Tivoli software

IBM

IBM Tivoli CCMDB Implementation Recommendations

- Gather requirements and document processes
- Extend the CCMDB data model
- Customize processes for your needs

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

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM Tivoli CCMDB Implementation Recommendations

May 2008

SG24-7567-00

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

First Edition (May 2008)

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

Contents

Figures

Notices

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

Redbooks (logo) ®	Extreme Blue™	Redbooks®
z/OS®	Informix®	Tivoli®
DB2®	IBM®	WebSphere®
DRDA®	IMS™	
Enterprise Asset Management®	Maximo®	

The following terms are trademarks of other companies:

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

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

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

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

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

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

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

Preface

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 valuable to those who want to plan for, customize, and use the IBM Tivoli CCMDB product to automate and manage change and configuration processes in their environments. It includes three parts:

- Understanding and documenting requirements: Provides the reader with the context around typical client requirements for change and Configuration Management and describes the IBM Tivoli Unified Process Composer that is used to document these processes.
- Using and customizing the CCMDB data model: Provides important details about the data model, its key concepts, and how one can enhance the data environment through federation.
- CCMDB Process Engine and PMPs: Describes details about the underlying process engine and the Change and Configuration Process Management Products. In addition, this part describes how the reader can customize the default PMPs to meet client requirements.

A companion book, *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565, provides a more general overview of the CCMDB product and information related to planning and installation of the product.

The team that wrote this book

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

Figure 1 Left to right: Michael Brokmann, Rainer Hoppen, Annelie Meels-Kurz, Rosemeire Oikawa, Tadeu Stellet Teixeira, Douglas Barranqueiros Gomes, Arsalan Lodhi, Kapil Madaan, Bart Jacob (not shown, Scott Dickerson)

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

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

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

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

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

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

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

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

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

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

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

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

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

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Part 1

Understanding and documenting requirements

1

CCMDB overview

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

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

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

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

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

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

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

This book focuses on:

- Gathering and documenting requirements
- Working with and extending the data model
- Understanding and customizing the Change and Configuration Process Management Programs

We highly recommend that this book be used in conjunction with *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565, which provides a more general overview of the CCMDB product and information related to planning and installation of the product.

2

Understanding client requirements

This chapter provides an overview of points to consider when planning an implementation of CCMDB from both the business and processes perspective.

2.1 Governance

Organizations that wish to successfully reach their strategic goals need an explicit understanding of governance and their approach to it. Unfortunately, many individuals are confused about exactly what governance is, and what constitutes good governance, why organizations should care about governance, and why IT governance is becoming important.

Within IBM, a widely accepted definition for IT governance is:

- Governance that pertains to an organization's information technology activities and the way those activities support the goals of the business
- Decision making rights associated with IT as well as the mechanisms and policies used to measure and control the way IT decisions are made and carried out within the organization

2.1.1 Why governance matters

Focus on an enterprise's core competencies has lead to an increase in the outsourcing of non-core competencies (and thus leaving other functions and competencies to other companies). This outsourcing is producing a growing interdependency between organizations. The consequence of this is that resources are often organizational and geographically dispersed, sometimes across different countries and even across different continents. As an enabler of this interdependency, IT operations is critical. A good IT governance is key to making certain that IT services are delivered with acceptable quality and availability. The key elements that define good governance are:

- Focuses on achieving strategic goals.

  IT Governance specifies the rules and procedures for making decisions. It also provides the structure through which the objectives are set and the means for controlling and monitoring the performance of those objectives.

- Helps an organization reach its goal.

  IT Governance monitors whether outcomes are in accordance with plans. Governance is the mechanism by which individuals are motivated to align their actual behaviors with others to achieve a common goal. Governance encompasses policies, processes, and people.

- Benefits business and creates value.

  Good IT governance improves perceived quality of services. Quality is determined by how policies and processes are implemented and how people are led. Quality is also determined by the goals that are achieved and how they are achieved (within planned budget and time).

- Helps mitigate risks.

  Governance mitigates risks by enabling good communication and establishing effective measurement and control.

Note: CCMDB enables the controlling of established governance processes through the Process Manager Product (PMP), which enables management to implement a service management process flow. CCMDB has Configuration and Change Management.

2.2 The need for a Change Management process

According to the ITIL V3 Service Transition book, what all high-performing IT organizations have in common is a culture of Change Management that prevents and deters unauthorized change. Those organizations also “trust but verify” by using independent detective controls to reconcile production changes with authorized changes, and by ruling out change first in the repair cycle during outages. Finally, these organizations also have the lowest mean time to repair (MTTR). Auditors will appreciate that in these high-performing IT organizations, Change Management is not viewed as bureaucratic, but is instead the only safety net preventing them from becoming a low-performer. In other words, IT management owns the controls to achieve its own business objectives, efficiently and effectively.

With businesses depending on IT services, it is vital that changes that may potentially affect the production environment are properly evaluated and approved before their implementation, so a Change Management process is necessary.

According tot he ITIL V3 Service Transition book, achieving a change success rate over 70 percent is possible only with preventive and detective controls.

2.2.1 Value to business

Reliability and business continuity are essential for the success and survival of any organization. Service and infrastructure changes can have a negative impact on the business through service disruption and delay in identifying business requirements, but Change Management enables the service provider to add value to the business by:

- Prioritizing and responding to business and customer change proposals
- Implementing changes that meet the customers' agreed service requirements while optimizing costs

- Contributing to meet governance, legal, contractual, and regulatory requirements
- Reducing failed changes and therefore service disruption, defects, and re-work
- Delivering change promptly to meet business time frames
- Tracking changes through the service life cycle and to the assets of its customers
- Contributing to better estimations of the quality, time, and cost of change
- Assessing the risks associated with the transition of services (introduction or disposal)
- Aiding productivity of staff through minimizing disruptions due to high levels of unplanned or “emergency” change and hence maximizing service availability
- Reducing the *mean time to restore service* (MTRS) through quicker and more successful implementations of corrective changes
- Liaising with the business change process to identify opportunities for business improvement

Definition: Change Management will control all changes to all configuration items (CIs) in the managed environment by:

- Ensuring standardized methods, processes, and procedures are used for all changes from the request for change to the post-implementation review
- Facilitating efficient and prompt handling of all changes
- Minimizing the impact of change-related incidents upon service quality, thus improving the day-to-day operations of the organization
- Ensuring that all changes are assessed, approved, implemented, and reviewed in a controlled manner

This process is responsible for controlling and managing requests for change (RFCs) to the IT environment, from inception through implementation.

2.2.2 Steps for implementing change

Change Management should be implemented in conjunction with Configuration Management in order to ensure that impact assessments will be accurate before approving and implementing changes. Items to consider include policies, objectives, scope, inputs, and outputs.

Policies

Again, from the ITIL V3 Service Transition book, policies that support Change Management include:

- Creating a culture of Change Management across the organization where there is zero tolerance for unauthorized change
- Aligning the service Change Management process with business, project, and stakeholder Change Management processes
- Prioritization of change, for example, innovation versus preventive versus detective versus corrective change
- Establishing accountability and responsibilities for changes through the service life cycle
- Segregation of duty controls
- Establishing a single focal point for changes in order to minimize the probability of conflicting changes and potential disruption to the production environment
- Preventing people who are not authorized to make a change from having access to the production environment
- Integration with other service management processes to establish traceability of change, detect unauthorized change, and identify change related incidents
- Change windows and enforcement and authorization for exceptions
- Performance and risk evaluation of all changes that impact service capability
- Performance measures for the process, for example, efficiency and effectiveness

Objectives

The objectives of a Change Management process could include:

- Facilitating the timely introduction of business benefit and enhanced user productivity
- Minimizing the risk of disruption to IT services
- Minimizing incidents caused by changes
- Ensuring the accurate assessment of the cost of proposed changes before they are incurred
- Allowing the absorption of changes at the rate required for business and technical purposes
- Generating enhanced perception of the quality of IT

- Balancing the business need for innovation with the business need for stable IT service, by using standard and repeatable methods for everything that occurs from the RFC to the PIR.

Scope

The process starts with the recognition of the need to put in place and define a management system to control change, including procedures and policies; it ends with the change being installed and activated.

The process includes managing changes from the creation of a request, its assessment, through to a deployment monitoring and post-implementation review.

The process also encompasses trend analysis and measurement reporting.

Typically, scope changes include hardware, communications equipment and software, system software, live application software, all documentation and procedures associated with running, and supporting and maintaining the production environment, which includes:

- Planned changes, standard changes (pre-approved by policy), and emergency changes (policy exception request)
- Application and infrastructure changes
- Establishing both recurring and one-time only schedules (change windows) during which changes may be performed without negatively affecting projected availability or SLA commitments
- Enforcement of standard methods and procedures from request for change through post implementation review
- Establishing regular meetings and communication schedules to evaluate proposed changes and schedules
- Control and management of the implementation of those changes that are subsequently approved
- Maintenance of open channels of communications to promote smooth transition when changes take place
- Increased visibility and communication of changes to both business and support staff

Examples of items that are excluded:

- The process of Change Management is principally managing the change; the process does not include the technical design and testing of the change.
- The process does not include the actual implementation of the change, but manages and coordinates the implementation of the change that may be performed by calling another process, for example, release management.
- Changes to ongoing projects are out of scope.
- Configuration Management.
- Hardware faults or repairs that do not alter the form fit or function of a tracked CI.
- Solution development and testing.

Inputs and outputs

A Change Management process can include the following inputs and outputs.

- Inputs
  - Service Request
  - RFC
  - Operational Schedules
  - Asset Deployment Items and Data
  - Project Plan
  - Validated Solution Design
  - Accepted Solution
  - Configuration Information
  - Release Acceptance Request
  - Implementation Progress Data
- Outputs
  - Project Proposal
  - Implemented Change
  - Change Information
  - Asset Deployment Inquiries and Requisitions
  - Change Implementation Communication
  - Forward Schedule of Change
  - Incident
  - CI Data Update Package
  - Release Acceptance
  - Authorized RFC
  - Closed RFC

Activities

The following is a list of key activities involved in a Change Management process:

- Establish Change Management Framework
- Accept and Categorize Change
- Assess Change
- Approve and Schedule Change
- Coordinate Change Implementation
- Prepare, Distribute, and Implement Change
- Review and Close Change
- Monitor and Report Change Management
- Evaluate Change Management Performance
- Controls
- Service Catalog
- IT Strategy
- SLAs OLAs UCs
- Architecture Baselines and Roadmaps
- IT Plan
- IT Management Ecosystem
- Compliance Plans and Controls
- Configuration Baseline Report

2.2.3 Change Management measurements

Effective day-to-day operation and long-term management of the Change Management process requires use of metrics and measurements throughout the process. There are also a variety of reports that need to be defined, executed, and distributed to enable the management of problems:

- Number of changes approved, rejected, deferred, and implemented in the period in total, and by CI type and service
- Breakdown of reasons for change
- Number and percent of successful changes
- Percent of emergency changes
- Number and percent of approved changes that are backed out, with reasons for the back out
- Number of incidents tracked to changes with severity levels
- Number and trends in RFCs
- Size of change backlog
- Breakdown of changes by Type and Service Area (all categories)
- Number of changes handled per change team member

- Number of changes that avoid the process
- Value of process improvement recommendations
- Human effort required to perform process activities
- Elapsed time and costs required to perform process activities
- Number of changes that avoided the process

Critical success factors

Some of the critical success factors in implementing a Change Management process are:

- Change policies are clear and known and they are rigorously and systematically implemented.
- Change Management is strongly integrated with Release Management and is an integral part of Configuration Management.
- There is a rapid and efficient planning, approval, and initiation process covering identification, categorization, impact assessment, and prioritization of changes.
- Automated process tools are available to support workflow definition, pro-forma work plans, approval templates, testing, configuration, and distribution.
- Expedient and comprehensive acceptance test procedures are applied prior to making the change.
- A system for tracking and following individual changes, as well as change process parameters, is in place.
- A formal process for hand-over from development to operations is defined.
- Changes take the impact on capacity and performance requirements into account.
- Complete and up-to-date application and configuration documentation is available.
- A process is in place to manage co-ordination between changes, recognizing interdependencies.
- An independent process for verification of the success or failure of change is implemented.
- There is segregation of duties between development and production.

Performance metrics (COBIT)

- Number of different versions installed at the same time
- Number of software release and distribution methods per platform
- Number of deviations from the standard configuration
- Number of emergency fixes for which the normal Change Management process was not applied retroactively
- Time lag between the availability of the fix and its implementation
- Ratio of accepted to refused change implementation requests
- Percent of changes recorded and tracked with automated tools
- Percent of changes that follow formal change control processes
- Ratio of accepted to refused change requests
- Number of different versions of each business application or infrastructure being maintained
- Number and type of emergency changes to the infrastructure components
- Number and type of patches to the infrastructure components

Outcome metrics (COBIT)

- Reduced number of errors introduced into systems due to changes
- Reduced number of disruptions (loss of availability) caused by poorly managed change
- Reduced impact of disruptions caused by change
- Reduced level of resources and time required as a ratio to number of changes
- Number of emergency fixes
- Application rework caused by inadequate change specifications
- Reduced time and effort required to make changes
- Percent of total changes that are emergency fixes
- Percent of unsuccessful changes to the infrastructure due to inadequate change specifications
- Number of changes not formally tracked or not reported or not authorized
- Backlog in the number of change requests
- Number of disruptions or data errors caused by inaccurate specifications or incomplete impact assessment

2.2.4 Roles and functions

The following is a list of the roles and functions typically associate with Change Management:

- Change Advisory Board
- Change Manager
- Change Requester
- Change Assignee
- Change Analyst
- Change Controller
- Change Coordinator
- Change Tester)
- Change Approver
- Change Implementer

2.2.5 Questions to ask when implementing a change process

- What will be the benefits of improving the process maturity to the desired goal?
- What are the risks of not reaching the goal?
- What inhibitors stand in the way of reaching those goals?
- What are areas of short term improvement that could be started right away without additional resources and with minimal negative impact?

2.3 The need for a Configuration Management process

According to ITIL V3 service transition book, no organization can be fully efficient or effective unless it manages its assets well, particularly those assets that are vital to the running of the customer's or organization's business. This process manages the service assets in order to support the other service management processes.

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. It also ensures that releases into controlled environments and operational use are done on the basis of formal approvals. It provides a configuration model of the services, assets, and infrastructure by recording the relationships between service assets and configuration items.

2.3.1 Value to business

Optimizing the performance of service assets and configurations improves the overall service performance and optimizes the costs and risks caused by poorly managed assets, for example, service outages, fines, correct license fees, and failed audits.

Configuration Management provides visibility of the accurate representation of a service, release, or environment that enables:

- Better forecasting and planning of changes
- Changes and releases to be assessed, planned, and delivered successfully
- Incidents and problems to be resolved within the service level targets
- Service levels and warranties to be delivered
- Better adherence to standards and legal and regulatory obligations (less non-conformance)
- More business opportunities able to demonstrate control of assets and services
- Changes to be traceable from requirements
- The ability to identify the costs for a service

Configuration Management definition: To identify, control, maintain, and verify the versions of configuration items (CIs) and their relationships in a logical model of the infrastructure and services.

Configuration Management provides IT infrastructure control through the identification, registration, monitoring, and management of:

- All the configuration items of the IT infrastructure in scope
- All configurations, versions, and their documentation
- All changes, errors, service level agreements, and history of the components in general
- Relationships between the different components
- Exceptions between configuration records and the real infrastructure

Configuration Management provides a sound basis for other processes such as Incident, Problem, Change, and Release Management by providing accurate information about all CIs.

2.3.2 Steps for implementing Configuration Management

To ensure quality for IT services, Configuration Management supports other processes as the basis for critical items.

Policies

Policies include:

- Ensuring that asset and Configuration Management operations costs and resources are commensurate with the potential risks to the services
- The need to deliver corporate governance requirements, for example, software asset management or Sarbanes-Oxley
- The need to deliver the capability, resources, and service warranties as defined by the service level agreements and contracts
- The requirement for available, reliable, and cost-effective services
- The requirement for clear economic and performance criteria for interventions that reduce costs or optimize service delivery, for example, lower maintenance costs
- The application of whole-life cost appraisal methods
- The transformation from “find and fix” reactive maintenance to “predict and prevent” proactive management
- The requirement to maintain adequate asset and configuration information for internal and external stakeholders
- The level of control and requirements for traceability and auditability
- The application of continual improvement methods to optimize the service levels, assets, and configurations
- Provision of accurate asset and configuration information for other business and Service Management processes
- Integration of asset and Configuration Management with other processes
- Migration to a common asset and Configuration Management architecture
- Level of automation to reduce errors and costs.

Objectives

The objectives of a Configuration Management process could include:

- To identify, capture, and organize configuration information
- To account for all the IT assets and configurations within the organization and its services

- To provide accurate information about configurations and their documentation to support all the other service management processes
- To verify the configuration records against the infrastructure and correct any exceptions
- To provide a sound basis for any processes requiring configuration information, including Incident Management, Problem Management, Change Management, and Release Management
- To enable the correction of any exceptions related to configuration records and the corresponding CIs themselves, by verifying the configuration records against the infrastructure

Scope

Configuration Management covers the identification, recording, and reporting of IT components, including their versions, constituent components, states, and relationships to other IT components and business uses. Items that should be under the control of Configuration Management include hardware, software, systems, services, and associated documentation.

Given the definition above, it should be clear that Configuration Management is not synonymous with Asset Management, although the two disciplines are related. Asset Management is a recognized accountancy process that includes depreciation accounting. Asset Management systems maintain details on assets above a certain value, their business unit (affiliation), and their location. Configuration Management also maintains relationships between assets, which Asset Management usually does not.

While different technologies and practices are sometimes applied in context, the scope of Configuration Management encompasses solution development and test environments as well as IT infrastructure and operational environments.

It is important to define the scope that both change and configuration processes will cover. For Configuration Management, for example, when CIs are defined at the wrong level with too much detail, staff become involved in unnecessary work. With too little detail, there is inadequate control.

A questionnaire may be used to determine the scope of the processes to be implemented.

Typical items included in the scope of a Configuration Management process are:

- Establishing naming conventions for Configuration Items and relationships
- Designing, creating, populating, and updating the Configuration Management Data Base (CMDB)
- Supporting Configuration Item audits

- Identifying Configuration Item interdependencies
- Linking Configuration Item changes to specific RFCs
- Defining and reporting Configuration Baselines.

Examples of items often excluded are:

- Asset Management
- Inventory Tracking
- Procurement of Configuration Items
- Tuning and Installing Configuration Items

Inputs and outputs

A Configuration Management process can include the following inputs and outputs:

- Inputs
  - Authorized RFCs
  - Closed RFC
  - Validated Solution Design
  - CI Data Update Package
  - Asset Information
  - Configuration Information Request
- Outputs
  - RFC
  - Configuration Baseline Report
  - Configuration Information

Activities

The following is a list of key activities involved in a Configuration Management process:

- Establish Configuration Management Framework
- Identify Configuration Items
- Control Configuration Items
- Report Configuration Status
- Verify and Audit Configuration Items
- Evaluate Configuration Management Performance

2.3.3 Configuration Management measurements

Effective day-to-day operation and long-term management of the Configuration Management process requires the use of metrics and measurements throughout the process. There are also a variety of reports that need to be defined, executed, and distributed to enable the management of problems:

- Number of times the configuration is not as authorized
- Incidents or problems tracked to wrong changes
- RFCs that failed due to bad data in CMDB or wrong impact assessment
- Cycle time to approve or implement changes
- Unused licenses
- Exception reports from audits

Critical success factors

Some of the critical success factors in implementing a Configuration Management process include:

- Owners are established for all configuration elements and are responsible for maintaining the inventory and controlling change.
- Configuration information is maintained and accessible, based on up-to-date inventories and a comprehensive naming convention.
- An appropriate software library structure is in place, addressing the needs of development, testing, and production environments.
- There exists a release management policy and a system to enforce it.
- Record keeping and physical custody duties are kept segregated.
- There is integration with procurement and Change Management processes.
- Vendor catalogues and configuration are aligned.
- Configuration baselines exist, identifying the minimum standard components and integration requirements, consistency, and integration criteria.
- An automatic configuration detection and checking mechanism is available.
- An automatic distribution and upgrade process is implemented.
- There is zero tolerance for illegal software.

Performance metrics

Key performance metrics may include:

- Percent of configuration components for which data is kept and updated automatically
- Frequency of physical verifications

- Frequency of exception analysis, addressing redundancy, obsolescence, and correction of configuration
- Time lag between the modification to the configuration and the update of records
- Number of releases
- Percent of reactionary changes
- Average time period (lag) between identifying a discrepancy and rectifying it
- Number of discrepancies relating to incomplete or missing configuration information
- Percent of configuration items in line with service levels for performance, security, and availability

2.3.4 Roles and functions

The following is a list of the key roles and functions associated with Configuration Management:

- Configuration Manager
- Configuration Librarian
- Configuration Administrator
- Configuration Auditor
- Configuration Reporter
- Configuration Software License Administrator
- CI Owner
- Inventory Manager
- Documentation Coordinator

2.3.5 Questions to ask when implementing a configuration process

- What will be the benefits of improving the process maturity to the desired goal?
- What are the risks of not reaching the goal?
- What inhibitors stand in the way of reaching those goals?
- What are the areas of short term improvement that could be started right away without additional resources and with minimal negative impact?

2.4 Summary

This chapter has provided an overview of some of the accepted concepts and best practices associated with implementing Change and Configuration Management processes. They can be used as a starting point in planning an implementation of these processes supported by such tools as the Tivoli Change and Configuration Management Database product.

3

IBM Tivoli Unified Process Composer process mapping and design

As you plan for a deployment of tools to support Change and Configuration Management, the processes to be used by the enterprise must be well documented. Employees should have access to the process documentation and understand how the various artifacts, tools, and individuals play a role in those processes.

This chapter gives a detailed explanation on how to create and document processes using IBM Tivoli Unified Process (ITUP) Composer.

3.1 Key concepts and terminology

Figure 3-1 shows the ITUP Composer main elements and their relationship. The following sections in this chapter explain in detail the meaning of each concept and the role they play.

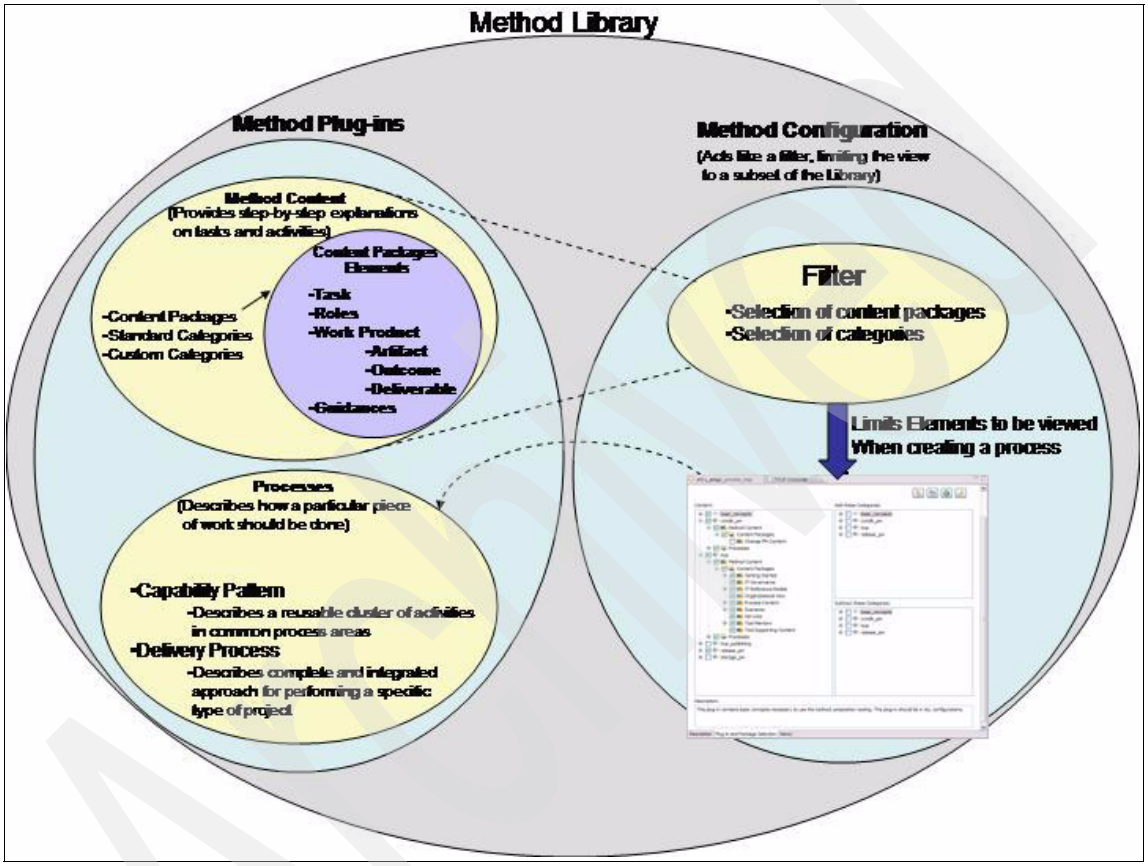


Figure 3-1 ITUP Composer general overview diagram

3.1.1 Method library

A method library is a physical container for method plug-ins and method configuration definitions. All method elements are stored in a method library.

Much like a library has books, a method library has method plug-ins. Where a library book is made up of sections or chapters and content within those chapters, method plug-ins are made up of method content and processes.

Method content contains content packages and both standard and custom categories, while processes structure this content into process fragments called *capability patterns* and full life cycle processes called *delivery processes*.

A method library also has one or more method configurations that filter the library and provide smaller working sets of library content for the user.

3.1.2 Method plug-ins

Method plug-ins are the basic mechanism for separating base content from custom content. Base content is often supplied as a read-only resource in the public domain, intended to be reused and customized in local projects. When creating a customized method plug-in, it is possible to separate the new content from the original library content.

All content is organized in method plug-ins. A method plug-in is a container for method packages which, in turn, contain the method and process content. Method plug-ins and method packages allow the organization of the new content at a level of granularity that suits the needs for authoring and reusing the content.

When creating a method plug-in, one will usually want to reuse content in other plug-ins. The content in these other plug-ins may be modified or extended to add new customized content. When creating a plug-in, any number of other plug-ins can be referenced. A method plug-in can also be stand alone and not reference other plug-ins.

3.1.3 Method content

Method content provides step-by-step explanations, describing how specific development goals are achieved, independent of the placement of these steps within a development life cycle. Processes take these method elements and relate them into semi-ordered sequences that are customized to specific types of projects.

Method content elements are:

Tasks 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.

Roles A role is a well-defined set of related skills, competencies, and responsibilities. Roles can be filled by one person or multiple people. One person may fill several roles.

Work Products A work product is a general term for task inputs and outputs, and descriptions of content elements that are used to define anything used, produced, or modified by a task. The three types of work product are *artifact*, *outcome*, and *deliverable*.

Guidances Guidance is a general term for supplemental information that can be added to most method and process elements. See 3.1.6, “Guidance” on page 29 for more details.

A process engineer creates these elements, defines the relationships between them, and then categorizes them. Method content provides step-by-step explanations, describing how specific development goals are achieved independently of the placement of these steps within a development life cycle. Processes take these method elements and relate them into semi-ordered sequences that are customized to specific types of projects.

For example, a software development project that develops an application from scratch performs development tasks such as “Develop Vision” or “Use Case Design” similar to a project that extends an existing software system. However, the two projects will perform the tasks at different points in time with a different emphasis. That is, they perform the steps of these tasks at different points of time and perhaps apply individual variations and additions.

Method content elements are contained within method content packages that, in turn, are contained within a method plug-in. In order to separate custom content from original content, a new method content should always be created in a new method plug-in that is produced. Creating method content in a method plug-in also allows one to update a custom library with new releases of the basic library without affecting the content that you have created in your own plug-ins.

3.1.4 Method content package

Method content is organized into content packages that are contained in method plug-ins. Before creating a content package, a method plug-in should be created.

A new method content package and method content should always be created in a method plug-in that is produced. This separates custom content from original content shipped with the tool and allows updating custom library with new library releases without affecting the content created in custom plug-ins.

A method content package is a container for method elements. Elements are organized in method packages to structure a large scale of method content and processes as well as to define a mechanism for reuse. Method elements from one package can reuse elements from other packages by defining a link between

them. For example, a work product defined in one package can be used as an input for tasks defined in another package, ensuring that no redundant definitions of the same elements are required. Also, maintenance of method content is greatly improved, as changes can be performed in only one place.

Although a method package is a container for method elements, its structure is broken down into smaller packages to better organize the content, such as content packages, standard categories, and custom categories.

3.1.5 Method configurations

A method configuration is a selection of method plug-ins and method packages in a method library.

A method configuration defines a working set of packages within the method library that limits the view to a subset of the library. Elements that comprise the selected configuration are displayed in the configuration view. Method configurations are used for creating processes and for publication by defining which elements are published in HTML and which are not.

A method configuration consists of the following components:

- A description of the configuration.
- A selection from the set of plug-ins and packages of which elements are defined to be part of the configuration.
- A selection of categories of which categorized elements are added to the set of elements of the configuration in addition to the elements of the selected plug-ins and packages.
- A selection of categories of which categorized elements are subtracted from the set of elements of the configuration defined earlier.
- A selection of views to be published on the Web site.

In a method configuration, it is possible to select and deselect content packages, process, and categories available in the method library's set of plug-ins. The selections may help determine the content of a published Web site. A configuration is given a name and then saved so it can be changed and then republished at a later date.

Before creating a method configuration, the needs and goals for the configuration should be assessed.

There are two ways to create a method configuration:

- Creating a new method configuration from scratch
- Creating a method configuration by copying an existing configuration

Configurations can be created by selecting plug-ins and packages and then adding or subtracting specific elements in content categories. This provides a way to remove whole groups of elements, such as all work products in a specific domain, or all tasks in a specific discipline.

Configurations will be specified in the following four-step procedure:

1. Select the plug-ins to be considered for the configuration definition. All additional selections in the following steps 2 to 4 must be included in these plug-ins only. If categories will be selected in steps 3 and 4 that are comprised of elements that are defined inside of these plug-ins as well as elements that are defined in other plug-ins, then configurations will only consider the elements that are within these plug-ins.
2. Select physical packages to be included in the configuration definition. As a refinement to the method plug-in selection, the specific method packages determine which packages should be included into the interpretation of the configuration. For a selected package, every element directly residing inside that package shall be interpreted as part of the configuration.
3. Select logical categories to be added to the configuration definition. As an additional refinement to the category definition created with Steps 1 and 2, one can select custom or standard categories whose elements shall be interpreted as part of the configuration as well. While step 2 required that all elements that are physically stored within the same package shall be part of the configuration, this step allows adding individual elements grouped into a logical category to a configuration.
4. Select logical categories to be subtracted from the configuration definition. As an additional refinement to the category definition created with steps 1 to 3, one can select custom or standard categories whose elements shall not be part of the configuration. In other words, one can subtract sets of individual elements from a configuration by grouping them into a category and listing this category in this step.

Advantages of this approach include the following:

- Increased flexibility in selecting complete packages
- Ability to remove individual elements from a configuration
- Ability to remove whole categories from a configuration in a single operation

3.1.6 Guidance

Guidance is a general term for supplemental information that can be added to most method and process elements.

Adding guidance is an easy way to tailor information for specific projects. For example, a type of guidance called a guideline could be associated to a work product that explains how a project uses that work product. See 3.1.8, “Method content variability” on page 31 for more information about attaching guidance to specific types of elements.

Guidance elements can also be associated with other guidance elements.

Types of guidance

The following guidance types can be added to method and process elements:

Checklist A specific type of guidance that identifies a series of items that need to be completed or verified. Checklists are often used in reviews such as walkthroughs or inspections.

Concept A specific type of guidance that outlines key ideas associated with basic principles underlying the referenced item. Concepts normally address more general topics than guidelines and span across several work product or tasks or activities.

Estimating Guideline A specific type of guidance that provides sizing measures or standards for sizing the work effort associated with performing a particular piece of work and instructions for their successful use. It may be comprised of estimation considerations and estimation metrics.

Example A specific type of guidance that provides an example of a completed work product.

Guideline Provides additional detail on how to perform a particular task or grouping of tasks, or provides additional detail, rules, and recommendations on work products and their properties. Among other items, it can include details about best practices and different approaches for doing work, how to use particular types of work products, information about different subtypes and variants of the work product and how they evolve throughout a life cycle, discussions on skills the performing roles should acquire or improve upon, and measurements for progress and maturity.

Practice	Represents a proven way or strategy of doing work to achieve a goal that has a positive impact on work product or process quality. Practices are defined orthogonal to methods and processes. They could summarize aspects that impact may different parts of a method or specific process.
Report	A predefined template of a result that is generated on the basis of other work products as an output from some form of tool automation. An example for a report would be a use case model survey, which is generated by extracting diagram information from a graphical model and textual information from documents and combines these two types of information into a report.
Reusable Asset	Provides a solution to a problem for a given context. The asset may have a variability point, which is a location in the asset that may have a value provided or customized by the asset consumer. The asset has rules for usage that are the instructions describing how the asset should be used.
Supporting Material	Used as a catch all for other types of guidance not specifically defined elsewhere. It can be related to all kinds of content elements, including other guidance elements.
Template	A specific type of guidance that provides for a work product a predefined table of contents, sections, packages, or headings, a standardized format, as well as descriptions of how the sections and packages are supposed to be used and completed. Templates cannot only be provided for documents, but also for conceptual models or physical data stores.
Term Definition	Defines concepts and is used to build up the Glossary. A term definition is not directly related to content elements, but its relationship is being derived when the term is used in the content elements description text.
Tool Mentor	A specific type of guidance that shows how to use a specific tool to accomplish some piece of work, either in the context of, or independent from, a task or activity.
White Paper	A special concept guidance that has been externally reviewed or published and can be read and understood in isolation from other content elements and guidance.

3.1.7 Process

A process describes how a particular piece of work should be done. The work may have a relatively small scope, in which case it can be described as a *capability pattern*, or may address a full project life cycle, in which case it can be described as a *delivery process*. A process can reuse method elements and combines them into a structure and sequence for carrying out work.

There are two main types of processes: a capability pattern and a delivery process. A capability pattern is a special process that describes a reusable cluster of activities in common process areas, while a delivery process describes a complete and integrated approach for performing a specific type of project.

Each time a task is included in a process, a reference object to that task is created in the context of the process. This is called a *task descriptor*. The same task can be referenced any number of times in the same process. In other words, one task can have many task descriptors. A task descriptor can also modify the base task without actually changing the task. For example, roles and work products can be added or suppressed, and steps can be suppressed or re-sequenced.

Roles and work products can also be included in processes as role descriptors and work product descriptors. Roles and work products can be customized to fit with the content of the process in which they are used.

3.1.8 Method content variability

Method content variability allows elements in one content package to modify or reuse elements in other content packages without directly modifying the original content. Variability provides a mechanism for making changes to the published Web site while keeping the components separate and optional.

Variability allows customizing configurations that use method content and processes that are owned by others and cannot be directly modified. When content packages are upgraded, they can be imported and customizations made earlier can be reapplied in a single step without having to go through each element.

Variability generally affects two characteristics of a method element: its attributes and its relationships with other content elements. If an element supports variability, the specification is shown at the bottom of the element's description view.

There are three factors to be considered when using variability:

- Attributes: Element data types such as Main Description.
- Incoming Associations: Associations from other elements. The associated element may have one or more references to the subject element.
- Outgoing Associations: Associations to other elements. The subject element many have one or more references to the associated element.

Variability type

Variability type describes how one element affects another through variability associations. The five types of variability associations are listed here:

Not Applicable The element is a base element and does not affect another element through variability. This is the default value of an element's variability type.

Contributes A contributing element adds to the base element. The base appears in the published Web site but the contributing element does not. In and out relationships from the contributing element are added to the base. Text from the contributing element is appended to corresponding base sections.

Replaces The element replaces parts of the base element. The replacer appears in the published Web site but the base element does not. Out relationships in the replacer are left untouched, and the base's are ignored. In relationships from the base are added to the replacer. Text in the replacer is left untouched, and the base's text is ignored.

Extend An extending element inherits characteristics of the base element. Both the extender and the base appear in the published Web site. Out relationships from the base are added to the extender. In relationships in the extender are left untouched, and the base's are ignored. Text is added from the base if the extender does not have a value defined for the given section.

Extends and Replaces This variability relationship combines the effects of the extends and replace variabilities into one variability type. While the replaces variability completely replaces all attributes and outgoing association instances of the base variability element with new values and instances, or removes all values or association instances if the replacing element does not define any, extends and replaces variability only replaces the values that have

been redefined and leaves all other values of the base element as is.

3.1.9 User roles and role-specific tasks

There are four primary roles performed by users of this application:

- Method Author
- Process Author
- Process Configurator
- Practitioner

Method Author

The Method Author uses the tool on a regular basis to provide standard processes for use in an organization. The Method Author uses the full functionality of the tool to:

- Create plug-ins
- Create new method elements
- Extend existing method elements
- Create reusable capability patterns by reusing method elements
- Create delivery processes by reusing capability patterns and method elements
- Create custom categories for use as views in a configuration
- Create and modify configurations
- Publish configurations or processes

Process Author

The Process Author's goal is to produce a delivery process for their project(s) by reusing method elements. The Process Author uses the tool occasionally, as project needs dictate, typically supporting one or, more likely, several projects by specifying the processes to be followed. The Process Author uses the process authoring and configuration publishing functionality of this tool to:

- Create plug-ins
- Create reusable capability patterns by reusing method elements
- Create delivery processes by reusing capability patterns and method elements
- Create custom categories for use as views in a configuration

- Create and modify configurations
- Publish configurations or processes

Process Configurator

The Process Configurator's goal is to produce a delivery process for their project(s) by rapidly leveraging ready-made plug-ins. The Process Configurator uses this tool occasionally, as project needs dictate, typically supporting one or several projects by specifying the process for the projects. The Process Configurator uses the configuration publishing functionality in this tool to:

- Create and modify configurations
- Publish configurations or processes

Practitioner

A Practitioner's goal is to correctly use the organization's processes and best practices effectively. A Practitioner uses a published configuration on a regular basis driven by the work being performed to view processes and methods.

3.2 Creating a method plug-in

Because the plug-ins shipped with the method library are locked and read-only, changes, additions, and extensions to existing method content and processes must be placed in custom created method plug-ins. However, it is possible to use various capabilities to logically merge plug-in contents into other plug-ins allowing you, as a result, to publish extended methods and processes that seamlessly incorporate new method elements. To create new method plug-ins, the steps described in the following sections should be followed.

3.2.1 Using the Method Plug-in Wizard

Create a new method plug-in using the New Method Plug-in wizard. To open the wizard, select **File** → **New** → **Method Plug-in**. Specify at least the name for the new method plug-in and select the check box for any other method plug-ins whose content needs to be extended or reused. Figure 3-2 on page 35 shows the window of the Method plug-in Wizard.

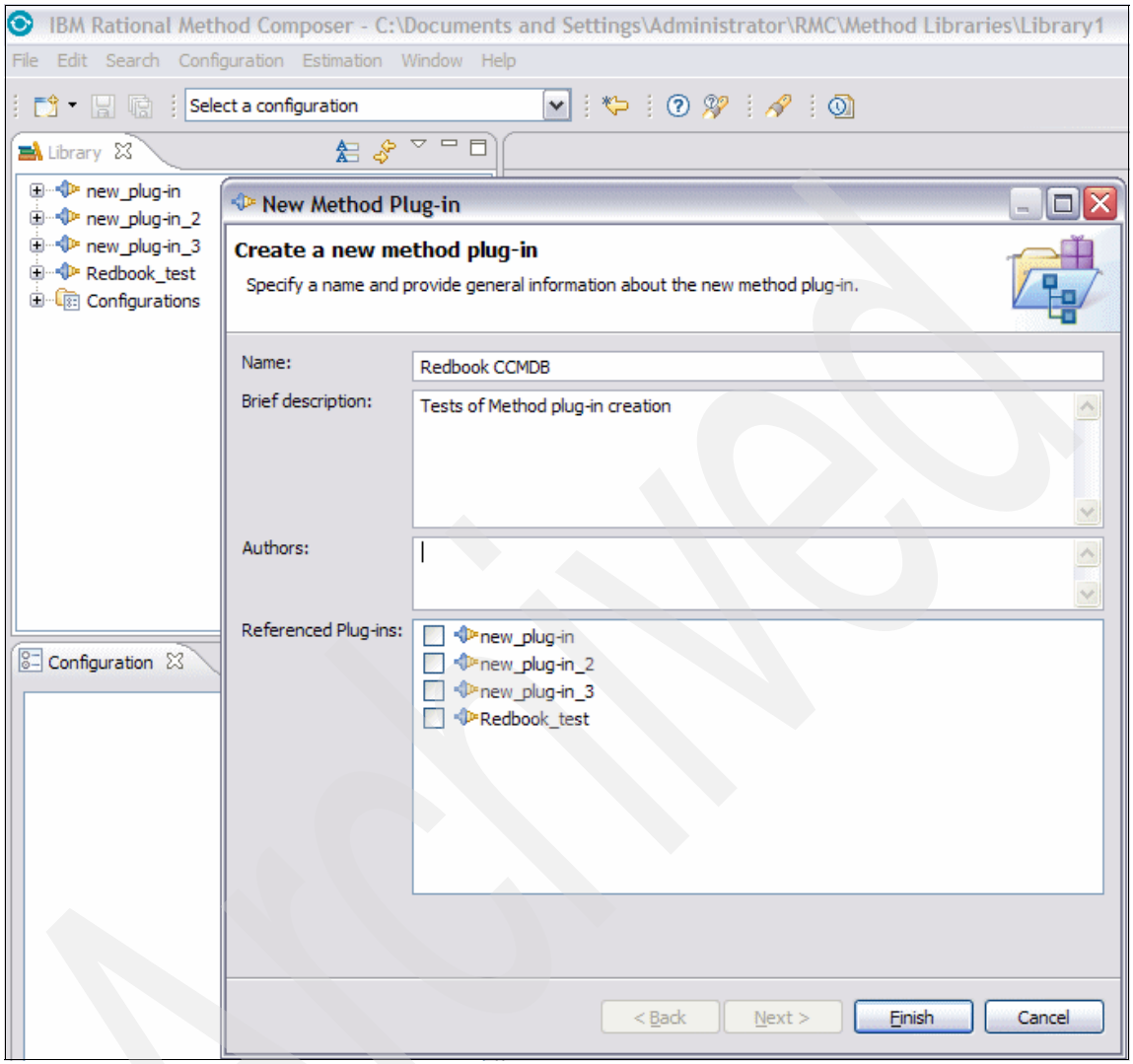


Figure 3-2 New Method plug-in window

3.2.2 Opening the method plug-in editor

After creating the plug-in, its specifications can be changed in the method plug-in editor. In the library view, double-click the method plug-in that was just created to open the editor. Figure 3-3 shows the Method plug-in editor in the ITUP Composer.

