design your CI space with. Instead the tooling provides a partial sandbox environment where the Changes you make are stored locally until you update your server, but communication to retrieve server specific information occurs during your design process. This means that you must always have a server available to you in order to make Changes to a local CI space. Just viewing the local CI space does not require a server.

Deployer's Workbench user interface

In this section we explain the Deployer's Workbench user interface.

Elements of the Eclipse Workbench

Within an Eclipse Workbench, there is usually a typical set of UI elements that provide the overall look and feel as well as function for the user (Figure A-1).

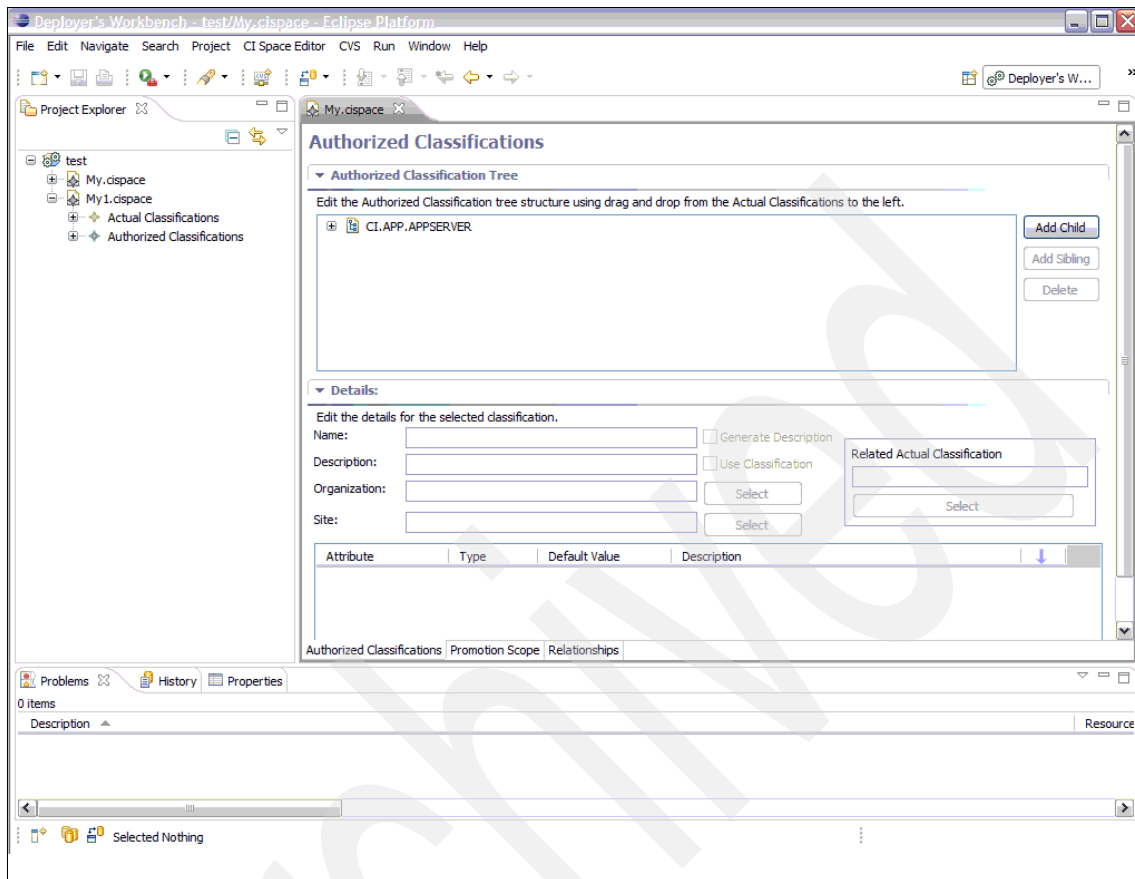


Figure A-1 Elements of the Eclipse Workbench

The following items are the five basic elements of an Eclipse workbench UI, illustrated in Figure A-1:

- ▶ **Navigator:** This element allows the user to navigate projects, files, models and any other piece of information that a plug-in wants to provide within this context.
- ▶ **Editor Space:** This area is typically to the right of the navigator and can contain one or more editors. Editors have a command stack and will undo, redo, dirty and save events.
- ▶ **Perspective:** A perspective is a configuration of UI elements that make up a particular task set for a user. In this case the Deployer's Workbench perspective provides just what the user needs to work with workbench tools such as the CI space tooling.

- ▶ **Editor Pages:** Within Eclipse many editors provide multiple editor panes that provide mechanisms to edit the data within the data set in various ways.
- ▶ **Information Views:** Information views in Eclipse provide a heads up view to information within the context of your current selection within the UI or due to the actions you are performing within the UI. For example, the Properties view can show you properties associated with a selection that you have made somewhere within the editor or navigator.

Let's look at each of the Eclipse elements within the context of Deployer's Workbench and how each has been refined for use. A step by step tutorial for creating a simple authorized CI space can be found later in this section.

Project Explorer

The project navigator for Deployer's Workbench allows the user to select a .cispace file to edit, as well as explore the models within the particular .cispace file (Figure A-2).

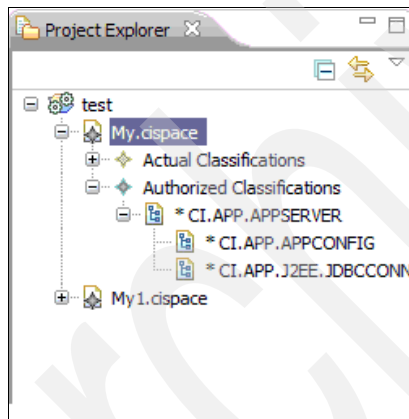


Figure A-2 Project Explorer

There are two CI spaces in CCMDDB, the actual and the authorized space. The explorer allows the user to examine each space within a tree-like context. This provides a visual view into the overall structure of the two spaces.

The explorer will also provide indications or clues as to the state of a particular element in your CI space based on Changes to the file name. In the preceding screen capture, the * indicates that the elements have been changed within the sandbox, but have not yet been updated on the server.

CI Space Editor

The CI Space Editor allows you to edit the authorized CI space for a particular .cispac file. You can have multiple CI Space Editors open at the same time, one per available .cispac file. To open the editor, just double-click the .cispac file in the Project Explorer (Figure A-3).

The screenshot shows the 'My.cispac' window with the 'Authorized Classifications' tab selected. The window is divided into three main sections: 'Authorized Classification Tree', 'Details', and 'Attributes'.

Authorized Classification Tree: This section allows editing the tree structure using drag and drop. It contains a tree view with a single node 'CI.APP.APPSERVER'. To the right of the tree are three buttons: 'Add Child', 'Add Sibling', and 'Delete'.

Details: This section allows editing the details for the selected classification. It includes fields for Name, Description, Organization, Site, Primary Related Actual Classification, and Secondary Related Actual Classifications. The 'Name' field is 'CI.APP.APPSERVER', 'Description' is 'APP.APPSERVER', and 'Primary Related Actual Classification' is 'APP.APPSERVER'. There are checkboxes for 'Generate Description' and 'Use Classification'. There are also 'Select', 'Add', and 'Delete' buttons.

Attributes: This section allows editing the attributes for the selected classification. It contains a table with columns: Attribute, Type, Default Value, and Description. The table has two rows: 'MODEBJECT_DISPLAYNAME' with Type 'ALN' and 'MODEBJECT_DESCRIPTION' with Type 'ALN'. There are checkboxes in the Description column for each row. At the bottom of the window, there are three tabs: 'Authorized Classifications', 'Promotion Scope', and 'Relationships'.

Attribute	Type	Default Value	Description
MODEBJECT_DISPLAYNAME	ALN		MODEBJECT_DISPLAYNAME
MODEBJECT_DESCRIPTION	ALN		MODEBJECT_DESCRIPTION

Figure A-3 CI Space Editor

As shown in the figure, the CI Space Editor has three editor pages (tabs): Authorized Classifications, Promotion Scope, and Relationships. Each of these editors provides you with a way to edit each of the three major aspects of the authorized CI space. You can switch between the editors by selecting the appropriate tab at the bottom of the CI Space Editor window.

Authorized Classifications Editor Page

This editor page allows you to build your authorized classification hierarchy based on the actual classifications within your model as well as new classifications of your own design. This editor page is broken down into two distinct pieces, the Authorized Classification Tree and the Details section.

The details section of this editor page, provides the ability for you to edit the details of whatever classification you have selected in the foregoing classification tree. There is a field in the details section called Primary Related actual Classification. This field shows what actual classification the selected authorized classification was derived from.

For any authorized classification that represents a derived actual classification, this field must be set. This allows the CI space editor keep track of the derived mappings so that attribute and relationship copies can correctly occur. The Secondary Related actual Classifications allows you to create a secondary set of authorized to actual classification mappings.

This allows you, for example, to create a single computer system authorized classification that actually relates to all of the other types of computer system actual classifications.

Important: Attempting to change the Org or Site of a classification will require network access to your CCMDB server in order to display the correct site specific values.

Promotion Scope Editor page

As shown in Figure A-4, Promotion Scope Editor provides you the ability to edit the promotion scope whose starting point authorized classification is the selected authorized classification in the project explorer.

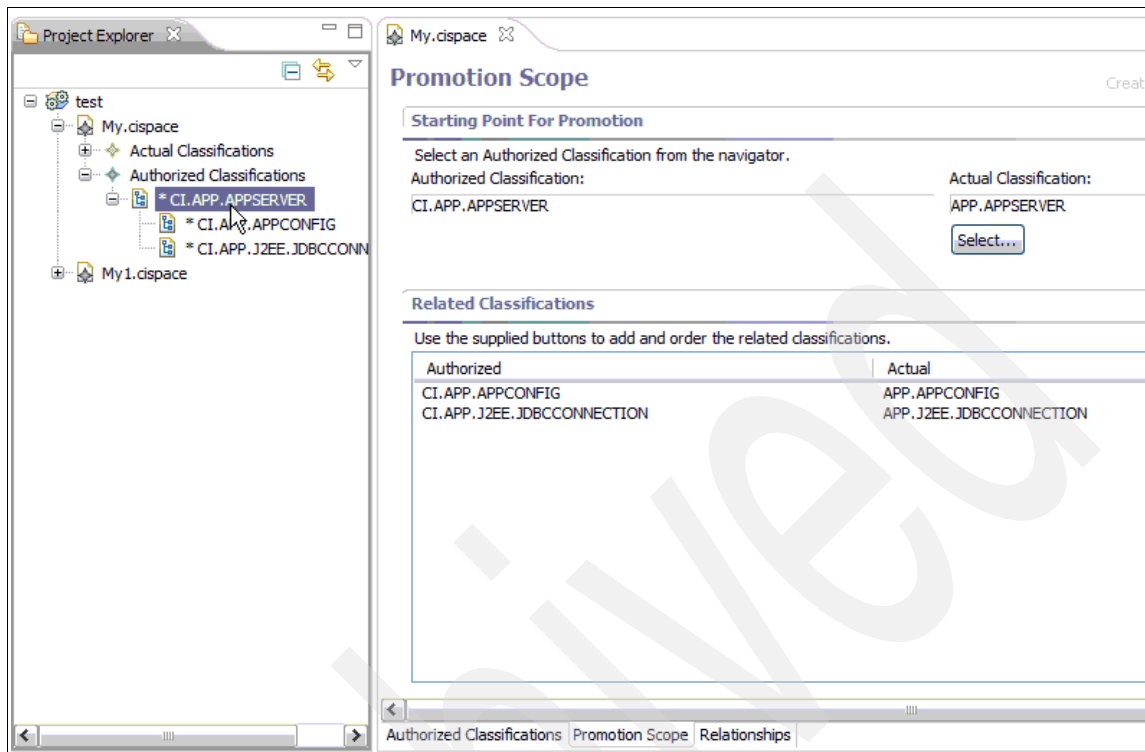


Figure A-4 Promotion Scope Editor page

As you can see, the selected classification in the project explorer is CI.APP.APPSERVER, and the promotion scope page is showing the current promotion scope for that particular authorized classification as the starting point. If you select a classification that has not yet had a promotion scope created, all of the fields will be blank and the Create Promotion Scope button at the top of the page will be enabled.

After you have created a promotion scope, using the button at the top of the page, you can add additional classification mappings to the scope using the Add button to the right of the table. This will add a new row to the table that you can then click to edit the particular row's column value.

Relationships Editor page

Figure A-5 shows the Relationships Editor page provides a diagram view of the relationships between the authorized classifications as seen in the context of the selected promotion scope. Similar to the Promotion Scope Editor page, an authorized classification with a promotion scope must be selected for this editor to display data.

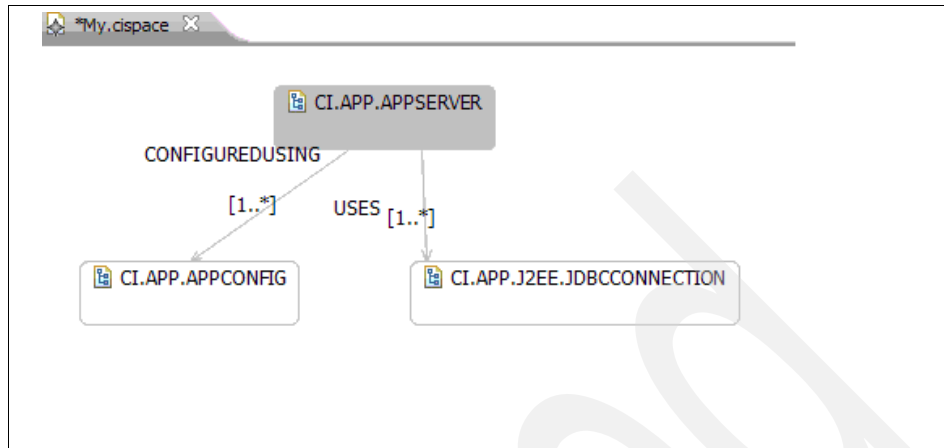


Figure A-5 Relationships Editor page

Deployer's Workbench perspective

The perspective supplied by the Deployer's Workbench plug-ins enable many of the needed actions to support the behavior of the editor and project explorer. Always make sure that this perspective is selected as you edit your CI space. If the perspective is not enabled, then the various features contributed to Eclipse by the plug-ins will not be enabled. Examples of odd behavior due to this might include not being able to examine the models in a .cispace file from within the project explorer.

Problems and properties views

Two of the informational views at the bottom of the workbench provide important information to you while you edit your CI space:

- ▶ The problems view provides you a view of all of the current errors in the CI space as detected by the CI space tooling.
- ▶ The properties view will show you the properties of whatever element you currently have selected within an editor or the explorer. Note that only elements selected in an editor allow the properties to be edited.

CI space designer preferences

The CI space designer currently has a limited set of preferences available. These can be accessed by choosing the **Preferences** option in the main Window menu.

After the preferences dialog is presented, expand the Deployer's Workbench item to access the preferences specific to the CI Space Designer.

There are two preferences that can be customized:

Attribute filters:

- ▶ When copying a classification from the actual to authorized space, all of the attributes are copied from the actual classification into the authorized one. However there might be a set of common attributes that you might not want to be copied over. These might include modification dates or other attributes. These attributes can automatically be filtered out when the copy occurs by updating the attribute filter list in the CI space designer preferences.
- ▶ Expand the **CI Space Designer** item in the preference dialog and select the **Attribute Filter** item. By default, the CI Space Designer ships with a set of common attribute filter strings. When filtering occurs, each string in the list is checked against the attribute being copied to see if the attribute's name contains the filter string. For example, an attribute with the name `MODELOBJECT_BIDIFLAG` is filtered out because the string `BIDIFLAG` is contained in the filter list.

Relationship editor preferences:

- ▶ Also, there is a set of preference pages for the relationships editor. These preference pages define the look and feel of the elements (font and color), connections, print setup, rulers, and grids.
- ▶ The user does not need to modify these preferences at this time.

Creating a new project in the Deployer's Workbench

Because Deployer's Workbench is based on Eclipse, you must create a new project and .cispaces file inside your workspace before you can import or edit any data.

Deployer's Workbench has its own specific kind of project type that provides additional functionality over a simple project. When working with Deployer's Workbench you must work within the context of a Deployer's Workbench project type.

1. To create a project of this type, select **File** → **New** → **Project...**, as shown in Figure A-6.

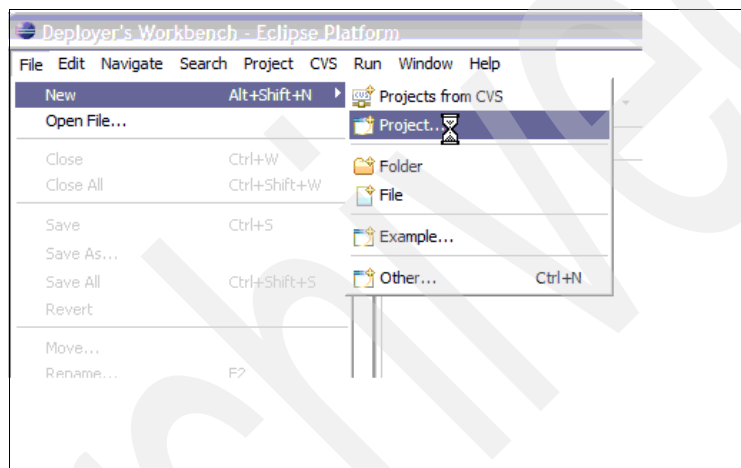


Figure A-6 Creating a new project

2. Next select the **Deployer's Workbench Project** from the list of available project types, as shown in Figure A-7.

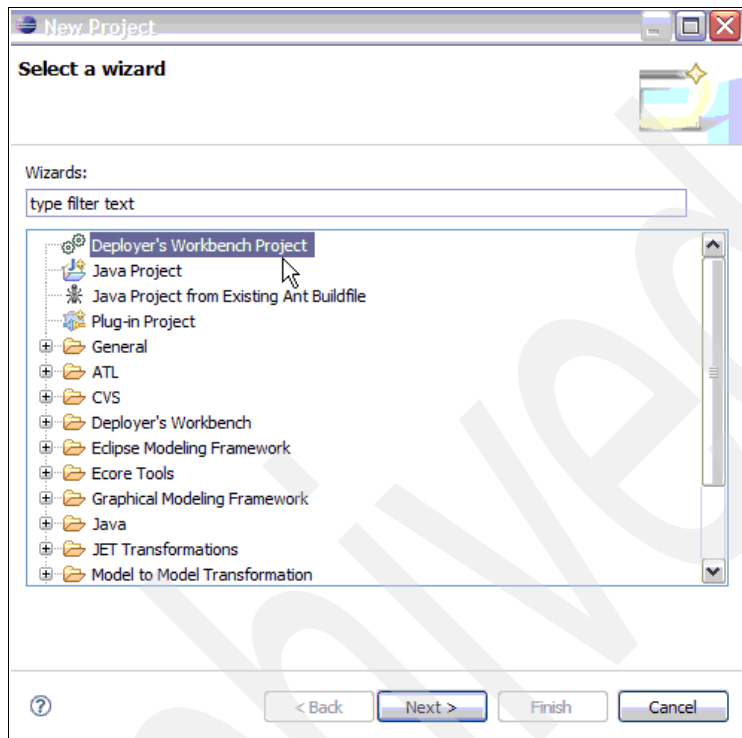


Figure A-7 Creating a new project

3. Give the project a name and click **Finish**, as shown in Figure A-8.

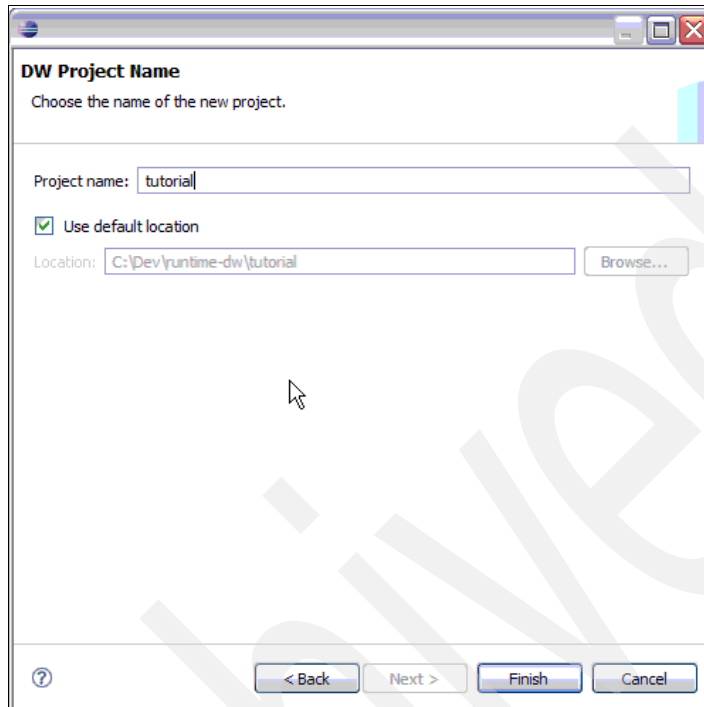


Figure A-8 Creating a new project

A new empty project will be created within your workspace. After you have created a project, you must now create the .cispaces file that will contain your sandbox.

Creating a new CI space file

All sandbox data is stored in a special xml file that ends with .cispace.

1. To create a new empty .cispace file, select your new project, right-click and select **New** → **Other...** as shown in Figure A-9.

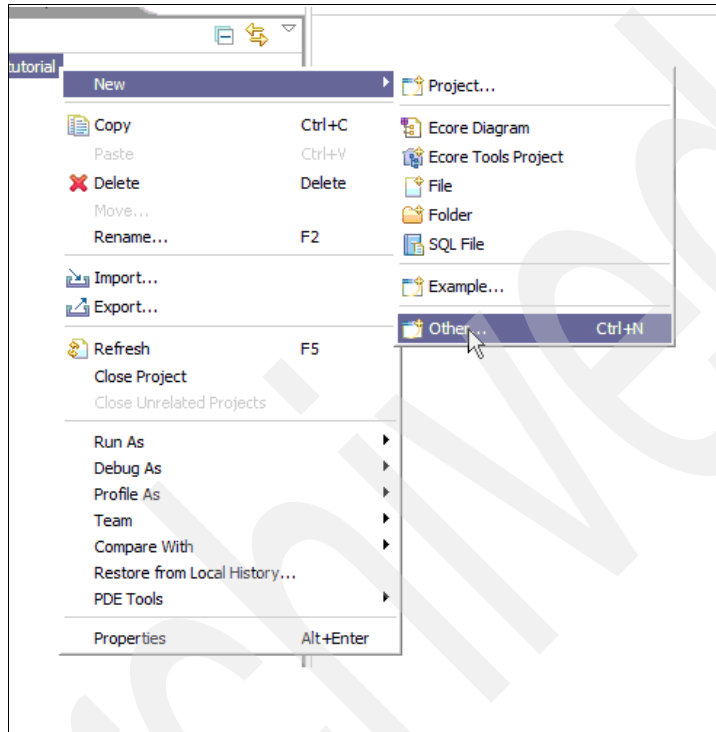


Figure A-9 Creating a new CI space file

2. Locate the **Deployer's Workbench category**, and expand it. Then select the **CI Space item** in the tree, and click **Next** as shown in Figure A-10.