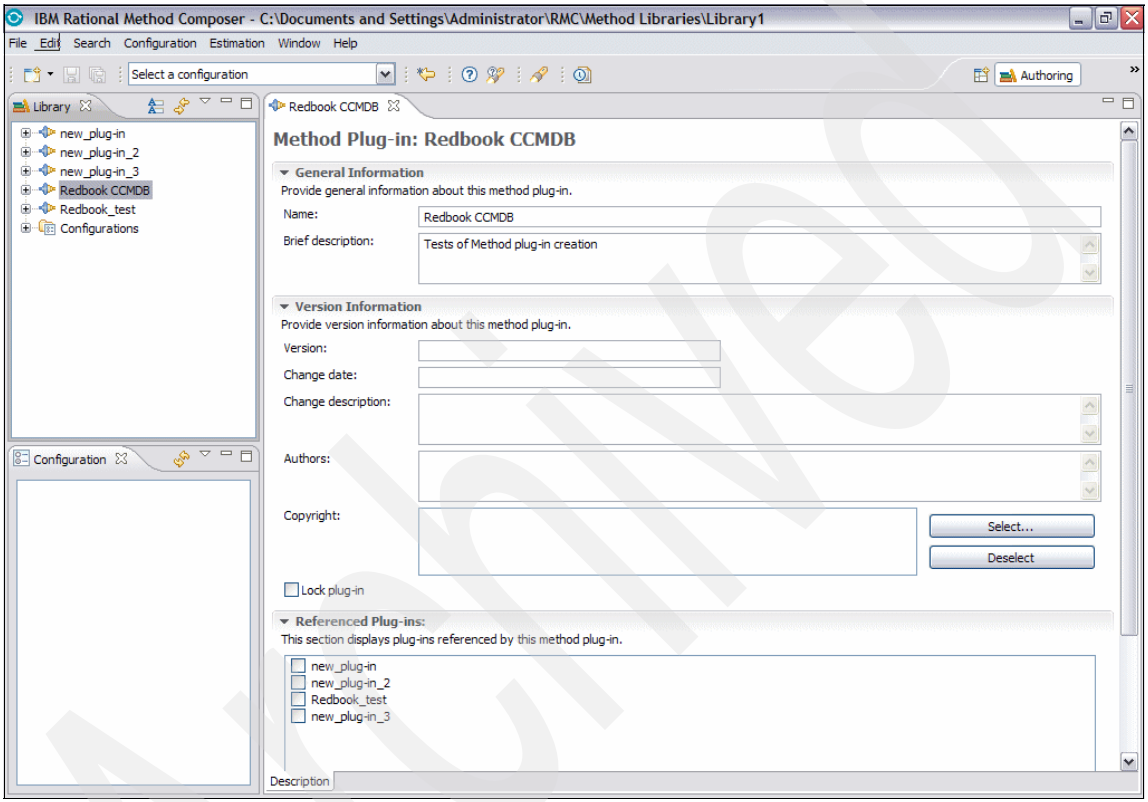


Figure 3-3 Method plug-in editor

3.2.3 Creating a new method configuration

Because the method library can contain large numbers of elements, one may want to limit the work to a user-defined subset of the library called method configuration. A method configuration defines a working set of packages within the method library that helps limit the view to a subset of all elements. Method configurations are not only used for creating processes, but also for publication, because a configuration defines which elements will be published in HTML and which will not.

The elements that are part of a selected configuration are displayed in the configuration view. Using the configuration view, one can browse the collection of method elements that are part of the selected configuration, and populate processes by dragging elements from the configuration view into the process editor.

Before creating a method configuration, the needs and goals for the configuration should be determined. One scenario for creating a method configuration is that an already created new method plug-in exists and there is a need to define method elements that extend the already existing plug-in. In this case, a configuration that includes the new plug-in and the existing plug-in should be created.

Another scenario for creating a method configuration is where a new configuration has to be defined for publication purposes on existing plug-ins, defining which elements to publish. For example, if the current set of configurations available does not meet the needs, an existing configuration could be either customized or a completely new configuration could be created.

To create a method configuration by copying an existing configuration, go to “Creating by copying an existing configuration” on page 38.

To create a new method configuration, skip to “Creating from scratch” on page 40.

Creating by copying an existing configuration

Expand the “Configurations” package at the end of the Library view. Right-click the method configuration that should be copied, and click **Copy** from the menu. Right-click the configurations package and click **Paste** from the menu. Figure 3-4 shows how to copy an existing method configuration.

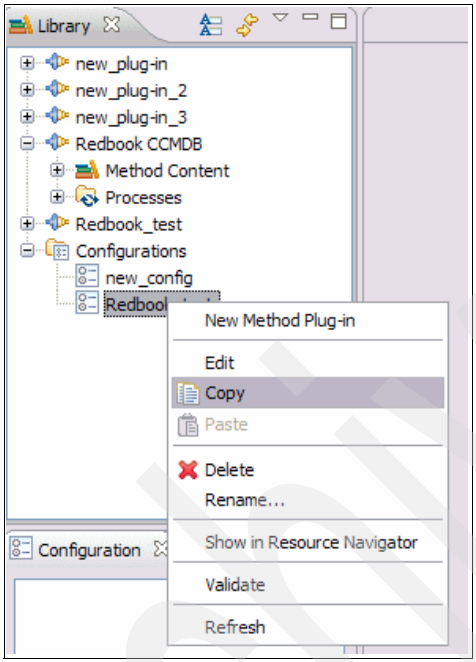


Figure 3-4 Creating method configuration by copying existing configuration

A dialog prompts you for a new configuration name. Provide a name that reflects the character or purpose of this configuration. Figure 3-5 shows an example of the dialog. To continue specifying your method configuration, go to “Specifying the method configuration” on page 41.

Figure 3-5 Providing a name for the new configuration

Creating from scratch

Click the **Plug-in and Package Selection** tab in the method configuration editor to go to the configuration specification form. This form displays a list of all method plug-ins and, for every plug-in, all of its content packages and processes. Use the check boxes to add or remove plug-ins, packages, and processes to or from the configuration. See Figure 3-6 and Figure 3-7 on page 41 for examples.

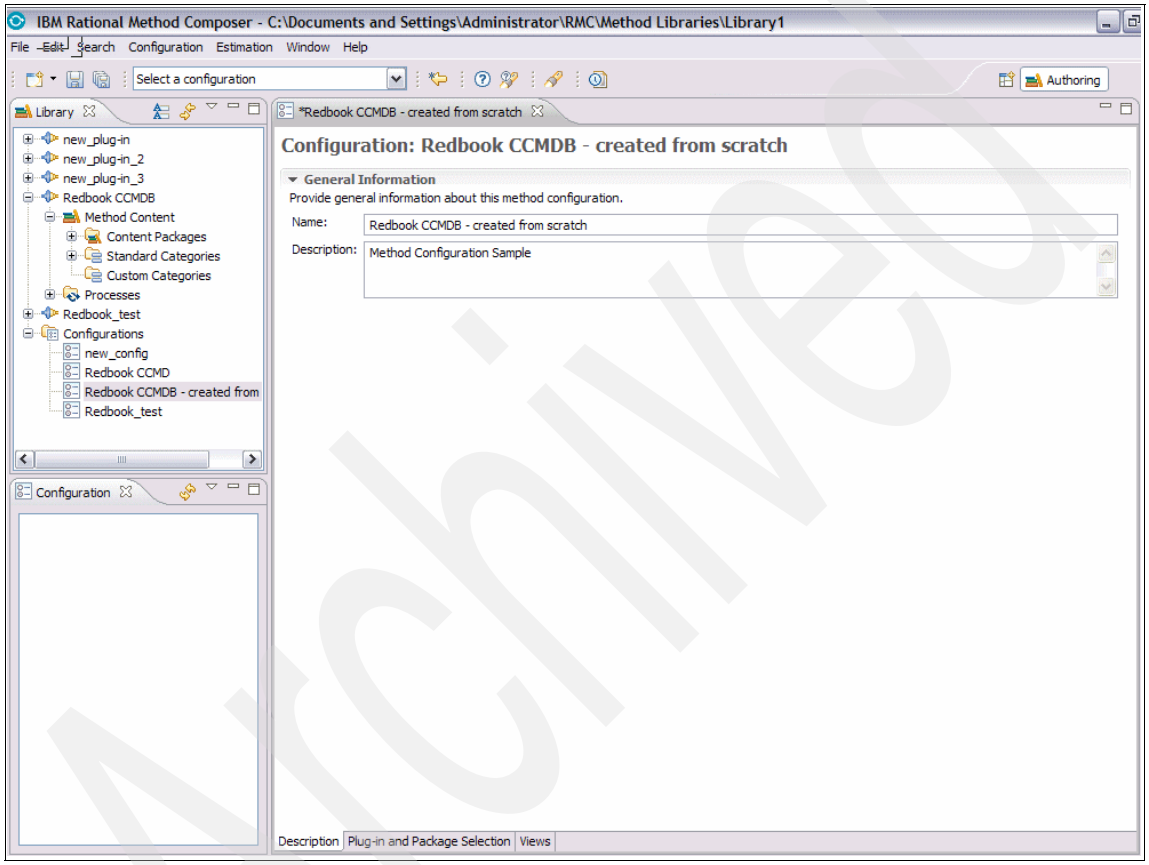


Figure 3-6 Creating method configuration from scratch

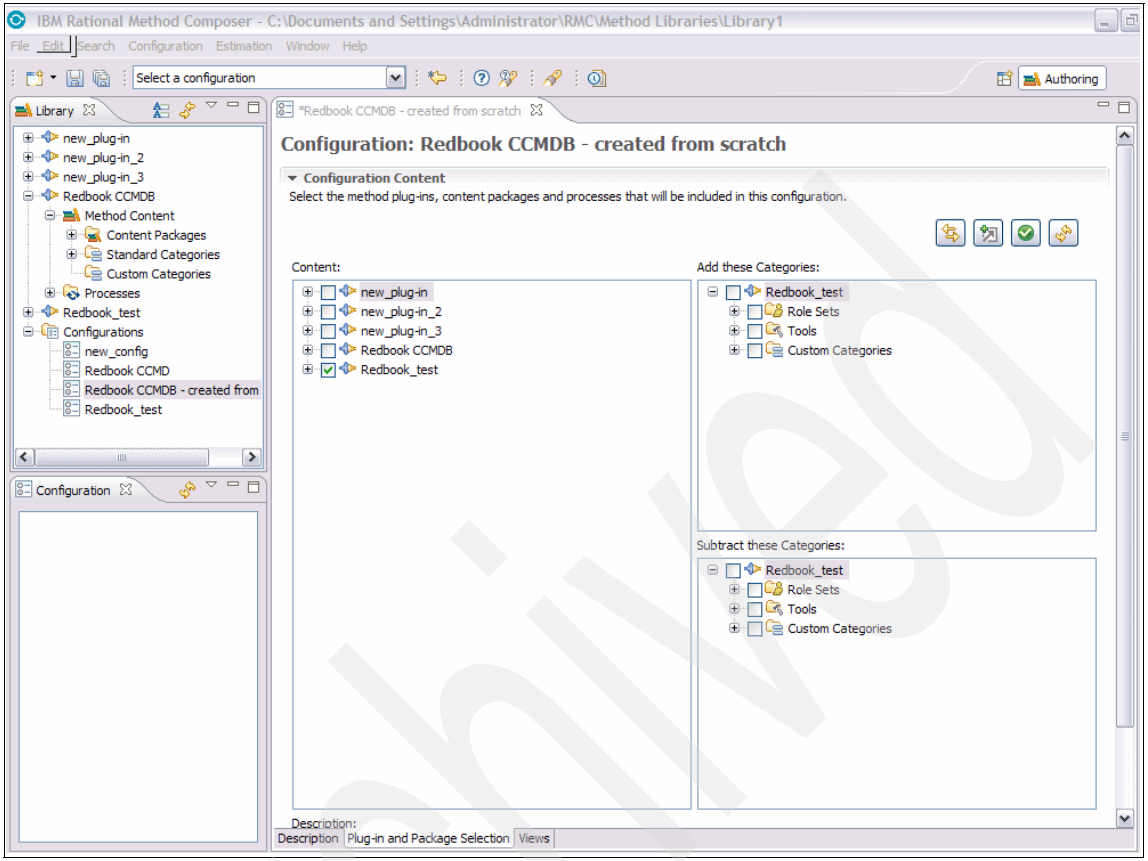


Figure 3-7 List of elements that can be reused on the new method configuration

Specifying the method configuration

Click the **Plug-in and Package Selection** tab in the method configuration editor to go to the configuration specification form. This form displays a list of all method plug-ins and for every plug-in all of its content packages and processes. Use the check boxes to add or remove plug-ins, packages, and processes to or from the custom configuration. Figure 3-7 shows an example of the Plug-in and Package Selection window.

3.2.4 Previewing method configuration in configuration view

It is possible to immediately preview the new method configuration using the configuration view. Refresh the configuration view by selecting the view's **Refresh** option on the menu. Drill into the tree structures displayed by the configuration view to see elements included in the configuration.

The configuration view shows the content elements in a library filtered by a configuration. Figure 3-8 shows how to refresh newly created method configuration.

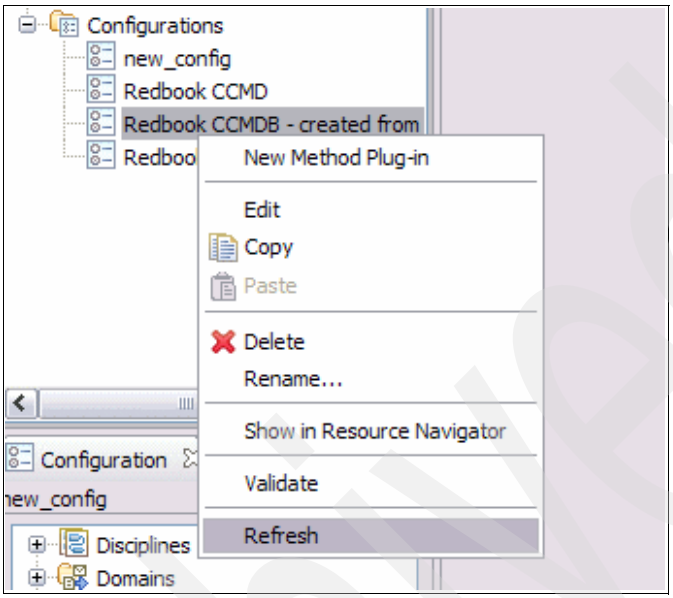


Figure 3-8 Refresh method configuration in configuration view

To select the new method configuration, select it in the toolbar drop-down menu, as shown in Figure 3-9 on page 43.

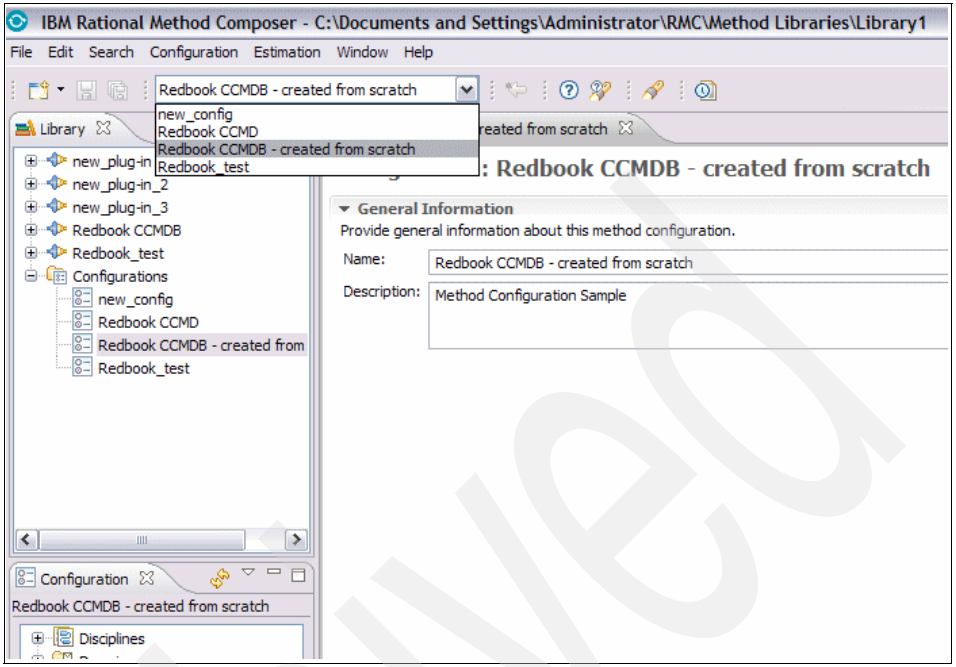


Figure 3-9 Selecting the method configuration to view

3.2.5 Defining navigation views for the method configuration

A navigation view is a navigation tree browser for a configuration published as HTML. Every published configuration can have several views that are displayed as stacked tree browser tabs. The structure of the navigation view is defined as custom categories. A custom category is a user-defined collection of categorizing elements, which may itself contain subcategories. This structure is what defines the structure for the tree browser. Therefore, to define a navigation view, select a custom category and all of this categories' sub-elements that make up the tree browser structure displayed by the view.

To add navigation views to the configuration, click the **Views** tab in the configuration editor. Use the **Add View** and **Remove View** buttons to select the custom categories you want to add and remove as a view, respectively. Click the tab of the views you just added to preview.

Figure 3-10 shows how to add a navigation view.

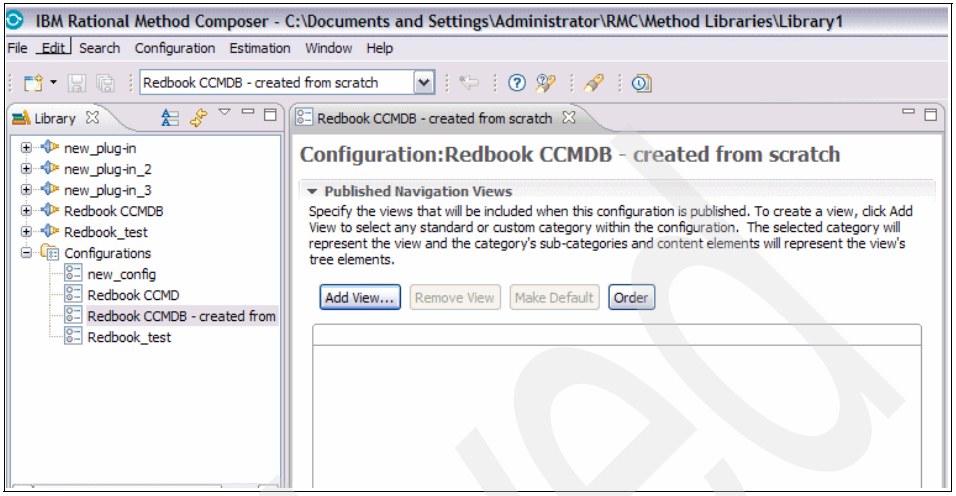


Figure 3-10 Adding navigation view

To select a view to display as the start-up view, click the **Make Default** button. The start-up view is the first view shown when a published configuration is displayed when starting up. Figure 3-11 on page 45 shows a window with examples of options that can be selected to represent the view.

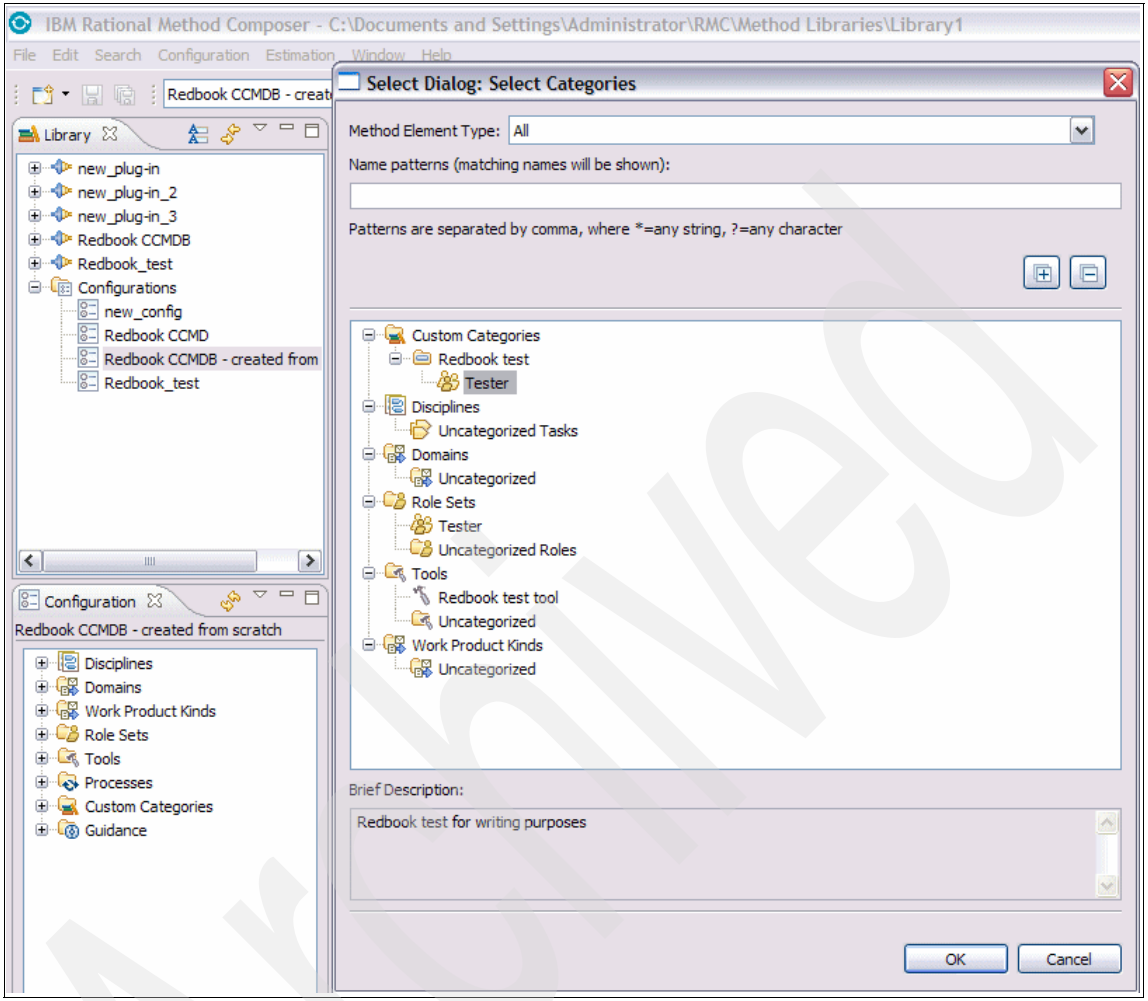


Figure 3-11 Selecting categories to represent the view

3.3 Adding new method content

Method content should always be created in a produced method plug-in. This separates new custom content from content that was reused from third parties and allows updating your own library with new releases of such third-party plug-ins without affecting the content that was created in custom plug-ins.

All plug-ins shipped with the method library are protected from direct modification. Creating new elements in custom plug-ins and then relating those elements to the elements in the locked plug-in allows tailoring the contents of the locked plug-in for custom use. A method plug-in must be created prior to adding new method content.

3.3.1 Creating a new content package

Content packages are used to group related method content together. Because content packages are selectable at publication time, it is a good practice to group content that needs to be published together into the same content package.

Find the custom created method plug-in in the Library view. Drill into the plug-in's packages to locate the package called “Content Packages.” This package contains all packages that can contain method elements. Select and expand a package in the “Content Packages” hierarchy in which to create a new element, or to create a new content package. Right-click a package and select **New** → **Content Package**.

An editor opens so that a unique name can be provided for the package and to briefly describe its purpose. Figure 3-12 shows how to create a new content package.

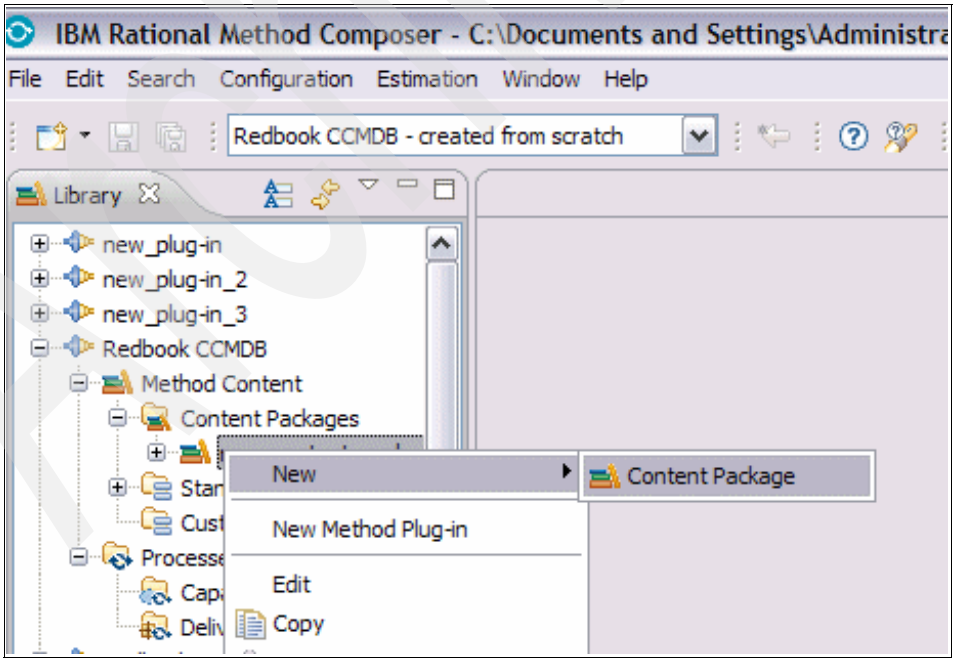


Figure 3-12 Creating a new content package

3.3.2 Creating a method content package element

In an expanded content package within the library view, right-click any one of roles, tasks, work products, or guidance to create any of these content elements. Select **New** and then select the concrete type of the element that will be created (for example, for work products, choose between artifact, outcome, or deliverable). The new element is created and its respective editor is opened. Figure 3-13 shows how to create new method content element.

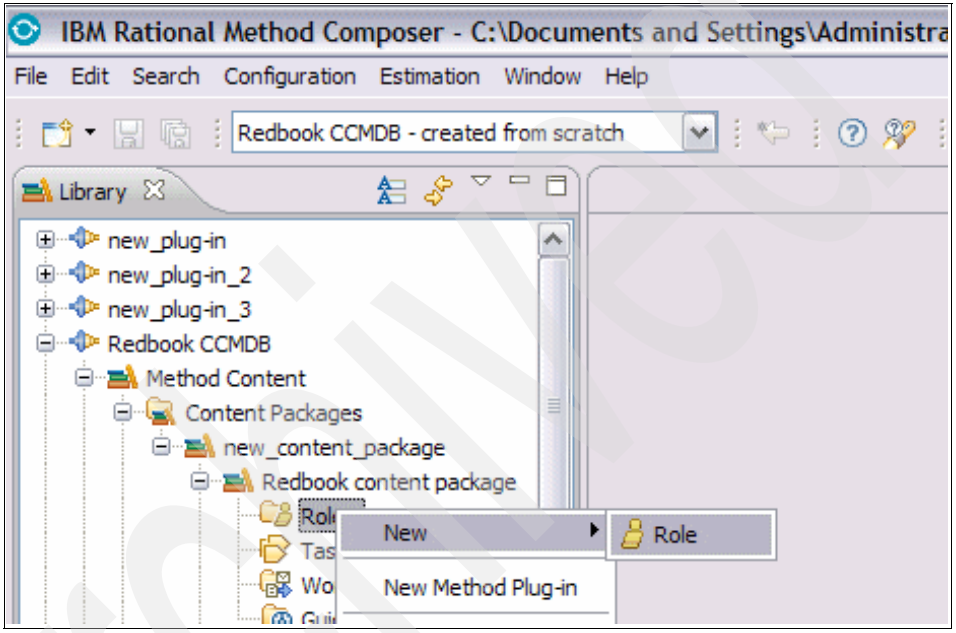


Figure 3-13 Creating new method content element

Detailing a created method content package element

Use the fields in the content element editor to specify the content element details. Start by assigning a unique “Name” to the element and giving it a “Presentation Name” that will be used as the external visible name when other elements refer to this element or when the element is published.

Every element owns several specific content fields distributed on several stacked editor tabs and sections within these tabs that you can use for your descriptions. Figure 3-14 shows an example on detailing a new admin role.

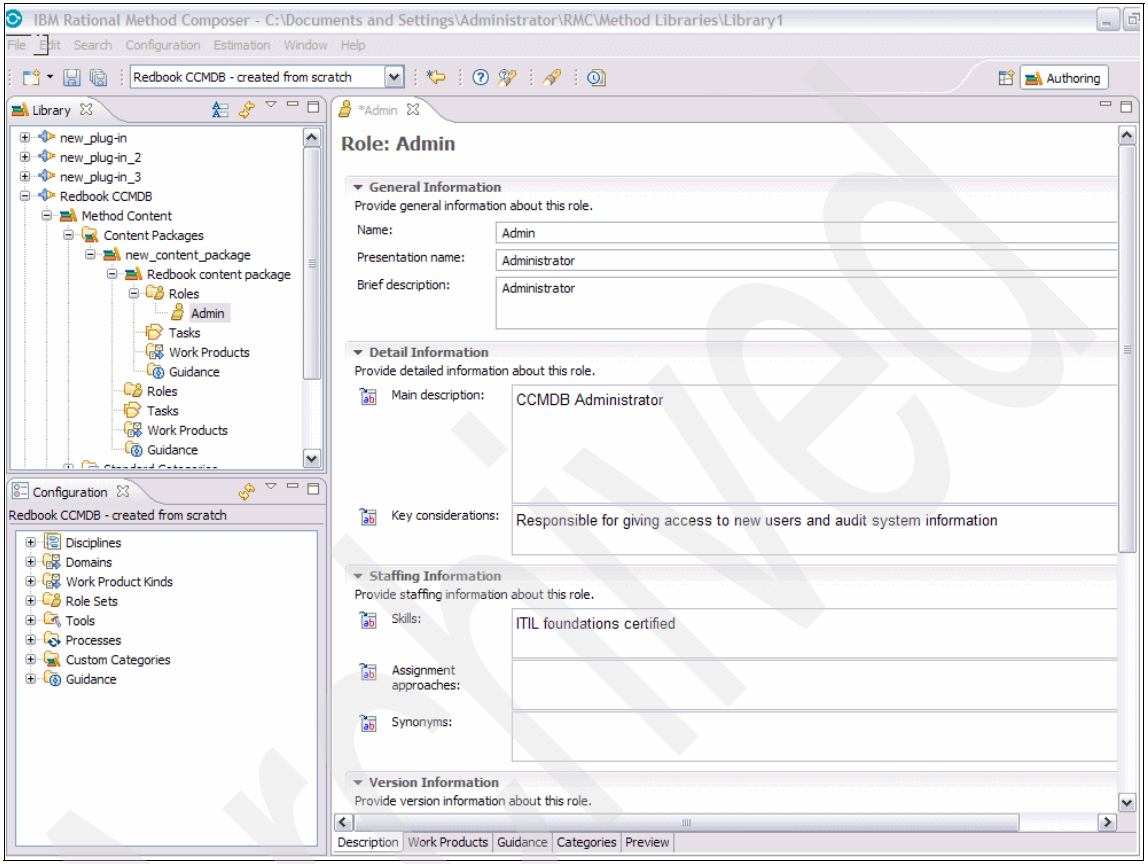


Figure 3-14 Created method element

3.4 Creating a process

There are two main types of processes: capability patterns and delivery processes. A capability pattern is a special process that describes a reusable cluster of activities in common process areas, while a delivery process describes a complete and integrated approach for performing a specific type of development project.

3.4.1 Selecting/creating a default method configuration for the process

A process can contain content from many different method plug-ins. The content elements from these plug-ins, such as tasks or work products, applied to the process through drag and drop, can have many contributions or replacements. Such contributions or replacements may provide additional relationships that need to be considered for creating the process elements with their mirrored set of relationships.

For that reason, it is important to define a configuration that defines the visible set of elements and relationships when the process is authored. This process authoring configuration is referred to as the “Default configuration” for the process and should define the largest reasonable set of method plug-ins, content packages, and other processes from the method library that will be referred to by the process at some point.

In addition to the default configuration, a process can be linked to many additional method configurations that have been verified to also produce valid results. However, all other valid configurations need to define subsets of the default configuration. In other words, it is not possible to link a method configuration to a process that refers to elements that are not part of the default configuration, because such elements were not considered when the process was created.

Process elements that refer to content packages that are defined outside of the scope of such a configuration will not be shown in the process when published or used under such a configuration. This allows you to easily hide content from a process by moving content packages in or out of the related configuration.

Therefore, before creating a process, review the list of configurations in the library view and decide which configuration to use. If necessary, open the configurations and examine their specification. If a fitting configuration that defines the right set of elements cannot be found, create a new method configuration.

3.4.2 Choosing a method plug-in to hold a process

Because it is not possible to add processes to write-locked third-party method plug-ins, a process needs to be created in one of the custom created method plug-ins. Therefore, it is best to create a process within the plug-in in which it is going to be used. For example, if there is a need to develop a set of capability patterns to use to assemble a delivery process, try to maintain all of the capability patterns in the same method plug-in.

In the library view, select a plug-in from the list of available method plug-ins. Icons that are dimmed have been locked for modification and cannot be used. Figure 3-15 shows how to select a plug-in.

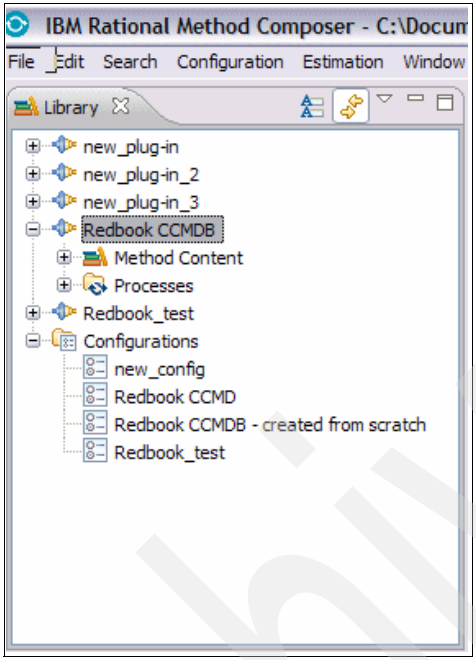


Figure 3-15 Selecting a method plug-in

3.4.3 Finding or creating a process package

Processes can be organized with process packages to increase maintainability and to make it easier for the process user to browse and find them. Be aware that it is possible to create capability patterns only in a capability patterns package or sub-package, and delivery processes only in a delivery process package or sub-package.

Using the library view, review the structure of process packages available in the method plug-in selected or created previously, and then select one of the packages present as a container for the process. Alternatively, it is possible to create a new process package by right-clicking a capability pattern or delivery process package or sub-package and then selecting **New** → **Process Package**. In the window that opens, specify the name of the package and click **OK**.

3.4.4 Creating the capability pattern or delivery process

To create a new capability pattern or delivery process, right-click the selected or newly created process package and select **New** → **Capability Pattern** or **New** → **Delivery Process**. In the window that opens, specify the process name and default configuration and click **OK**. The process is created and the editor is opened. Figure 3-16 shows a sample window for creating a new capability pattern.

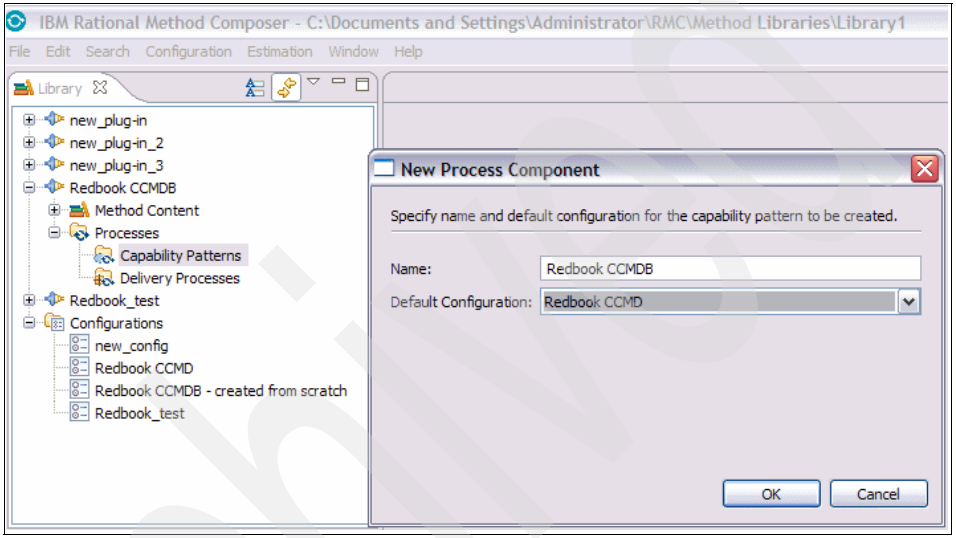


Figure 3-16 Creating a new capability pattern

3.4.5 Documenting a process

With the process editor opened in the Description tab, document the process using the available text fields. At a minimum, provide a presentation name and a brief description for the process. Figure 3-17 shows a sample window for documenting a process.

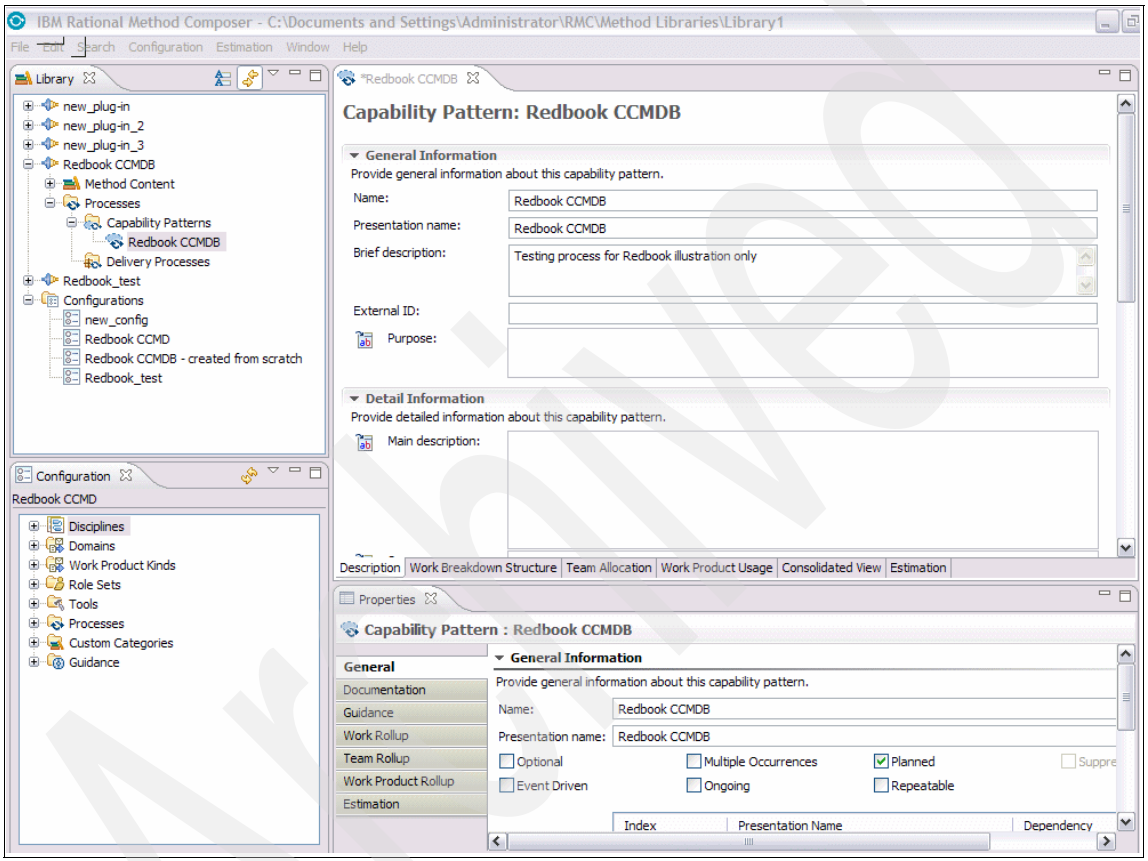


Figure 3-17 Creating a capability pattern

3.4.6 Process authoring views

A process can be developed from three different views:

Breakdown Structure

You can create a process by defining a work breakdown structure. Create iterations and activities first and then populate activities by either applying tasks from method content or applying capability patterns.