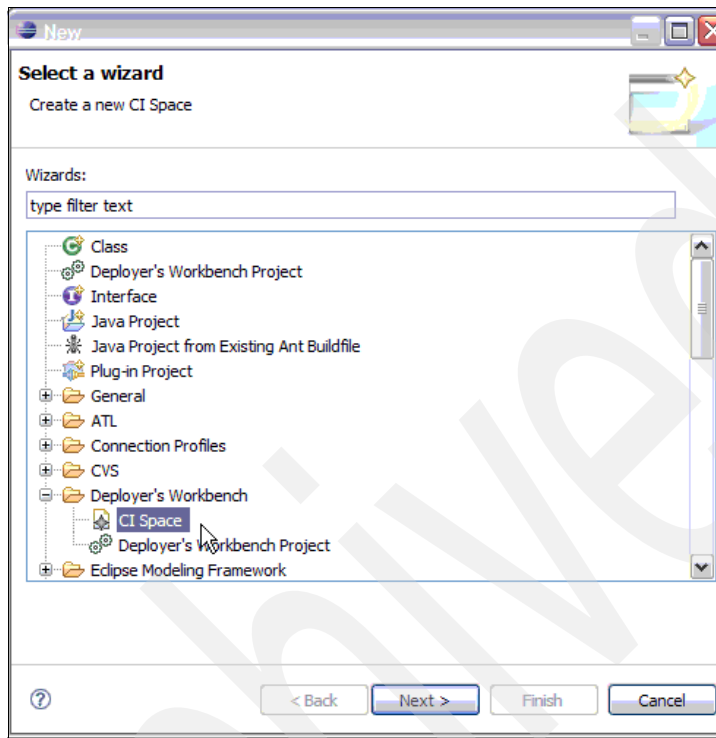


Figure A-10 Creating a new CI space file

3. Next, give the file a name, and click **Finish**.

Important: The file name must end in .cispace.

After the file has been created, your Project Explorer will look something like Figure A-11.

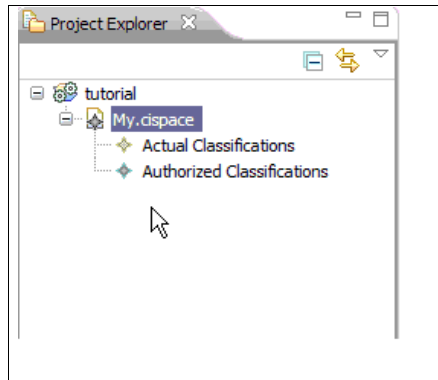


Figure A-11 Creating a new CI space file

As you can see, there is no data within the actual classifications or authorized Classifications items. You will either import or create this information within the workbench.

Synchronizing the CISPAC with the CCMDB

As mentioned earlier, the CI space design tooling provides a partial sandbox environment. The environment allows the user to make as many changes to their local copy of the authorized space prior to actually pushing those changes to the server. Because this work is done in a sandbox, various steps are required to synchronize the sandbox with a running CCMDB server.

Connection setup

Before synchronizing the sandbox, the connection settings for the sandbox must be set. The connection settings are specific to each individual sandbox in the design tooling. That is, each .cispac file contains its own connection settings to allow the user the opportunity to work across multiple servers if needed.

To access the connection settings, you must access the properties of the appropriate .cispac file. To access the connection properties, select the .cispac file, right-click, and select the **Properties** option. Alternatively, you can use the key shortcut, ALT+Enter.

In the CI Spaces Server Properties window, each of the fields must be filled in correctly. The following fields must be provided:

- Database Type: You must select the correct database type for your CCMDB installation. Currently only DB2 and Oracle are supported.

- ▶ Database URL: This is the URL used by CCMDB to access the database. You can find this exact URL within your CCMDB preferences.
- ▶ Schema Owner: This value can also be found in the CCMDB preferences.
- ▶ Username: This value is the database user name with authority to make changes to data within the database. This value can also be found in the CCMDB preferences.
- ▶ Actual Classification Root: This value is the root classification ID of the actual classification tree (CDM) imported into CCMDB by the ITIC adapter. (You must have the actual tree imported into CCMDB before using this tool.)
- ▶ Authorized Classification Root: This value is the single root of your authorized classification tree. If you have not created one in CCMDB, it will be created automatically for you based on this value. If you already have one in CCMDB, make sure to set this field to the correct value.

You can test the connection by using the Test button towards the bottom of the properties window.

Importing the actual classification tree

Before you can edit the CI space in your sandbox, you must import the actual classification tree. This step provides the design tooling the needed information to allow you to browse and copy classifications from the actual to the authorized classification trees:

1. In order to import these classifications, expand the .cispac file, and right-click the **Actual Classifications** element, as shown in Figure A-12. Select the only action available to you: **Import Actual Classification Tree**.

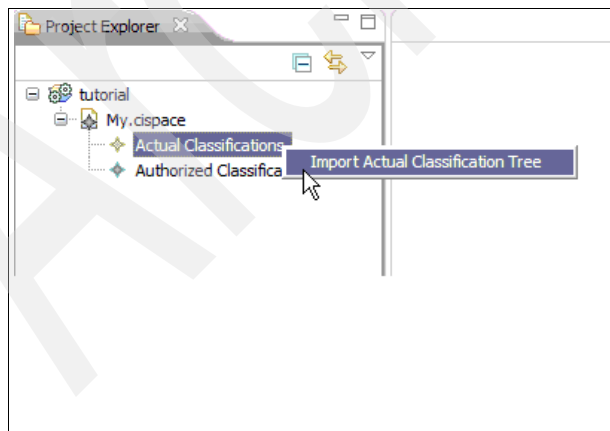


Figure A-12 Importing the actual classification tree

2. As shown in Figure A-13, if this is the first time that you have connected to your server, you will be prompted for a password. Note that the password is not stored anywhere, so each new Eclipse session will require you to enter the password once.

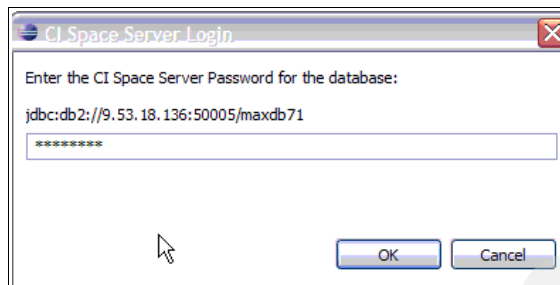


Figure A-13 Importing the actual classification tree

3. After the import completes, the project explorer will contain the entire actual classification tree as seen in Figure A-14.

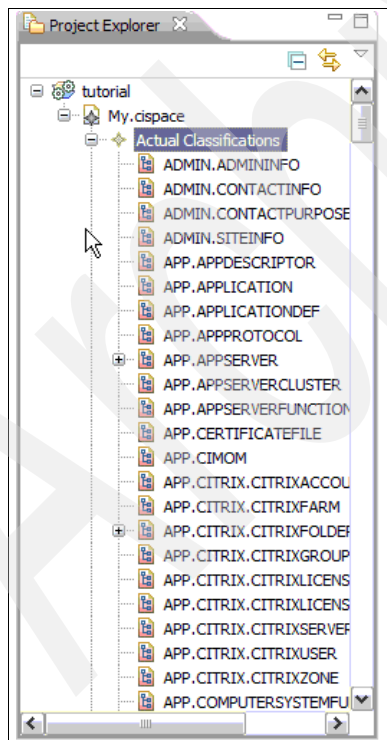


Figure A-14 Importing the actual classification tree

Attention: If your “Actual Classification Root” in the connection properties is incorrect, the import might not discover any information from your server.

Importing existing CI space

If your server was previously configured with authorized classifications, promotion scopes and relationship rules, you must import that preexisting information before beginning to work with the CI space designer. This insures that you are working with the latest information about your server configuration.

Similar to importing the actual classification tree, you will need to expand the .cispac file in the project explorer, but this time you will need to right-click the **Authorized Classifications** item and select **Import Authorized CI Space**. This is shown in Figure A-15.

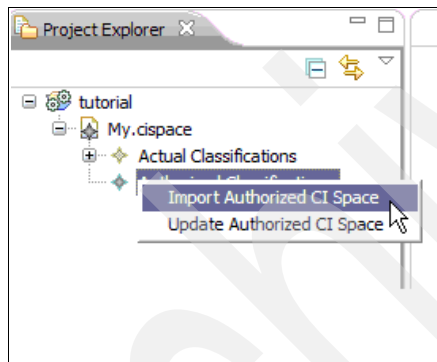


Figure A-15 Importing existing CI space

As with the actual classification import, after the import has completed, you will see the authorized classification hierarchy within the project explorer.

Attention: If your “Authorized Classification Root” in the connection properties is incorrect, the import might not discover any information from your server.

Updating the server

After you have made all of the Changes required to your sandbox, you can “push” these Changes back into CCMDDB. Similar to the import of the authorized classification tree, the “Update Authorized CI Space” action is within the contextual menu on the “Authorized Classifications” item in your project explorer. This is shown in Figure A-16.

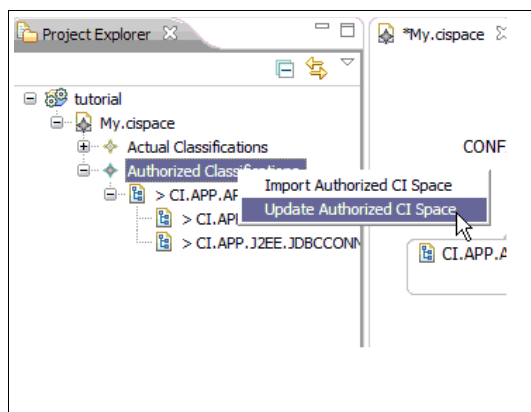


Figure A-16 Updating the server

These are known limitations at the time of writing:

- ▶ **Limited Synchronization:** The current implementation of this tool does not provide a full synchronization with your server. What this means is that you cannot compare what you have in your sandbox against what exists currently on the server.
- ▶ **All or Nothing:** When updating your server, you are not given the opportunity to only update a subset of the configuration information stored in your sandbox.
- ▶ **No Conflict Detection:** The current implementation does not check to see if another Change has occurred on the system because the last time you might have imported the authorized classification information. This behavior currently follows the behavior of the existing web UI.

Default authorized CI space definitions

There are two default best-practice authorized CI space definitions which are included with the Deployer's Workbench.

These best-practice authorized CI space definitions were developed through extensive review of the authorized CI spaces created by various customers and IBM Services teams. You can very quickly deploy them as they are into a development environment and use them in promotions in order to see the resulting sets of authorized CIs that are created. You can use either definition as the starting point of a project and add, delete, or modify elements to make it suit your exact needs. Be sure to delete CI types and attributes that you will not use in your business processes, in order to avoid having extraneous data in your authorized CI space.

The two best-practice authorized CI space definitions delivered as library objects, either of which can be imported into a Deployer's Workbench project. They include the same set of authorized CI classifications; the comprehensive definition includes broader promotion scopes. They are:

- ▶ **Simple Best Practice** - This CI space contains a limited set of promotion scopes, with which many deployments begin in their first deployment phase. It is designed for the deployer who wants to start with a small set of managed CI types and expand the list over time. It includes Computer Systems, Application Servers, Business Applications and Business Services, as well as a small set of attributes and relationship rules. Relationship rules have been modified to ensure that all CIs can be promoted successfully from the Computer System, Business Application, and Business Service classifications.
- ▶ **Comprehensive Best Practice** - This CI space includes a large set of promotion scopes, in an attempt to include everything that is commonly used in Change Management and other processes. It is designed for the deployer who wants to begin with a broad set of managed CI types. It includes a large set of attributes and relationship rules, though not all that are defined in the Common Data Model, to ensure that it can be used by most businesses. The promotion scope includes all CI classifications included in this CI space. Relationship rules have been modified to ensure that all CIs can be promoted successfully from the Computer System, Business Application, and Business Service classifications.

To use either of these Best Practice CI spaces, you simply drag and drop the relevant item from the Library into the CI space in the Project. This process is illustrated below.

Figure A-17 shows the dragging of the Simple Best Practice CI space, in your Project's CI space.

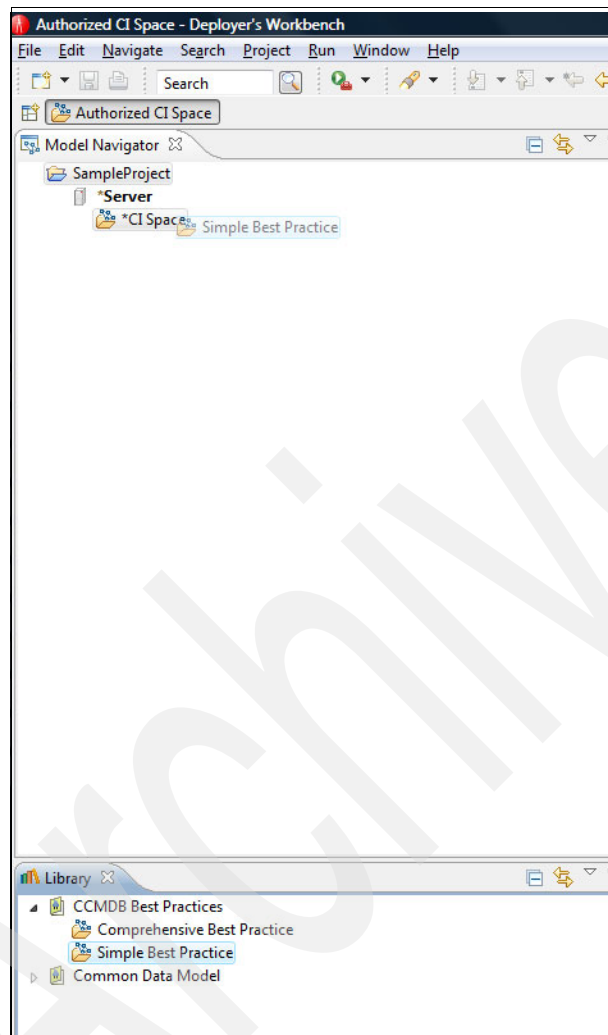


Figure A-17 Simple Best Practice CI space

After you let go of the mouse, the sample CI space will be copied into your project as shown in Figure A-18 through Figure A-20.

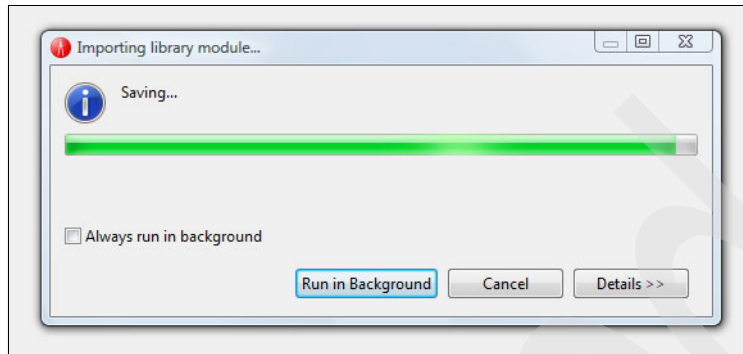


Figure A-18 Sample CI space will be copied into your project -1/3

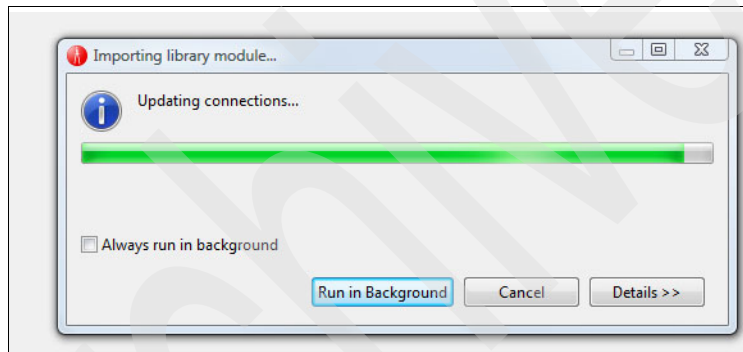


Figure A-19 Sample CI space will be copied into your project -2/3

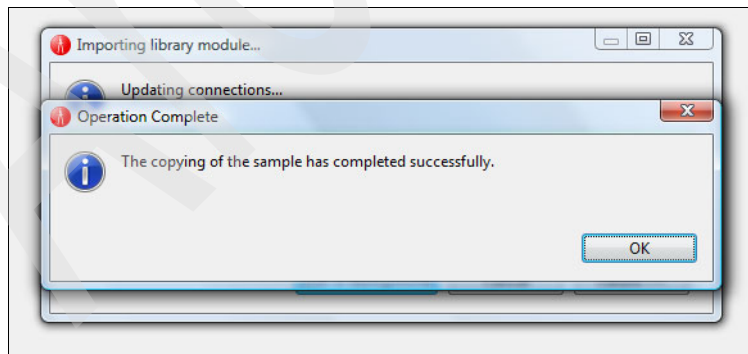


Figure A-20 Sample CI space will be copied into your project -3/3

Your Project will now contain actual and authorized classifications as shown in Figure A-21 through Figure A-24.