Refer to 3.4.7, “Developing the work breakdown structure” on page 53 to learn how to work with this view.

Team Allocation You can create a process by defining which teams and roles shall participate in activities and finding work products and tasks from there. Refer to 3.4.8, “Developing the team allocation structure” on page 57 to learn how to work with this view.

Work Product Usage You can create a process by defining which work products should be created in activities and finding tasks and roles from there. Refer to 3.4.9, “Developing the work product usage structure” on page 60 to learn how to work with this view.

3.4.7 Developing the work breakdown structure

Before starting, it is important to make sure that the method configuration selected in the tool bar is the same as the configuration that was selected as the default configuration for the process. Figure 3-18 shows how to check if the method configuration selected and default configuration for the process are the same.

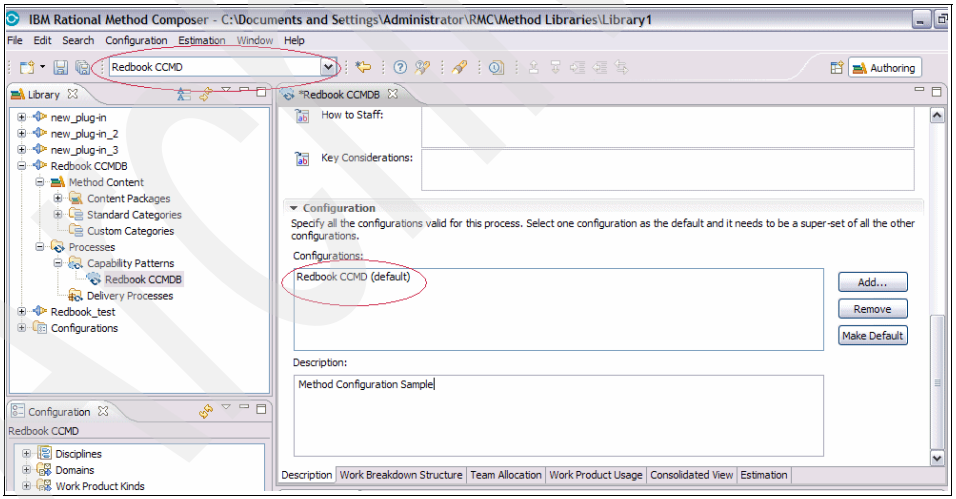


Figure 3-18 Checking the selected configuration and the default configuration for the process

To access the work breakdown structure editor, select the **Work Breakdown Structure** tab in the process editor.

Right-click the element in the first row of the breakdown structure and select **New Child** → **Activity** to create a new activity. Alternatively, you can create a phase or iteration, depending on the scope of your process. If needed, create more activities to set up your breakdown structure. Activities can be nested inside each other. Figure 3-19 shows a sample window for creating a new child activity.

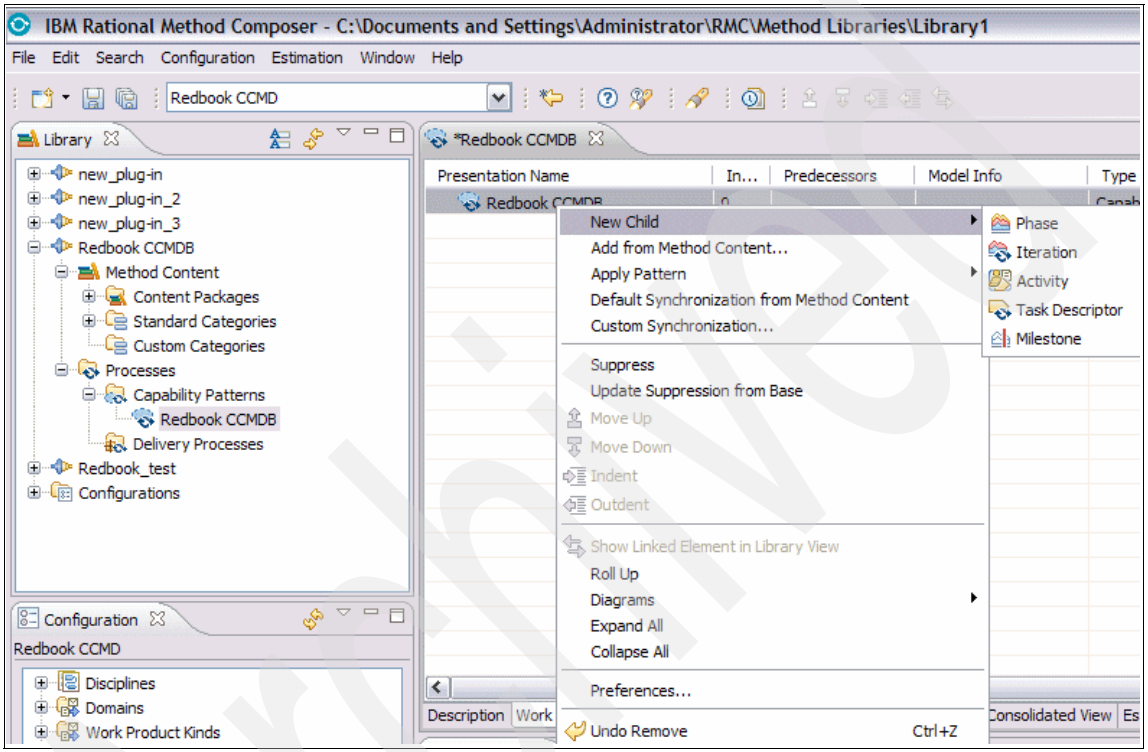


Figure 3-19 Creating a new child activity

Review the list of tasks in the configuration view. In this view, tasks are sorted by discipline. Drill into the disciplines hierarchy to see which tasks are available in this configuration. Select a task that you want to add to your breakdown structure and drag it on top of one of the activities that you just created. The task is added as a so-called task descriptor (an occurrence of a task in one specific activity). Figure 3-20 shows how to drag a discipline from the configuration view into work breakdown structure.

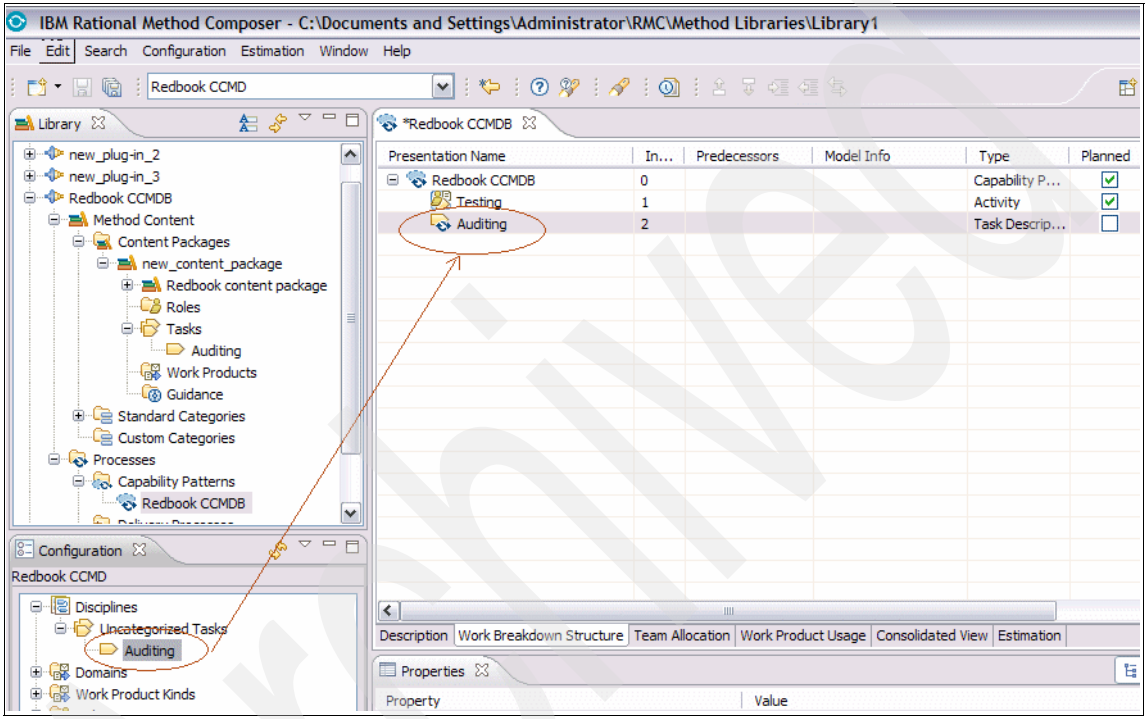


Figure 3-20 Dragging a discipline into a work breakdown structure

Review the task descriptor's details in its properties view. If the properties view is not displayed, then select the task in the Work Breakdown Structure editor, right-click, and select **Properties**. Use the tabs on the side of the properties view to review different aspects of the task descriptor. Figure 3-21 shows where task descriptor properties are displayed.

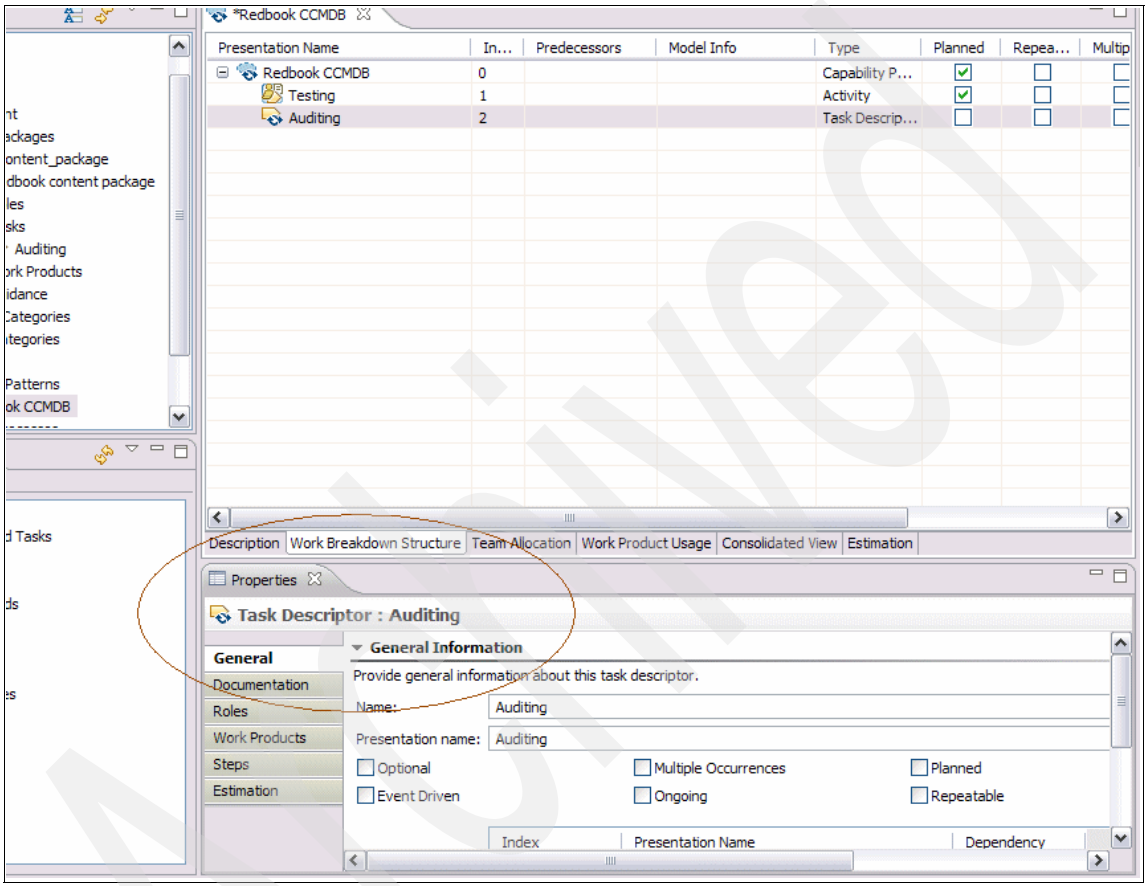


Figure 3-21 Reviewing task descriptor details

The application allows performing individual modifications of the task descriptor in the properties view. For example, changing the presentation name, adding textual descriptions, changing performing roles, changing the inputs and outputs, and so on. When changing the task descriptor's relationships in the property window tabs roles or work products, new elements can either be added from the method content by using the **Add** button or a task descriptor can be connected with tasks already present in this activity.

Rather than dragging tasks one by one, it is also possible to apply whole capability patterns or activities from other processes available in the current method configuration. Select a capability pattern or any activity of such a pattern or delivery process available in the configuration view and drag it on top of an activity within the process breakdown in the process editor.

Continue adding more tasks, activities, or patterns to the activities, or switch to the team allocation tab to add roles or to the work product usage tab to add work products.

3.4.8 Developing the team allocation structure

Before starting, it is important to make sure that the configuration selected in the tool bar is the same as the configuration that was selected as the default configuration for the process. Figure 3-18 on page 53 shows how to check if the method configuration selected and the default configuration for the process are the same.

In the process editor, click the **Team Allocation** tab to open the team allocation editor.

Right-click the element in the first row of the breakdown structure and select **New Child** → **Activity** to create a new activity. Alternatively, a phase or iteration can be created, depending on the scope of the process. If needed, create more activities to set up your breakdown structure. Activities can be nested inside each other.

Figure 3-22 shows how to create a new child activity.

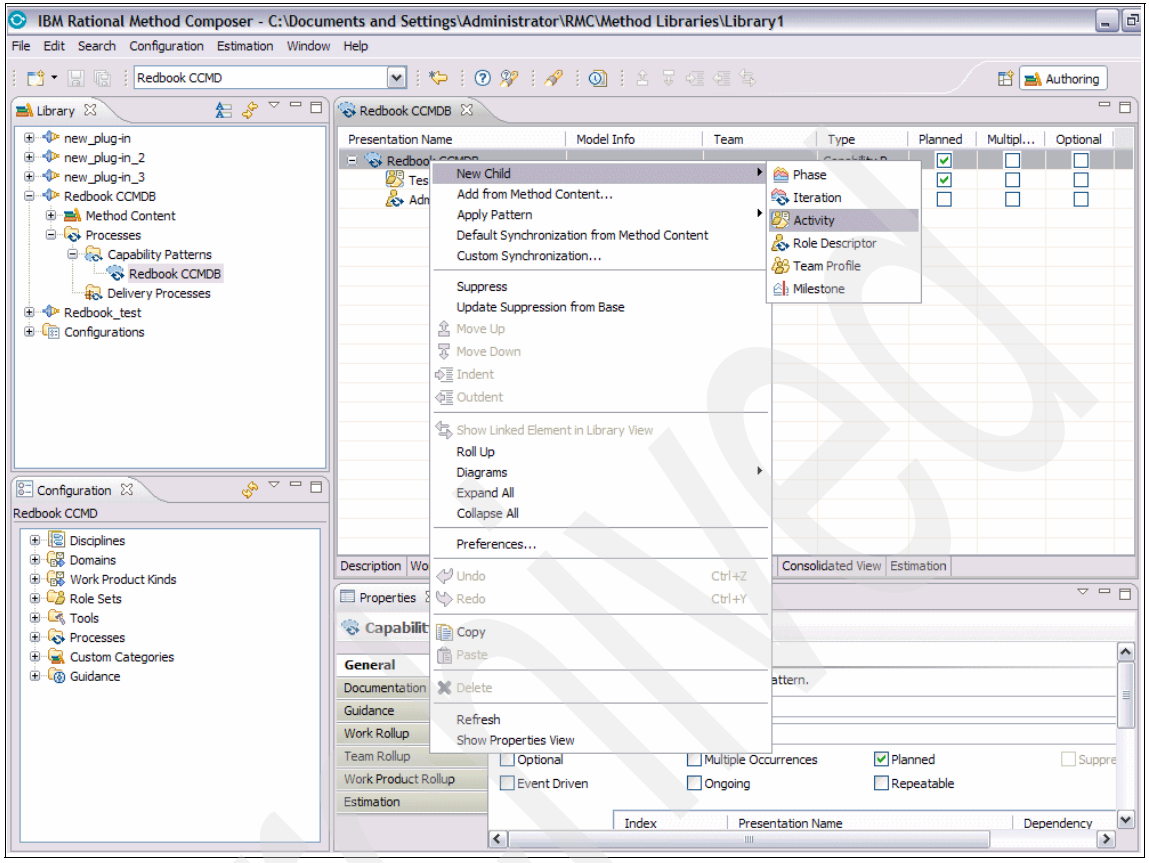


Figure 3-22 Creating a new team allocation child activity

Roles can be added directly to the activities now. In the configuration view, review the list of roles. In this view, roles are organized into role sets. Drill into the role sets hierarchy to see which roles are available in this configuration. Select a role to add to the activity and drag it on top of the activity created earlier. The role is added as a role descriptor (an occurrence of a role in one specific activity).

If the role that was just dragged has responsibility relationships to defined work products, a wizard opens and prompts you to add any of the work products to the process. Select zero to many work products and then click **OK**. Figure 3-23 on page 59 shows an example.

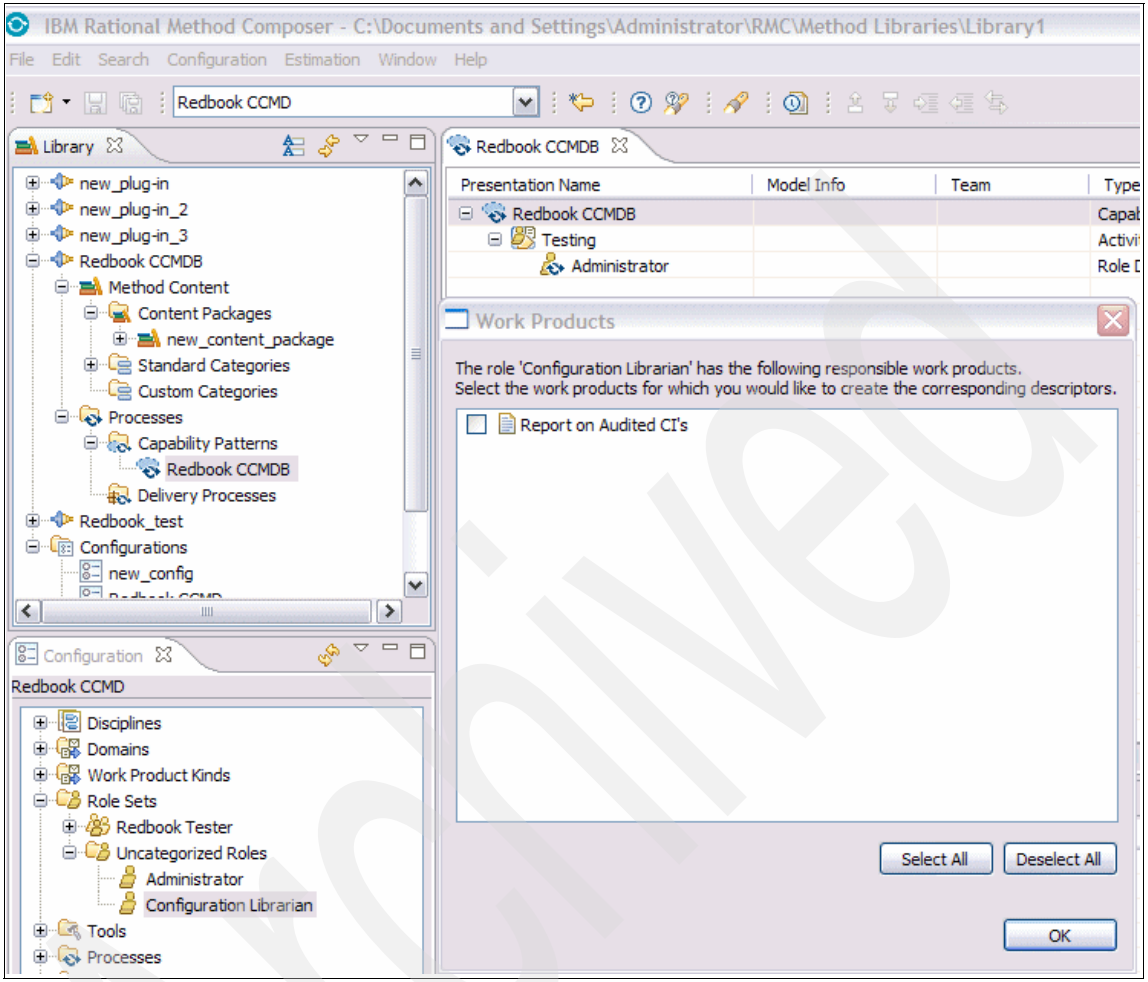


Figure 3-23 Selecting a work product on team allocation structure

For each selected work product, the wizard window prompts you to select tasks that produce these work products. Again, select zero to many tasks and then click **OK** to add these elements to the process.

Review the role descriptor's details in its properties view. If the properties view is not displayed, then select the role in the breakdown structure editor, right-click, and select **Properties**. Use the tabs on the side of the properties view to review different aspects of the role descriptor. It is possible to also perform individual modifications of the role descriptor, such as change the presentation name, add textual descriptions, change the work products the role is responsible for, and so on.

When changing the role descriptor's relationships in the property window tab's roles or work products, new elements from the method content can be added by using the **Add** button, or you can connect the role descriptor with work products already present in this activity.

Continue adding more roles to the activities, or switch to the work breakdown structure tab to add tasks or to the work product usage tab to add work products.

3.4.9 Developing the work product usage structure

Before starting, it is important to make sure that the configuration selected in the tool bar is the same as the configuration that was selected as the default configuration for the process. Figure 3-18 on page 53 shows how to check if the method configuration selected and the default configuration for the process are the same.

In the process editor, click the **Work Product Usage** tab to open the work product usage editor.

Right-click the element in the first row of the breakdown structure and then select **New Child** → **Activity** to create a new activity. Alternatively, you can create a phase or iteration, depending on the scope of your process. If needed, create more activities to set up your breakdown structure. Activities can be nested inside each other. Figure 3-24 on page 61 shows how to create a new child activity.

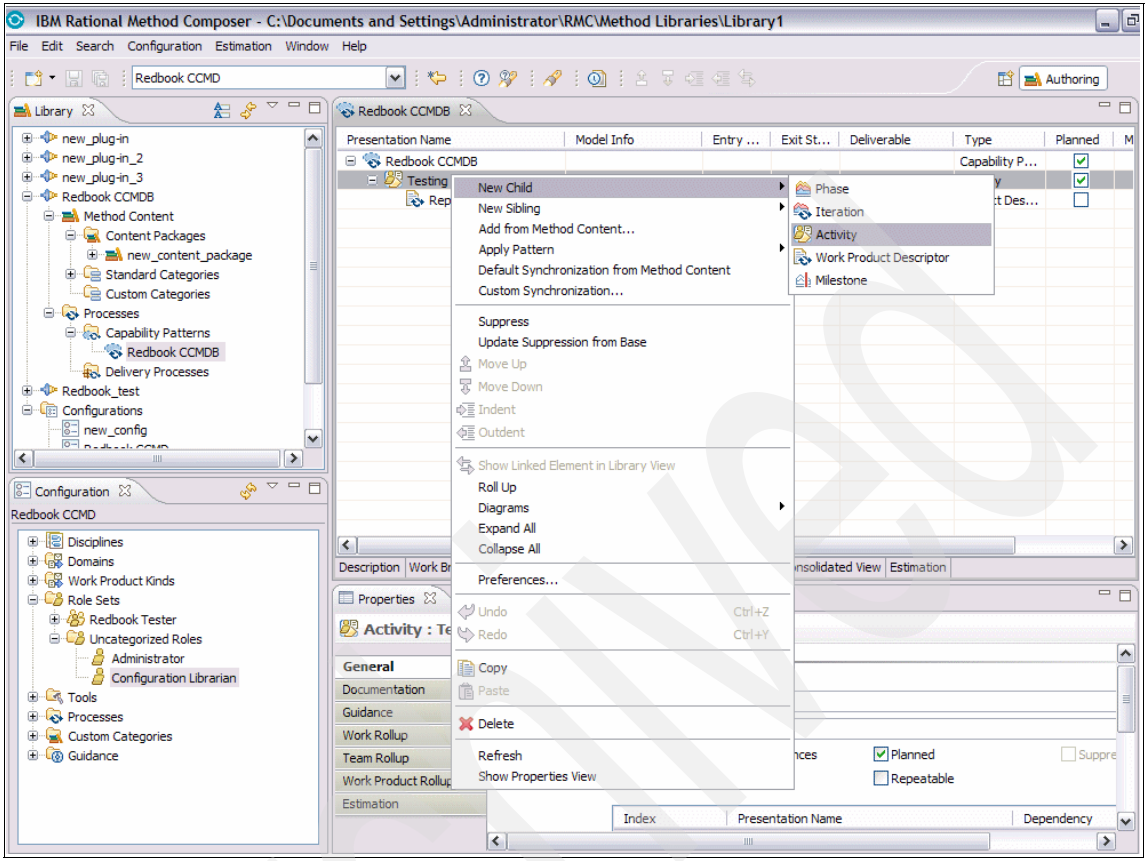


Figure 3-24 Creating a new work product usage child activity

Review the list of work products in the configuration view. In this view work, products are sorted by domain in addition to work product types. Drill into either of these hierarchies to see which work products are available in this configuration. Select a work product to add to an activity and then drag it on top of the activity created earlier. The work product is added as a work product descriptor (an occurrence of a work product in one specific activity).

Review the new work product descriptor's details in its properties view. If the properties view is not displayed, then select the work product descriptor in the process editor, right-click, and select **Properties**. Use the tabs on the side of the properties view to review different aspects of the work product descriptor. It is possible to also perform individual modifications of the role descriptor, such as change the presentation name, add textual descriptions, add entry and exit states, and so on.

When changing the role descriptor's relationships in the property window tab's roles or work products, new elements can be added from your method content by using the **Add** button, or by connecting the role descriptor with work products already present in this activity.

Continue adding more work products to the activities, or switch to the work breakdown structure tab to add tasks or the team allocation tab to add roles.

3.4.10 Applying a capability pattern or capability pattern activity

It is not necessary to develop a process from scratch by adding descriptors one by one as described in the previous steps. It is possible to reuse existing capability patterns or even capability pattern parts.

A capability pattern is a special process that describes a reusable cluster of activities in common process areas. Capabilities patterns express and communicate process knowledge for a key area of interest, such as a discipline, and can be directly used by a process practitioner to guide his work. They are also used as building blocks to assemble delivery processes or larger capability patterns ensuring optimal reuse and application of the key practices they express.The same pattern can be applied several times to the same process and define local modifications to each individual pattern application. In this way, it is possible to express specific changes each time the pattern is being performed throughout the life cycle that a process represents.

Finding an activity to which a pattern will be applied

Given that a process is already opened in the process editor, switch to the work breakdown structure tab and review the process. Find the location to apply a capability pattern. A capability pattern has to be applied to one specific activity (including an iteration or phase, which are special activities) in a process. Such an activity can either be defined locally in the process (presented as a name in a standard black font) or an activity that was added to the process by applying another capability pattern (presented as a name in a green-italic font).

If a pattern needs to be applied to a local activity (black font), go to "Selecting a capability pattern in the configuration view" on page 63. If there is a need to apply a pattern to an activity from another pattern (green italic font), go to "Contributing to an activity from another pattern" on page 62.

Contributing to an activity from another pattern

If a pattern needs to be applied to an activity from another pattern (recognizable by the green italic font), then an activity contribution needs to be created so that local changes can be made to the activity.

Find the activity's parent element. If this element is a not a local element (green italic font), then a contribution needs to be created first for this parent and so on (if the parent's parent is not local, then create a contribution to the parent's parent first, and so on)

To create a contribution to a non-local activity, right-click the non-local activity and then click **Contribute**. Do this with all parents, top-down, until you reach the activity to which the pattern should be applied. After clicking **Contribute**, the activity become local and is presented with a standard black font.

Selecting a capability pattern in the configuration view

Find a capability pattern in the configuration view. Expand the package **Processes** → **Capability Patterns** and its sub-packages. Select the pattern that should be applied to your process.

Applying the capability pattern or capability patterns activity

It is possible to apply either the whole capability pattern to the process or just one or more activities from it. To apply the whole capability pattern, drag the pattern over the activity selected earlier in your process. To apply one or more activities of the capability pattern, select them in the configuration view. Multiples selection can be done by pressing Ctrl + select, or Shift + select to capture an all-inclusive section.

Drag the selected items to the activity selected earlier in the process. After dragging the pattern or activities, the application will prompt you to apply the pattern or activities using Extends (dynamic binding) or Copy. (For more details about these two choices, see “Process Authoring Overview” in the ITUPC online help). Click your choice.

As an alternative to dragging the capability pattern, you can right-click the activity to which the pattern will be applied and select **Apply Pattern** → **Copy...** or **Apply Pattern** → **Extend...**.

Making local changes to a pattern application

If the pattern was applied by copying, you can freely make modifications to the copied elements. If the pattern was applied by extends (dynamic binding), you can still provide local additions to the pattern by defining a contribution to the pattern's activities. Follow the instructions in “Contributing to an activity from another pattern” on page 62 to define such a contribution. After creating the local activity contribution, additional descriptors or patterns can be added to this activity.

Suppressing pattern elements

To suppress elements, such as descriptors or activities of a dynamically bound pattern (through extends), right-click the element and select **Suppress**.

3.5 Working with processes

ITUP Composer can be used as a way to document and detail processes managed by CCMDB. Using the change process explained in 8.2, “Change Management Process Manager” on page 243 as an example, elements defined in CCMDB can be also created with more details in ITUP Composer and reference documents can be included as part of the definition of each element. The following sections show ITUP Composer sample windows for this change process.

3.5.1 Change process sample

Using the Job Plan presented in 8.2, “Change Management Process Manager” on page 243 as our sample, here are some of the steps that could be taken in ITUP Composer for the change process.

Artifacts

A Change Management process artifact may be detailed and documented in ITUP Composer. Figure 3-25 shows an example of this artifact in the left side of Figure 8-9 on page 247. The artifact, with all the inputs and outputs that are involved in receiving, approving and planning a change, is called “Request for Change” in the example.

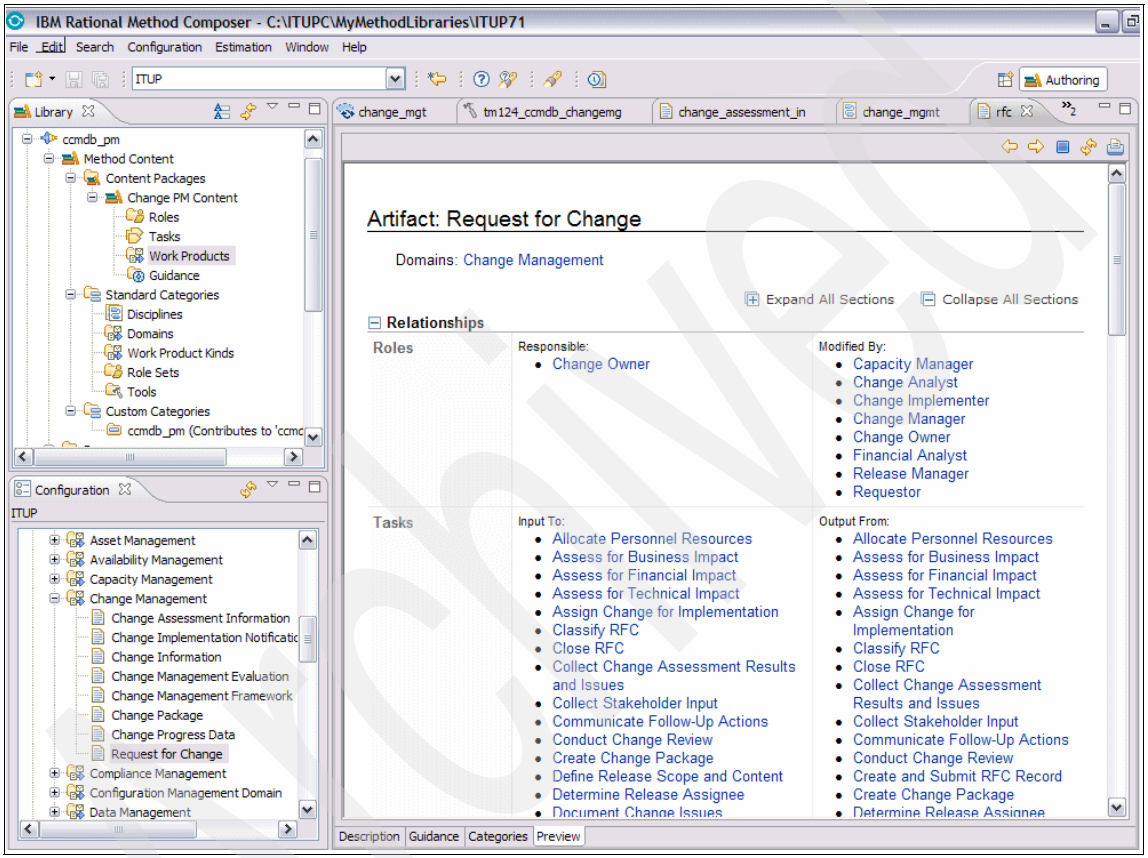


Figure 3-25 Request for Change artifact

Display all steps of a process

ITUP Composer allows viewing of all of the tasks of a process, including the predecessors, primary performers, inputs, and outputs for each task. Figure 3-26 shows an sample window for the Change Management process.

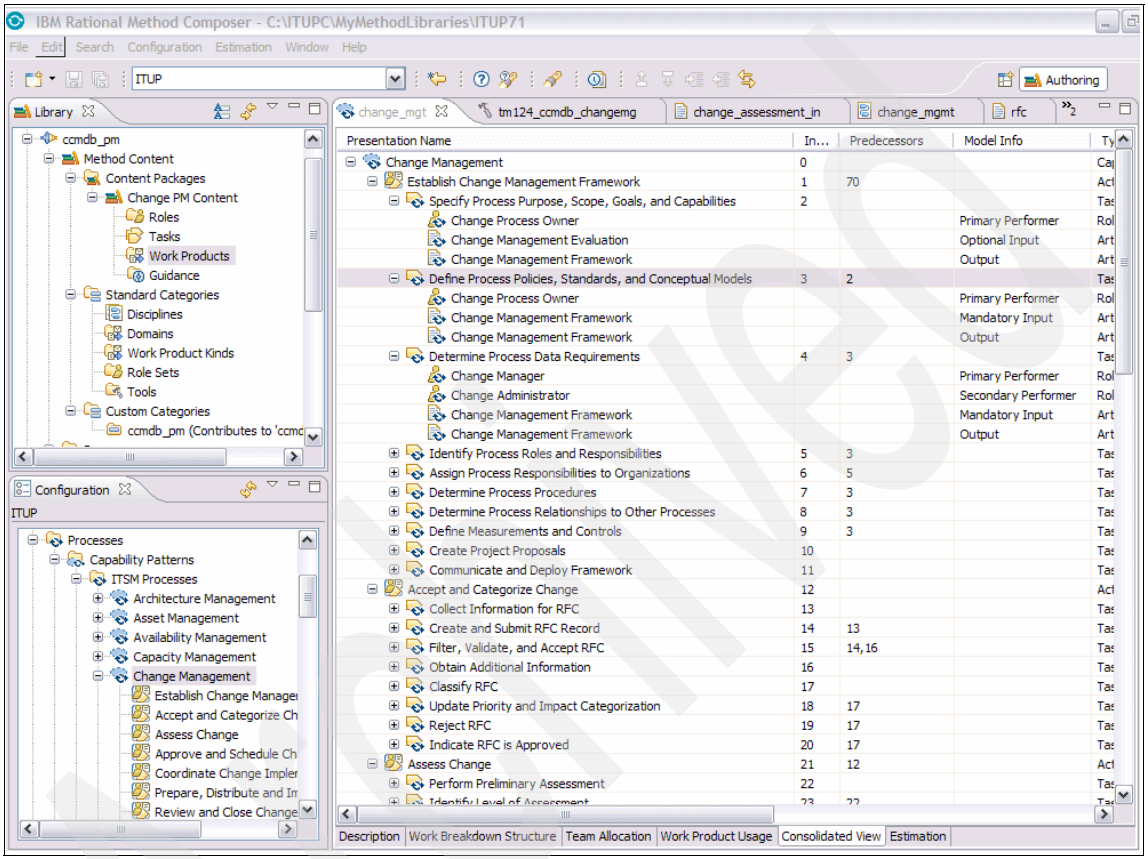


Figure 3-26 Change Management process tasks

View reference material using tool mentors

With ITUP Composer Tool Mentors, it is possible to include additional reference text for processes. Figure 3-27 on page 67 shows an example of reference material for the Change Management process.

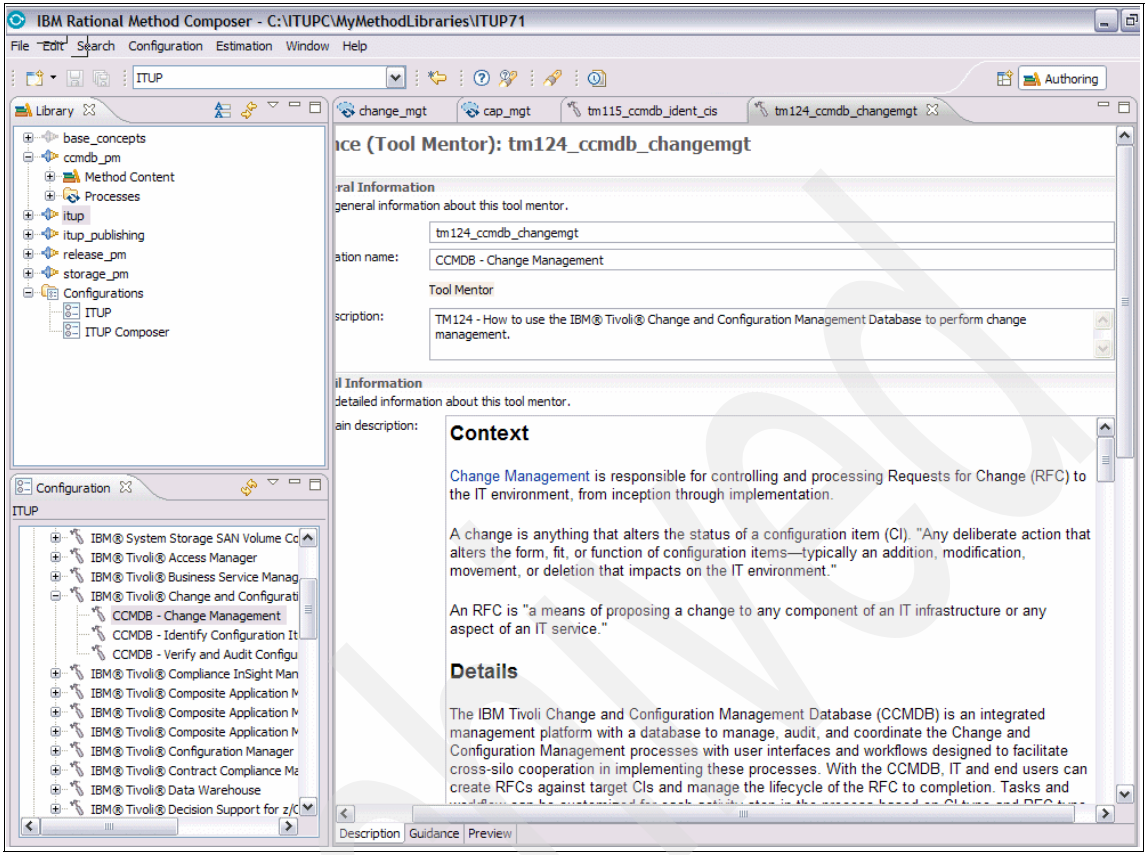


Figure 3-27 Tool Mentor reference material for the Change Management process

3.6 Summary

This chapter has provided an overview of how the ITUP Composer would be used to document processes such as Change and Configuration Management. Proper documentation and making that documentation available to those who use or are affected by the process is a critical factor in the successful deployment and use of a process. A tool such as ITUP Composer can make this documentation process easier and deliver high quality and usable documentation.

Part 2

Using and customizing the CCMDB Common Data Model

4

Data layer scenarios

The CCMDB data layer contains three data spaces that are all aligned to the Common Data Model (CDM) specification: discovered, actual, and Authorized CI data spaces. The CDM is a logical representation of CI object classes and their relationships. It is the metamodel, that is, the definition that prescribes, for example, the object types, the attributes, their relationships, and the cardinalities of the relationships. While the CDM is a conceptual representation, in this chapter we explain how the data is persisted in the CCMDB according to the rules of this model.

We explain the reason for having different data representations as well as the relationship between the different representations of configuration items in *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565.

In this chapter, we explain how the data model is implemented and what needs to be done in order to extend the model. We describe the major table structures inside the process layer database that hold the information about the model itself as well as the configuration item instance data.

Following the description of how the model is implemented, some important use case scenarios are described in detail throughout this chapter:

- The configuration of the ITIC adapters in order to bring CDM type and instance data over from the discovered to the Actual CI space.

- The overall topic of promotion. Promotion is referred to as the transfer of CI data from the actual to the Authorized CI space. There are two options for configuring the promotion:
  - Dual class approach
  - Manage with CI Hierarchy approach

  The reason for promoting data from the actual to the Authorized CI space is to make the data available for the different process manager products, such as Change or Configuration Management.
- How to configure Launch in Context definitions in the Launch in Context application in order to allow to launch in context from the Authorized or Actual CI application to external systems. The Launch in Context facility is used internally within CCMDB to launch from the authorized or actual view of configuration items to the more detailed view of the discovered configuration items.
- The steps necessary to configure a federation scenario. We use a step by step example of how to link the Actual CI application at runtime to an external data source without having to import the external data physically into the CCMDB data layer implementation.

4.1 Implementation of Actual and Authorized CI spaces

The overall CCMDB data layer is physically implemented in two different databases:

- The TADDM database, which holds the discovered configuration items
- The process layer database, which keeps the Actual and Authorized CI data

Our focus in this chapter is on the implementation of the data model inside the process layer database. We describe the major tables of the CCMDB schema that are used for persisting actual and authorized configuration item instance data, including their relationships. In addition, key table structures that are used to persist the data model itself are explained. This includes specification of object or class types, their respective attributes, and relationships between the class structures. All tables that we refer to in this chapter are physically part of the *MAXDB71 database*, which is the default name for the CCMDB process layer database.

Important: We do not provide a complete entity relationship diagram or data model of the CCMDB. Our intention is to highlight the major tables that are used for persisting CI data in the various data spaces of the CCMDB. The data schema is definitely using more tables in the context of CI data then described in this chapter. This chapter is intended to ease your burden when you are, for example, using the various CI-related applications, adapting an application user interface layout using the application designer application, setting up a federation scenario, or when you must extend the data model.

As a user or administrator, you are usually working with various applications provided by the process manager products, such as Change or Configuration Management or the base services layer. Applications like the change, the actual configuration items, the authorized configuration items, the classification, the relationships or the collection application are examples of application that are used to work on or administer the data inside the various tables of the CCMDB data schema.

Applications never use the data in the various tables directly. They always are associated with one or more Maximo® Business Objects (MBO). An MBO is a Java™ object with business logic that encapsulates a database table.

Figure 4-1 shows an overview of the major tables of the *MAXDB71 database* and its relationship to the major applications involved in using and administering CI data in the actual and authorized data spaces of the CCMDB.

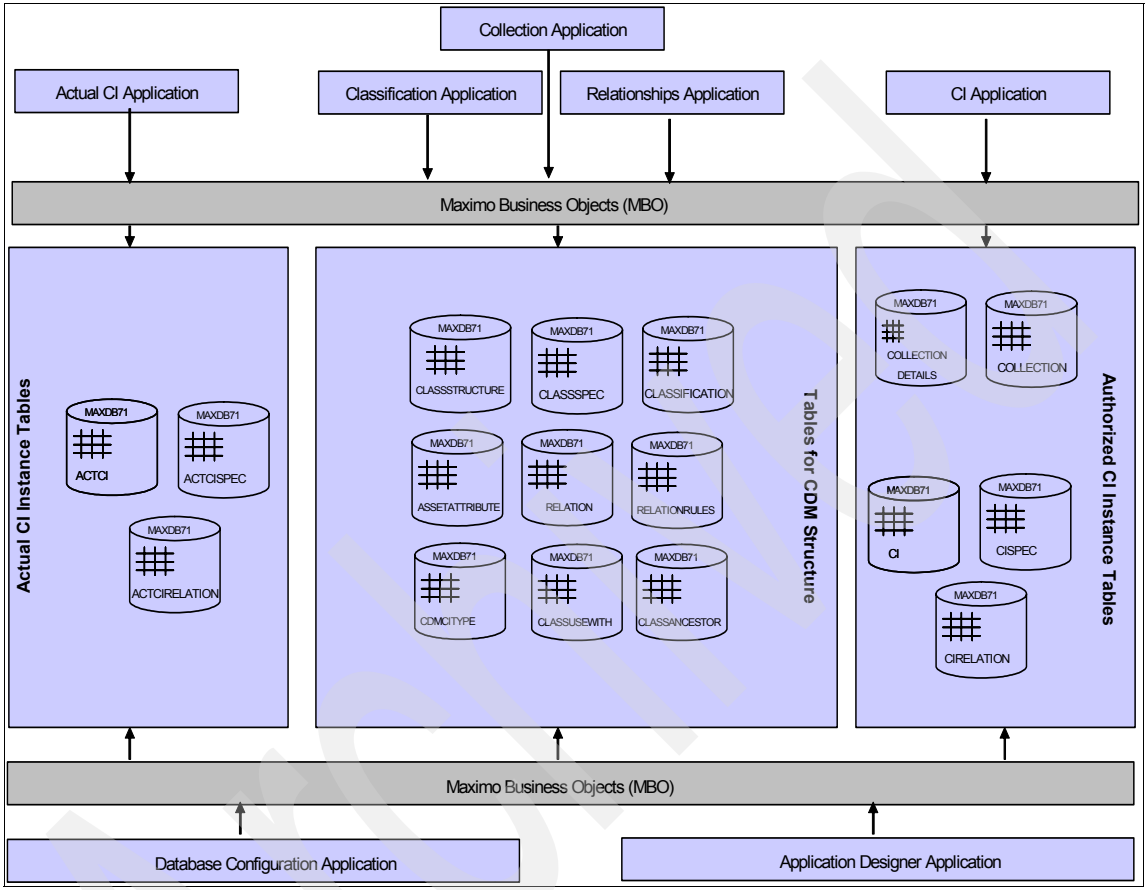


Figure 4-1 Major CCMDB Tables for Classification, Actual, and Authorized CI Data Spaces

The overview of the various tables in Figure 4-1 is divided into three different segments:

- The left box represents the major tables to persist actual configuration item data.
- The right box represents the major tables to persist authorized configuration item data.
- The box in the middle represents the major tables to persist the data model itself, object or class types, relationships and their cardinalities, attribute definitions, and classification structures.

As mentioned in this chapter already, applications always access the data layer using a Maximo Business Object.

The Database Configuration Application is the primary application to define the relationship between a database table and an MBO. There is a one-to-one relationship between an object and a table.

The Database Configuration application is to maintain the structure of the database tables. For example, you use it to change attribute definitions, set the field length and format of certain fields, or set up electronic audit records for certain fields. Please note that an attribute in the context of the Database Configuration application refers to a column of the appropriate database table. This is different from an attribute definition in the context of a configuration item. The attributes or fields that you maintain in the Database Configuration application are those fields that appear on the user interface of the various applications.

Figure 4-2 shows the ACTCI table encapsulated by the ACTCI object.

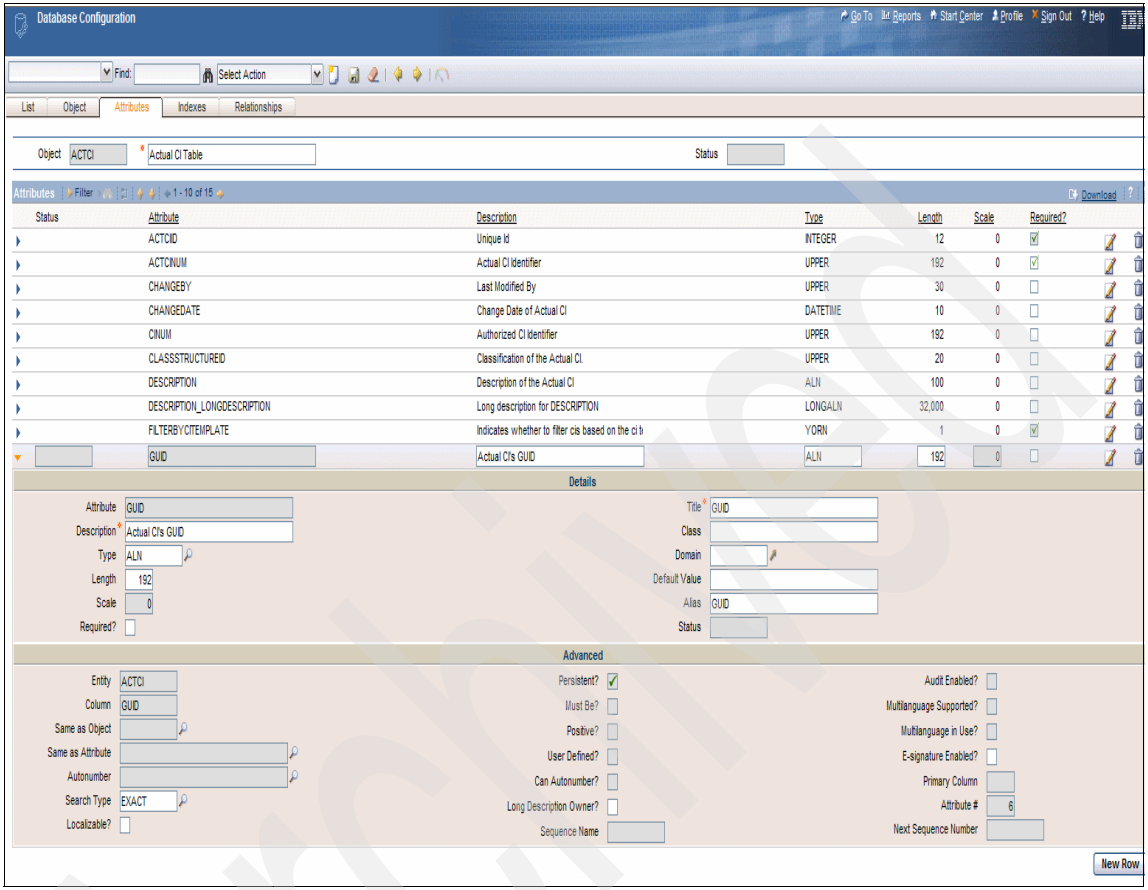


Figure 4-2 Database configuration application for ACTCI table

The listed attributes like ACTCIID, ACTCIINUM, or GUID refer to the columns of the ACTCI table in the MAXDB71 database. Each attribute, such as the GUID attribute, has a type, length, and various further characteristics. The attributes of the database object (the columns of the database table) can be selected or deselected in the Application Designer application when laying out the user interface of an application.

Each database object can have relationship definitions to other objects. The main purpose for these relationship is to leverage them in the user interface design of your application. Based on a relationship definition, attributes from a different database object can be transparently shown in the context of the application using the primary database object.

Figure 4-3 reveals an example of a database relationship definition for the ACTCI object to the CI object.

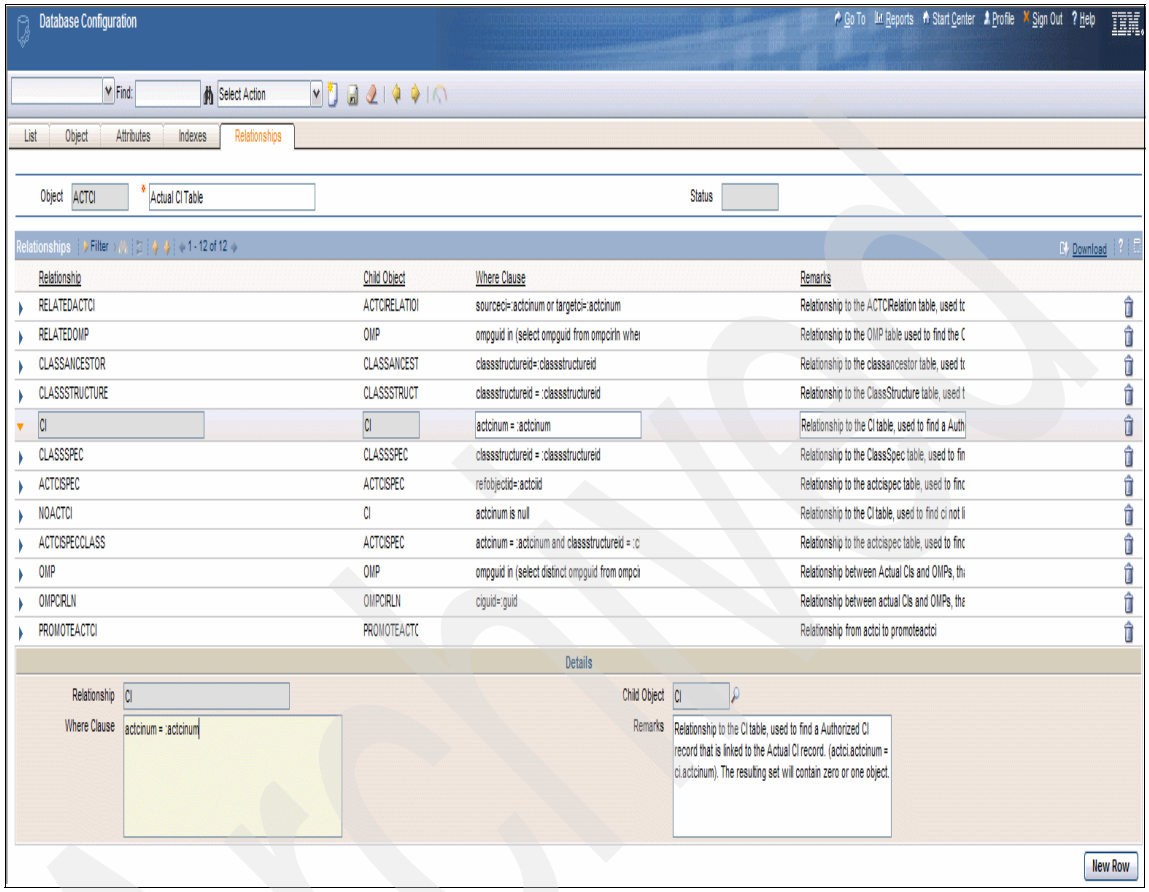


Figure 4-3 Relationship definitions of ACTCI object

The ACTCI object owns multiple relationship definitions to different database objects, for example, the CI object, as outlined in Figure 4-3. The relationship definition links attributes of the database objects together in the *Where Clause* definition field. By linking the objects, the application user interface allows you to show value fields of the related object. For the example shown in Figure 4-3, the related record of the CI table can be shown in an application that is using the ACTCI table. By default, this is the Actual Configuration Items application.

Important: Please note that the relationship definition is not equal to a definition of a foreign key relationship at the raw database layer. These kind of definitions are designed by IBM development and are not exposed through any CCMDB application.

In the Application Designer application, you leverage the objects defined in the Database Configuration application. The following example shows the definition of the Actual Configuration Items application layout and reveals that the primary or main object is *ACTCI*. Since a relationship to the *CI* object is defined, the attributes or fields of the CI table can be used in the Actual Configuration Items application.

You can define which field or attribute (a column in a table) you want to present to the user of the application. This includes fields enabled by the relationship definition. For a concrete example of using a relationship definition to an external database table, please refer to Chapter 6, “Implementing federation” on page 139.

Here we explain the purpose of the major tables outlined in Figure 4-1 on page 74.

We initially explain the tables that hold the CDM classification schema. In other words, these tables keep the rules and syntax of what CI types, what CI attributes, or which relationships between CI types are defined. These tables do not persist the CI instance data itself.

CLASSIFICATION table

The CLASSIFICATION table is primarily populated by the TADDM ITIC CI type adapter which brings over all CDM class definitions from TADDM into the process runtime database.

The CLASSIFICATION table keeps a list of all class types according to the CDM without a hierarchy definition between the different class types. Figure 4-4 on page 79 shows a query extract of the CLASSIFICATION table inside the DB2® Control Center.

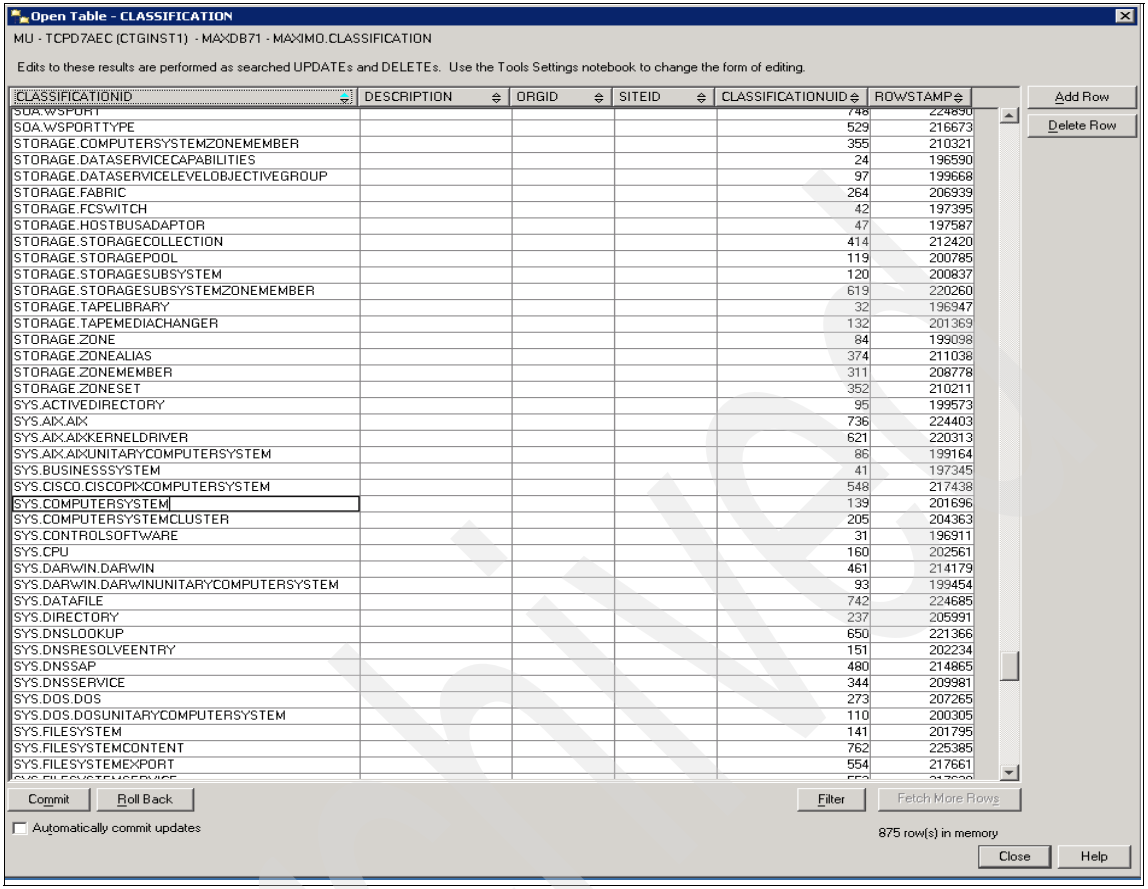


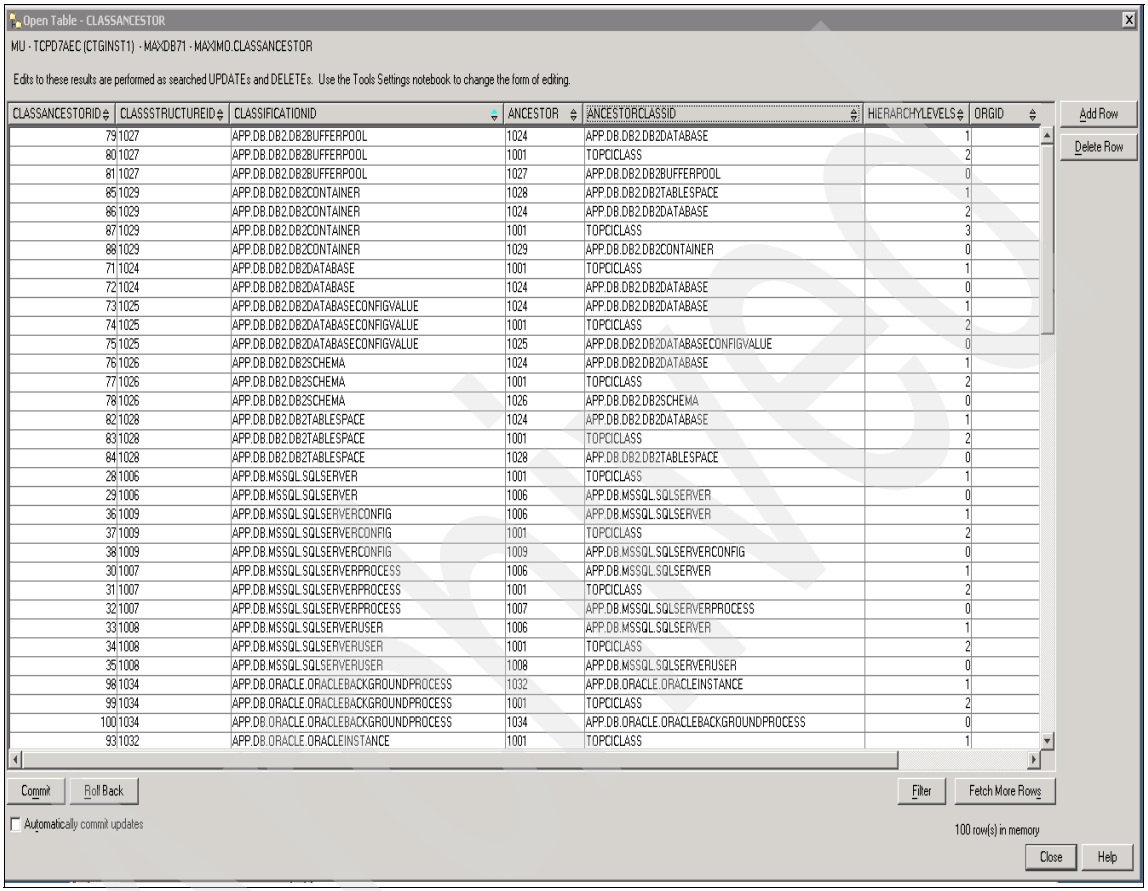
Figure 4-4 CLASSIFICATION table

The sequence in Figure 4-4 is in alphanumerical order in the CLASSIFICATIONID column. The CI class SYS.COMPUTERSYSTEM is highlighted, so you also can see various storage related CI class types.

The CLASSIFICATIONUID column keeps a unique sequential number, which is the primary key of this table.

CLASSANCESTOR table

The CLASSANCESTOR table (Figure 4-5) keeps track of the class hierarchy. It keeps an entry for each direct and indirect ancestor of a class up to the top of the hierarchy. The table is populated by the ITIC TADDM CI types adapter.



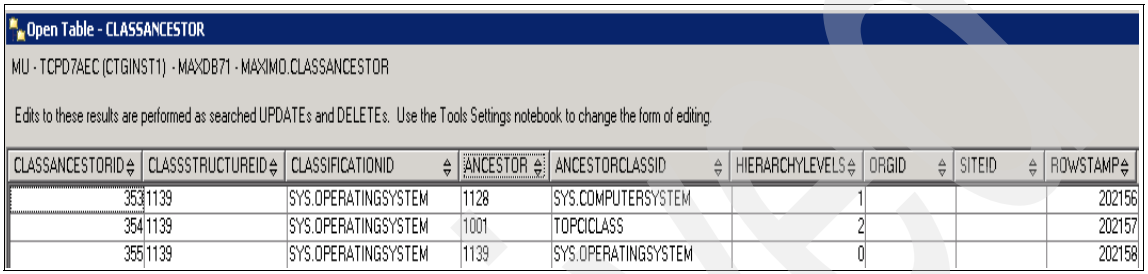
CLASSANCESTORID	CLASSSTRUCTUREID	CLASSIFICATIONID	ANCESTOR	ANCESTORCLASSID	HIERARCHYLEVELS	ORGID
79	1027	APP.DB.DB2.DB2BUFFERPOOL	1024	APP.DB.DB2.DB2DATABASE	1	
80	1027	APP.DB.DB2.DB2BUFFERPOOL	1001	TOPCICLASS	2	
81	1027	APP.DB.DB2.DB2BUFFERPOOL	1027	APP.DB.DB2.DB2BUFFERPOOL	0	
85	1029	APP.DB.DB2.DB2CONTAINER	1028	APP.DB.DB2.DB2TABLESPACE	1	
86	1029	APP.DB.DB2.DB2CONTAINER	1024	APP.DB.DB2.DB2DATABASE	2	
87	1029	APP.DB.DB2.DB2CONTAINER	1001	TOPCICLASS	3	
88	1029	APP.DB.DB2.DB2CONTAINER	1029	APP.DB.DB2.DB2CONTAINER	0	
71	1024	APP.DB.DB2.DB2DATABASE	1001	TOPCICLASS	1	
72	1024	APP.DB.DB2.DB2DATABASE	1024	APP.DB.DB2.DB2DATABASE	0	
73	1025	APP.DB.DB2.DB2DATABASECONFIGVALUE	1024	APP.DB.DB2.DB2DATABASE	1	
74	1025	APP.DB.DB2.DB2DATABASECONFIGVALUE	1001	TOPCICLASS	2	
75	1025	APP.DB.DB2.DB2DATABASECONFIGVALUE	1025	APP.DB.DB2.DB2DATABASECONFIGVALUE	0	
76	1026	APP.DB.DB2.DB2SCHEMA	1024	APP.DB.DB2.DB2DATABASE	1	
77	1026	APP.DB.DB2.DB2SCHEMA	1001	TOPCICLASS	2	
78	1026	APP.DB.DB2.DB2SCHEMA	1026	APP.DB.DB2.DB2SCHEMA	0	
82	1028	APP.DB.DB2.DB2TABLESPACE	1024	APP.DB.DB2.DB2DATABASE	1	
83	1028	APP.DB.DB2.DB2TABLESPACE	1001	TOPCICLASS	2	
84	1028	APP.DB.DB2.DB2TABLESPACE	1028	APP.DB.DB2.DB2TABLESPACE	0	
28	1006	APP.DB.MSSQL.SQLSERVER	1001	TOPCICLASS	1	
29	1006	APP.DB.MSSQL.SQLSERVER	1006	APP.DB.MSSQL.SQLSERVER	0	
36	1009	APP.DB.MSSQL.SQLSERVERCONFIG	1006	APP.DB.MSSQL.SQLSERVER	1	
37	1009	APP.DB.MSSQL.SQLSERVERCONFIG	1001	TOPCICLASS	2	
38	1009	APP.DB.MSSQL.SQLSERVERCONFIG	1009	APP.DB.MSSQL.SQLSERVERCONFIG	0	
30	1007	APP.DB.MSSQL.SQLSERVERPROCESS	1006	APP.DB.MSSQL.SQLSERVER	1	
31	1007	APP.DB.MSSQL.SQLSERVERPROCESS	1001	TOPCICLASS	2	
32	1007	APP.DB.MSSQL.SQLSERVERPROCESS	1007	APP.DB.MSSQL.SQLSERVERPROCESS	0	
33	1008	APP.DB.MSSQL.SQLSERVERUSER	1006	APP.DB.MSSQL.SQLSERVER	1	
34	1008	APP.DB.MSSQL.SQLSERVERUSER	1001	TOPCICLASS	2	
35	1008	APP.DB.MSSQL.SQLSERVERUSER	1008	APP.DB.MSSQL.SQLSERVERUSER	0	
98	1034	APP.DB.ORACLE.ORACLEBACKGROUNDPROCESS	1032	APP.DB.ORACLE.ORACLEINSTANCE	1	
99	1034	APP.DB.ORACLE.ORACLEBACKGROUNDPROCESS	1001	TOPCICLASS	2	
100	1034	APP.DB.ORACLE.ORACLEBACKGROUNDPROCESS	1034	APP.DB.ORACLE.ORACLEBACKGROUNDPROCESS	0	
93	1032	APP.DB.ORACLE.ORACLEINSTANCE	1001	TOPCICLASS	1	

Figure 4-5 CLASSANCESTOR table

The table keeps an entry for the class itself and each of its ancestors in the hierarchy. At the top of the window, above the class type, APP.DB.DB2.DB2BUFFERPOOL reveals three entries, one for itself, one for its direct parent class, which is APP.DB.DB2.DB2DATABASE, and finally the top class, TOPCICLASS. The HIERARCHYLEVELS column keeps a value that reveals how many levels up in the hierarchy the ancestor class is away from the respective class. The ANCESTOR and ANCESTORCLASSID columns keep values referring to the the ancestor class identifier, which is the CLASSTRUCTURE identifier of the class, and the name of the ancestor class.

Note: The hierarchy level refers to the depth level that you configure in the ITIC type adapter. If you configure a depth level of three in the ITIC adapter, the highest number of hierarchy levels up in the chain to the highest ancestors should not surpass three.

Another example is the SYS.OPERATINGSYSTEM class. It is two levels down in the hierarchy from the top. It is a child class of SYS.COMPUTERSYSTEM and indirectly related to TOPCICLASS (Figure 4-6).


Open Table - CLASSANCESTOR

MU - TCPD7AEC (CTGINST1) - MAXDB71 - MAXIMO.CLASSANCESTOR

Edits to these results are performed as searched UPDATEs and DELETEs. Use the Tools Settings notebook to change the form of editing.

CLASSANCESTORID	CLASSSTRUCTUREID	CLASSIFICATIONID	ANCESTOR	ANCESTORCLASSID	HIERARCHYLEVELS	ORGID	SITEID	ROWSTAMP
353	1139	SYS.OPERATINGSYSTEM	1128	SYS.COMPUTERSYSTEM	1			202156
354	1139	SYS.OPERATINGSYSTEM	1001	TOPCICLASS	2			202157
355	1139	SYS.OPERATINGSYSTEM	1139	SYS.OPERATINGSYSTEM	0			202158

Figure 4-6 SYS.OPERATINGSYSTEM class type in CLASSANCESTOR table

A hierarchy level of 0 always indicates a pointer to the class itself.

CLASSSTRUCUTURE table

The CLASSSTRUCTURE table leverages the unrelated class type entries in the CLASSIFICATION table in order to make them available to the classification application. The classification application is used to present the classes in a defined structure to the user according to the hierarchy of the class types. The table is populated by the ITIC TADDM CI types adapter.

Figure 4-7 shows the CLASSSTRUCTURE table.

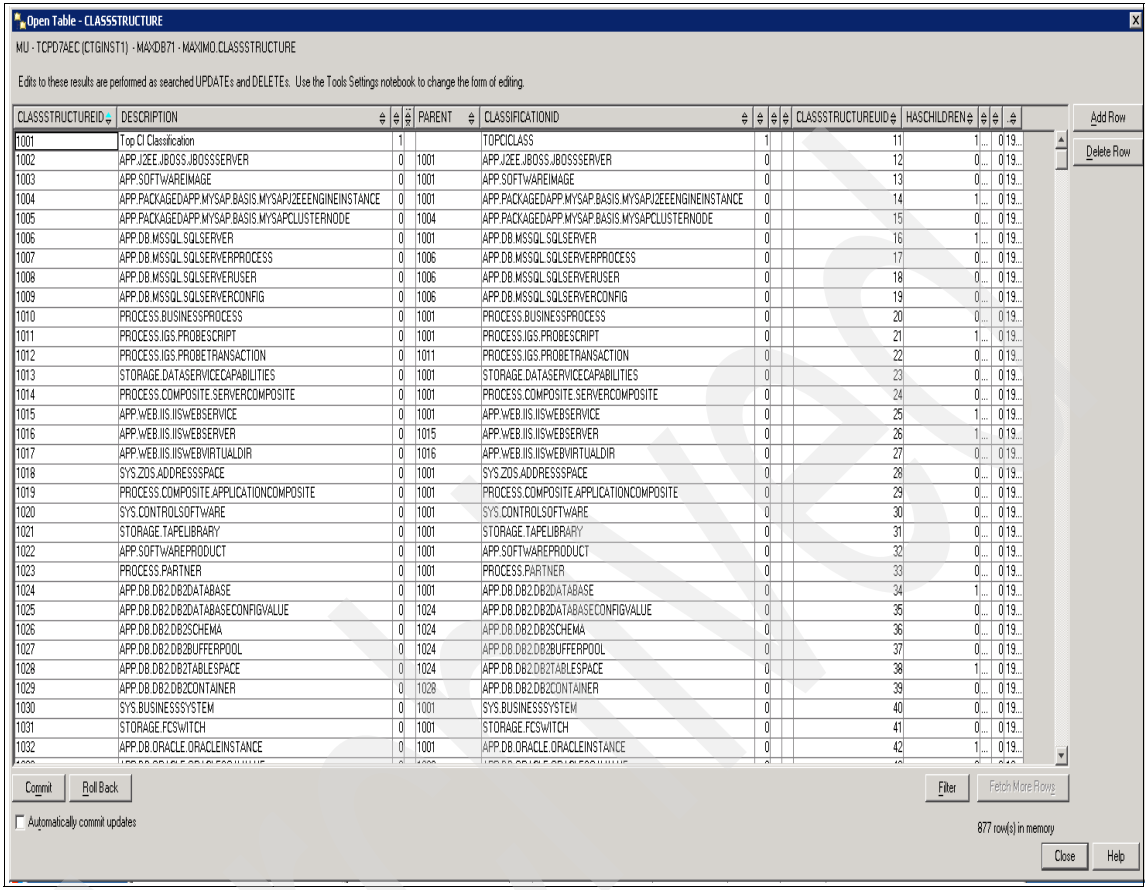


Figure 4-7 CLASSSTRUCTURE table

You can see that the column CLASSSTRUCTUREID is a sequence number given to the single classifications. The TOPCICLASS has the lowest sequence number.

The parent column refers to the CLASSSTRUCTUREID of the parent class, while the HASCHILDREN column indicates if the class has child classes.

If we bring up the CLASSSTRUCTUREID table object definition in the Database Configuration application, the columns of the database table are shown as attributes, as shown in Figure 4-8 on page 83.

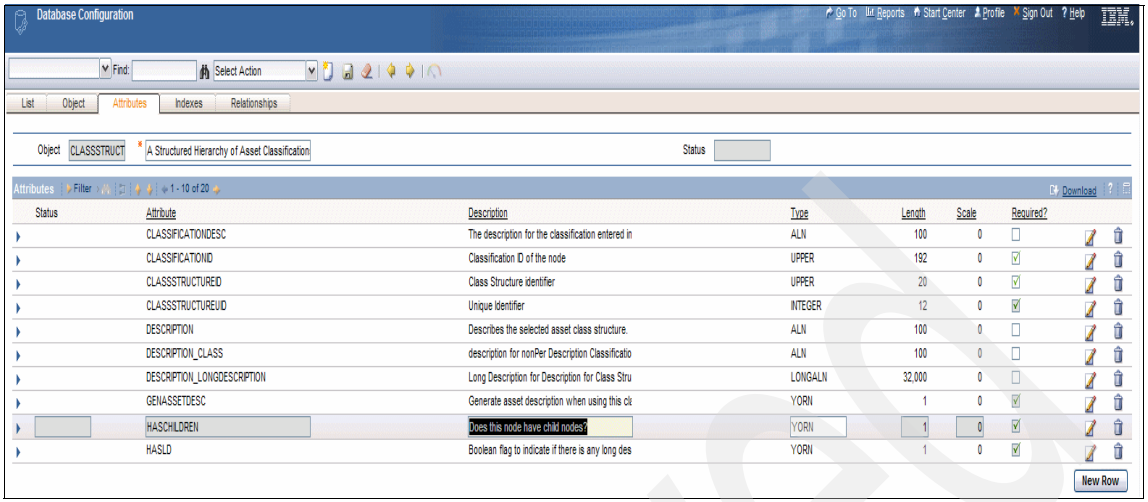


Figure 4-8 CLASSSTRUCTURE Object in database configuration application

You can, for example, recognize the HASCHILDREN attribute that you see in Figure 4-7 on page 82 as a column of the database table. HASCHILDREN, as well as most of the other attributes, are defined as persistent attributes. This means the values of the instance values will be persisted in the database table.

In addition to persistent attributes, an attribute value can be calculated at runtime from the values of other attributes. The HIERARCHYPATH attribute is an example of a non-persistent attribute. You cannot find a related table in the CLASSSTRUCTURE table, as it is calculated by applying the Java class specified in the class field of the attribute definition, as shown in Figure 4-9.

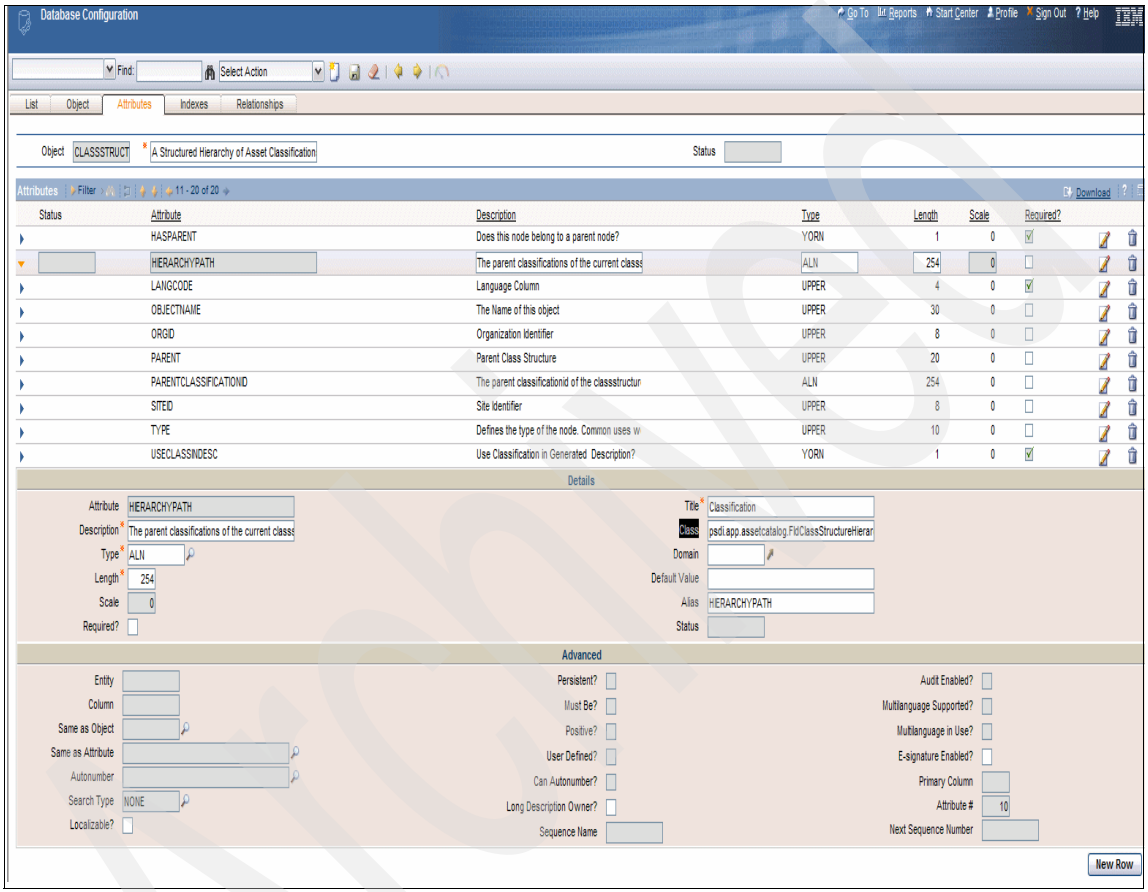


Figure 4-9 Non persistent attribute HIERARCHYPATH

The purpose of the calculation is to present the classification structure according to the parent child relationships of the class types in a path separated structure in the classification application, as shown in Figure 4-10 on page 85.

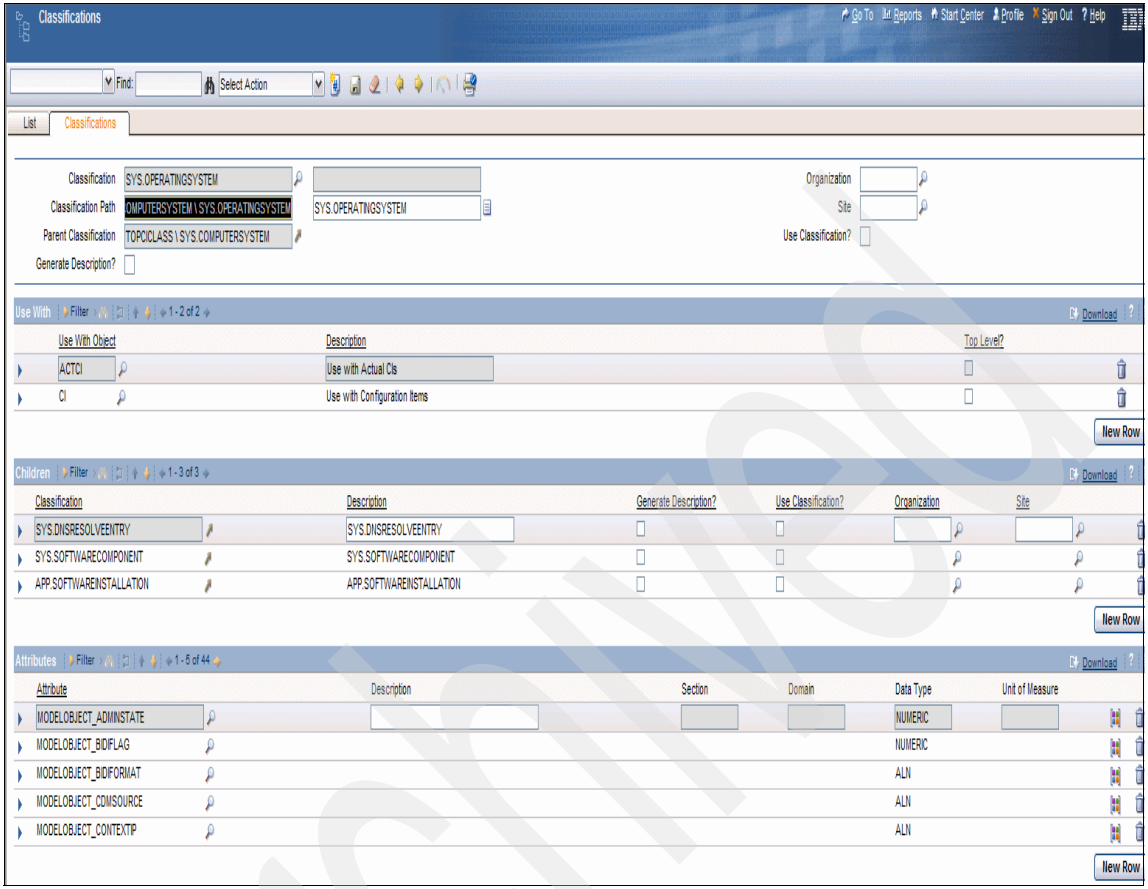


Figure 4-10 Classification structure in classification application

The Classification Path field holds the complete path of the classification up to the upper most ancestor.