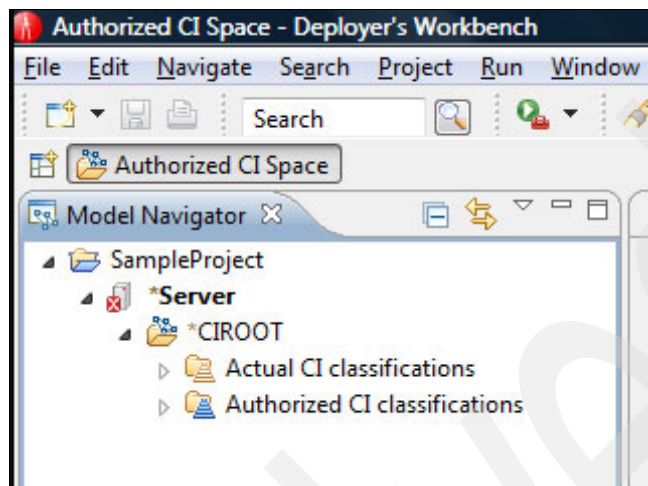


Figure A-21 Actual and authorized classifications 1/4

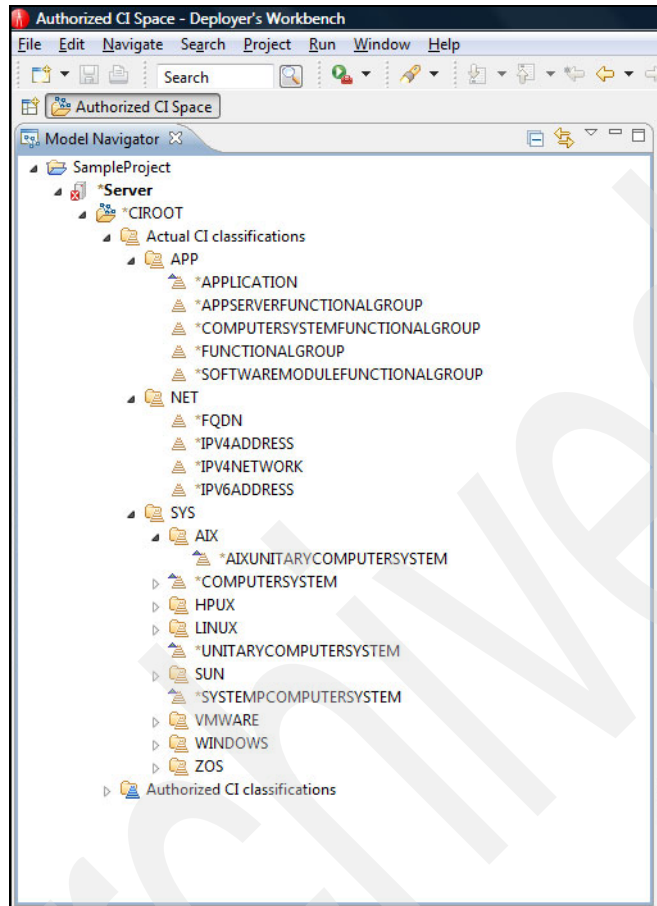


Figure A-22 Actual and authorized classifications 2/4

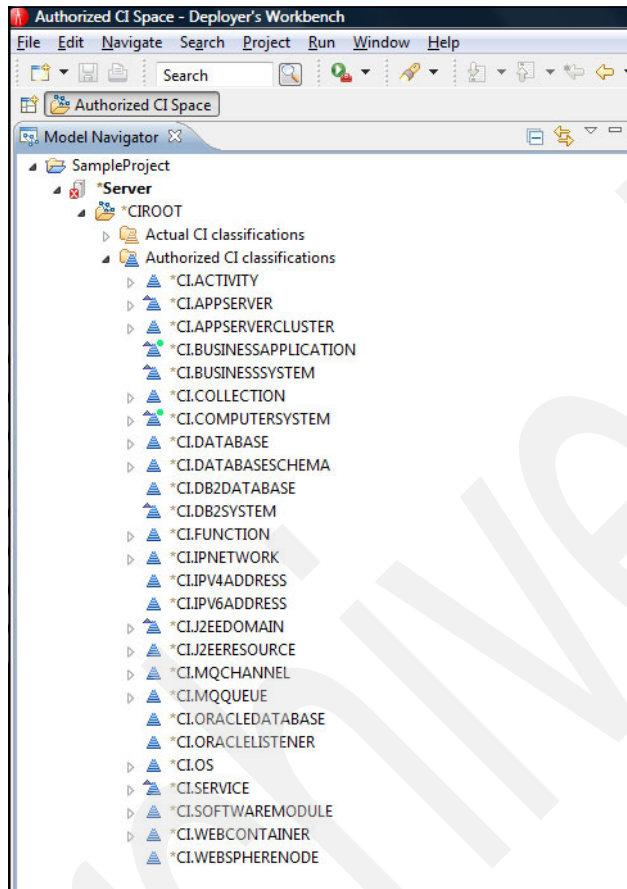


Figure A-23 Actual and authorized classifications 3/4

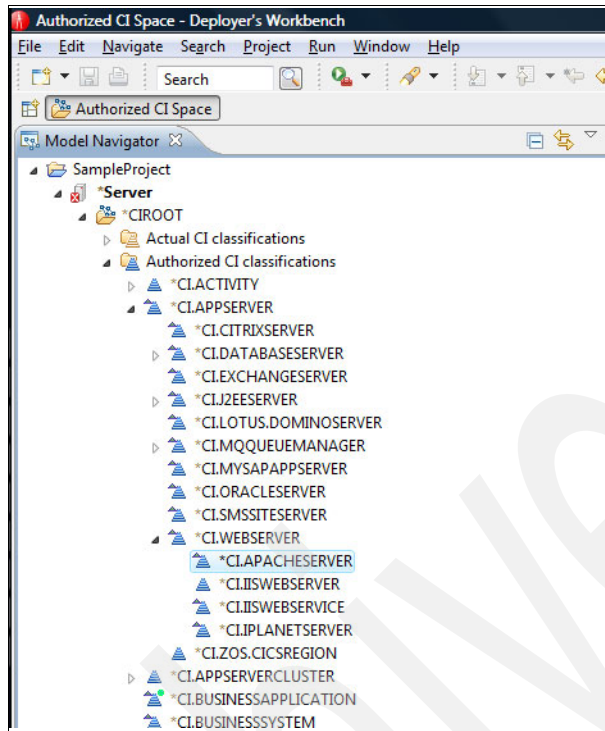


Figure A-24 Actual and authorized classifications 4/4

Figure A-25 shows the dragging of the Comprehensive Best Practice CI space, in your Project's CI space.

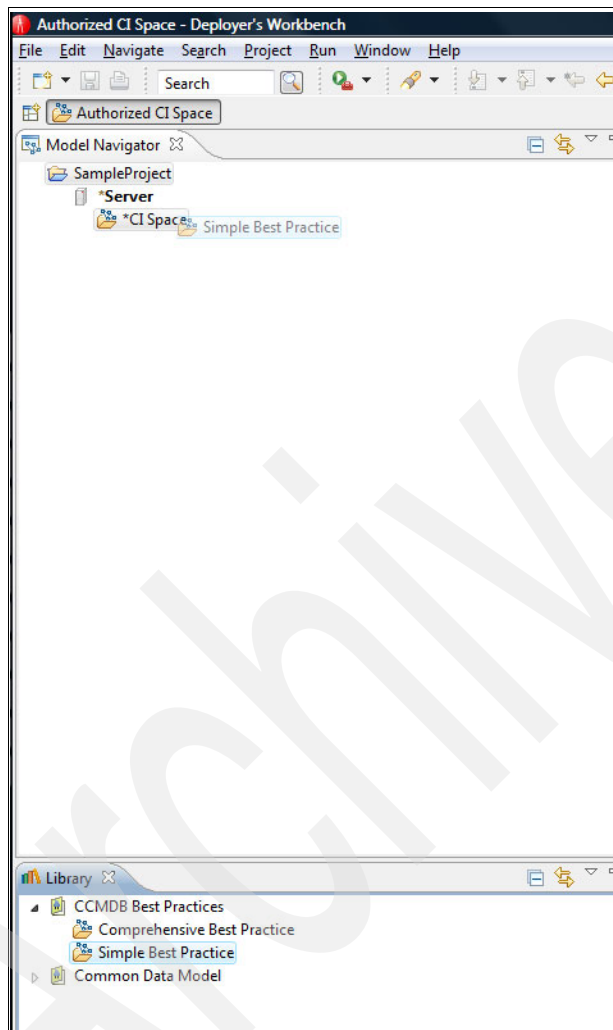


Figure A-25 Comprehensive Best Practice CI space

After you let go of the mouse, the Comprehensive CI space will be copied into your project, which will now contain actual and authorized classifications as shown in Figure A-26 through Figure A-27.

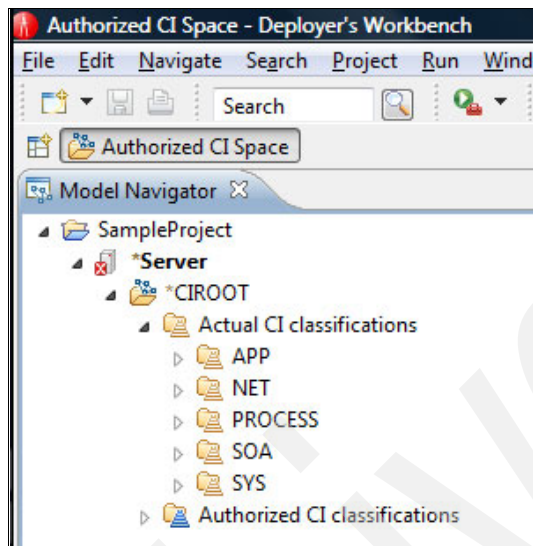


Figure A-26 Sample CI space will be copied into your project - 1/2

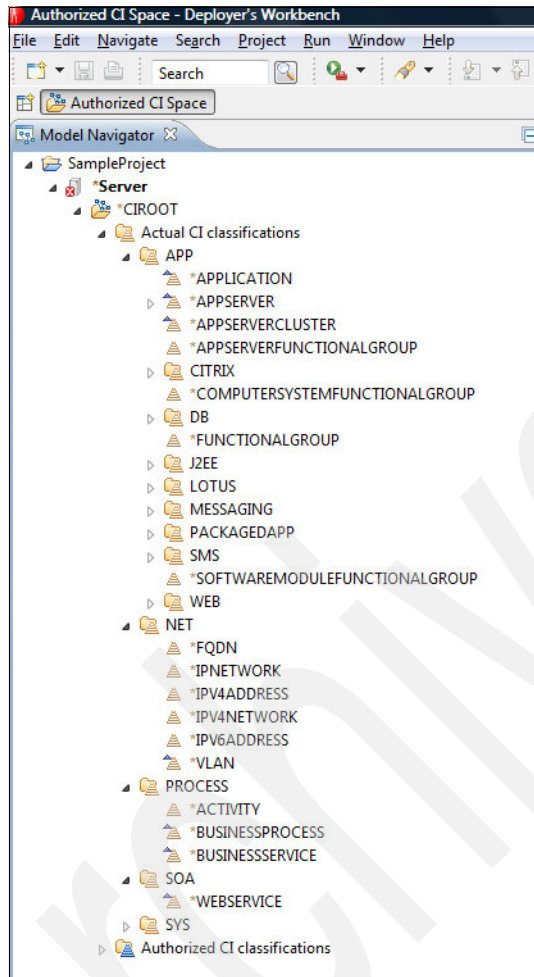


Figure A-27 Sample CI space will be copied into your project -2/2

Editing the authorized classification tree

Editing the authorized classification hierarchy is the first main editing step that must be taken to create a usable CI space for CCMDb. This step requires that a valid network connection exists between you and your CCMDb server in order to properly copy information from the actual classifications while creating the authorized classifications.

Important: You will not be allowed to edit the authorized classification tree unless you have imported the actual classification hierarchy first.

Creating the classification tree

In order to edit or create your classification tree, you must open the editor for your .cispac file. This can be done by double-clicking the file, or choosing **Open With** → **CI Space Editor** from the contextual menu in the project navigator.

The authorized classification tree is where you will build your overall authorized classification hierarchy. There are two ways you can add classifications to this hierarchy:

1. Click the **Add Child** or **Add Sibling** button to create a new empty classification. The location of the classification in the tree is dependant on what classification is currently selected in the authorized classification tree itself. This option is useful if you want to organize your classification hierarchy with a custom set of classifications that have no corresponding actual classification.
2. Drag and drop actual classifications from the project explorer to your left (from within the same .cispac model file) into the authorized classification tree view. This drag and drop operation will copy all of the classifications you've selected, and optionally their children as well, into the selected position in your overall hierarchy. This operation will not only copy the basic classification information, but also the attributes from the actual to the authorized equivalent.

Access: This drag and drop operation requires that you have network access to a CCMDB server, so that it can copy the correct information into your sandbox.

When you drag and drop, you will see an indicator that shows you where your drop will occur.

In Figure A-28 you can see a black line indicating the drop point. On the left most end of the line, there is a small bracket or arrow. Where this bracket/arrow is in relation to the nearest node indicates where in the hierarchy the drop will take place. In Figure A-29 the drop will create a new top level authorized classification on that is a sibling to CI.APP.DB.DB2.DATABASE.

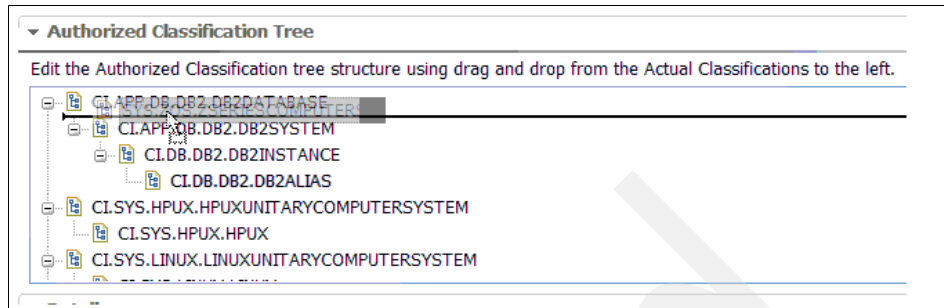


Figure A-28 Creating the classification tree

Within this same tree you also have the option to rearrange the overall structure. This can be done by dragging items within the tree itself from one parent to another.

As shown in Figure A-28, when you select a specific authorized classification in the authorized classification tree, the details section below the tree will be filled in with values from that particular classification.

There is a field in the details section called Related Actual Classification. This field shows what actual classification the selected authorized classification was derived from. For any authorized classification that represents a derived actual classification, this field must be set. This allows the CI space editor keep track of the derived mappings so that attribute and relationship copies can correctly occur.

Access: Attempting to change the Org or Site of a classification will require network access to your CCMDDB server in order to display the correct site specific values.

The attributes table contains the attributes for this classification. Most of these columns are self explanatory, however the last column requires discussion. The last column indicates that whatever rows have been checked, those attributes will be propagated downward to child classifications in the classification hierarchy. If you see the blue downward pointing arrow at the beginning of the attribute row, this indicates that the attribute was propagated to the selected classification from a parent classification.

If at any time you accidentally delete an attribute from a classification, you can recover that attribute by copying it back from the originating actual classification. To perform this action, you must find and select the actual classification in the project explorer. Then select the properties view at the bottom of the workbench.

3. In Figure A-29, APP.APPSERVER was selected in the project explorer, and the properties view has been selected. Within the properties view are two sub-tabs on the left hand side. Selecting the **Attributes** tab, as seen here, will show you all of the selected actual classification's attributes. At this point, you can now find the attribute you need to copy, and drag and drop it from the table in the properties view into the attributes table in the foregoing details section.