If we press Alt-F1 in the Classification Path field in order to reveal the database attribute behind the entry field, we can see that the calculated non-persistent HIERARCHYPATH attribute of the CLASSSTRUCTURE object is used, as shown in Figure 4-11.

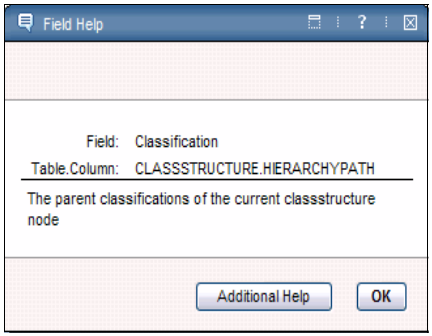


Figure 4-11 Classification Path field attribute definition

CDMCITYPES table

The main reason for the CDMCITYPES table is to keep track if a class type called CLASSSTRUCTUREID is active or not.

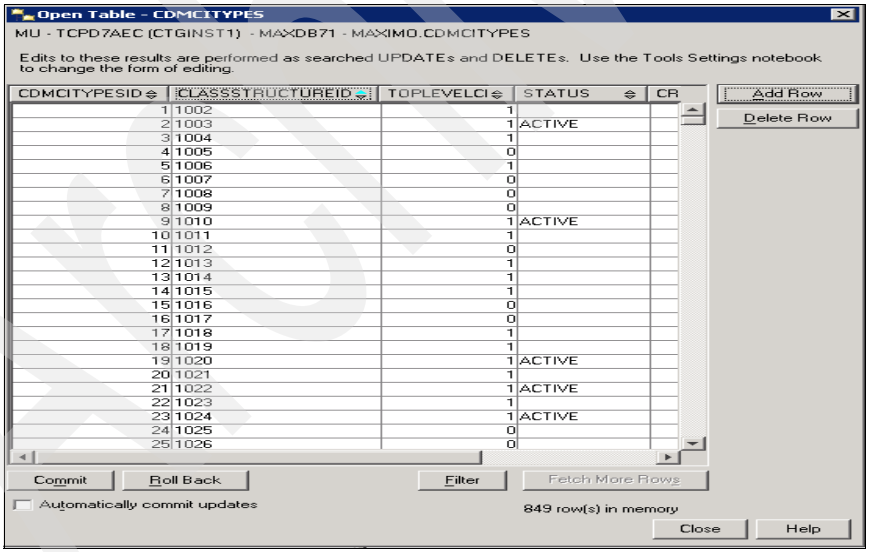


Figure 4-12 CDMCITYPES table

You have to activate a specific class type after the types have been populated into the database through the ITIC TADDM CI type adapter. The activation is a way to control which class types instance data gets populated into the actual

configuration item tables through the ITIC TADDM Actual CI adapter. The less classes you activate, the less instance data gets populated into the Actual CI space.

You control the activation of the CI types in the CI Types application, as shown in Figure 4-13.

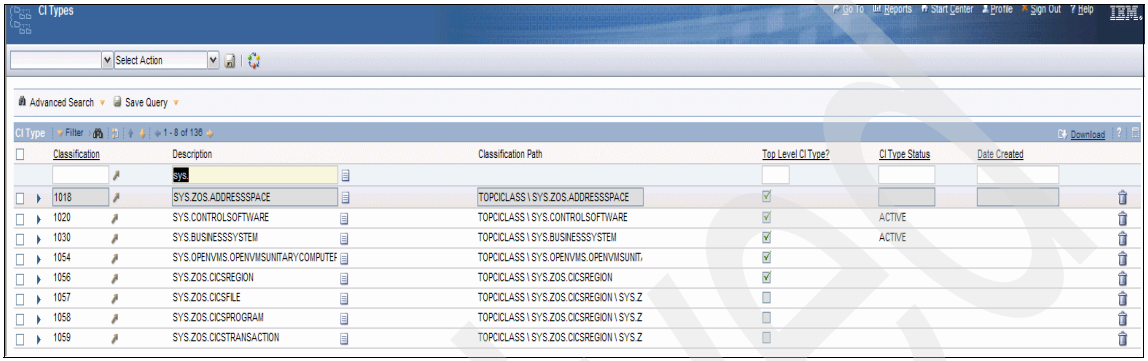


Figure 4-13 Activation of CI types in the CI Type application

CLASSUSEWITH table

Another impact of activating a CI type in the CI Types application is to associate the class type with the ACTCI object (Figure 4-14). This allows you to work with the instance data of the specific class type within the Actual Configuration Items application.

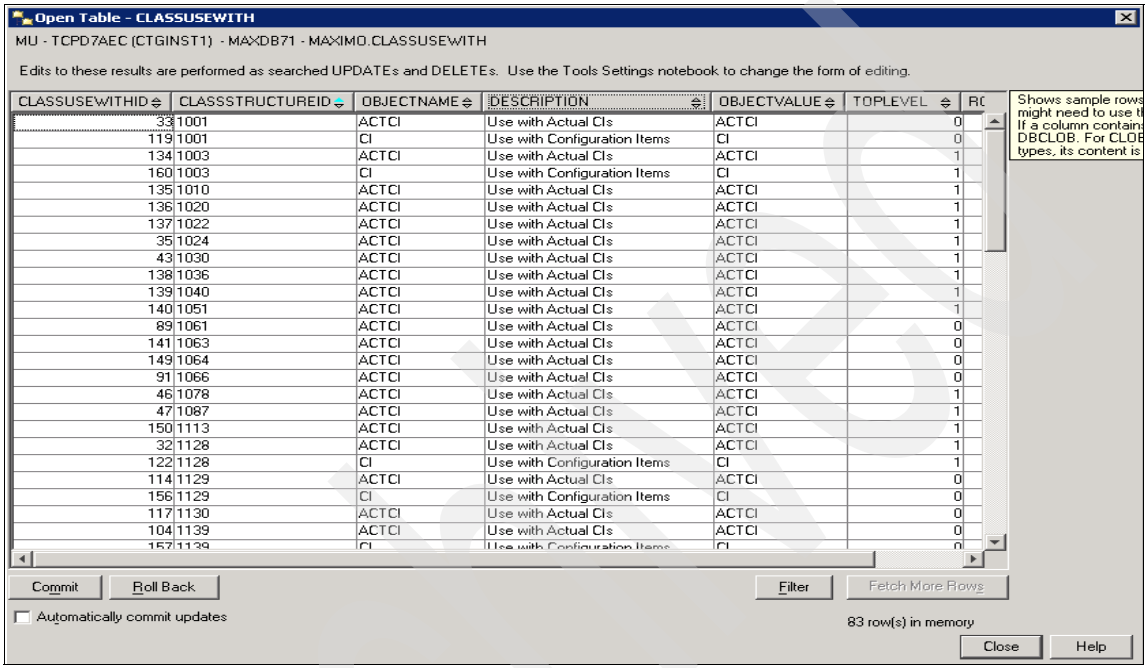


Figure 4-14 CLASSUSEWITH table

The CLASSUSEWITH table holds a record for each object that is allowed to work with the instance data related to the respective class types. In Figure 4-14, you can see, for example, that most of the class types, specified by the CLASSSTRUCTUREID, are allowed to be accessed through the ACTCI object, while just a few are allowed to be accessed through the CI object. The ACTCI and CI object are the main objects for the Actual Configuration Items and Configuration Items application.

The classification application reveals the objects that are allowed to access data of specific object classes (see Figure 4-15 on page 89).

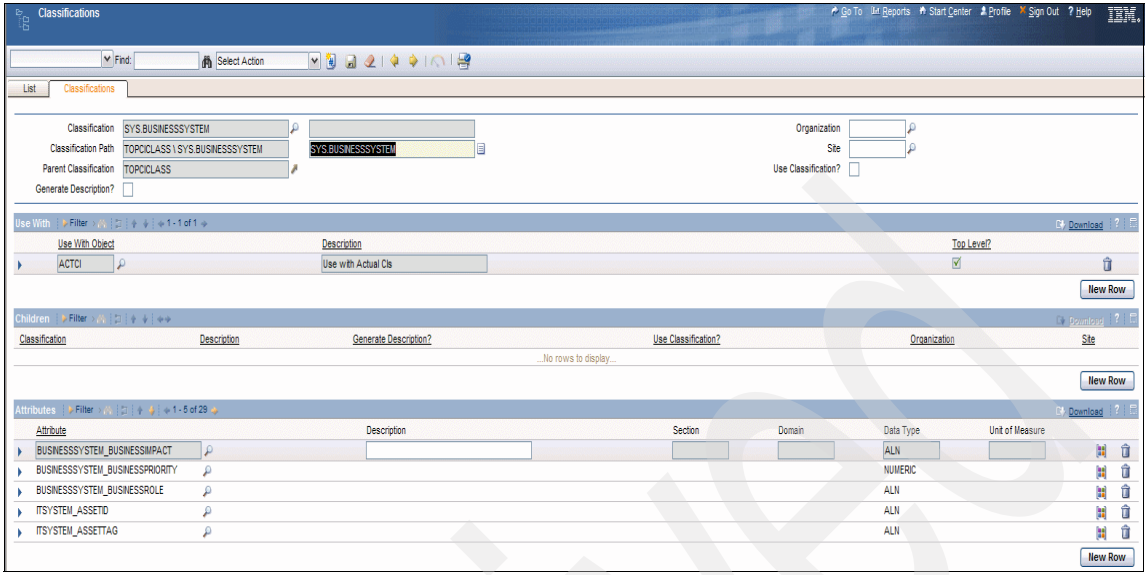


Figure 4-15 Use with Object field in classification application

The ACTCI object is automatically added to the Use with Object field in case you activate a CI type in the CI Types application. The SYS.BUSINESSSYSTEM class is shown with an entry for the ACTCI object in the Use with Object field. In Figure 4-13 on page 87, we show that this class has been activated in the CI Types application.

ASSETATTRIBUTE table

All the tables we have explained so far are related to class type definitions. We did not explain where attribute definitions related to the various class types are kept.

The ASSETATTRIBUTE table keeps track of all attribute definitions that are transferred from TADDM through the ITIC TADDM CI Type adapter. Please bear in mind that we are still talking about schema definition tables, and not yet referring to tables that keep the instance records of the data.

Similar to the CLASSIFICATION table for class type definitions, the ASSETATTRIBUTE table keeps records for all attribute definitions.

Figure 4-16 reveals an extract of the attribute definitions that are transferred from TADDM. Each attribute definition has a type definition and a unique ID.

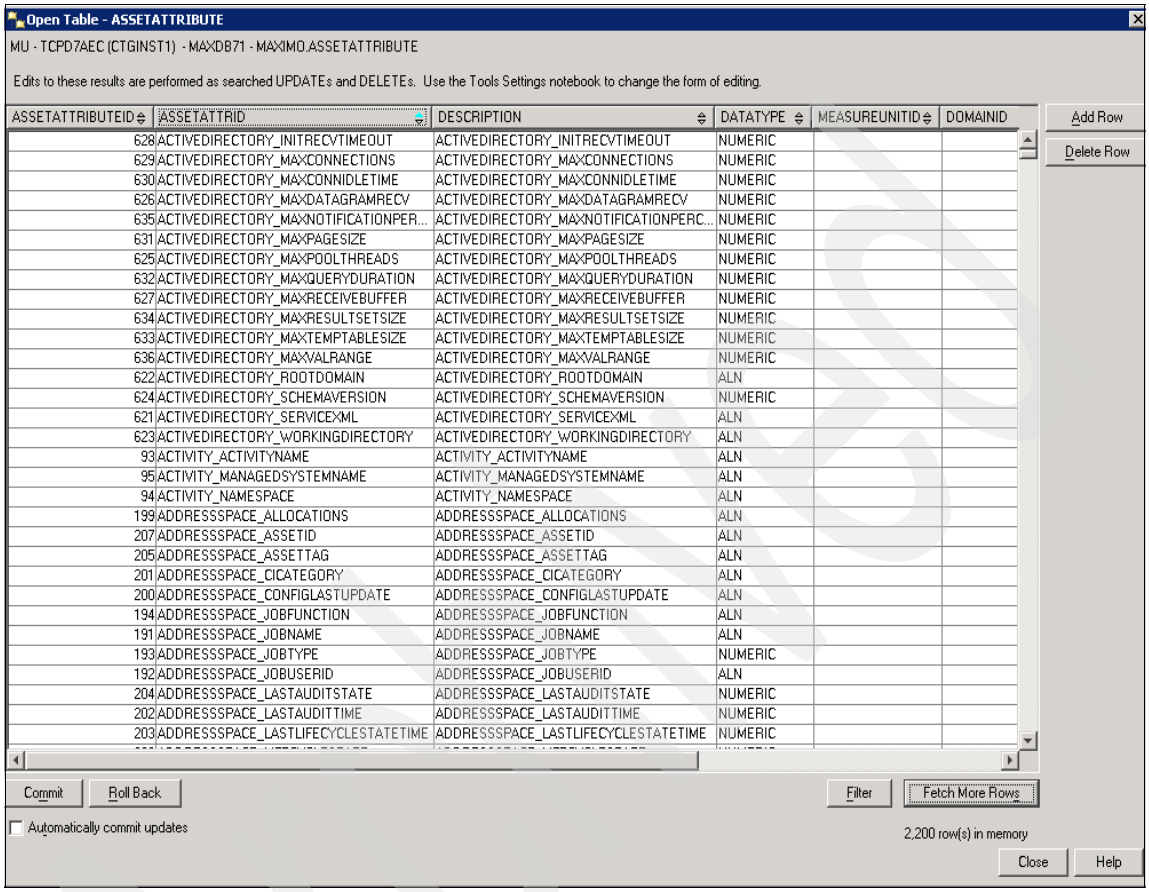


Figure 4-16 ASSETATTRIBUTE table

You see various attributes related to class types like Active Directory® or z/OS® address spaces in Figure 4-16.

CLASSSPEC table

While the ASSETATTRIBUTE table keeps records for each attribute definition, the CLASSSPEC table make these definitions available to the classification application and associates them with a CLASSSTRUCTUREID, which represents a class type (see Figure 4-17 on page 91).

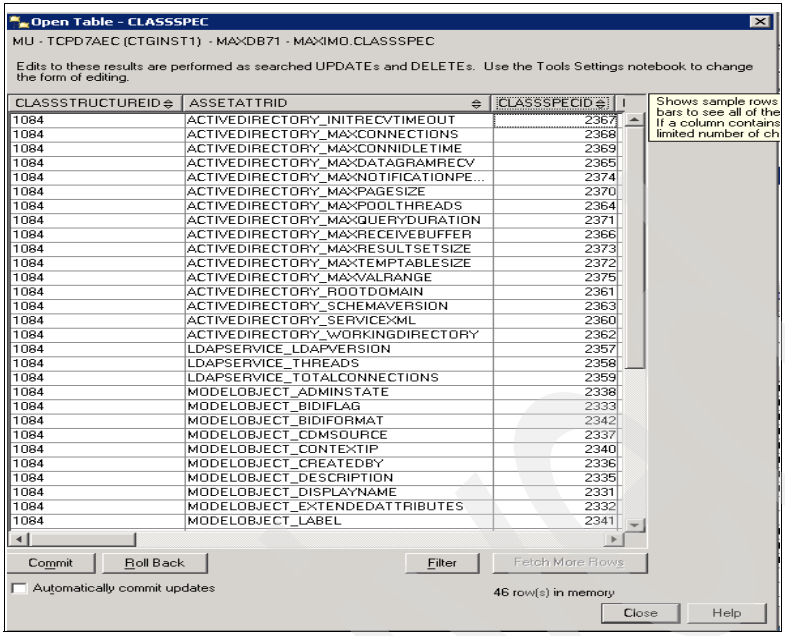


Figure 4-17 CLASSSPEC table

In the CLASSSPEC table, you also find the ASSETATTRID value of the ASSETATTRIBUTE table.

In the classifications application, the attribute definitions have the settings shown in Figure 4-18.

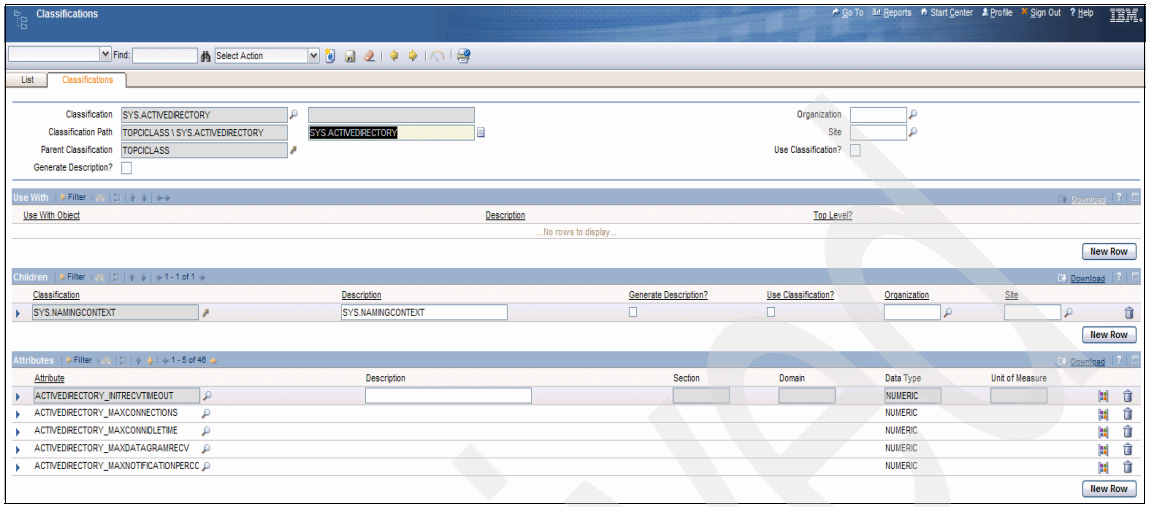


Figure 4-18 Attribute definitions in classification application

Attributes related to the SYS.ACTIVEDIRECTORY class type are shown in Figure 4-18.

RELATION table

After we explained where class type and attribute definitions are kept, we now explain where relationship definitions are kept in the database. Relationship definitions according to the CDM are class types themselves, so you can also see them in tables like CLASSIFICATION and CLASSSTRUCTURE.

The RELATION table keeps track of the relationship definitions that are populated by the ITIC TADDM CI Type adapter.

You can see in Figure 4-19 that 90 predefined relationship definitions are kept in the table. You can add your own relationship definitions in the Relationships application, but initially the relations are brought over from TADDM. The TYPE column holds the definitions if the relation is defined as unidirectional or bidirectional. The RELATION table also has a CLASSSTRUCTUREID columns, which is a hint that the relationship is a class type by itself.

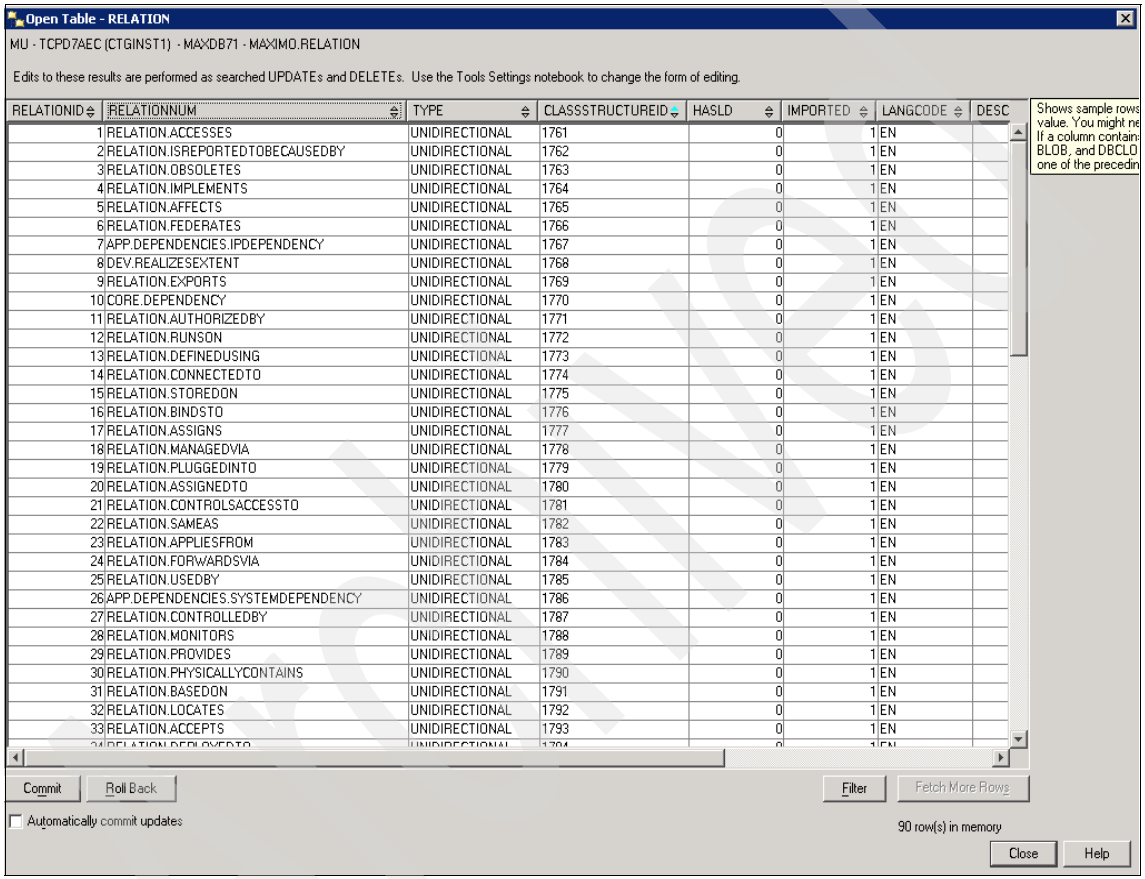


Figure 4-19 RELATION table

Figure 4-20 shows an excerpt from the CDM Web application showing all default relationship definitions. The CDM Web application represents all CCMDB class types, attributes, and relationship definitions in a graphical way.

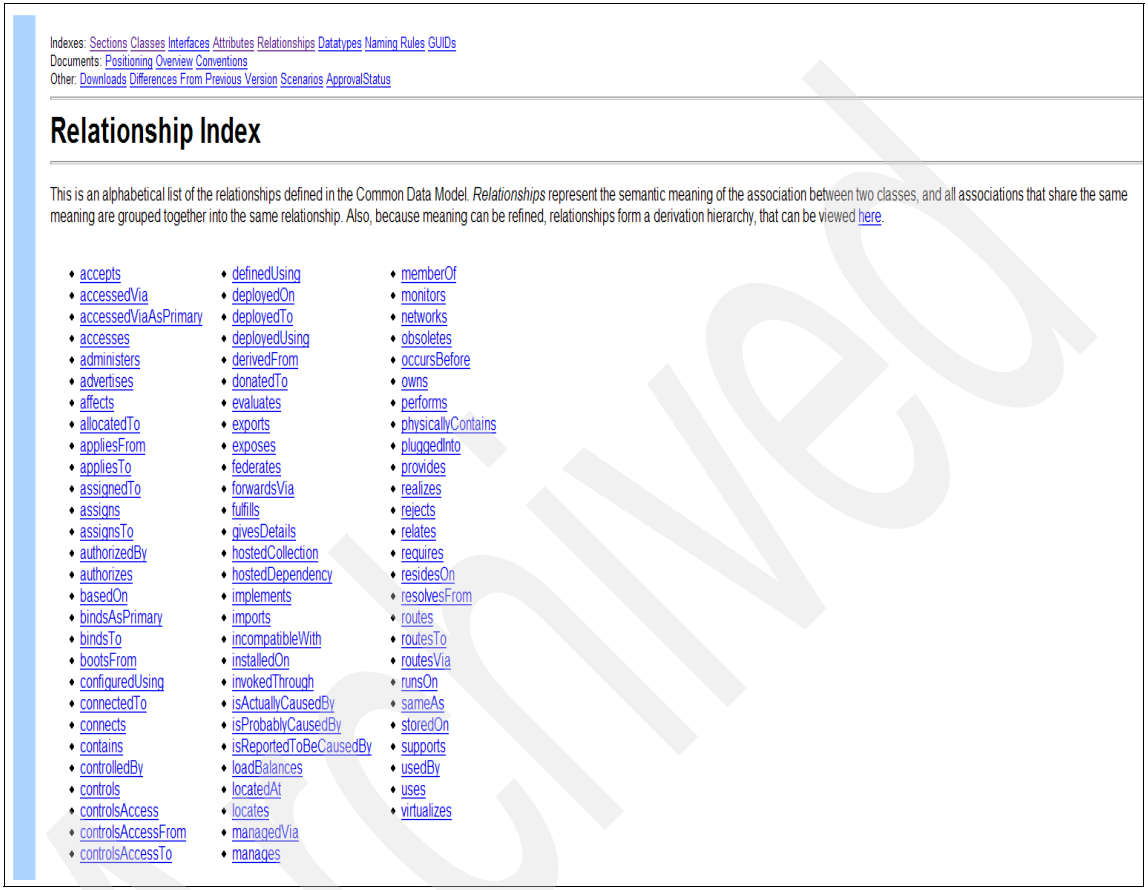


Figure 4-20 CDM Relationship Index

These are the relationship definitions that get imported into the RELATION table by the ITIC TADDM CI Type adapter. In the Relationships application, the definitions appear as shown in Figure 4-21 on page 95.

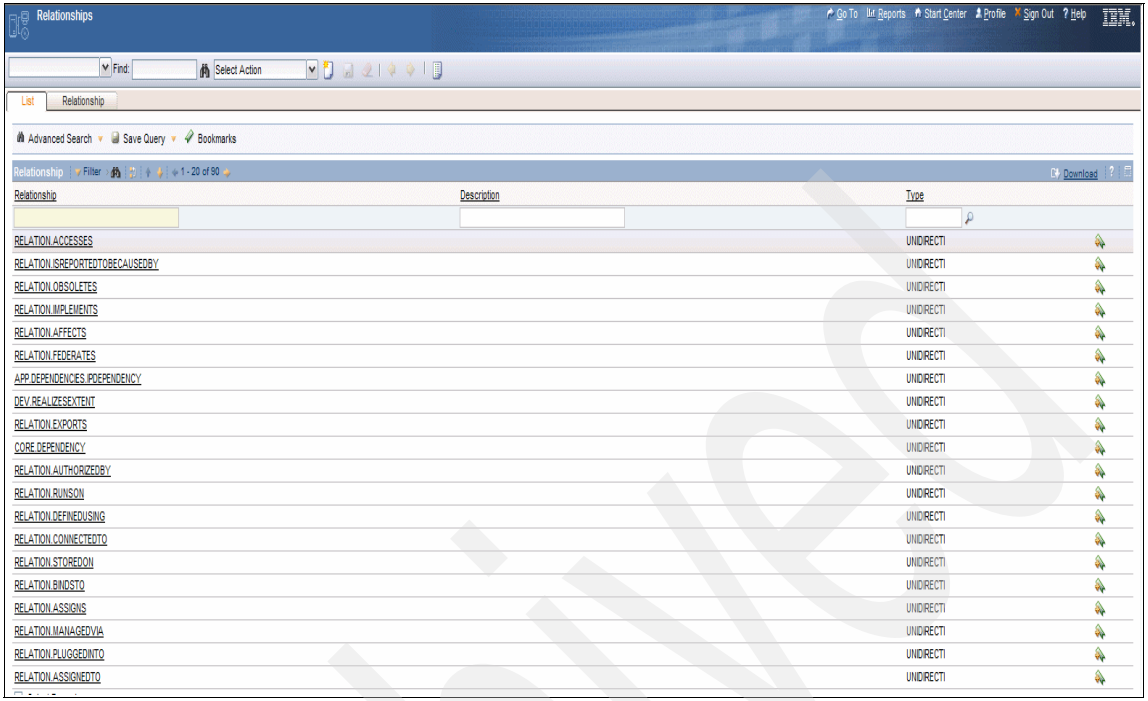
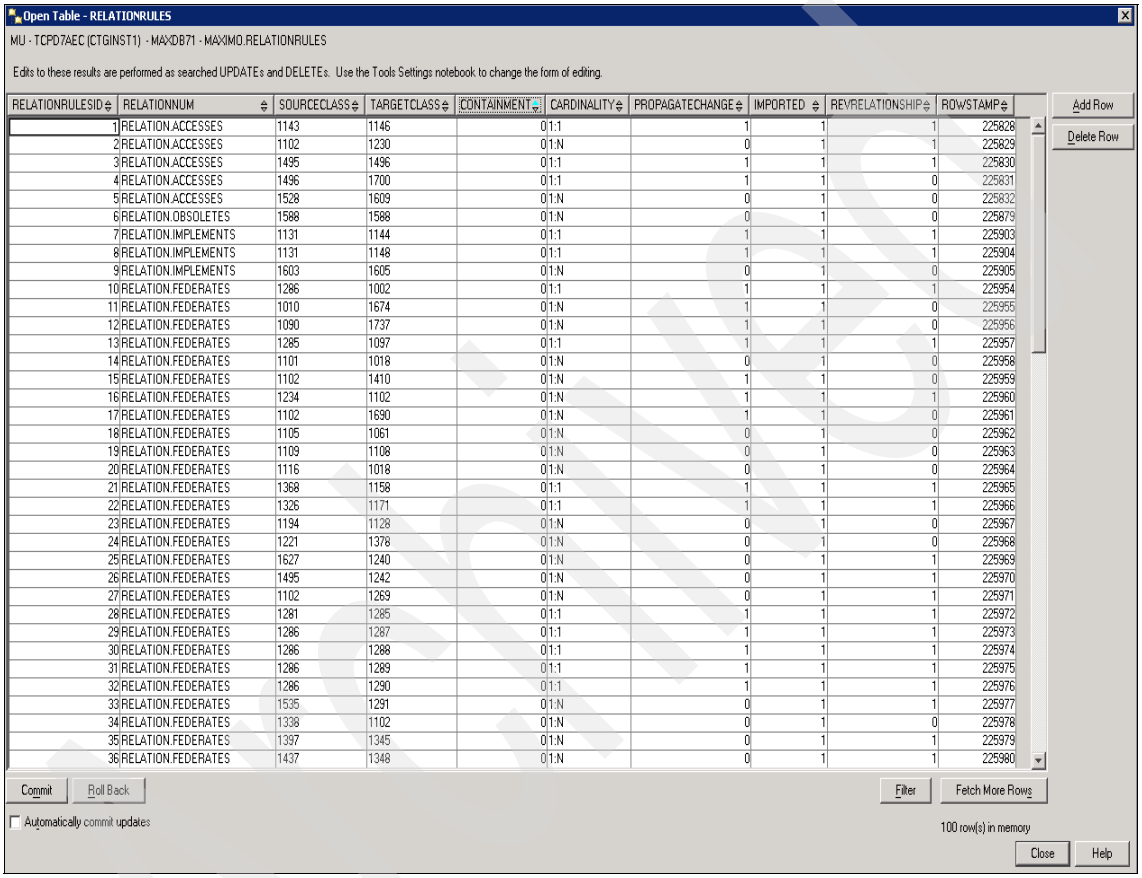


Figure 4-21 Relationship definitions in Relationships application

In the Filter row, you can see that there are 90 predefined definitions.

RELATIONRULES table

The RELATIONRULES table keeps track of which class types are using the various relations being defined in the RELATION table either as a source or target of the relationship type (Figure 4-22). By default, the definitions are populated by the ITIC TADDM CI Type adapter.



RELATIONRULESID	RELATIONNUM	SOURCECLASS	TARGETCLASS	CONTAINMENT	CARDINALITY	PROPAGATECHANGE	IMPORTED	REVRELATIONSHIP	ROWSTAMP
1	RELATION.ACCESSES	1143	1146	0	1:1	1	1	1	225828
2	RELATION.ACCESSES	1102	1230	0	1:N	0	1	1	225829
3	RELATION.ACCESSES	1495	1496	0	1:1	1	1	1	225830
4	RELATION.ACCESSES	1496	1700	0	1:1	1	1	0	225831
5	RELATION.ACCESSES	1528	1609	0	1:N	0	1	0	225832
6	RELATION.OBSOLETES	1588	1588	0	1:N	0	1	0	225879
7	RELATION.IMPLEMENTS	1131	1144	0	1:1	1	1	1	225903
8	RELATION.IMPLEMENTS	1131	1148	0	1:1	1	1	1	225904
9	RELATION.IMPLEMENTS	1603	1605	0	1:N	0	1	0	225905
10	RELATION.FEDERATES	1286	1002	0	1:1	1	1	1	225954
11	RELATION.FEDERATES	1010	1674	0	1:N	1	1	0	225955
12	RELATION.FEDERATES	1090	1737	0	1:N	1	1	0	225956
13	RELATION.FEDERATES	1285	1097	0	1:1	1	1	1	225957
14	RELATION.FEDERATES	1101	1018	0	1:N	0	1	0	225958
15	RELATION.FEDERATES	1102	1410	0	1:N	1	1	0	225959
16	RELATION.FEDERATES	1234	1102	0	1:N	1	1	1	225960
17	RELATION.FEDERATES	1102	1690	0	1:N	1	1	0	225961
18	RELATION.FEDERATES	1105	1061	0	1:N	0	1	0	225962
19	RELATION.FEDERATES	1109	1108	0	1:N	0	1	0	225963
20	RELATION.FEDERATES	1116	1018	0	1:N	0	1	0	225964
21	RELATION.FEDERATES	1368	1158	0	1:1	1	1	1	225965
22	RELATION.FEDERATES	1326	1171	0	1:1	1	1	1	225966
23	RELATION.FEDERATES	1194	1128	0	1:N	0	1	0	225967
24	RELATION.FEDERATES	1221	1378	0	1:N	0	1	0	225968
25	RELATION.FEDERATES	1627	1240	0	1:N	0	1	1	225969
26	RELATION.FEDERATES	1495	1242	0	1:N	0	1	1	225970
27	RELATION.FEDERATES	1102	1269	0	1:N	0	1	1	225971
28	RELATION.FEDERATES	1281	1285	0	1:1	1	1	1	225972
29	RELATION.FEDERATES	1286	1287	0	1:1	1	1	1	225973
30	RELATION.FEDERATES	1286	1288	0	1:1	1	1	1	225974
31	RELATION.FEDERATES	1286	1289	0	1:1	1	1	1	225975
32	RELATION.FEDERATES	1286	1290	0	1:1	1	1	1	225976
33	RELATION.FEDERATES	1535	1291	0	1:N	0	1	1	225977
34	RELATION.FEDERATES	1338	1102	0	1:N	0	1	0	225978
35	RELATION.FEDERATES	1397	1345	0	1:N	0	1	1	225979
36	RELATION.FEDERATES	1437	1348	0	1:N	0	1	1	225980

Figure 4-22 RELATIONSHIPRULES table

There is an entry in the table for each pair of source and target class that make use of the relationship definition. The source and target class types are referenced by their respective CLASSSTRUCTUREID of the CLASSSTRUCTURE table.

Each entry also specifies the cardinality of the relationship. The cardinality column defines if the relationship is one to one, one to many, many to one, or many to many.

The relationship rule definitions appear in the Relationship application as shown in Figure 4-23.

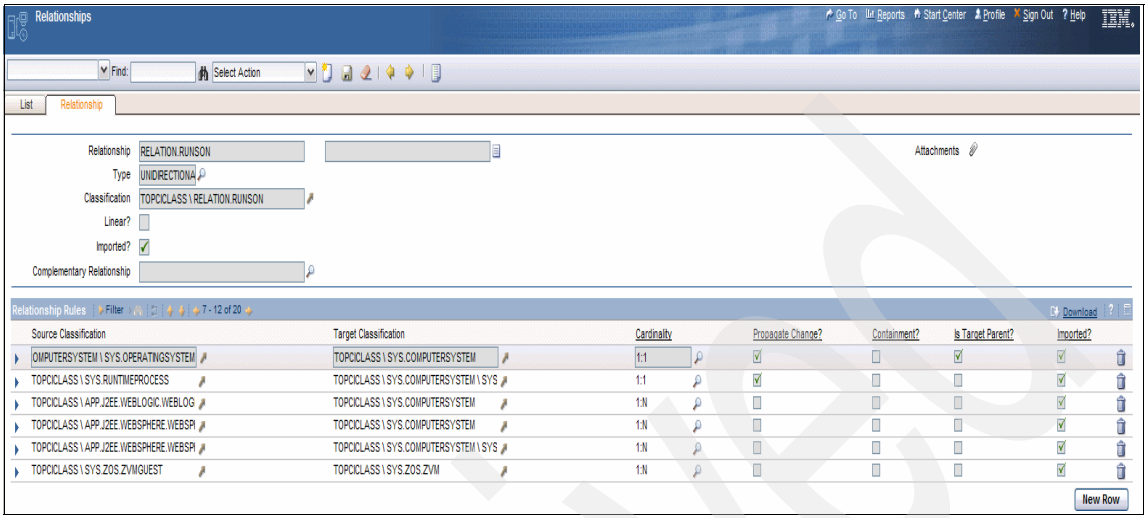


Figure 4-23 Relationship rules in Relationships application

You see that twenty relations between class types use the RUNSON relationship type. One of them is the relation between the SYS.OPERATINGSYSTEM class which runs on the SYS.COMPUTERSYSTEM class with a cardinality of 1:1.

The containment check box indicates if you can view CIs contained within another CI from within the Actual Configuration Items or Configuration Items application based on the specified relationship type.

If you want changes to a CI lower in a classification tree to be reflected in classifications higher in the tree, select the **Propagate Change?** check box. You can edit this field only if the Containment? check box is selected.

Relationships can be complementary in that one relationship implies another. For example, operating system A “installed on” computer B implies that computer B is “installed with” operating system A. “Installed on” and “installed with” are therefore complementary relationships.

We now explain the major tables that keep instance records of CI data. We explain the major tables for the actual as well as the Authorized CI data space representations.

ACTCI table

The ACTCI table keeps a record for each and every CI instance. The table is populated by the ITIC TADDM Actual CI adapter. This second ITIC adapter makes use of the previous populated class and attribute type definition records.

The ACTCI table stores a record for each CI without its related attributes.

See Figure 4-24 for more details.

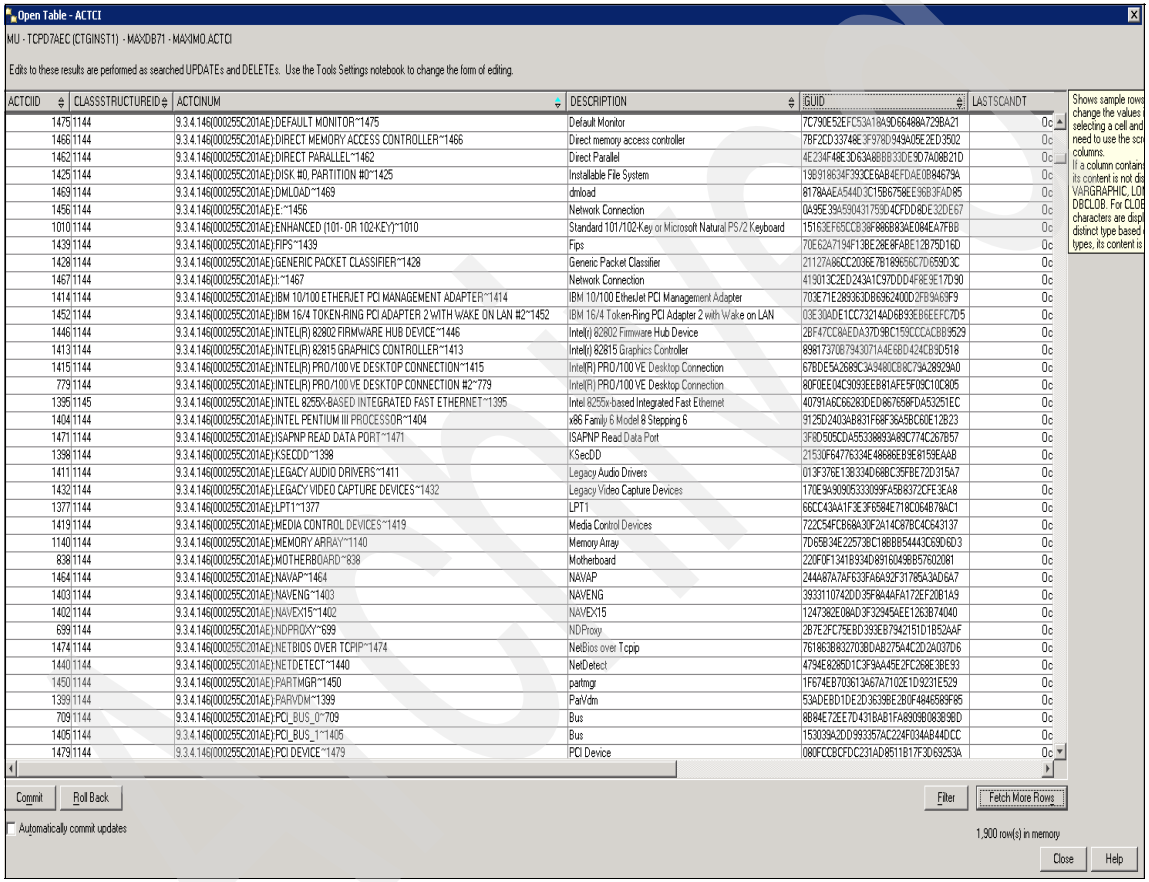


Figure 4-24 ACTCI table

The ACTCIID column stores a unique identifier for the record; the CLASSSTRUCTUREID column relates the attribute to the respective class type while the GUID column is keeping the unique identifier for the CI itself.