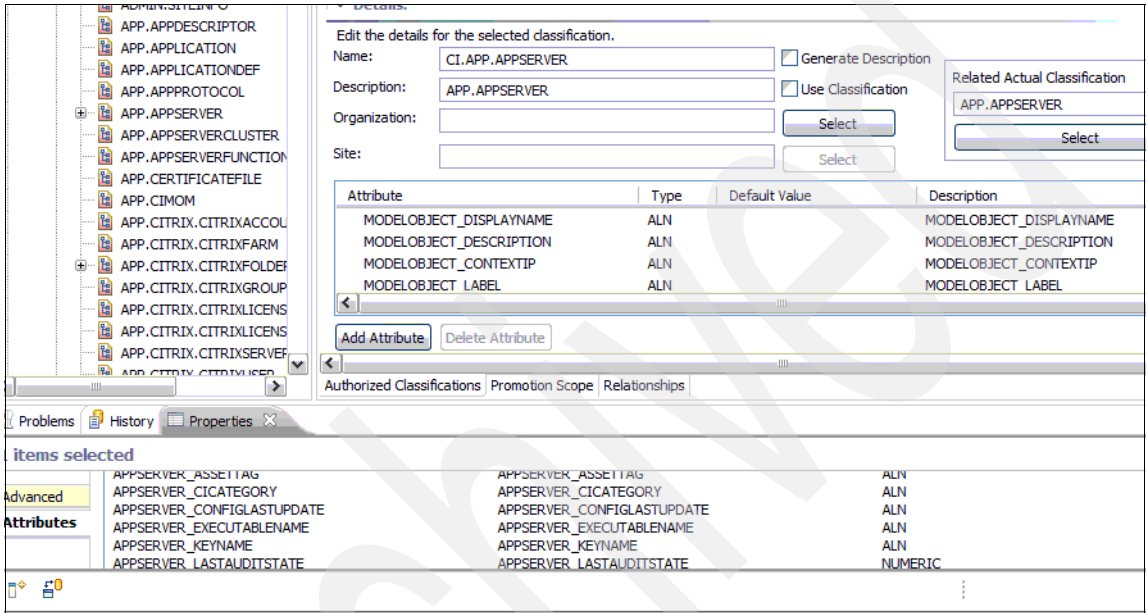


Figure A-29 Creating the classification tree

Editing the promotion scopes

Promotion scopes define what CIs will get promoted from the actual space to the authorized space within the context of the classifications defined within scope. All promotion scopes have a single authorized classification starting point that is unique among all other promotion scopes.

Because each authorized classification can only be the starting point for one promotion scope, the promotion scope editor's contents are keyed off of what authorized classification is selected in the project explorer. By changing the selected authorized classification, the promotion scope editor's contents will adjust accordingly.

The primary goal of this editor is to help the user quickly build up the set of authorized to actual classification mappings that must exist within this scope. When selecting an authorized classification to add to the scope, the editor will try to automatically determine what the actual classification side of the mapping ought to be.

Creating a promotion scope

Promotion scopes are primarily edited in the promotion scope editor. After you have switched to the promotion scope editor, search for and select the starting point authorized classification in the project explorer. If no promotion scope exists, you will see the Create Promotion Scope button enabled, as shown in Figure A-30, otherwise the Delete Promotion Scope button will be enabled.

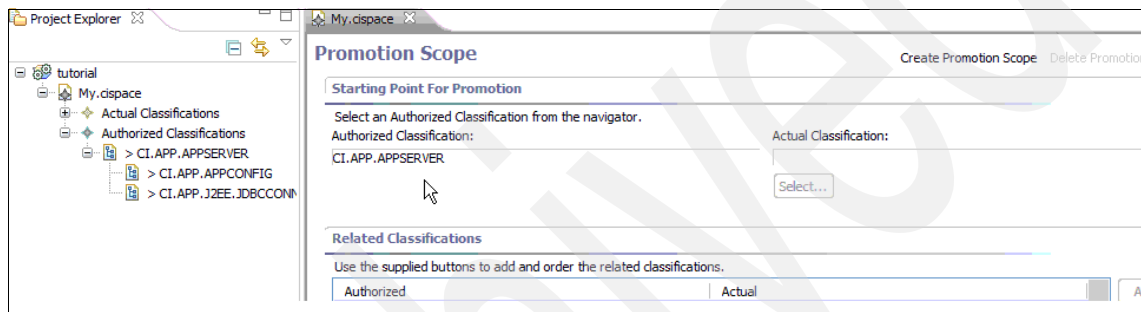


Figure A-30 Creating a promotion scope

Click the **Create Promotion Scope** button. This will create the underlying object, and automatically set the starting points actual classification mapping. This mapping is picked up from the related actual classification that was defined in the authorized classification editor.

Connection: As you add mappings to the scope, the editor will go back to your CCMDB server to acquire relationship rule information to copy into the authorized classification space. This requires a valid network connection to your CCMDB server.

Now that the starting point of the promotion scope has been set, you can add additional related classifications to the scope, as shown in Figure A-31.

The screenshot shows a web application window titled "MyDispace". The main content area is titled "Promotion Scope" and has two sub-sections: "Starting Point For Promotion" and "Related Classifications".

In the "Starting Point For Promotion" section, there is a text input field labeled "Authorized Classification:" containing the text "CI.APP.APPSERVER". To its right, there is a text input field labeled "Actual Classification:" containing the text "APP.APPSERVER". Below the "Actual Classification:" field is a "Select..." button.

In the "Related Classifications" section, there is a table with two columns: "Authorized" and "Actual". The table has one row with the text "Click to set" in both columns. To the right of the table are two buttons: "Add" and "Remove".

Figure A-31 Creating a promotion scope

To add additional related classifications, click the **Add** button to the right of the table at the bottom of the promotion scope editor. When you click the **Add** button, a new row will be added to the table with no values set, as shown in Figure A-32.

This screenshot shows the same "Promotion Scope" configuration window as Figure A-31, but with an additional row added to the "Related Classifications" table. The table now has two rows, both with "Click to set" in both the "Authorized" and "Actual" columns. The "Add" button is still visible to the right of the table.

Figure A-32 Creating a promotion scope

Click in the empty **Authorized** cell, then click the button in the right hand side to bring up a dialog to select the authorized classification, as shown in Figure A-33.

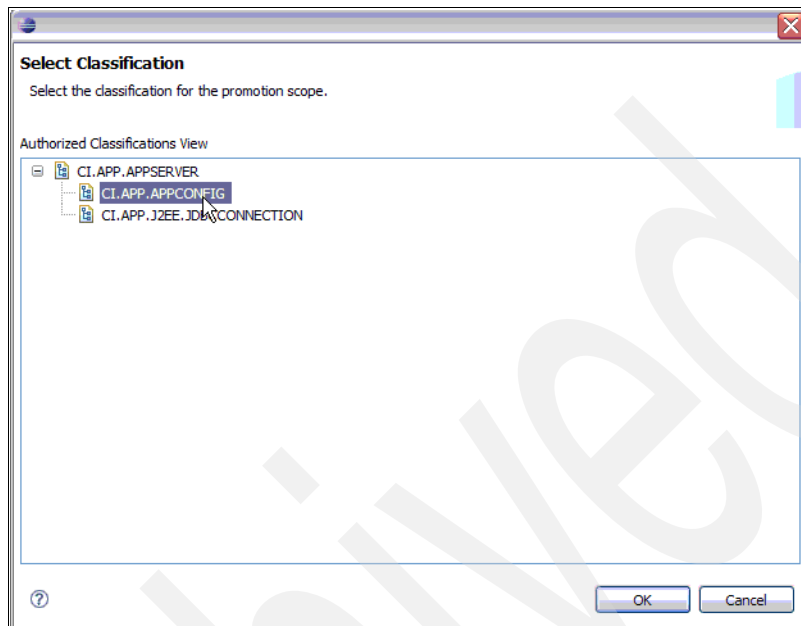


Figure A-33 Creating a promotion scope

After you have selected the authorized classification, the editor will attempt to automatically set the actual side of the mapping for you.

When you believe you have set up all of the mappings for the selected promotion scope, you will need to evaluate and edit the relationship rules within the context of the scope. This is done within the Relationships editor.

Editing the relationship rules

Similar to the promotion scope editor, the relationships editor is keyed off of the project explorer's selected authorized classification. If no authorized classification is selected, or if the selected authorized classification does not contain a promotion scope, the relationships editor will be blank.

If a valid authorized classification is selected, the editor will display a diagram similar to what is shown in Figure A-34.

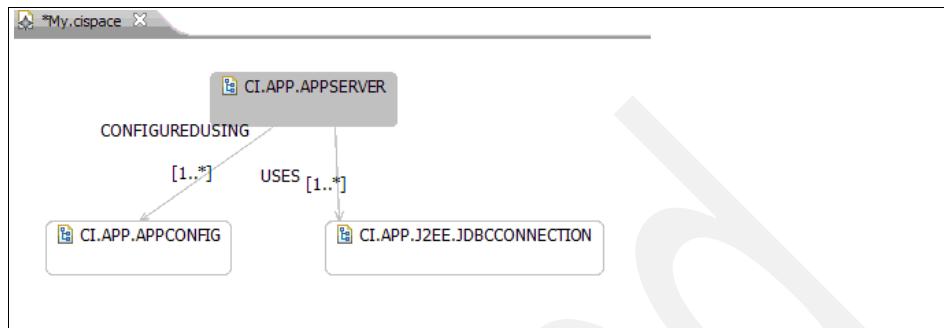


Figure A-34 Editing the relationship rules

This editor differs significantly from the other editors previously described. It provides a visual representation of how classifications relate to each other within the context of a promotion scope. In the foregoing example, you can see the top level classification:

CI.APP.APPSERVER **USES** CI.APP.J2EE.JDBCConnection

When you click a node in the diagram the relationships will highlight to show you all of the valid paths from that node to all related nodes. This visual clue helps you to understand what nodes in your diagram are unreachable by the relationships.

There are times where the elements in the diagram might not be visible on screen, or they are drawn on top of each other. This can be fixed by right-clicking in the editor pane, and selecting the **Arrange All** command.

As you can see from Figure A-34, the primary classification for this promotion scope is CI.APP.APPSERVER and it has two relationships connecting it to each of the other authorized classifications within the promotion scope. This provides you a visual representation of how the promotion of the CI will occur. If for whatever reason, you add an authorized classification to your promotion scope that cannot be reached by traversing the diagram from the starting point, an error will be indicated. You can also select any of the nodes in the diagram and the relationships between the nodes will be highlighted to indicate the valid paths starting at the selected node.

You can change particular attributes on a relationship by selecting the relationship line, and right-clicking, as shown in Figure A-35.