Please bear in mind that this table is not keeping track of attributes although the depth level of some of the records in Figure 4-24 on page 98 could lead to the impression that some of the records are attributes. They are not, even though the record with the ACTCIID 1404 9.3.4.146(000255C201AE):INTEL PENTIUM III PROCESSOR~1404 is a CI related to a CI type rather then an attribute definition.

In order to guarantee a uniqueness of the CI in the ACTCINUM column, the ACTCIID of the CI is appended to the name of the CI following a tilde sign.

In order to work with Actual CI data, you have to use the Actual Configuration Items application, as shown in Figure 4-25.

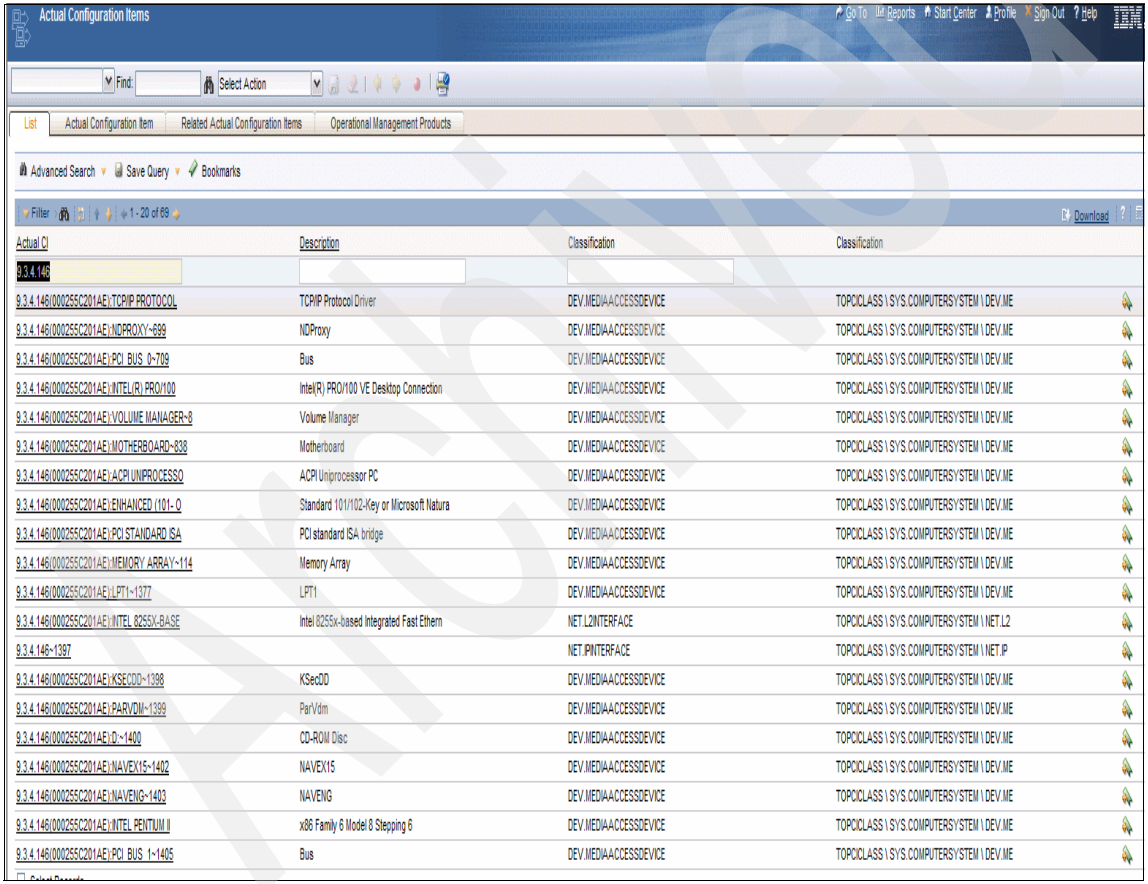


Figure 4-25 List of Actual CIs in actual configuration items application

Figure 4-25 shows a search result on the 9.3.4.146 IP Address, showing the same records as previously shown in Figure 4-24 on page 98.

ACTCISPEC table

The ACTCISPEC table is keeping track of attribute records that are related to the CI records itself (see Figure 4-26). This means that although classes and attributes are related and are presented to the user, they are kept in different tables of the database. The ACTCISPEC table makes use of the attribute definitions being kept in the CLASSSPEC table.

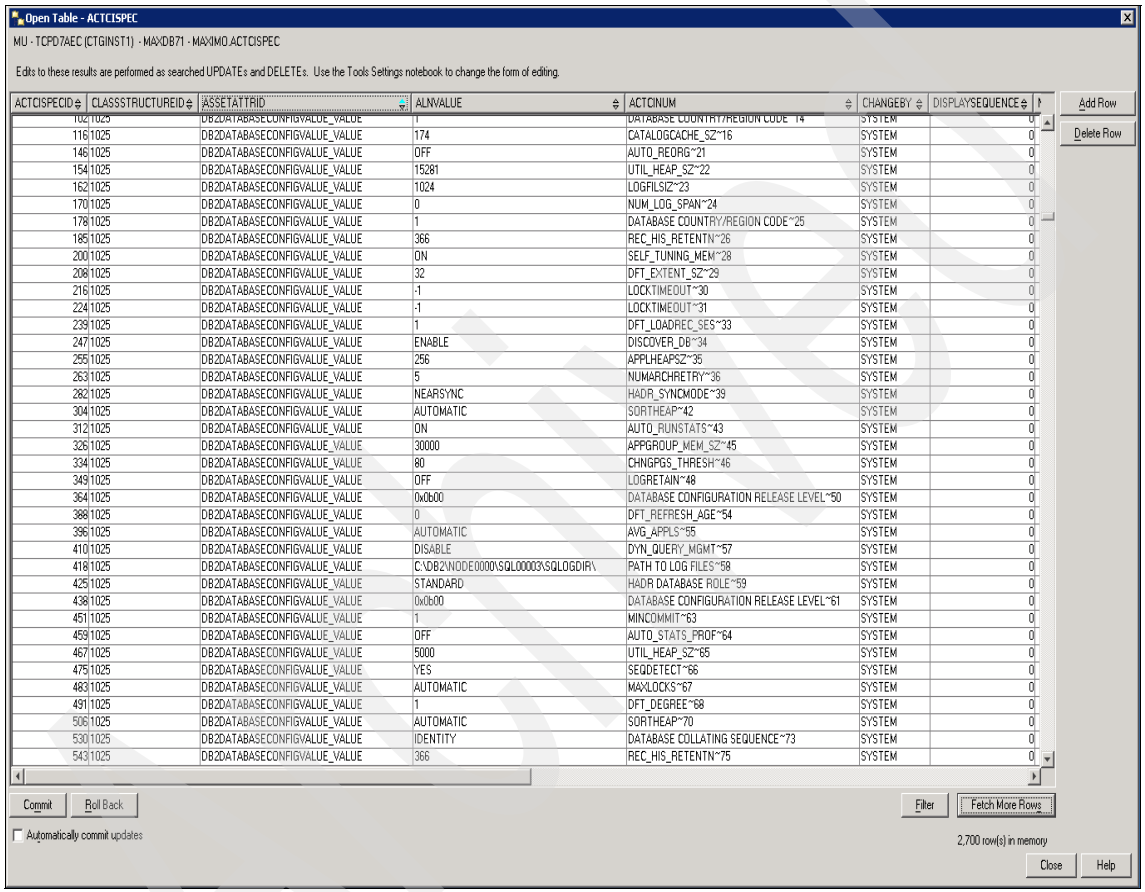


Figure 4-26 ACTCISPEC table

The CLASSSTRUCTUREID column is the key to relating the record to the CI class type, the ASSETATTRID column is keeping a value for the kind of attribute, while the ACTCINUM column is keeping track of which CI instance the attribute is related to. The ALNVALUE column stores the attribute value.

Figure 4-26 shows the data sorted by the ASSETATTRID column. Various records for different CIs are kept holding values for the

DB2DATABSECONFIGVALUES_VALUE attribute. Since attributes brought over by the ITIC adapter from TADDM are not unique across class types, they are named “Classname_Attributename”.

We can see attributes and their values for the instance data in the Actual Configuration Items application, as shown in Figure 4-27.

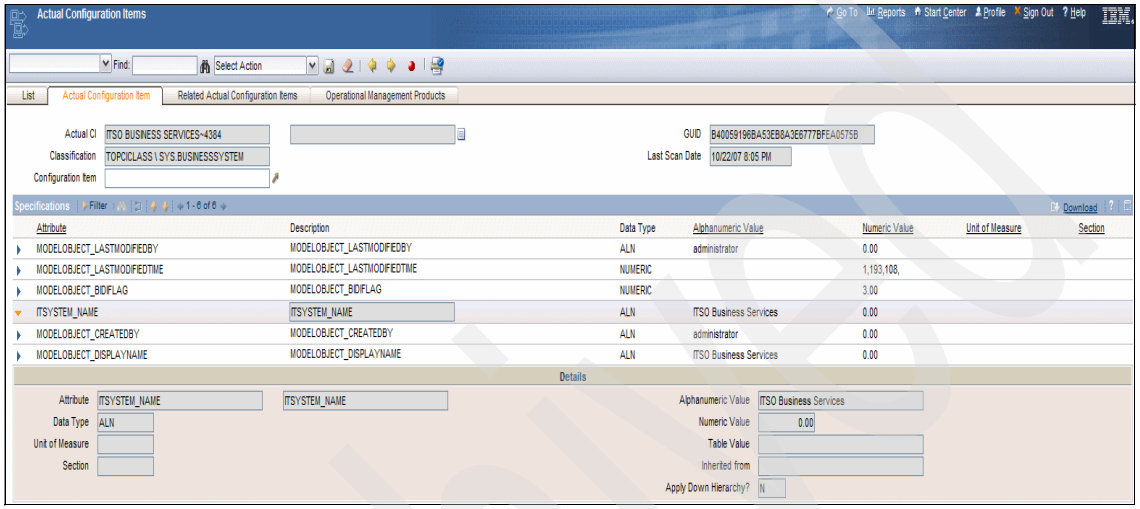


Figure 4-27 Attributes in the actual configuration items application

ACTCIRELATION table

Instance data leverages the relationship definitions in order to relate instances of CIs belonging to different class types to each other. The ACTCIRELATION table keeps track of these relationships, as shown in Figure 4-28.

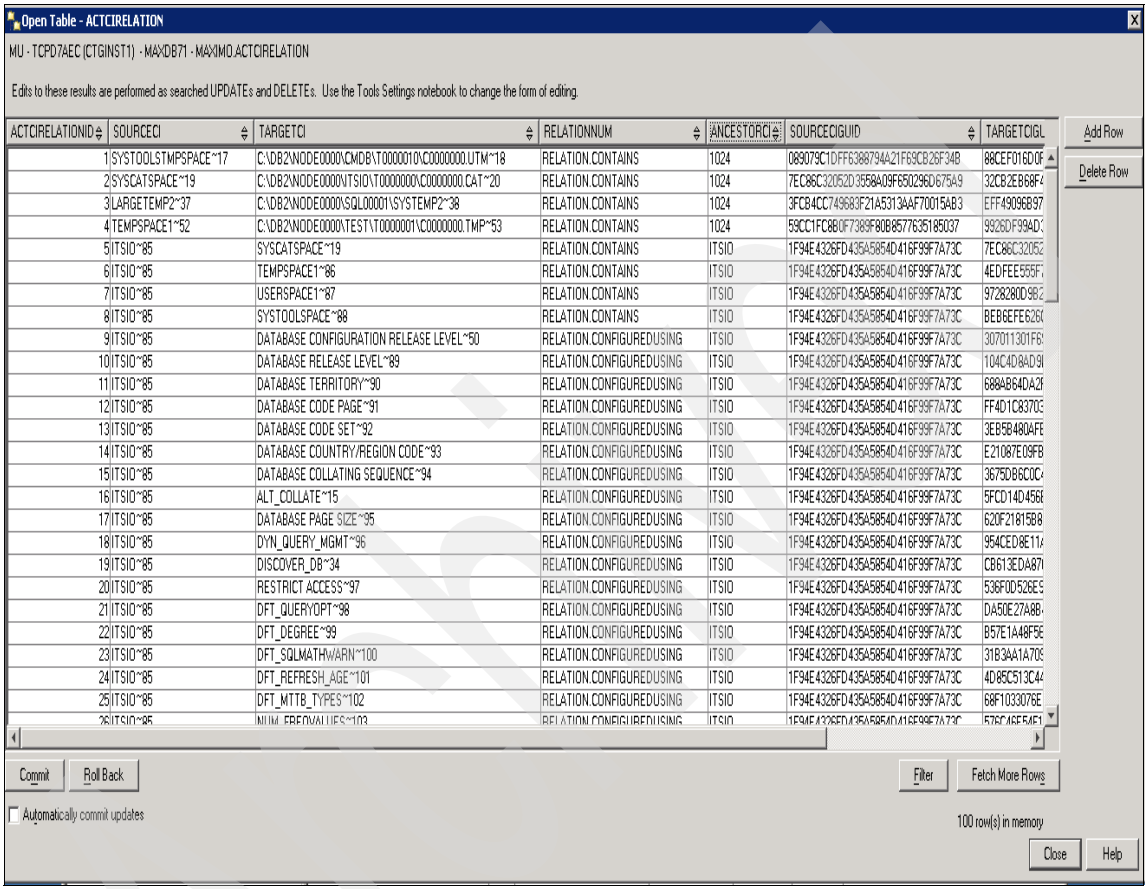


Figure 4-28 ACTCIRELATION table

Source and Target CIs as well as their respective GUIDs are kept in various columns, as shown in Figure 4-28. There is one entry in the table for each relationship type that a CI has with a related CI.

An example of a relationship view inside the Actual Configuration Items applications is shown in Figure 4-29 on page 103.

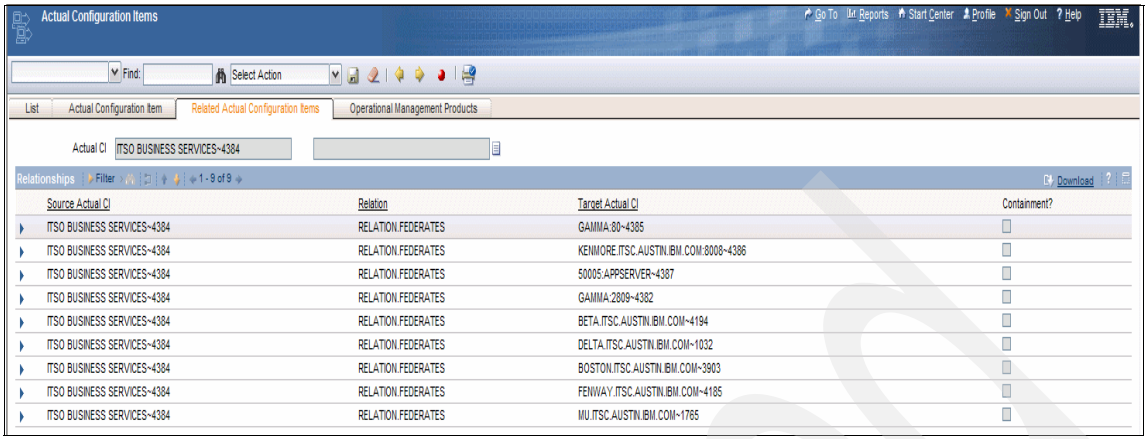


Figure 4-29 Relationship view in the actual configuration items application

You can see that the ITSO Business Services CI has a FEDERATES relationship to various target CIs. The target CIs are members of the defined business service CI. The instance data has been transferred through the ITIC TADDM Actual CI adapter.

CI table

In the same way that tables keep records for the Actual CIs, the Authorized CIs are persisted in equivalent tables. Authorized CI tables are not prefixed with the string “Authorized”; they are simply referred to as CI tables.

The CI table persists all instance records of Authorized CIs that either have been promoted from the Actual CI structures, have been manually entered into the system, or have been synchronized from external systems using the MEA integration technology.

For more details on promotion, please refer to Chapter 5, “CI promotion” on page 113.

Usually the promotion filters out a lot of details from the Actual CI data when transferring to the authorized tables. A decision is required as to which level of detail is required for the process manager products to be operated.

The CLASSSTRUCTUREID column is again used to relate the CI to the appropriate class type. You can also see in the ACTCINUM column that the Authorized CI is related to the Actual CI that it has been promoted from or related to manually (Figure 4-30).

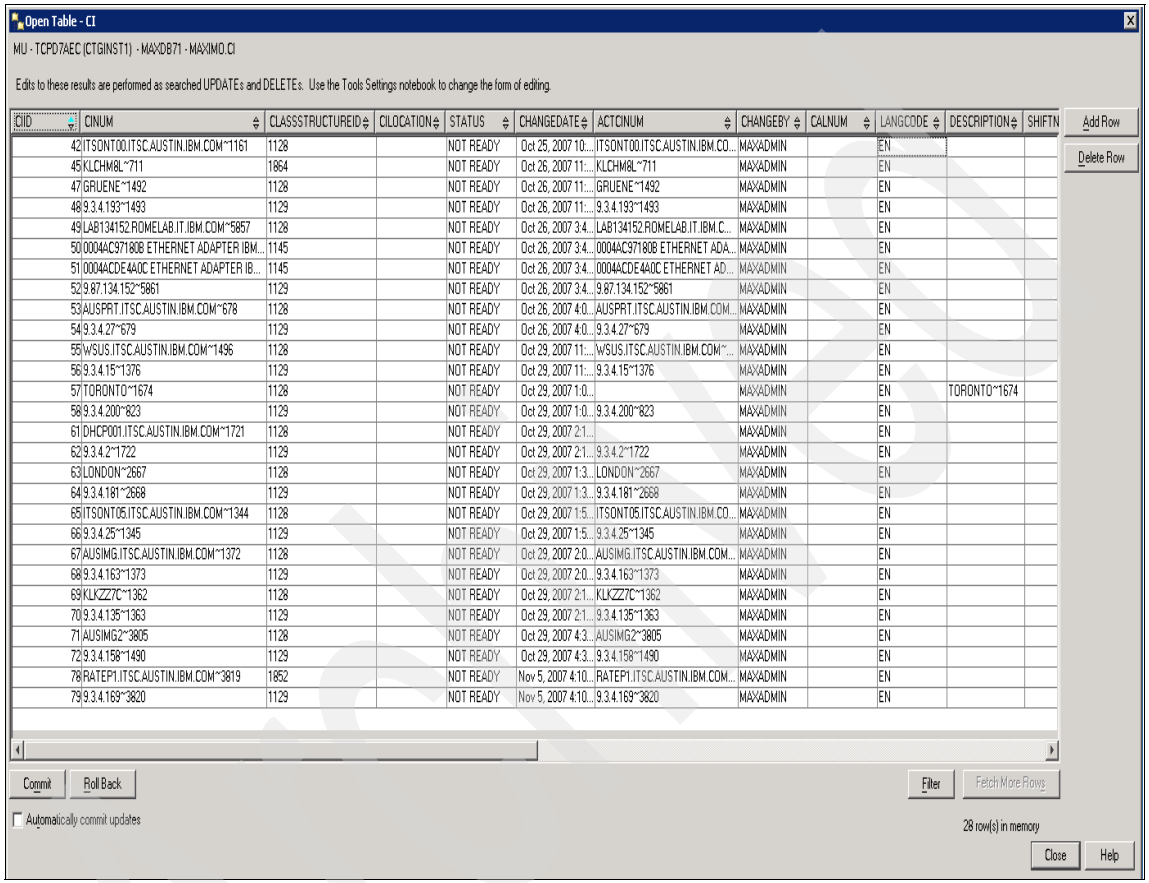
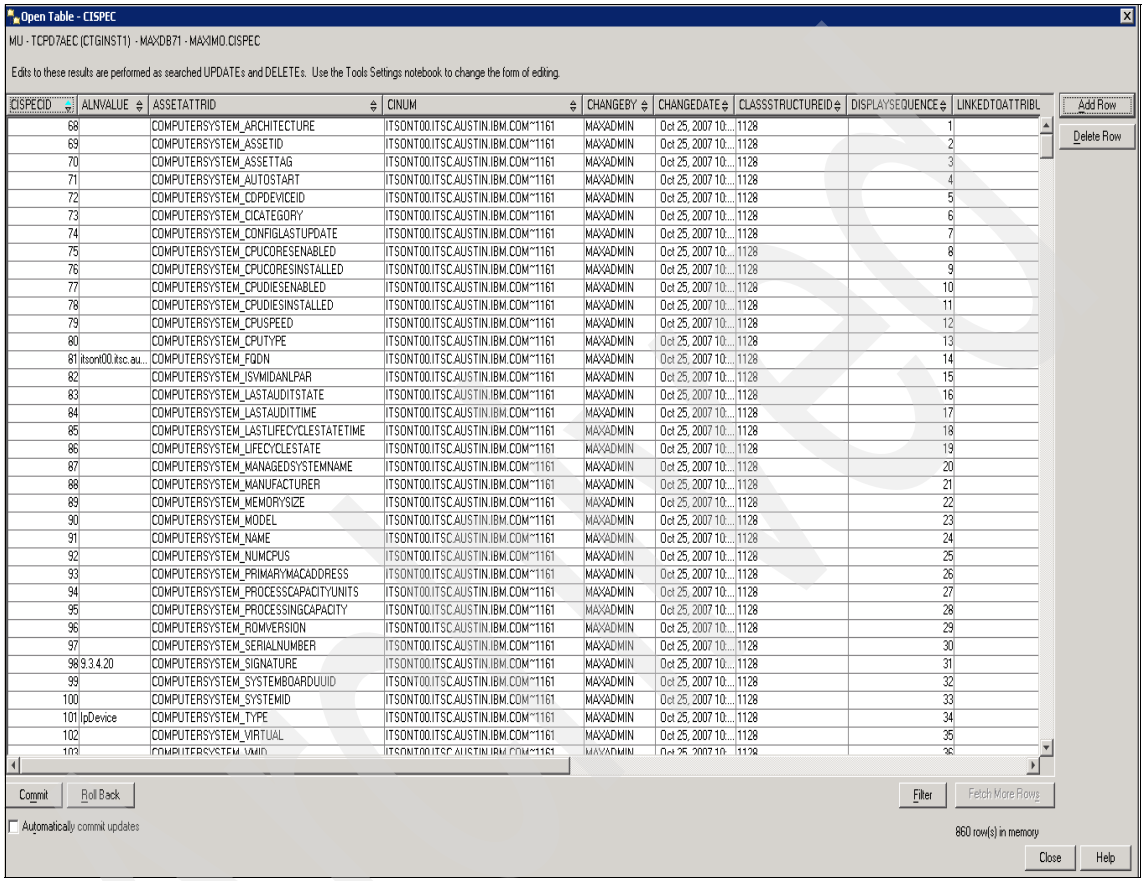


Figure 4-30 CI table

CISPEC table

In the CISPEC table, all attributes of the Authorized CI are kept. Figure 4-31 shows an excerpt of the attributes related to a specific CI identified by the CINUM column.



CISPECID	ALNVALUE	ASSETATTRID	CINUM	CHANGEBY	CHANGEDATE	CLASSSTRUCTUREID	DISPLAYSEQUENCE	LINKEDTOATTRIBL
68		COMPUTERSYSTEM_ARCHITECTURE	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	1	
69		COMPUTERSYSTEM_ASSETID	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	2	
70		COMPUTERSYSTEM_ASSETTAG	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	3	
71		COMPUTERSYSTEM_AUTOSTART	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	4	
72		COMPUTERSYSTEM_CDPDEVICEID	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	5	
73		COMPUTERSYSTEM_CICATEGORY	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	6	
74		COMPUTERSYSTEM_CONFIGLASTUPDATE	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	7	
75		COMPUTERSYSTEM_CPUCORESENABLED	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	8	
76		COMPUTERSYSTEM_CPUCORESINSTALLED	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	9	
77		COMPUTERSYSTEM_CPUDIESENABLED	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	10	
78		COMPUTERSYSTEM_CPUDIESINSTALLED	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	11	
79		COMPUTERSYSTEM_CPUSPEED	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	12	
80		COMPUTERSYSTEM_CPUTYPE	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	13	
81	itsont00.itsc.au...	COMPUTERSYSTEM_FQDN	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	14	
82		COMPUTERSYSTEM_ISVMIDANLPAR	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	15	
83		COMPUTERSYSTEM_LASTAUDITSTATE	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	16	
84		COMPUTERSYSTEM_LASTAUDITTIME	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	17	
85		COMPUTERSYSTEM_LASTLIFECYCLESTATETIME	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	18	
86		COMPUTERSYSTEM_LIFECYCLESTATE	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	19	
87		COMPUTERSYSTEM_MANAGEDSYSTEMNAME	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	20	
88		COMPUTERSYSTEM_MANUFACTURER	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	21	
89		COMPUTERSYSTEM_MEMORYSIZE	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	22	
90		COMPUTERSYSTEM_MODEL	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	23	
91		COMPUTERSYSTEM_NAME	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	24	
92		COMPUTERSYSTEM_NUMCPUS	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	25	
93		COMPUTERSYSTEM_PRIMARYMACADDRESS	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	26	
94		COMPUTERSYSTEM_PROCESSCAPACITYUNITS	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	27	
95		COMPUTERSYSTEM_PROCESSINGCAPACITY	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	28	
96		COMPUTERSYSTEM_ROMVERSION	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	29	
97		COMPUTERSYSTEM_SERIALNUMBER	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	30	
98	9.3.4.20	COMPUTERSYSTEM_SIGNATURE	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	31	
99		COMPUTERSYSTEM_SYSTEMBOARDUUID	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	32	
100		COMPUTERSYSTEM_SYSTEMID	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	33	
101	IpDevice	COMPUTERSYSTEM_TYPE	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	34	
102		COMPUTERSYSTEM_VIRTUAL	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	35	
103		COMPUTERSYSTEM_VMID	ITSONT00.ITSC.AUSTIN.IBM.COM~1161	MAXADMIN	Oct 25, 2007 10:...	1128	36	

Figure 4-31 CISPEC table

In the Configuration Items application, attributes, including their appropriate values, are as shown in Figure 4-32.

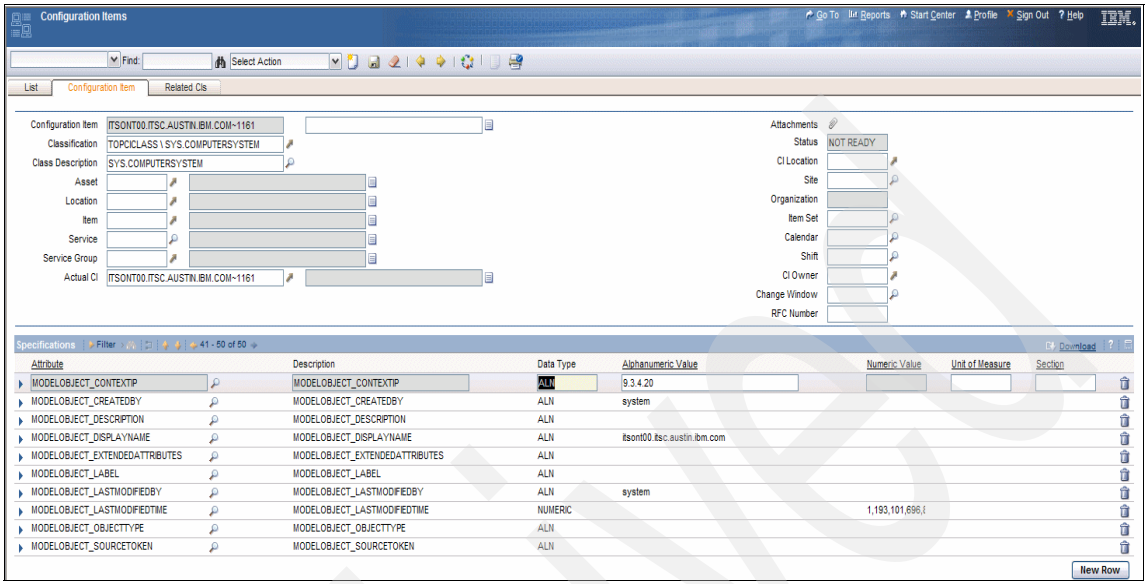


Figure 4-32 Attributes in configuration items application

Please note that not all attributes necessarily have to have values. If the schema has more attributes defined than attribute values have been discovered and promoted, no values are shown. You can enter attribute values manually in the Configuration Items application.

COLLECTION table

Collections are a way to group Authorized CIs. They can also be used within the Security Group application in order to provide access to a collection rather then single CI instances.

The COLLECTION table keeps records of defined collections, as shown in Figure 4-33 on page 107.

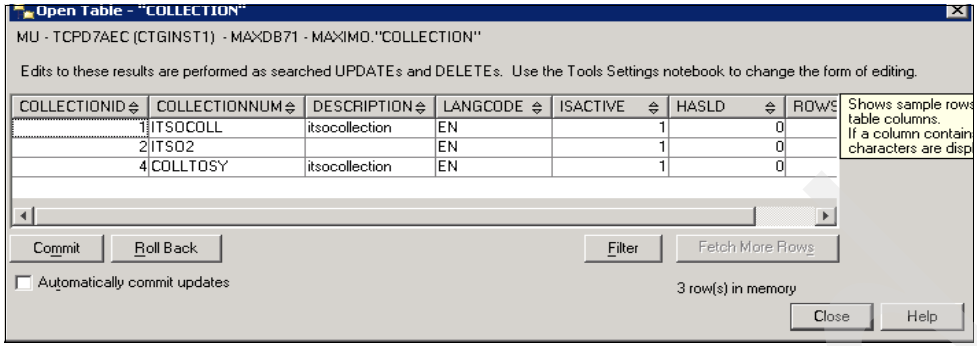


Figure 4-33 COLLECTION table

There are three collections defined according to Figure 4-33.

COLLECTDETAILS table

The COLLECTDETAILS table keeps track of the members of each of the defined collections, as shown in Figure 4-34.

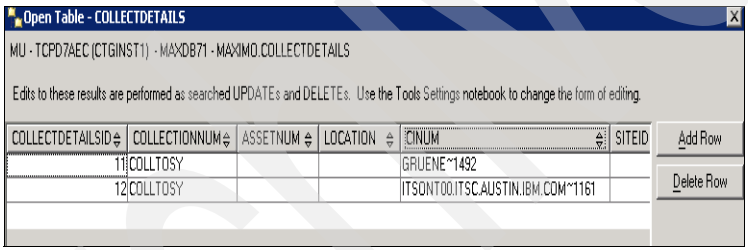


Figure 4-34 COLLECTDETAILS table

Only collections that have members defined have a record in the table. You can see that the *COLLTOSY* collection has two members defined. You manage collections in the Collections application.

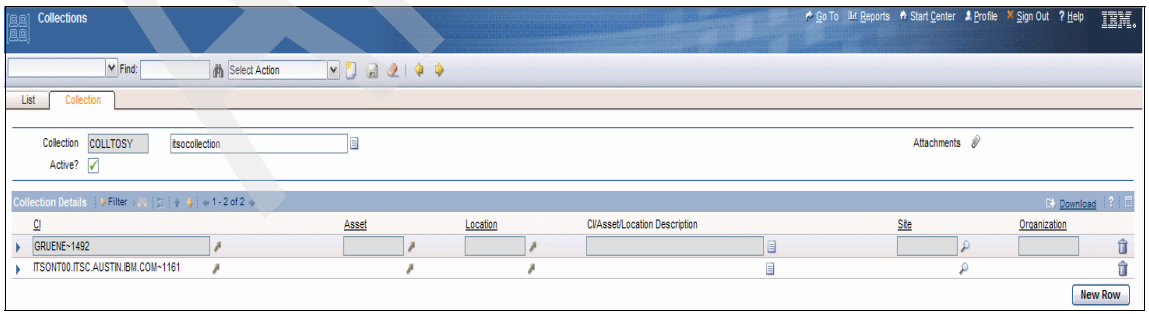


Figure 4-35 Collection member in collections application

4.2 Extending the model

In 4.1, “Implementation of Actual and Authorized CI spaces” on page 73, we explain the major tables of the database that are used to keep track of schema and instance data. In this section, we provide examples of adding a new class type as well as adding an attribute in the process layer database. You can extend the model on the TADDM side of the CCMDB solution as well and then use the ITIC TADDM adapters to synchronize the schema, but in this case we show how you can extend the schema from within the Classifications application.

4.2.1 Adding a new class type

In the Classifications application, we add a new class type using the classification path ITSO.CHILD.OF.COMPUTERSYSTEM. We add the new class type as a subclass of the existing SYS.COMPUTERSYSTEM class. In the Classification field, we add a name and have to select a parent classification by using the arrow symbol right next to the Parent Classification field.

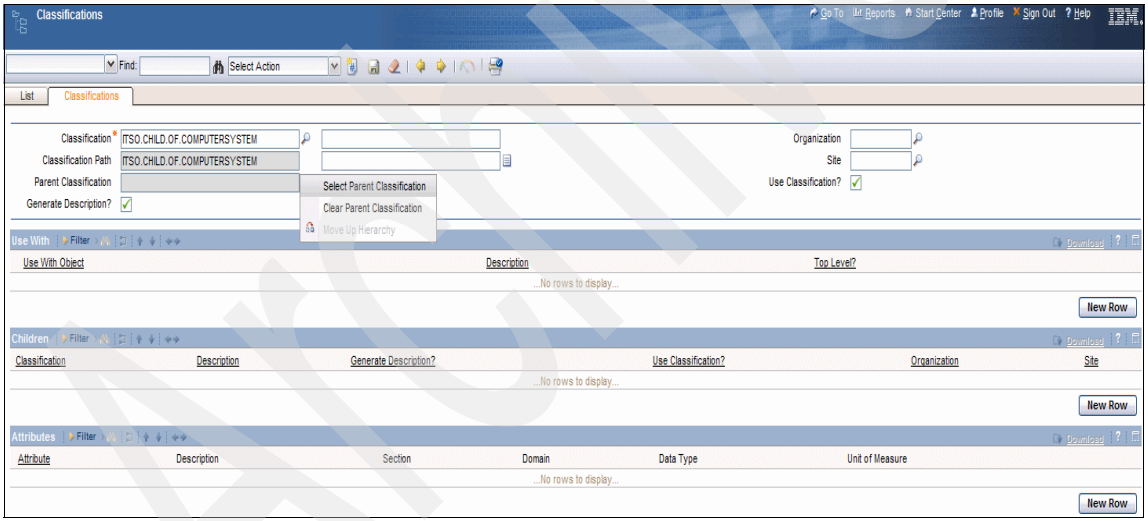


Figure 4-36 Extend class model in classifications application

A selection menu is shown in order to specify the classification of the parent classification class type (see Figure 4-37 on page 109).

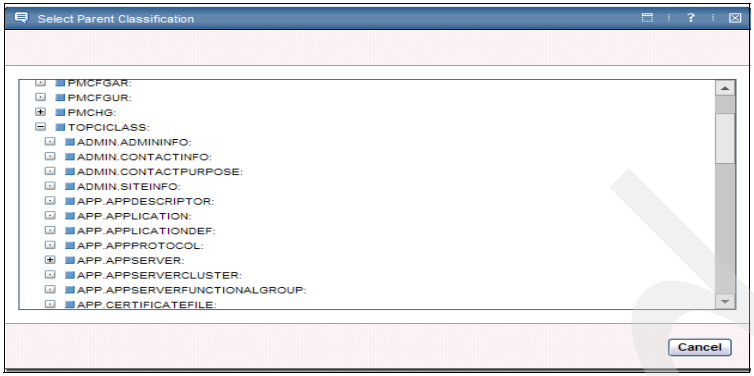


Figure 4-37 Select Parent Classification menu

In this case, we decide to use a classification path for the Authorized CI structure. After we selected the parent classification, the value of the classification path field changes in order to reflect the hierarchy of the Authorized CI classification structure (see Figure 4-38).

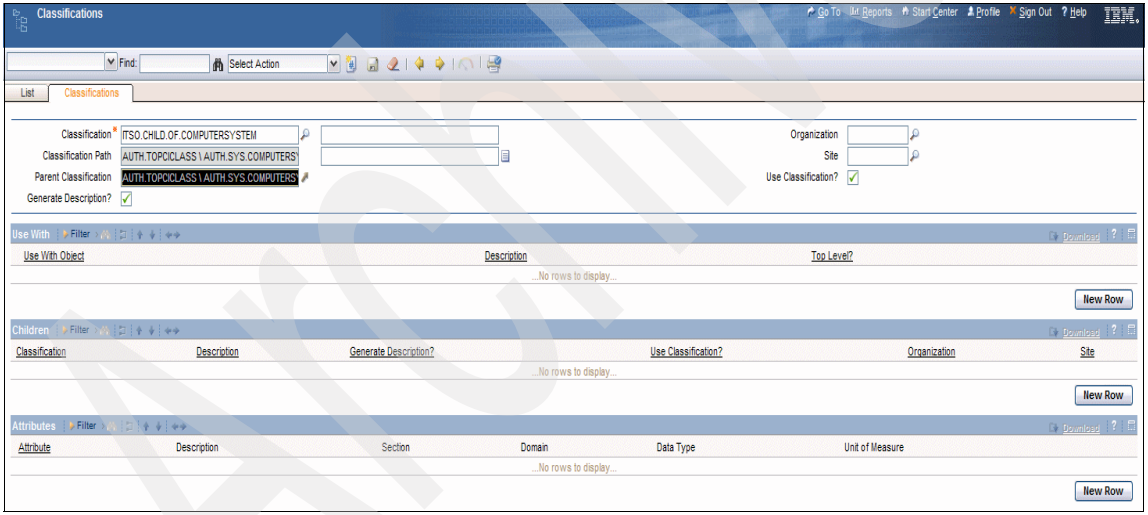


Figure 4-38 Parent Classification selected

You can see that the parent classification is different from the Actual CI classification structure in this case. We begin our classification structure for the Authorized CIs with the prefix “AUTH”. The top-level structure is AUTH.TOPCICLASS.

Saving the record automatically adds the CI object to the *Use With* list of the Classifications applications. This makes the new class type available to the CI object and all applications using the CI object, for example, the Configuration Items application.

When listing the definition of the AUTH.SYS.COMPUTERSYSTEM classification in the Classifications application, we can see the new class ITSO.CHILD.OF.COMPUTERSYSTEM listed in the Children section.

The new class has been successfully added as a child class to an existing class of the hierarchy.

4.2.2 Adding a new attribute

You would add new attributes to the Actual CI data space, for example, if you want to add some information to a CI that is not discoverable and for which no pre-existing attribute does exist. One use case would be to add an attribute for some information that you want to federate from a remote data source. Please refer to Chapter 6, “Implementing federation” on page 139 for more details and an example of how to set up federation.

In the Classifications application Attributes section, you have to click the **New Row** button in order to add the specifications of a new attribute, as shown in Figure 4-39 on page 111.