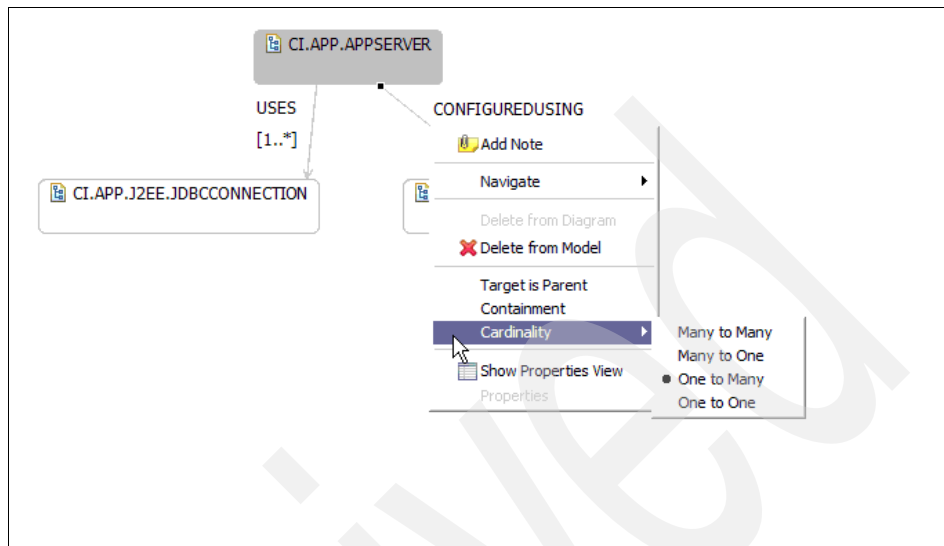


Figure A-35 Editing the relationship rules

You can change the Target is Parent, Containment, and Cardinality of a relationship. If you make a Change that invalidates the relationships due to a traversal error you will be notified as seen in Figure A-36.

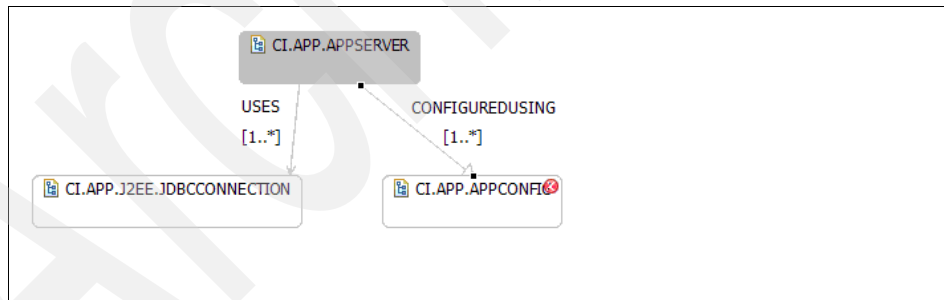


Figure A-36 Editing the relationship rules

This particular error was caused by the user setting the CONFIGUREDUSING relationship to Target is Parent. This causes the CI.APP.APPCONFIG classification to not be reachable.

If you accidentally remove a relationship that you actually needed, there is an easy way to restore relationships that used to exist between particular nodes in the diagram. Select two or more nodes in the diagram then right-click and select **Restore Relationships**.

Finally, you can also create new relationships by hand by using the RelationshipRule tool in the palette. This does require that you know precisely the settings needed to match the equivalent relationship in the actual classification tree.

Attention: From within this editor page, you are not able to delete authorized classifications from the promotion scope. You must edit those mappings from within the promotion scope editor page.

Data validation

The Deployer's Workbench CI space design tools contain a set of validation rules to help the user avoid common mistakes. These rules span from simple, to more analytical. There are two modes of validation:

- ▶ **Live:** Live validation occurs as you make changes to the model.
- ▶ **Batch:** Batch validation occurs when you save the editor to disk. Batch validation performs a model wide validation check to ensure that the model conforms to the defined model constraints.

These are two examples of current validation rules:

- ▶ **Empty fields:** There are a number of required fields within the overall model. Each of these fields are checked as you make changes and during batch validation. Examples of these fields are the name fields in Classifications, Attributes and Relationship Rules.
- ▶ **Accessible classification:** When building a promotion scope, validation is run to insure that all classifications within the scope can be accessed from the starting point of the promotion scope by the supplied relationship rules. This validation occurs during model edits as well as during batch validation.

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 publications

For information about ordering these publications, see “How to get Redbooks publications” on page 582. Note that various documents referenced here might be available in softcopy only.

- ▶ *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565

<http://www.redbooks.ibm.com/abstracts/sg247565.html>

Online resources

These websites are also relevant as further information sources:

- ▶ CCMDB Info Center:
http://publib.boulder.ibm.com/infocenter/tivihelp/v32r1/topic/com.ibm.ccmdb.doc/ccmdb_welcome.htm
- ▶ CCMDB Release Notes:
<http://www.ibm.com/support/search.wss?q=ccmdb72relnotes>
- ▶ CCMDB wiki website:
<https://www.ibm.com/developerworks/wikis/display/tivoliccmbd/Home>
- ▶ TADDM Info Center:
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.2/welcome.html
- ▶ IBM Software Support website for CCMDB:
<http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliChangeandConfigurationManagementDatabase.html>
- ▶ *IBM Tivoli Common Data Model: Guide to Best Practices*, REDP-4389
<http://www.redbooks.ibm.com/abstracts/redp4389.html>

How to get Redbooks publications

You can search for, view, or download Redbooks publications, Redpapers publications, Technotes, draft publications, and Additional materials, as well as order hardcopy Redbooks publications, at this website:

ibm.com/redbooks

Help from IBM

IBM Support and downloads:

ibm.com/support

IBM Global Services:

ibm.com/services

Index

A

- Accept Update CI Request 120
- Accepting or Rejecting the Process Request 129
- Accepting or Rejecting the RFC 179
- ACTCI object 448, 479
- ACTCI table 446
- action group
 - definition 45
- actions 55
 - in action groups 45
- actual CI 349
 - auditing 135
 - authorizing 127
 - checking 124
 - configuration 345
 - promoting 419, 435
 - space 345
- actual CI filter 345
- actual class structure 431
- actual configuration items application 133
 - Description field 479
 - primary MBO 478
- application server 474, 477
 - WebSphere Cell 12
- Approval 190
- assessments
 - Business 61
 - IT 61
- Assigning a Change Owner 181
- Attributes equality comparison rule 146
- Audit Request 128, 132
- authorized CI 15–16, 110, 347–349, 351, 420
 - auditing 127, 135
 - characteristics 110
 - checking 124
 - comparison 146
 - controlling 117
 - creating 348, 350
 - creating in a CCMDB database 417
 - data 347
 - data space 349–350
 - data structure 351
 - defining specific aspects 342

- hierarchy 419
- instance 350
- menu 440
- objects 349
- promoted CI 437
- promoting 350, 419
- promoting actual CI 441
- reconciliation 160
- scheme 351
- setting up comparison rules 145
- space 348, 350–351
- task filters 138

- authorized CI space 14, 16, 426
- authorized classification structure 430
- authorized space 418, 426, 430–431, 434–435, 437
 - selected actual CIs 418

B

- BIRT
 - reporting 533
 - reporting engine 12
 - reporting services 530
- bulkloader 9, 343
- BUSASSESS node 63
- Business Assessment Results 214
- Business Intelligence and Reporting Tools, see BIRT

C

- Calendar View 225
- Cancel an Outstanding Request 252
- Catalog Node and Database 451
- CCMDB
 - adding discovery domains 10
 - Change Assessment tab 52
 - Change Authorization tab 52
 - Change Template tab 52
 - Data Layer 6, 14
 - logical component overview 6
 - Process Layer 6, 10
 - process technology 39
 - workflows for the authorization process 69

- CCMDB Base Services 7, 11
- CCMDB environment 446–447, 469
- CCMDB overview 3, 417
- CCMDB solution 35, 349
- CCMDB Web User
 - user interface 8
- CDM 342
 - purpose 9
- Change Advisory Board (CAB) 169, 191
- Change Advisory Board (CAB) authorization 67
- Change Analyst 168, 186, 217
- Change and Release process interaction 258
- Change Assessment tab 52
- Change Authorization tab 52
- Change implementation 102
 - actual CI 110
 - applying a lifecycle 106
 - change impact analysis 215
 - change implementation schedule 225, 228, 241
 - CI 106
 - CI comparison 146
 - CI lifecycle 102–103
 - CI owner 112
 - CI view 228
 - defining 95
 - defining CI attributes 194
 - deleting an authorized CI 114
 - duplicating a CI 116
 - filter type 137
 - modifying CI attributes 246
 - new CIs 106
 - opening a CI 108
 - opening CI lifecycles 109
 - risks 168
 - selecting the source CI 114
 - updating a CI request 119
 - viewing audit results 158
- Change lifecycle flow 168
- Change Management 14, 167, 169–171, 224
 - activities 38
 - applicable functions 209
 - change impact analysis 215
 - CI auditing 345
 - configuration items 246
 - controlling authorized CIs 348, 418
 - generating a process request 79
 - generating process requests 16
 - interaction with other processes 246
 - ISM processes 349
 - job plan 243
 - performing Changes 78
 - PMP 7, 14, 16, 18
 - process 40, 168, 171
 - process flow 38
 - process flow definitions 55
 - process in CCMDB 177
 - Process Manager 167
 - receiving a process request 47
 - RFC for new release 252
 - roles 173
 - schedule 224
 - updating a CI request 119
 - using CI information for Changes 162
 - value 9
- Change Management PMP 16–17
- Change Manager 117, 169, 171, 179, 182, 206, 208, 224
 - scheduling an RFC 169
- Change Owner 168, 243
- Change Record 168, 246
 - new Task 192
- Change Request 40–41, 110, 184
- Change Template tab 52
- Change Window 112, 193, 225, 238–239, 241
 - repeating and custom scheduling 238
- Change Window Conflicts 241
- Change work order 46
- Changes and releases 250
- CI
 - actual 436
 - Actual and Authorized Versions 110
 - auditing 15
 - authorized 111, 418
 - authorized space 16, 437
 - Change association 259
 - Change implementation 246, 248
 - collections 116
 - comparison 147
 - configuration librarian 96
 - current status 80
 - Current Status Information 80
 - data 109, 136, 246, 345, 490–491
 - data exchange 7
 - data service management 345
 - data set 149
 - data set object 141
 - data structures 19
 - definition 94

- deleting 114
- dependencies 344
- discovered data 10
- discovering 109
- entering into CCMDB 9
- Filter 345
- information 246
- instance 345, 348
- job plan 86
- lifecycle 102
- maintaining 15
- modified by problem resolution 167
- Name 111
- number 111
- object
 - structure 349
- Physical And Logical Connections 95
- promoting an actual CI 435
- reconciliation history view 23
- Relevant Relationships 216
- RFC 15
- selecting 17
- state 418
- target 114
- targeted 40
- type 103, 419
- type adapter 11
- types 102, 106–107, 345–346, 417
 - data 19
 - default lifecycle 103
 - information 20
 - life cycle 106
- updating 114
- updating a process request 17
- CI data
 - space 342–343
- CI information 162
- CI Lifecycle 95, 103
- CI Lifecycle application 15
- CI Lifecycle management 103
- CI number 139, 228
- CI Owner 112, 126, 224
- CI record 136
 - specific site 136
- CI specification (CISPEC) 141, 149
- CI type
 - adapter 11
- class structure 139, 141–142, 150
- classifications application 112
- CMDB
 - definition 4
- CMS
 - objective 35
- COBIT 84
- Common Configuration Item PMP 7
- Common Data Model, see CDM
- Common PMP 7
- Common Process Manager Product 13
- comparison rule 135–136, 145–148, 154–156, 159
 - brief description 147
 - following information 156
 - special type 146
- comparison rules
 - types 146
- computer system 19, 344
 - Authorized CIs 348
 - authorized CIs 350
 - software components 344
- Condition Node 42
- configdb command 448, 475–476
- configuration item (CI) 167, 259
- configuration items application 108, 110, 126, 133, 135, 436, 438, 441
- Configuration Management 7, 10–12, 15, 55, 78–79, 246, 344–345, 348
 - process requests 15
- Configuration Management Data Base (CMDB) 95, 163, 223, 246
- Configuration Management PMP 14
- Configuration Management Roles 96
- Configuration Management System, see CMS 35
- Configuration Manager 16, 78, 96, 120, 123, 128–129, 169, 249
 - PMP 7
- Configuration Process Record 121–123
- Configure Parameters 450
- Control CI 15
- Create a new Release 261
- Create a Wrapper 459
- Create an Update CI Request 119
- Create new lifecycle 103
- cron task 135–136, 154, 156–158

D

- data classifications to be promoted 426
- Data Layer 6, 14, 351
- data model 11, 19, 109, 419

- data sources
 - federated 10
- database configuration application 12
- database table 446–448, 467, 469, 472, 478, 480, 491
- default state 102, 108
- define lifecycle transitions 106
- Define Relationships 113
- Defining CI Attributes Modifications 194
- Definitive Hardware Store 257
- Definitive Software Library (DSL) 257
- dialog box 106, 116, 156, 225, 239, 246, 249–252, 419, 437, 441
 - thorough description 246
 - upper right corner 246
- discovered CI space
 - Actual CI space 345
 - CI type 345
 - computer system 347
- discovered configuration items 343
- discovery domains
 - adding 10
- discovery environment 6, 19
- Discovery Library
 - Adapter 343
- DRDA 461
- duplicate authorized configuration item 116

E

- Emergency Change Advisory Board (ECAB)
 - authorization 67
- Emergency Change Job Plan (PMCHGEMERG) 60
- enhanced Telecom Operations Map (ETOM) 84
- escalation services 12
- eTOM 84
- external system 8

F

- FED_DATA table 447, 449, 451, 472, 476, 492
- federated data 10
- federated data source 447–448, 472, 488, 491–492
- Federation at the Database Layer 449
- Federation scenario 446
- federation server 444, 449, 451, 459, 465
 - user ID 465
- filter

- actual CI 345
- Filter clause 137, 147–148
- flow control, work orders 67
- Forward Schedule of Change (FSC) 169
- Full CI comparison
 - rule 147
 - specification 147

G

- graphical user interface (GUI) 85

H

- highlighted implementation task
 - impacted CI list 224
- host name 446, 463, 479, 482, 490–491
- hostname attribute 482

I

- IBM Service Management (ISM) 84–85, 95
 - Architectural context 85
- IBM Solution Console (ISC)
 - working with CCMDB 8
- IBM Tivoli Release Process Manager (TRPM) 255
- IdML 9
- impact analysis 17, 186, 189, 213, 215, 217–219, 222
- implementation note 187, 213, 223
- Implementation Task 186, 189, 193, 195, 213, 218, 221, 223–226, 241
 - required sequence 194
 - scheduled dates 223
- Implementing the Tasks 197
- Information Technology Infrastructure Library, see ITIL
- Initiate Tasks 123
- Integration Adapter 11, 19–20
 - TADDM 7
- Integration Composer 10, 19
 - TADDM 7
- Integration Module 15
- Integration services 12
- ISCHANGE workflow 43
- ISCONFIG workflow 43
- ISM 11
- ISMSUBMIT workflow 43
 - generating a Change work order 56
- ISPMF workflow 43

ISRELEASE workflow 43
IT Infrastructure Library 84
Item Specification (ITEMSPEC) 149
ITIC

 adapter 417
ITIL
 Change Advisory Board (CAB) authorization 67
 Emergency Change Advisory Board (ECAB)
 authorization 67
 service transition principles 37
ITUP 38

J

J2EE application
 job plan 18
job plan 90, 130, 256–257, 260
 application 87
 applying 122
 applying to a work order 47
 applying to configuration process 132
 assessing Changes 212
 assigning to a work order 54
 automated task 243
 Automatic Task 188
 automatic task 188
 calling an action 75
 classifying 88
 configuring for PMP 88
 defining 46
 defining process flow activities 67
 defining templates 49
 description 48
 initiating the first activity 185
 J2EE application 18
 nested 54, 92
 nesting 53, 67
 pre-approved 18, 49
 record 87, 93–94
 reviewing 130
 selecting 183
 standard 49
 standard Change 18, 51
 status 94
 task 88–90, 92, 124
 tasks 51, 80
 template 41, 64, 80
 template task definition 81

L

Launch in Context 15
lifecycle 135
 ID designation 104
lifecycle state 103, 105–106, 108
lifecycle transitions 106
link rule 135–136, 140, 144, 154, 156
 brief description 140
 component 140
 following information 156
 tab 144
logical component
 overview 6
Logical Management Operation (LMO) 15

M

Manage CI Collection 116
Managed Software Systems (MSS) 9
master workflow 43
Matches found comparison rule 146
MAXDB71 446
MAXDB71 database 446–447, 450–451, 460, 465, 476
Maximo Business Object (MBO) 447
 create 470
Maximo Enterprise Adapter (MEA) 349
MEA
 integration services 12
Modifying an existing lifecycle 107

N

NET.IPIN TERFACE 427
Nickname 467
Normal Change job plan (PMCHGNORML) 60

O

OMP
 connecting to a PMP 14
Operational Management Product (OMP) 109
Operational Management Product, see OMP

P

Performing Business Change Assessment 189
physical component 6, 344
PMCHGAP workflow 69
PMCHGAUTH workflow 69
 risk levels 69

- PMCHGBUSASSESTYPE domain 214
- PMCHGCRAWF workflow 78
- PMCHGCRWF workflow 79
- PMP 7
 - Change Management 7, 16
 - Common 7, 13
 - Common Configuration Item 7
 - Configuration Management 14
 - Configuration Manager 7
 - definition 13
 - process request 14
 - pmp.submit.workflow 56
- Post Implementation Review, see PIR
- PREVIEW node 62
- primary MBO 447–448, 478
- PRM-IT 37
- Process definition 84
- process environment 19–20
- process flow 35–36, 38, 41, 43, 46
 - components 35
 - supporting an external system 80
- Process Layer 6
 - database 12, 19–20
- Process Layer database
 - storing reports 12
- Process Manager 45, 83–84, 86
 - implementation 7
 - product 7, 45
 - product (PMP) 13
 - starting point 87
 - type 47, 119, 128, 177, 210
- Process Manager Product (PMP) 11
- Process Manager Role 85
- Process Reference Model for IT 37
- Process Request 14, 42, 47, 78–80, 87, 119, 124, 128–132, 168–169, 177, 179, 199–200, 209–210, 246, 248–250
- process request 14
- process runtime
 - environment 444, 492
- Progress Map
 - indicating process flow 43
 - route button 44
- Projected Service Outage (PSO) 169
- promotion 417

R

- Read-only field 142–143, 150

- Reconciliation 133
- Reconciliation Applications 134
- reconciliation task 134–137, 140, 145–146, 154–156, 158
 - link rule 140
 - link rules 140
 - optional component 145
 - required component 140
- Redbooks website 582
 - Contact us xv
- Register Server 462
- Relationship 478
 - relationship definition 481
 - relationship definition overview 478
- Release application 253
- Release Management 163, 169, 246, 250
- Release process 254, 261
- Release Process phases 257
 - Communicate, Prepare 257
 - Design, Build 257
 - Distribute, Install 258
 - Plan 257
 - Plan Rollout 257
 - Test, Accept 257
- Request for Change, see RFC
- Resolving Discrepancies 160
- RFC 112, 119, 184
 - accepting or rejecting 179
 - approving or rejecting 192
 - assessing 171
 - Change management process 16
 - Change PMP 120
 - changing a CI 15
 - closing 208
 - creating 16, 260, 265
 - creating from a release 258
 - evaluating 215
 - for protected state 102
 - in CCMDDB Data Layer 342
 - job plan 93
 - process request 40
 - reasons for request 167
 - requesting 56
 - reviewing 124
 - sending to Change Approver 169
 - submitting 37, 56, 177
- role
 - Process Manager 85
- route button

Progress Map 44
Route Tasks 188

S

Schedule creation 195
Scheduling Tasks 157
security group 16
Selecting an Appropriate job plan 183
Service Management 11, 345, 348, 350–351
Service Management architecture 7
service transition principles 37
stakeholders 257
Standard Change Job Plan (PMCHGSTAND) 60
Start Center 88, 96–97, 101, 129–130, 171–172, 179, 181, 186, 190, 197, 210
Submitting a Configuration Request 199
sub-tab 88, 212, 214
SYS.OPER ATINGSYSTEM
 class 428

T

TADDM
 federation capabilities 10
 IBM Redbooks 8
 Integration Adapter 7
 Integration Composer 7
TADDM server
 user interface 8
Target Analysis 218–219
target CI 114, 223–224
 Time Window View 227
target database 20
task definition
 Predecessor Definitions 80
Task Dependencies 54
task filter 135, 137, 154
 brief description 137
tasks predecessors
 defining 73
 relationships 73
tasks status 197
TECASSESS node 63
Technical Assessment Results 212
templates
 defining for process flows 49
Time Window View 226
Tivoli Application Dependency Discovery Manager (TADDM) 20, 109, 343

Tivoli Configuration Manager (TCM) 14, 258
Tivoli Provisioning Manager (TPM) 14
Tivoli Unified Process, see ITUP 38
top-level object 140, 146, 148
 class structure identifiers 142, 150
 specific attribute 146
Tracking the progress of a Change 242

U

user interface 10, 12, 84–86, 349–350, 470, 474, 483, 486, 490, 492
 Authorized CIs 349
 behavior 8
 CCMDB Web User 8
 TADDM server 8
user interface layer 8
User Mapping 465

V

value list 138, 141–143, 149–151
 system displays 150
Verify / Audit CI Process 127
verify and audit ci 15

W

WebSphere Cell
 application server 12
WebSphere Federation Server 9.1
 component 444
 implementation 444
WOACTIVITY object 65
WOCHANGE object 43, 57
work order
 Change 46
 child object 64
 creating 46
 definition 46
 field click 131
 flow control 67
 generating 47, 80
 record number 131
 starting tasks 67
work order plan
 description 48
Work Product 171, 221
workflow
 benefits 41

BUSASSESS node 63
changes in new CCMDB version 55
CLASSIFY subprocess 58
Condition Node 42
defining an assisted workflow 78
engine 49
generating a process request 80
ISCHANGE 43
ISCONFIG 43
ISMSUBMIT 43
ISPMF 43
ISRELEASE 43
master 43
PMCHGAP 69
PMCHGAUTH 69
PMCHGCRAWF 78
PMCHGCRWF 79
PREVIEW node 62
process 56
START node 43, 58
STOP node 43
TECASSESS node 63
using action groups 75
using an interaction node 79
VALIDATE node 58



Redbooks

IBM Tivoli Change and Configuration Management Database (CCMDB) V7.2.1 Implementation Guide

(1.0" spine)
0.875" <-> 1.498"
460 <-> 788 pages



IBM Tivoli Change and Configuration Management Database (CCMDB)

V7.2.1 Implementation Guide

Understand the CCMDB architecture and its implementation

Experiment with the CCMDB V7.2 and V7.2.1 new features

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). 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) V 7.2.1 product. It covers the new features available with CCMDB V7.2 and CCMDB V7.2.1, as well as details about the underlying components of the product and utilizing the product to support robust IT processes such as change and configuration management. It also focuses on details of the data model, process engine, and the Change and Configuration Management Process Management Programs (PMPs).

This book provides a reference for IT Specialists and IT Architects working with the CCMDB V7.2.1 product.

**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