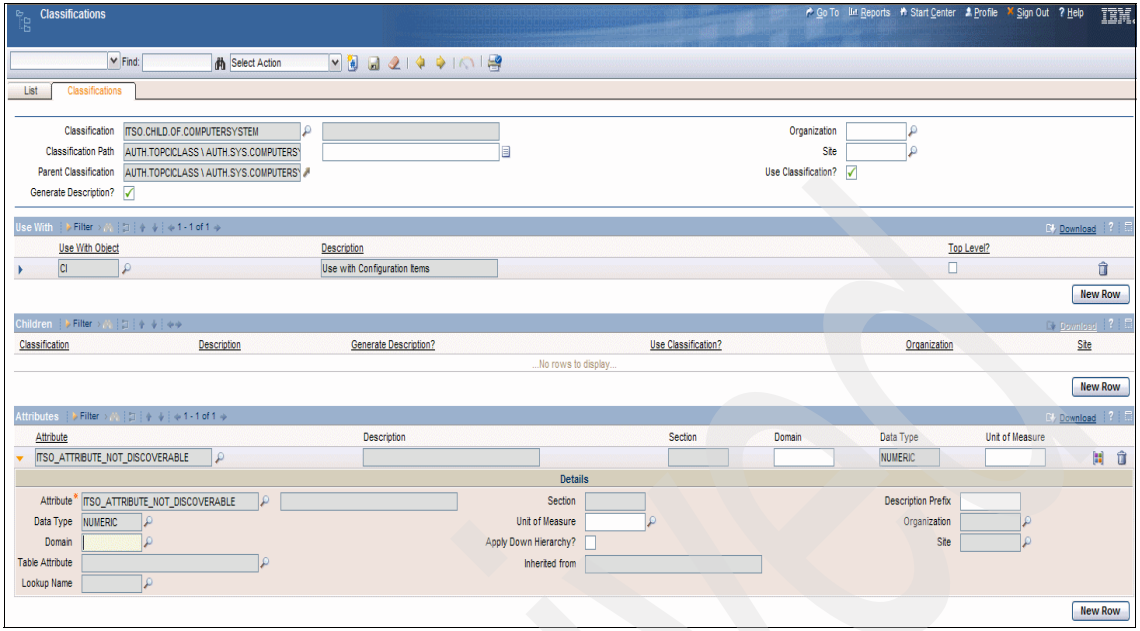


Figure 4-39 Adding an attribute in the classifications application

We add an attribute to our new CI class structure ITSO.CHILD.OF.COMPUTERSYSTEM named ITSO_ATTRIBUTE_NOT_DISCOVERABLE and give it a data type of NUMERIC.

In case you want to propagate the attribute to child classes of this class, you have to check the **Apply Down Hierarchy?** check box in the attribute definition section. This is the reason why no other attributes from ancestor classes are inherited to the ITSO.CHILD.OF.COMPUTERSYSTEM class. The attributes of the ancestor classes do not have the Apply Down Hierarchy? check box checked; you would add a new relationship type, its cardinality, as well as the specification of which class types make use of this relationship type in the Relationships application.

4.3 Other data related topics

For more details related to the data model in CCDM and its customization and usage, see the following:

- Promotion of Actual CIs to Authorized CIs: Chapter 5, “CI promotion” on page 113
- Federation of CCMDB data: Chapter 6, “Implementing federation” on page 139
- CCMDB Data Layer Concepts: *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565
- ITIC Customization: *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565

5

CI promotion

This chapter talks about the process called promotion and how this process can help you create Authorized CIs in the CCMDB database.

We have already discussed in previous chapters about the processes related to discovering CIs using the TADDM application and the creation of Actual CIs using IBM Tivoli Integration Composer (ITIC).

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.

Refer to Figure 5-1 to understand the different states of CIs in CCMDB.

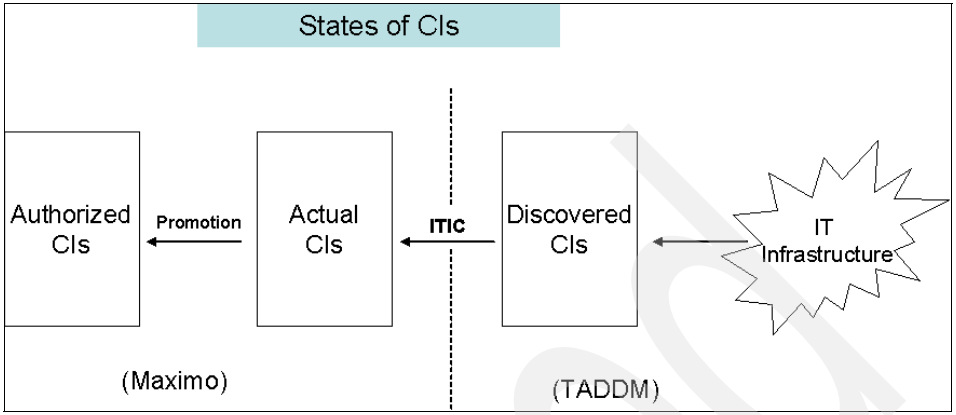


Figure 5-1 Configuration item states

The Discovered Configuration items and Actual CIs in your environment may have many attributes and relationships. In some cases you might not want to manage and control every attribute and relation for every CI that appears in the Actual space and would also like 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 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. Once 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.

Note: 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.

5.1 Promoting Actual CIs to Authorized CIs through using Authorized CIs hierarchies

Actual CIs are created in a hierarchy that follows the common data model. You might chose to import only some types of CIs. You also might chose 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. You build this hierarchy in the Classifications application. (For instructions, open the Classifications application and read the help topic titled “Manage CI hierarchies.” Note that you must start with the top level CI in your hierarchy, and that you must specify that the “Use With Object” is CI.)

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 hierarchy for a particular type of CI. For example, for Computer System, you might 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 hierarchy. 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 hierarchy. Since 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.

Refer to the flow chart shown in Figure 5-2 to promote the Actual CIs using authorized hierarchies.

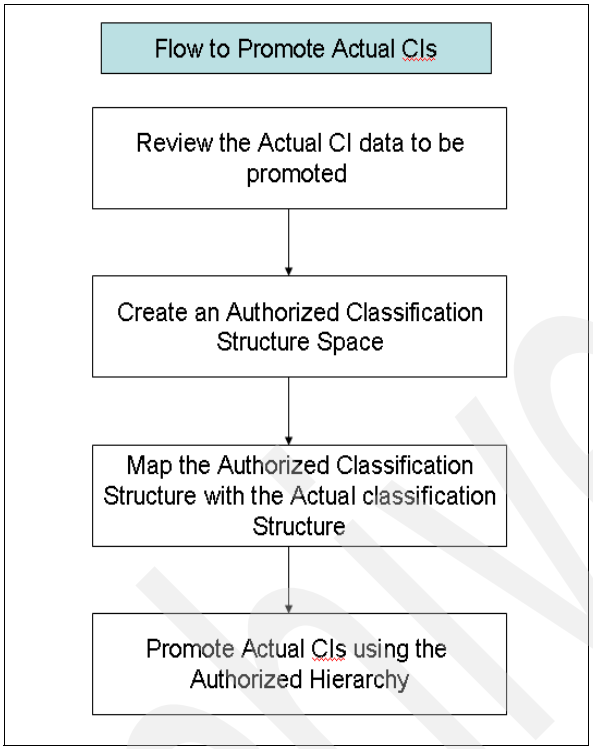


Figure 5-2 Flow to promote Actual CIs to Authorized CIs

5.1.1 Step by step procedure to promote CIs

The following steps are taken to promote an Actual CI to an Authorized CI.

Step 1: Review the Actual CI data classifications to be promoted

- At this point, you should 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 would hold Authorized CIs. We advise starting the investigation at the top-level classification and working downward.
- In Figure 5-3 on page 117, we have selected a Top-Level class (SYS.COMPUTERSYSTEM). Notice that this class has 14 children in the

actual space. For example, we have chosen two children classifications that we would bring over to our authorized space.

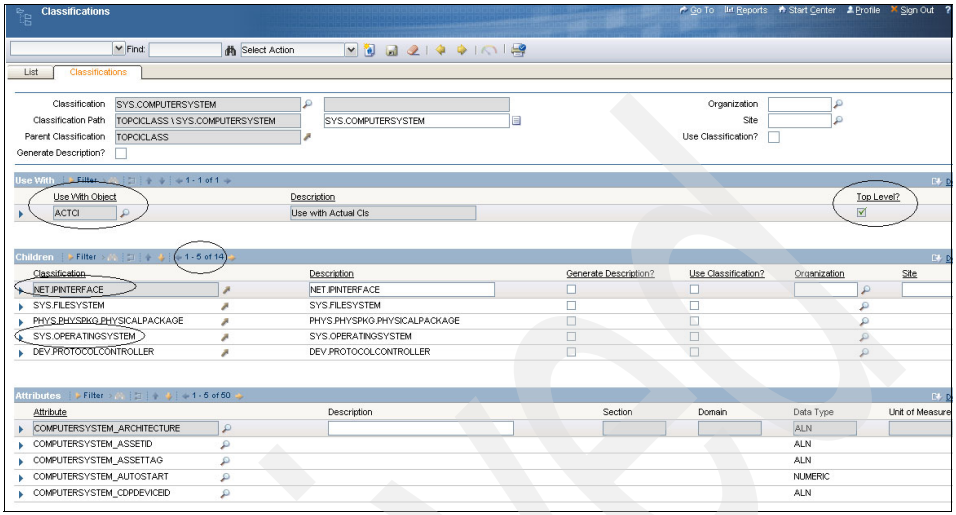


Figure 5-3 Choosing classifications for authorized space

- The next step is to look at the child classes. NET.IPINTERFACE does not have any children, so this leg of the hierarchy is complete.

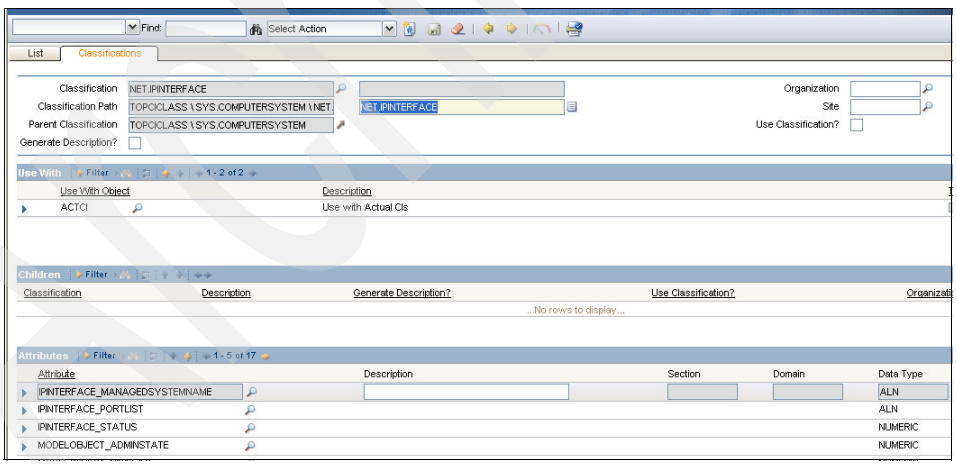


Figure 5-4 Viewing child classes

- We look at another child class of SYS.COMPUTERSYSTEM, SYS.OPERATINGSYSTEM (Figure 5-5). This class has three children. Let us say that we want to manage only SYS.SOFTWARECOMPONENT in our authorized space.

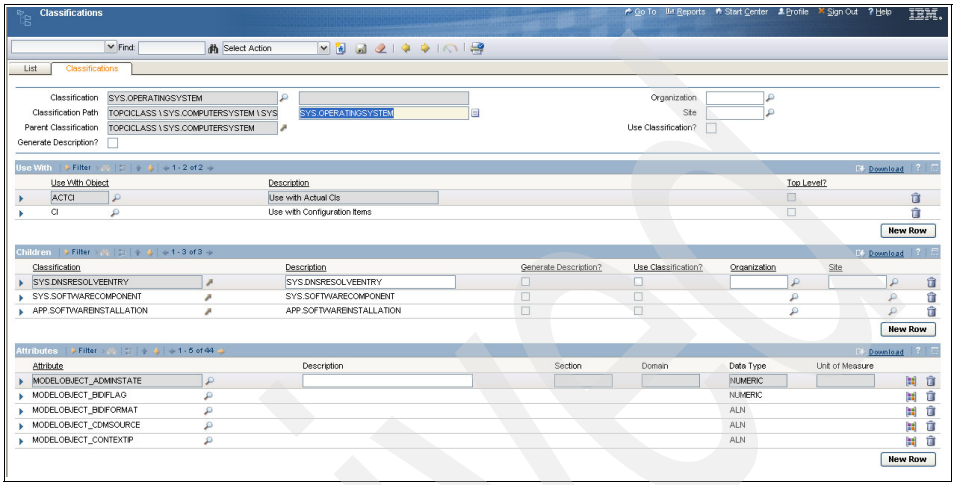


Figure 5-5 SYS.OPERATINGSYSTEM class

- 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.

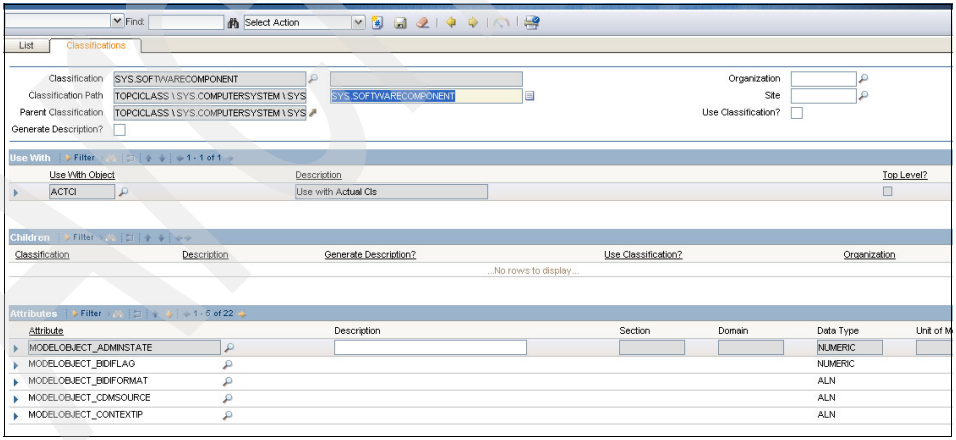


Figure 5-6 SYS.SOFTWARECOMPONENT class

Note: In real scenarios, you should gather all the information related to classifications according to your need. A real authorized structure may look different from the above examples.

Figure 5-7 shows the structure of our final authorized space.

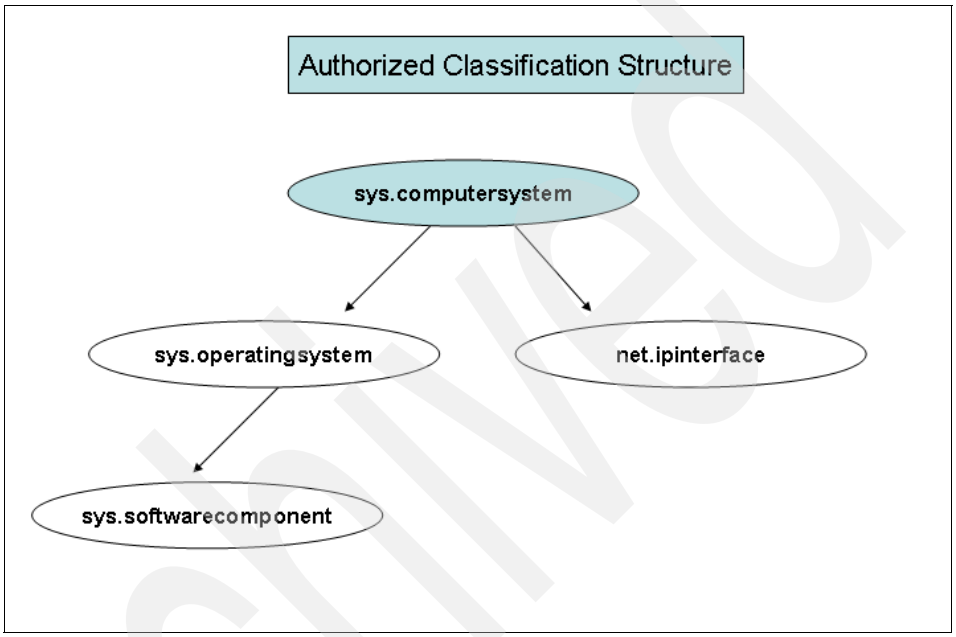


Figure 5-7 Sample authorized classification structure

This structure may look different in real scenarios, depending upon your need. In Figure 5-7, we have only one Top-Level class (sys.computersystem), but you may 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. This should be created manually. Similarly, you need to create other children authorized classes and you should also think of a naming convention to these authorized classifications. There is no restriction on the naming convention.

- Actual: TOPCICLASS \ SYS.COMPUTERSYSTEM \ SYS.OPERATINGSYSTEM
- Authorized: AUTH.TOPCICLASS \ SYS.COMPUTERSYSTEM \ SYS.OPERATINGSYSTEM

To create a classification, select **Administration** → **Classification**.

Note: 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

Once 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 would get promoted automatically provided they belong to the Authorized Classification Structure. Anything outside the authorized space will not get promoted.

- 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 5-8 on page 121. In order to map the authorized classification to the actual one, you have to select the **Manage CI Hierarchies** from the Action menu.

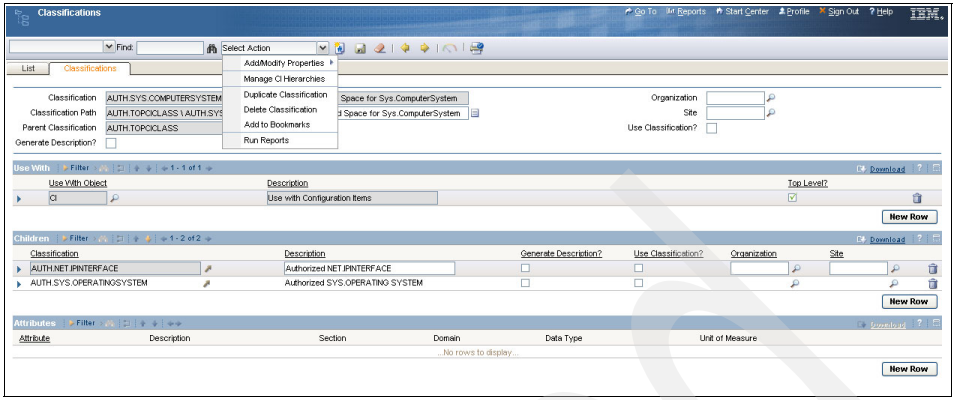


Figure 5-8 Classifications

- After selecting **Manage CI Hierarchies**, you will get the window shown in Figure 5-9.

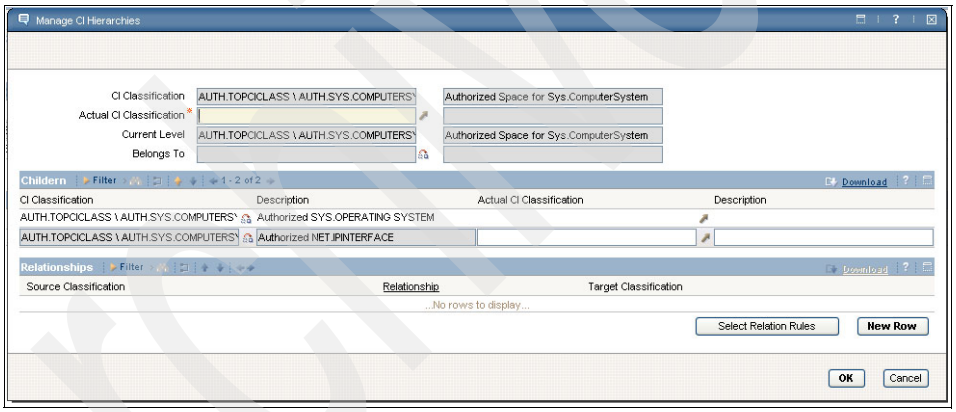


Figure 5-9 Manage classifications

- Here you can map the authorized classification to the respective actual classification. First, you should map the top-level classification and then the children classifications. In our example, the top-level authorized classification AUTH.TOPCICLASS \ AUTH.SYS.COMPUTERSYSTEM should be mapped to TOPCICLASS.SYS.COMPUTERSYSTEM (Actual CI classification).

  The children classification should also be mapped to their respective actual classes.

Refer to Figure 5-10 for more details.

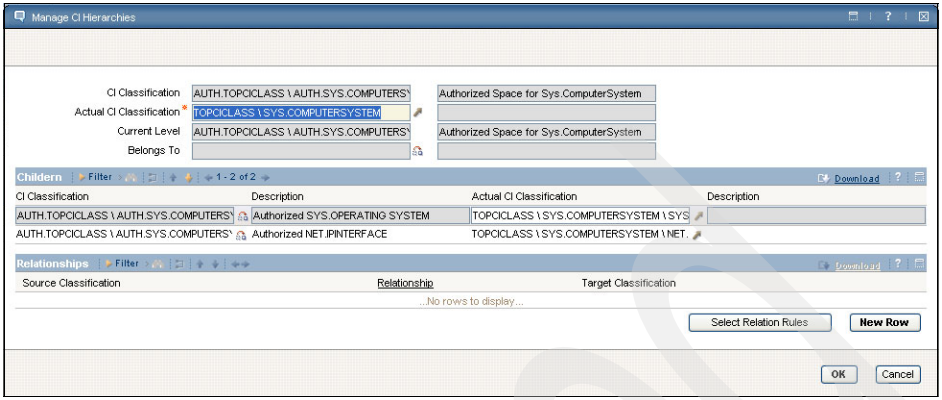


Figure 5-10 Manage CI hierarchies

- After mapping the Top-Level Classification and child classes, you should 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 would automatically establish the relations based on the relations between actual classifications. But you can also modify/create your own relationships that you would like to manage in your authorized space. To create new relations, click the **New Row** button under the Relationships section. Figure 5-11 shows the relationships that would be managed in an authorized space.

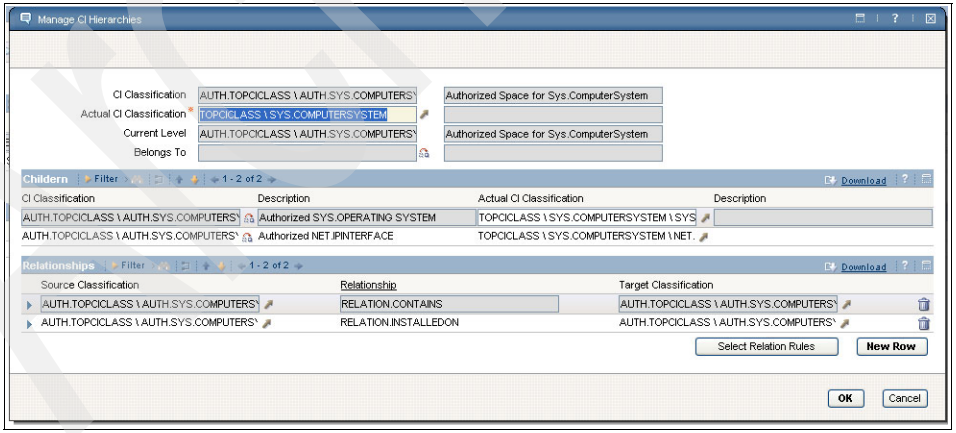


Figure 5-11 Relationships managed in an authorized space

- Repeat the above 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

Note: 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 classification.

In the above example, our top-level actual class is SYS.COMPUTERSYSTEM. We can promote any CI defined with this actual class. 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 5-12.

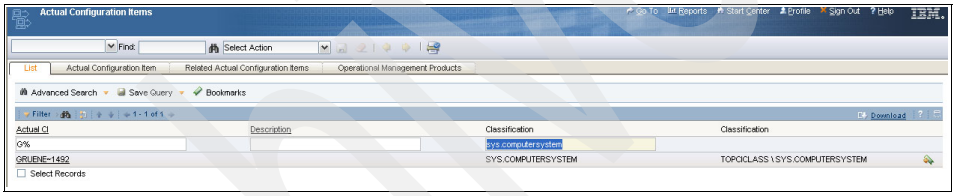


Figure 5-12 Actual CI

Click the **Actual Configuration Item** tab. This view shows the details about the respective Actual CI, as shown in Figure 5-13.

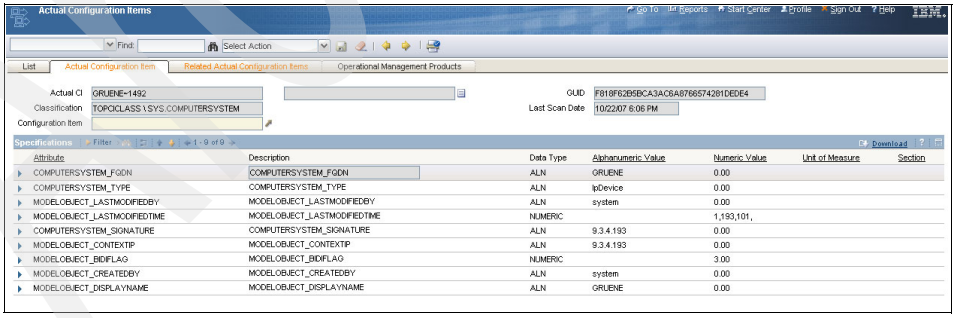


Figure 5-13 Details of Actual CI

Select **Select Action** → **Create Authorized Configuration Item** to promote the CI, as shown in Figure 5-14.

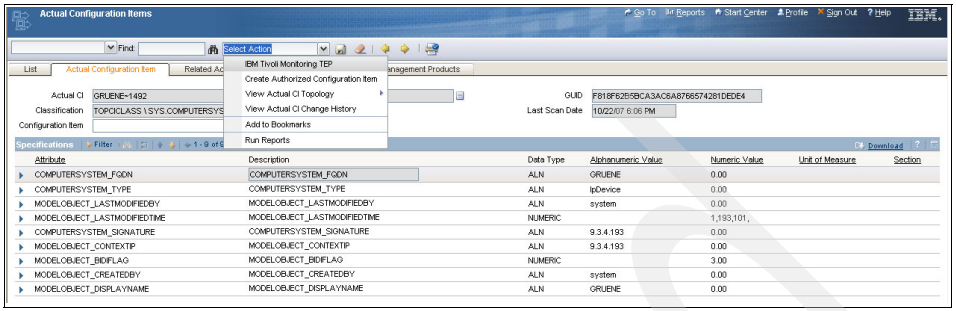


Figure 5-14 Selecting action to promote

This would show a dialog box, as shown in Figure 5-15. In this case, as you are promoting CIs with the authorized classification, the CI class should be the same as the defined authorized structure.

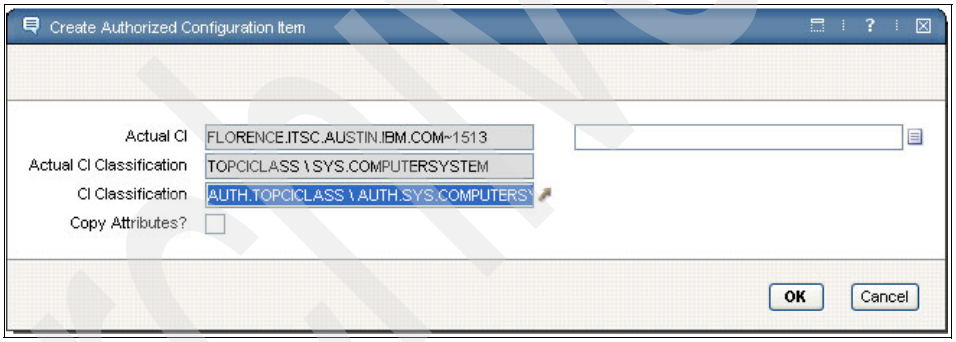


Figure 5-15 Promotion details

Note: If you want to promote an Actual CI with all its attributes values, then check the **Copy Attributes** check box. This would bring 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.

Click **OK** to start the promotion. This may take some time.

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 5-16 shows the Promoted CI in the Configuration Items application.

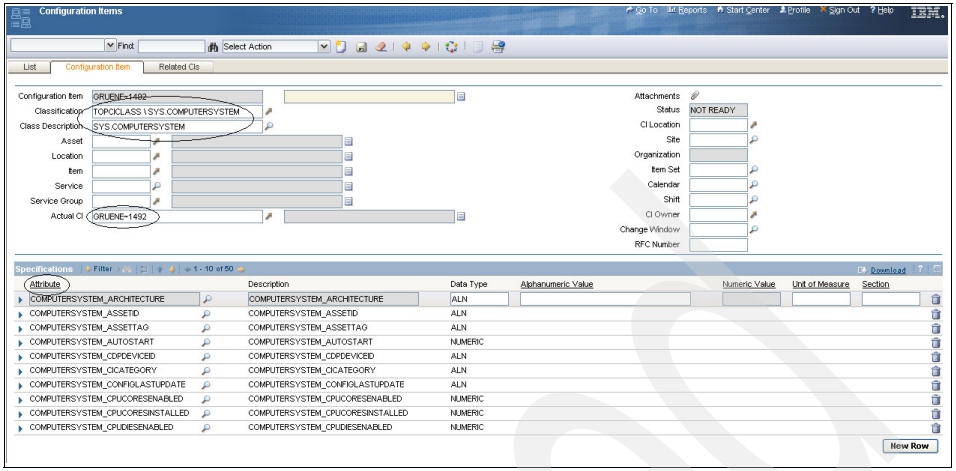


Figure 5-16 Viewing the promoted CI

You can review the desired results with Classification details, Associated Actual CI, 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 5-17.

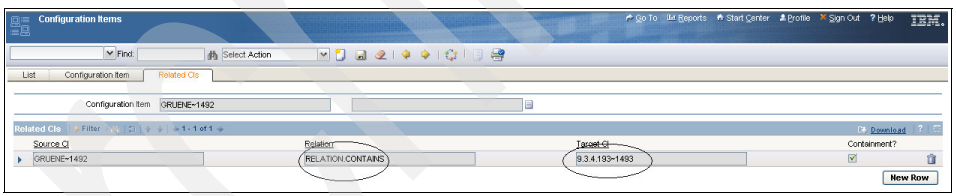


Figure 5-17 Viewing Related CIs

Promoting more than one CI at time

Note: 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, 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 5-18.

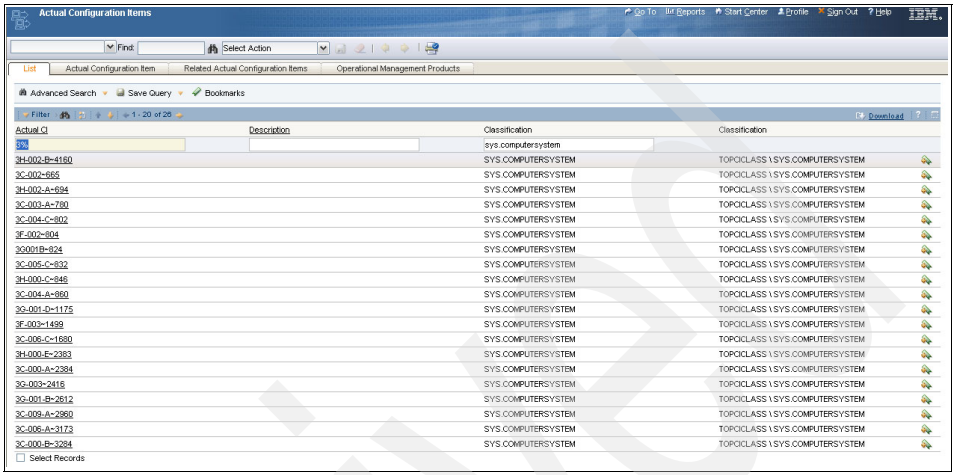


Figure 5-18 Results of CI query

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. Refer to the window shown in Figure 5-19.

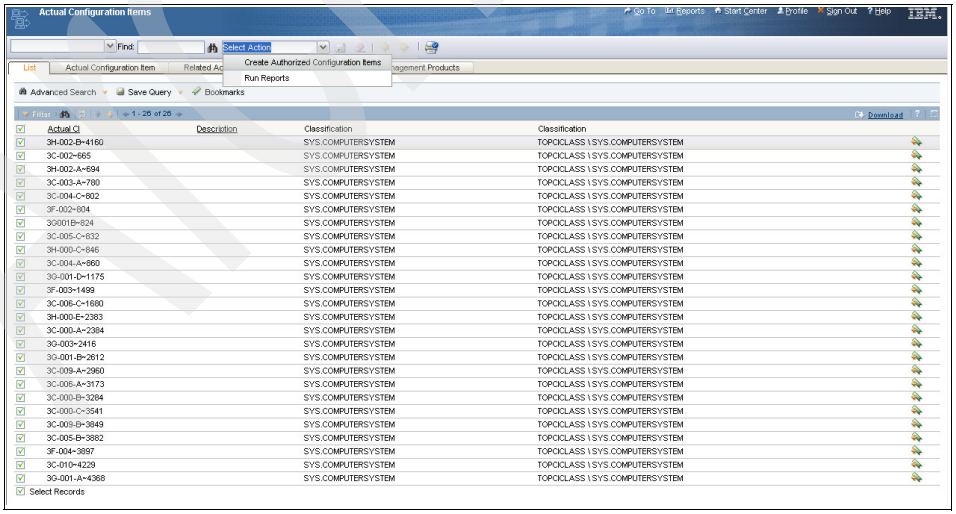


Figure 5-19 Create Authorized CIs menu

After selecting the **Authorized Configuration Items** option, you see dialog box shown in Figure 5-20. Specify the Authorized CI Classification name. The Copy Attribute option is optional, as explained previously.

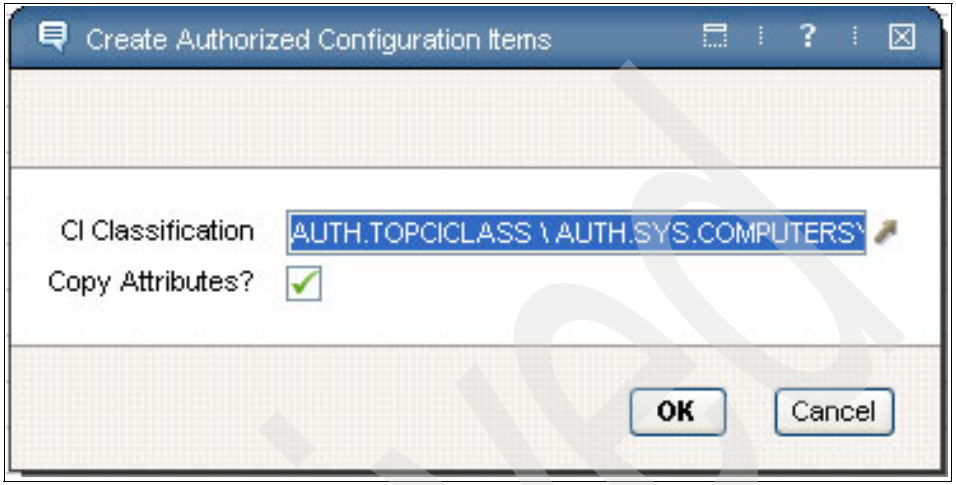


Figure 5-20 Create Authorized CI dialog

Click **OK** to start the promotion process. This may take some time.

The results can be reviewed in the Configuration Items application, as explained previously.

5.2 Promoting Actual CIs to Authorized CIs without authorized hierarchies

This section describes how you can promote CIs if you are not using the Authorized CI hierarchy method.

Figure 5-21 shows the flow to promote Actual CIs.

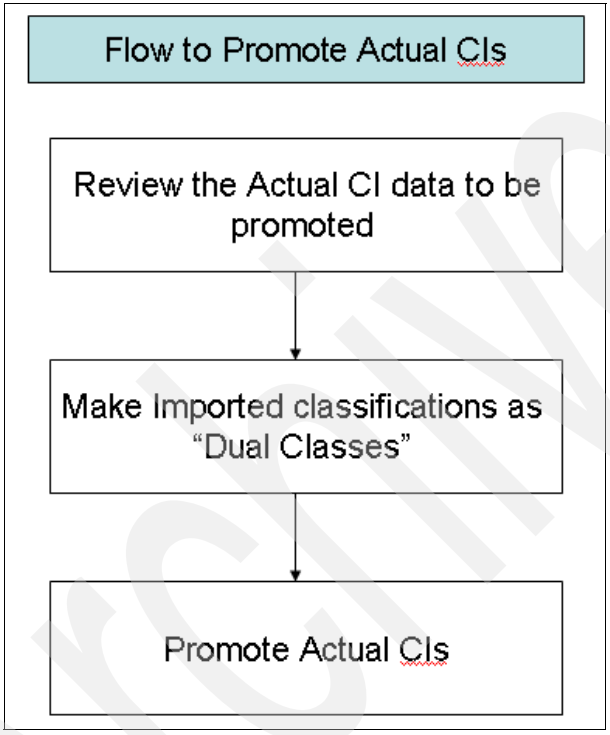


Figure 5-21 Flow for CI promotion

5.2.1 Step by step procedure to promote CIs

The following steps promote an Actual CI to an Authorized CI.

Step 1: Review the Actual CI data that you wish you promote

You can promote an Actual CI using the Actual CI application. The Actual CI application would let you promote only those Actual CIs defined as a top-level classification. To determine if this is true, query for the respective classification in

the Classification Application. Figure 5-22 shows the example of class = sys.computersystem; as you can see, it is a top-level class.

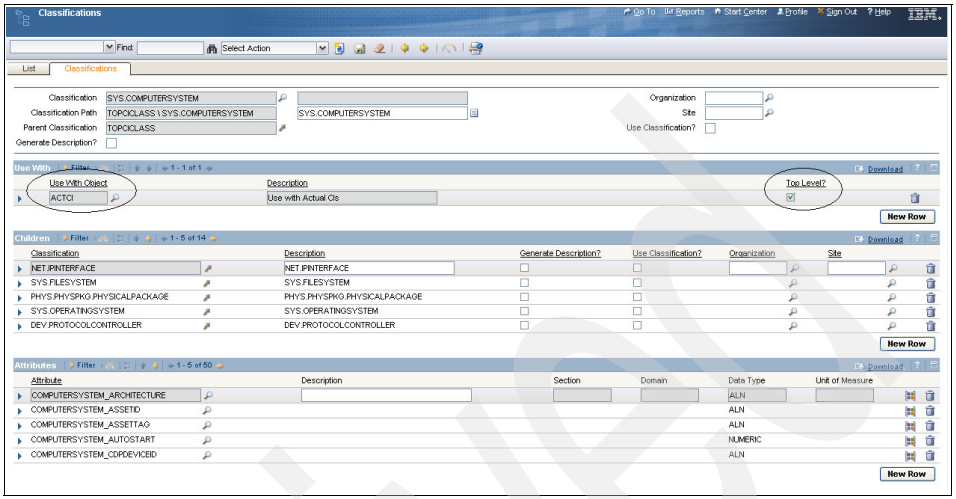


Figure 5-22 Classification query

In the Actual CI Application, query for the Actual CIs that you want to promote. We have already chosen the classification (sys.computersystem), so we can use it as a filter parameter to search for Actual CIs. Figure 5-23 shows the Actual CIs defined under the sys.computersystem classification.

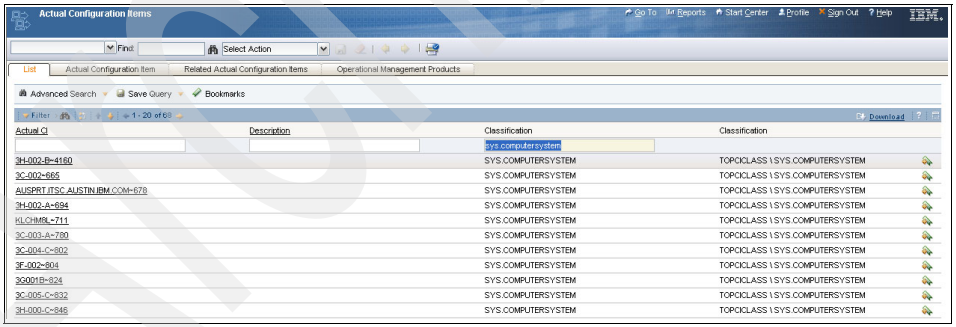


Figure 5-23 Query for Actual CIs

Select an Actual CI that needs to be promoted. Figure 5-24 shows an example.

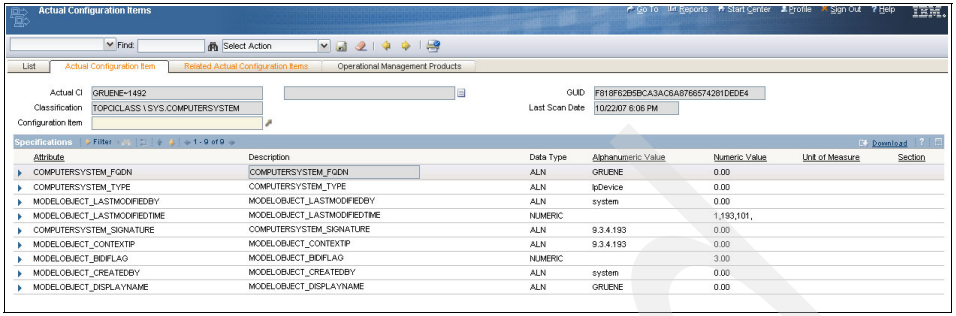


Figure 5-24 Selecting a CI for promotion

To review the related CIs, look at the Related Actual Configuration Items tab in Figure 5-25. This will show all the related CIs that we selected in Figure 5-24.

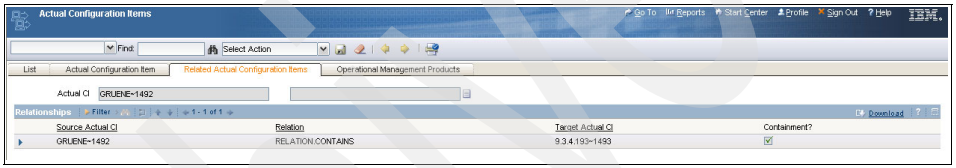


Figure 5-25 Viewing related CIs

While promoting a CI using dual classes (without using an authorized hierarchy), you will be promoting the Actual CI using the Actual CI class structure. So after promoting an Actual CI, you would get the exact replica of the entire Actual CI hierarchy in the authorized space. In this example, one Actual CI, one related CI, and one relation should be promoted.

Note: In real scenarios, you should gather all the information related to classifications for each related Actual CI. In the previous example, we have only one related CI.

Step 2: Establishing Imported classifications as dual classes

Before promoting the Actual CI, we have to make all related classification *dual classes*. By now all classes are defined in the Actual CI context. To promote the Actual CIs, we need to make them usable in both the Actual CI context and the Authorized CI context. In our example, the following are the classification details:

- Root Class: TOPCICLASS
- Top Level Actual CIs class: SYS.COMPUTERSYSTEM
- Relation: Related.Contains
- Related Actual CIs class: NET.IPINTERFACE

Note: In real scenarios, you should gather similar information related to the Actual CIs that are ready to be promoted.

Now you query the TOPCICLASS in the classification application to make it dual, as shown in Figure 5-26.

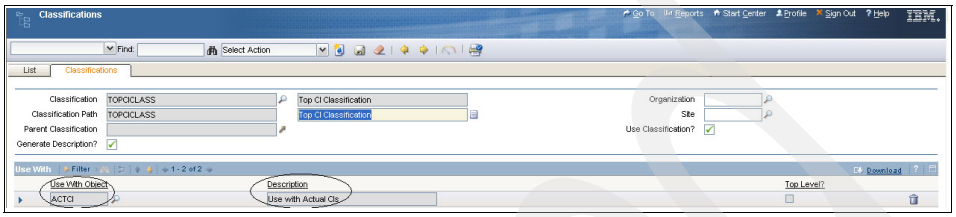


Figure 5-26 Query for TOPCICLASS

You can see the root class TOPCICLASS already has one record: Use with Actual CIs.

The next step would be to add one more record for Authorized CI (Use with Configuration Items) and this will make this classification a dual class. To do so, click **New Row** in the Use With section. You can add a new row with Object = CI (Use with Configuration). Refer to Figure 5-27 for more details.

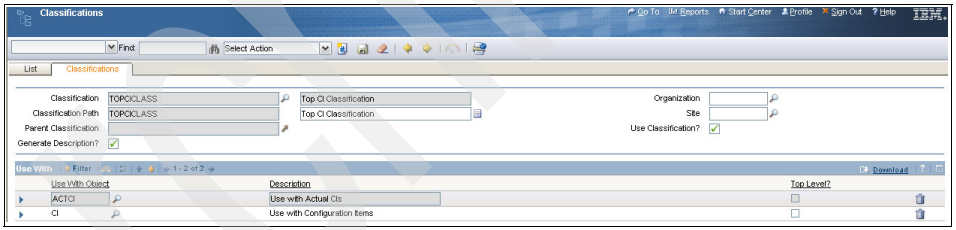


Figure 5-27 Adding an Authorized CI record

Perform the previous steps for the rest of the application classes, SYS.COMPUTERSYSTEM and NET.IPINTERFACE. Figure 5-28 and Figure 5-29 are sample windows for SYS.COMPUTERSYSTEM and NET.IPINTERFACE. You can see that both have been defined as dual class.

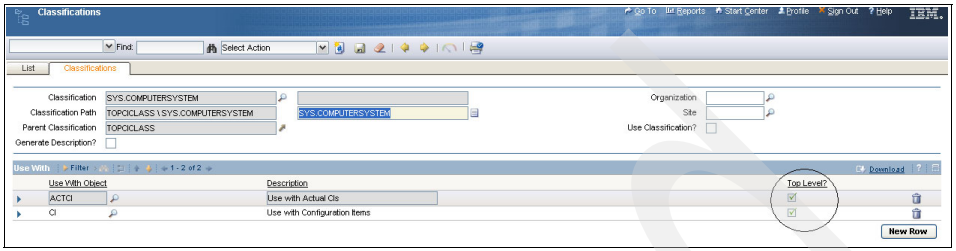


Figure 5-28 SYS.COMPUTERSYSTEM

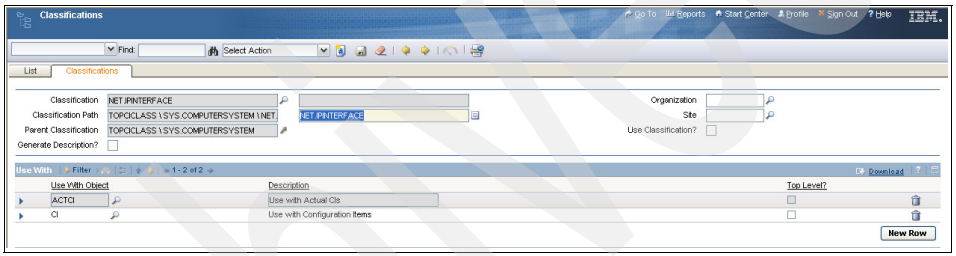


Figure 5-29 NET.IPINTERFACE

Note: In Figure 5-28, you can see that the top-level actual classification is SYS.COMUTERSYSTEM. You have to make sure that while you are making the top-level classification a dual class that the new record “Use with Configuration Items” is defined as top level.

Step 3: Promoting Actual CIs

There are two ways to Promote the Actual CIs:

- Promoting one CI at a time
- Promoting more than one CIs at time

Promoting one CI at a time

Note: 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 classification.

In our example, our Top-Level actual class is SYS.COMPUTERSYSTEM, and we can promote any CI defined with this actual class. You can query all CIs under the Actual Configuration Items Application and can pick the desired Actual CI. For example, we queried Actual CIs for class = SYS.COMPUTESYSTEM beginning with the letter G%, as shown in Figure 5-30.

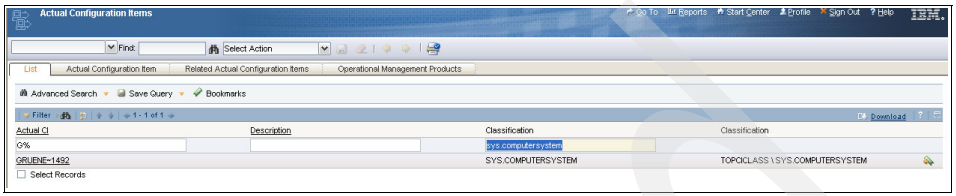


Figure 5-30 Query Actual CIs

Click the **Actual Configuration Item** tab. This view shows details about the respective Actual CI attributes, and so on, as shown in Figure 5-31.

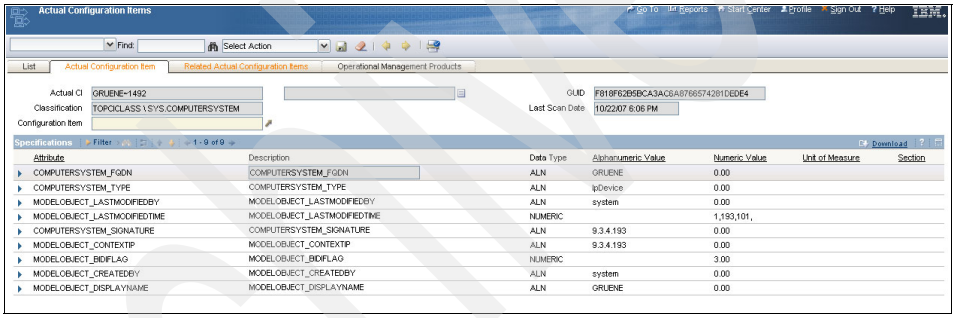


Figure 5-31 Actual CI details

Select **Select Action** → **Create Authorized Configuration Item** to promote the CI, as shown in Figure 5-32.

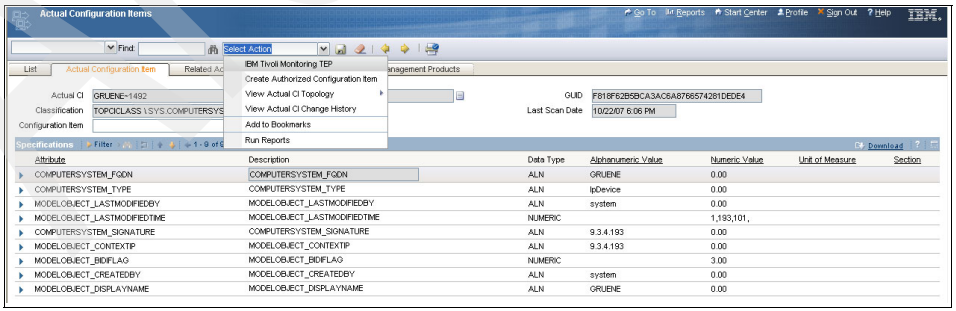


Figure 5-32 Create Authorized CI action

A dialog box appears, as shown in Figure 5-33. In this case, you are promoting without authorized classification, so Actual CI class and CI class would be the same as though you have already defined Actual CI class as dual.

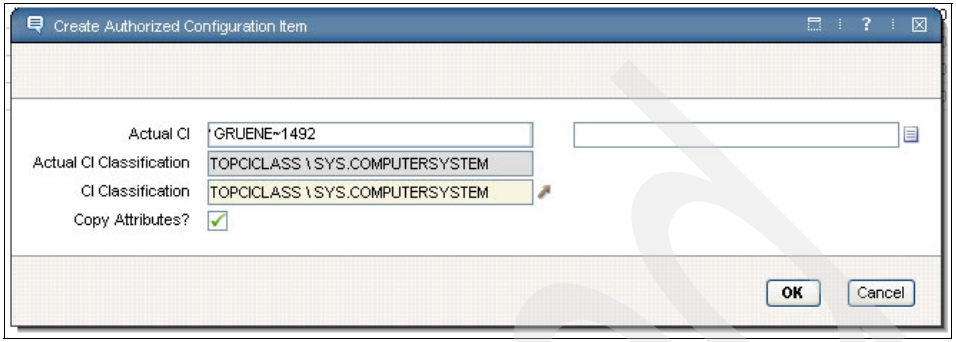


Figure 5-33 Create Authorized CI dialog

Note: If you want to promote an Actual CI with all its attributes values, check the **Copy Attribute** check box. This would bring all attributes associated with the CI to Authorized Space. These attributes values can be modified or deleted and new attributes can also be added.

Click **OK** to start the promotion. This may take some time.

After a successful promotion, the Actual CI can be seen in the Authorized CI's space. To view the promoted CI, go to the Configuration Item Application and query the CI, as shown in Figure 5-34.

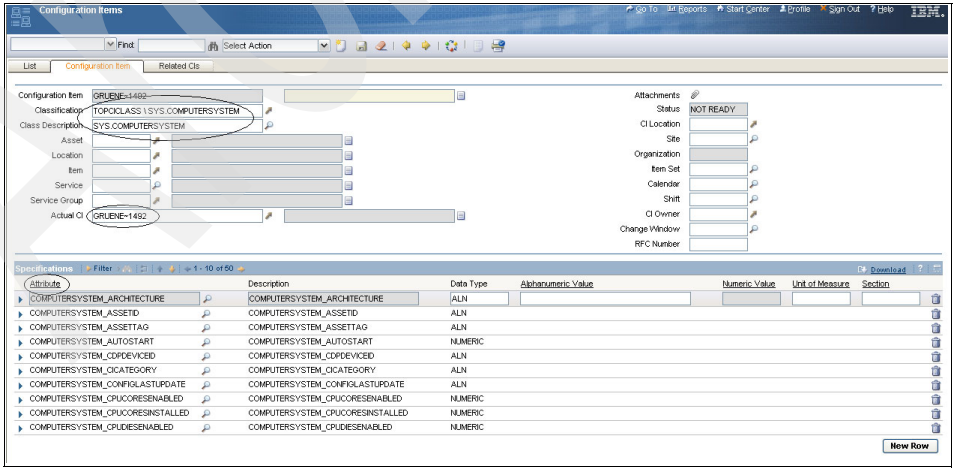


Figure 5-34 Viewing the Authorized CI

In Figure 5-34 on page 134, you can review the desired results, classification details, associated Actual CI, and attributes.

To check the related CIs and the relations, click the **Related CIs** tab. This tab shows that all related CIs and relations have been promoted successfully.

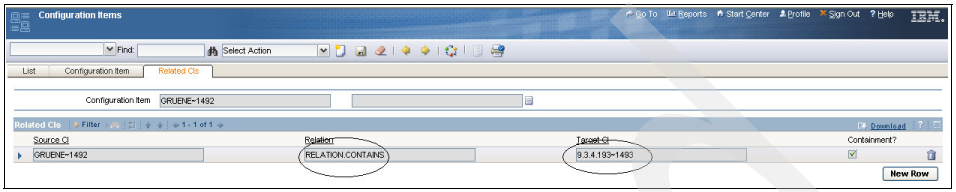


Figure 5-35 Viewing related CIs

Promoting more than one CI at time

Note: 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, go to the Actual Configuration Application and query the desired Actual CIs using filter criteria. For example, we have queried class= SYS.CLASSIFICATION and Actual CIs that begin with 3%, as shown in Figure 5-36.

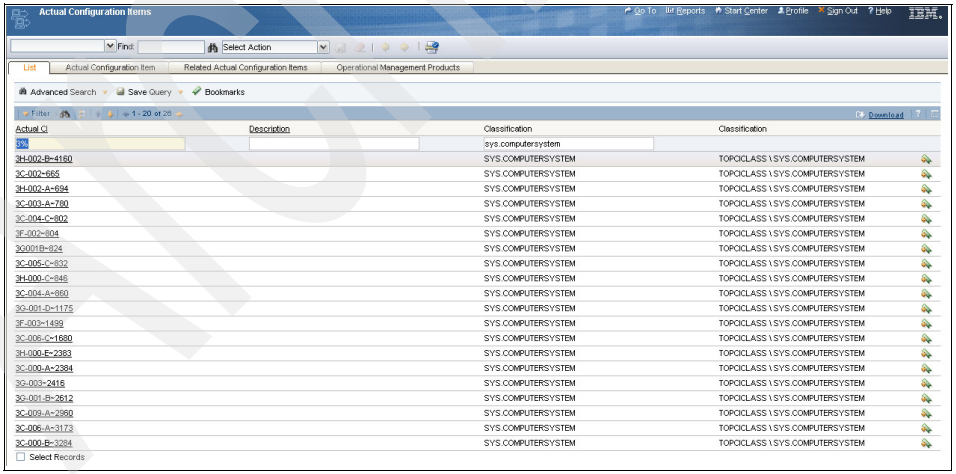


Figure 5-36 Query for multiple CIs

To promote multiple CIs, you must enable the check boxes. Click the Select Records check box at the bottom of the list, and select the **Create Authorized Configuration Items** option.

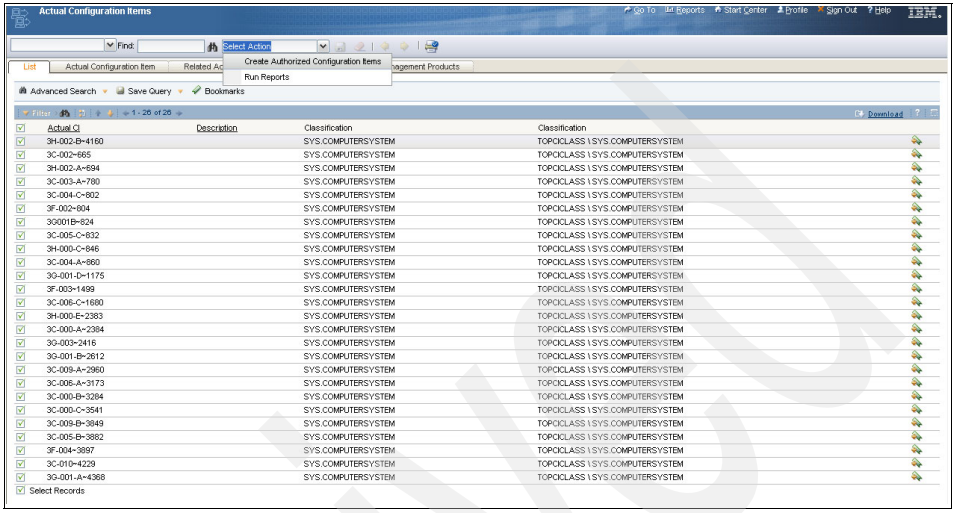


Figure 5-37 Selecting multiple records for promotion

After selecting the Authorized Configuration Items option, you see the dialog box shown in Figure 5-38. Specify the Authorized CI Classification name. As we are using dual class, it should be same as the Actual CI Class. The Copy Attribute option is optional.

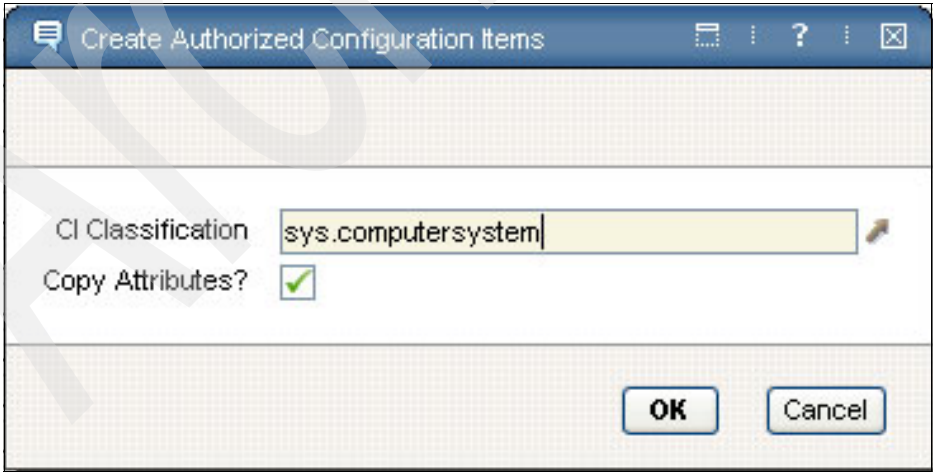


Figure 5-38 Promotion dialog

Click to **OK** to start the promotion process. This may take some time.

THe results can be reviewed in the Configuration Items application.

5.3 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.

6

Implementing federation

Federation, in contrast to synchronization, leaves the data within its source. No data is imported into CCMDB when using the federation approach. Within the CCMDB solution, both ways of dealing with data are supported.

In this chapter, we use a specific use case in order to show which steps you need to take if you want to expose a federated data source to CCMDB provided applications. In our case, we extended the *Actual Configuration Items* application in order to enrich the panels of this application with additional data from the federated data source. But once you exposed the federated data to the CCMDB process runtime environment, you can use it with any application, including the PMP provided applications, such as Change or Configuration Management.

Note: Please note that this chapter is not talking about how to use the federation approach inside the TADDM discovery environment.

There are slight differences in how you have to set up the federation depending on your CCMDB runtime environment and the target data source 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, is your target data source either a DB2 or Informix® database?

- If using DB2, is your target data source a relational data source like Oracle, Microsoft® SQL Server, or Sybase? Is the data source a mainframe database like VSAM, IMS™, or from Software AG? Is the data source falling into the range of Excel®, a flat file (for example, comma separated file), an XML file, a Web Service, a WebSphere® Business Integrator application, or an ODBC data source?

If your CCMDB process runtime database (also referred to as the Maximo database) is DB2 and your target data source is either DB2 or Informix, then you do not have to install anything in addition to what was installed with CCMDB. DB2 is capable of federating DB2 and Informix data sources without any additions.

If your CCMDB process runtime database (also referred to as the Maximo database) is DB2 and your target data source is anything in the range of the data sources listed above except 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 package and is actually an extension to DB2 specifically for federation purposes.

Note: 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.

If your CCMDB process runtime database is Oracle, you are required to have a federation solution provided by Oracle. Oracle provides two solutions to federate data: *Oracle Generic Connectivity* and *Oracle Transparent Gateway*. These two solutions allow you to access non-Oracle systems from an Oracle based (CCMDB) environment. The base Oracle environment provides the ability to federate other Oracle environments. IBM does not provide the Oracle federation extensions as part of the CCMDB V7.1 package.

Figure 6-1 on page 141 summarizes the concept of federation if using DB2 as the runtime database, also referred to as the federation server, if it is the entity that is used to federate. Figure 6-1 on page 141 depicts both options, using DB2 or using DB2 and WebSphere Federation Server as an add-on.

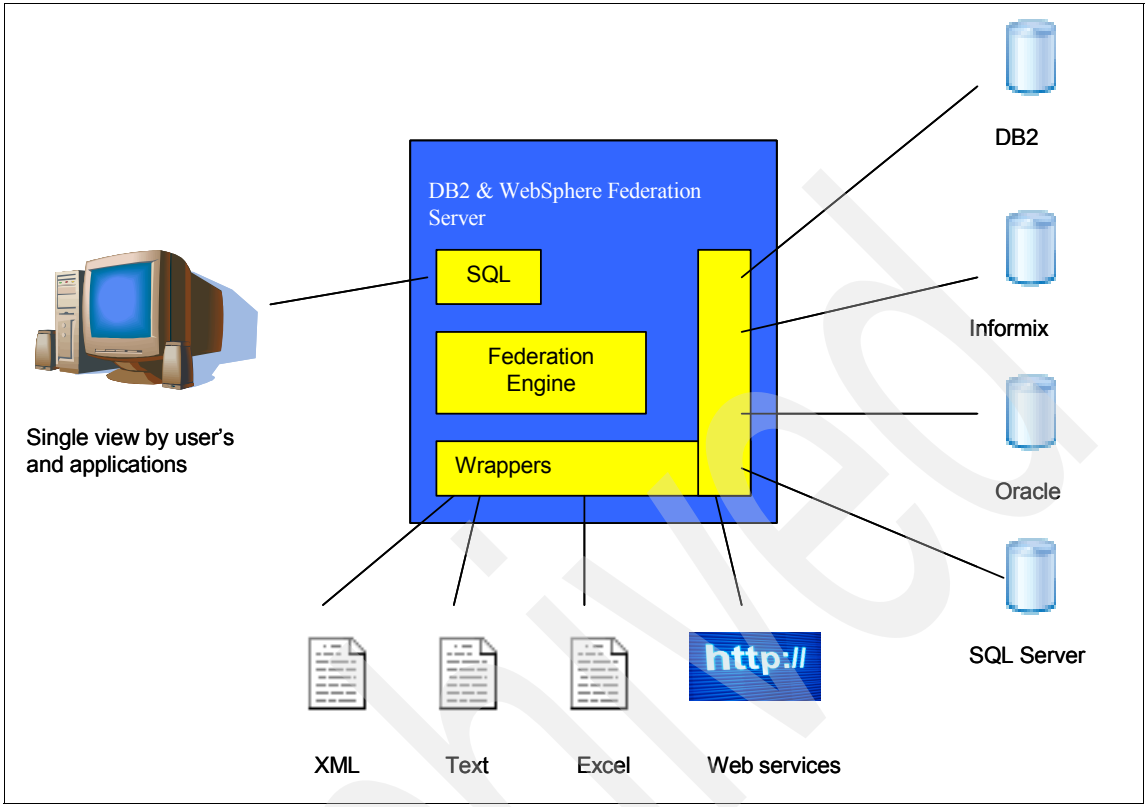


Figure 6-1 DB2 and WebSphere Federation Server

Wrappers, as shown above, are used by the federation server to communicate with and retrieve data from remote data sources. They abstract the communication protocol and access mechanism of the remote data source to the federation server.

If you would federate from an Oracle database instance, conceptually the picture would look like similar, except you would use Oracle provided technology to set up the federation and have a smaller number of target data sources to federate. Please check your Oracle documentation to get the latest list of supported data sources that you can be federated using either the Oracle Generic Connectivity or Oracle Transparent Gateway solution.

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 DB data source. We *do not* show how to install WebSphere Federation Server V9.1 or any Oracle technology.

Tip: As part of the CCMDB V7.1 deliverables, a best practice toolkit will be provided. Part of this toolkit is a whitepaper on Data Federation for CCMDB that talks about the WebSphere Federation Server 9.1 implementation as well as using Oracle technology.

After describing the scenario we are using, we guide you through the configuration steps you have to take in the database layer and in the process environment in order to enrich your CCMDB applications with federated data.

6.1 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 6-2 gives an overview of the scenario architecture that we are using in our lab environment.

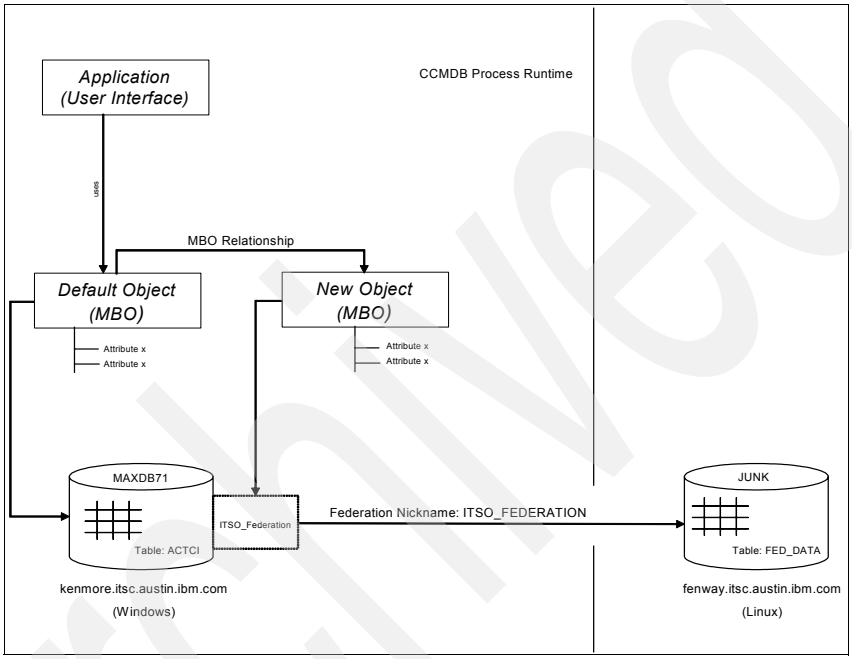


Figure 6-2 Lab environment for federation

Our CCMDB 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 CCMDB 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 leverage 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.

Once 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 leverage the federated data source to define a new Maximo Business Object (MBO).

Note: We do not explain the basic concepts of Maximo Business Objects in this chapter. Please refer to the product documentation. In short, 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, since we assume the remote data source already exists in your environment. We describe each step in more detail in the upcoming sections of this chapter:

1. Setting up 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.

6.2 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 guide you through the steps, 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 (see Figure 6-3).

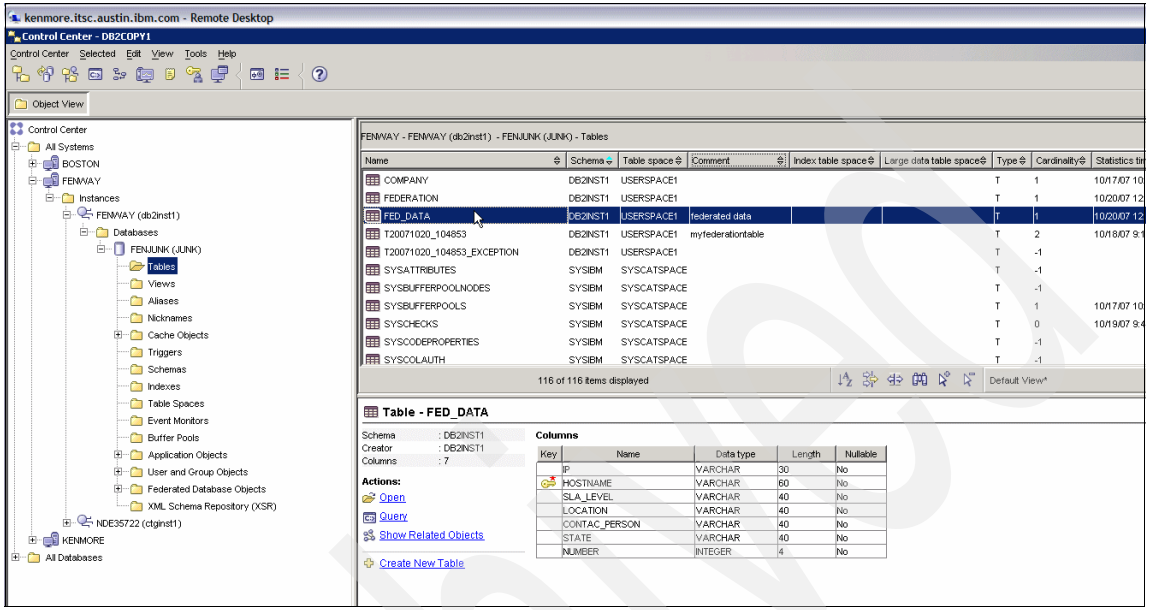


Figure 6-3 FED_DATA table

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, open the DB2 Control Center and search for the instance (by default, the instance name is CTGINST1).

Select the instance, right-click it, and select **Configure Parameters** (Figure 6-4 on page 147).

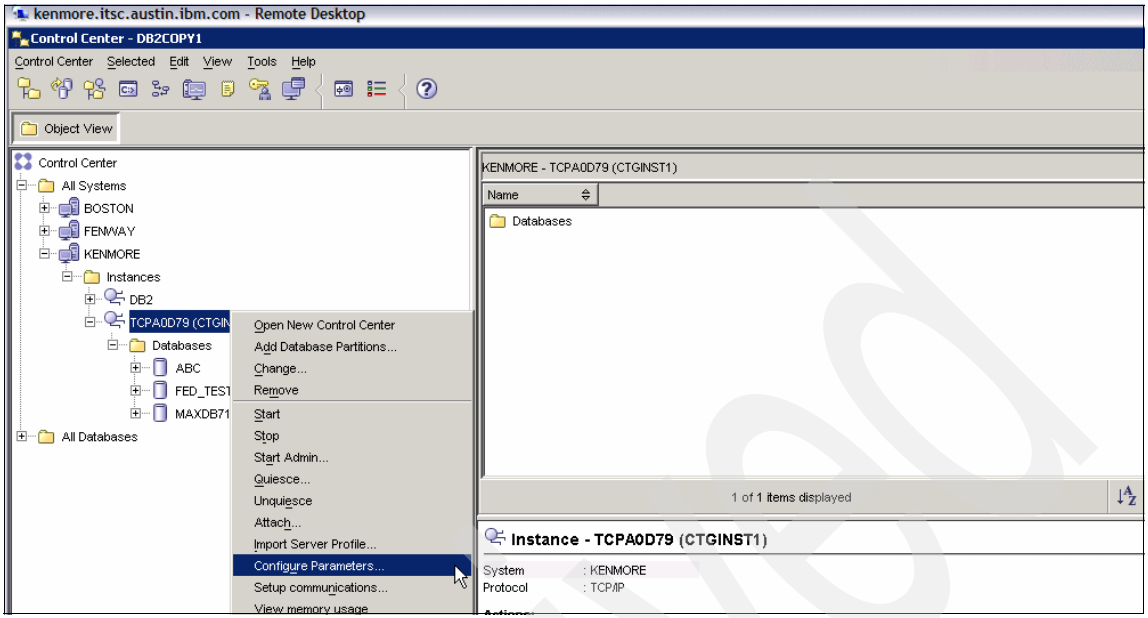


Figure 6-4 Configure parameters option

Search for the keyword FEDERATED and set the value to **Yes**. Click **OK** in order to save your modifications (Figure 6-5).

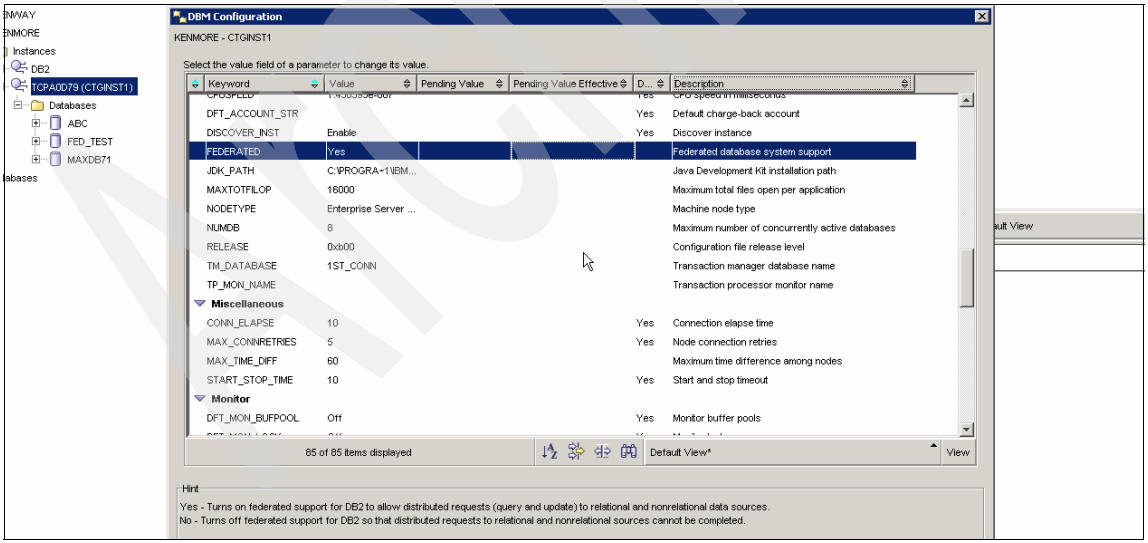


Figure 6-5 Setting the Federated option

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.

6.2.1 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.

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 6-6.

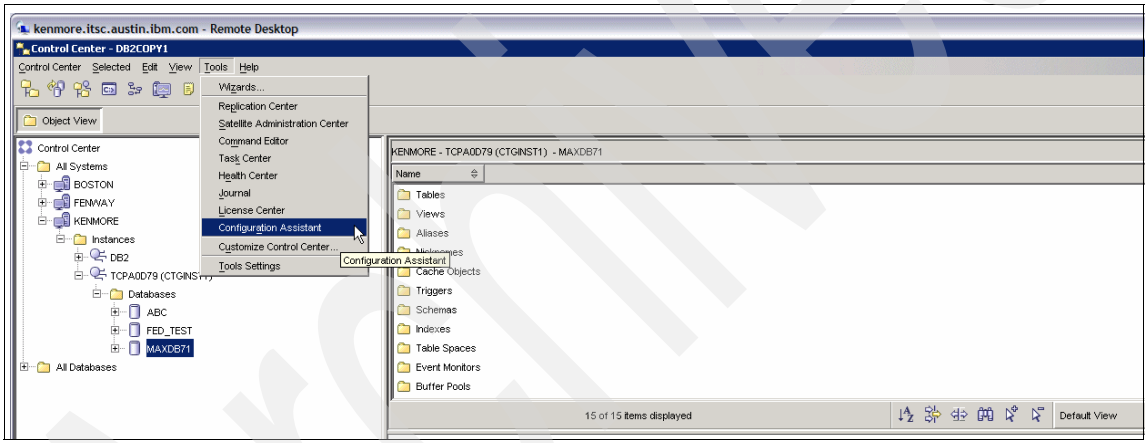


Figure 6-6 Opening the configuration assistant

This opens up the Configuration Assistant Dialog. Right-click in the white space of this dialog and select **Add Database using Wizard...**, as shown in Figure 6-7 on page 149.

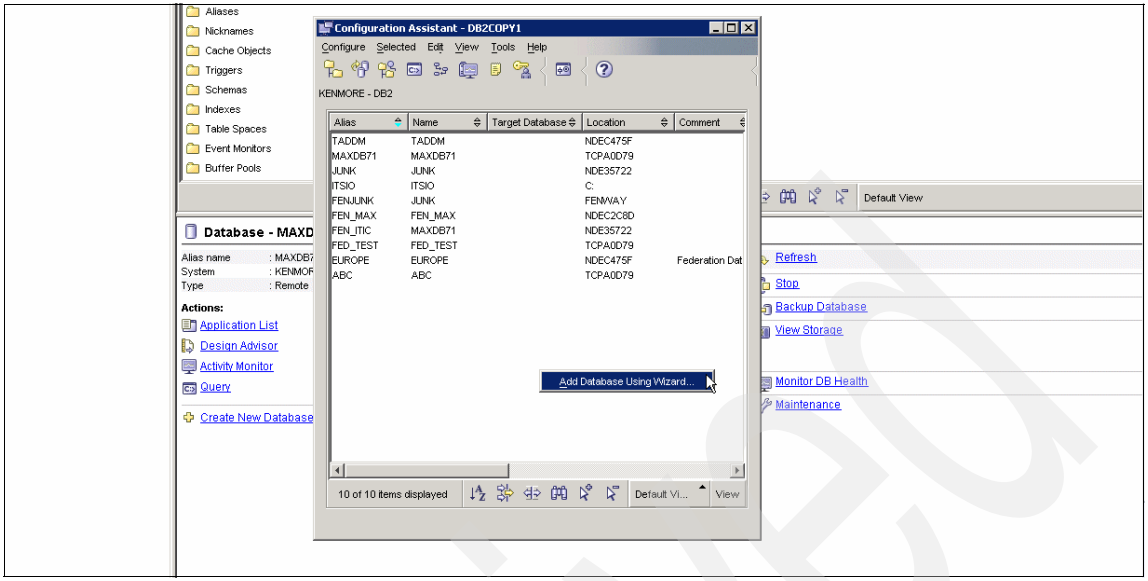


Figure 6-7 Selecting the Add Database Wizard

This brings up the Add Database Wizard. Select the **Search the network** radio button, as shown in Figure 6-8.

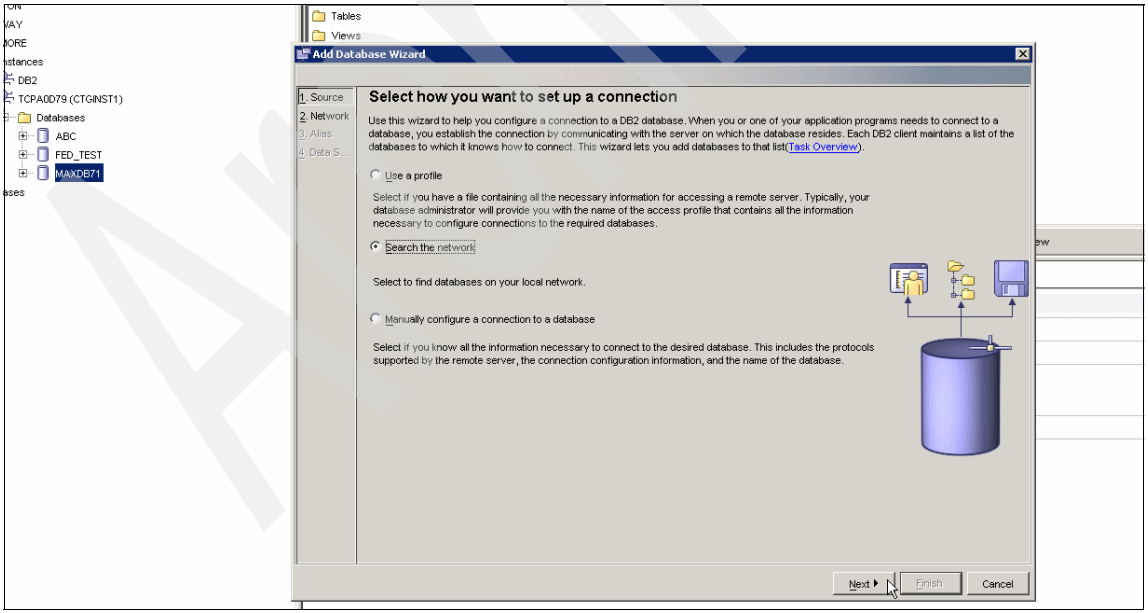


Figure 6-8 Select Search the network

Click **Next**, and the dialog box shown in Figure 6-9 appears.

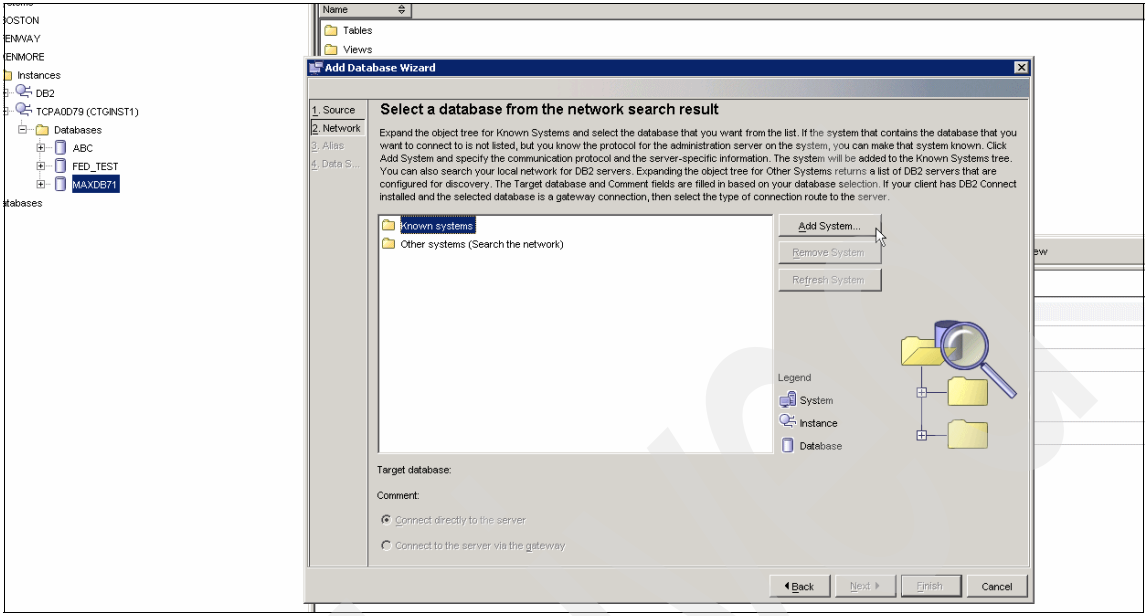


Figure 6-9 Select Add System

Click the **Known systems** folder and click the **Add System...** button. The dialog box show in Figure 6-10 on page 151 appears.

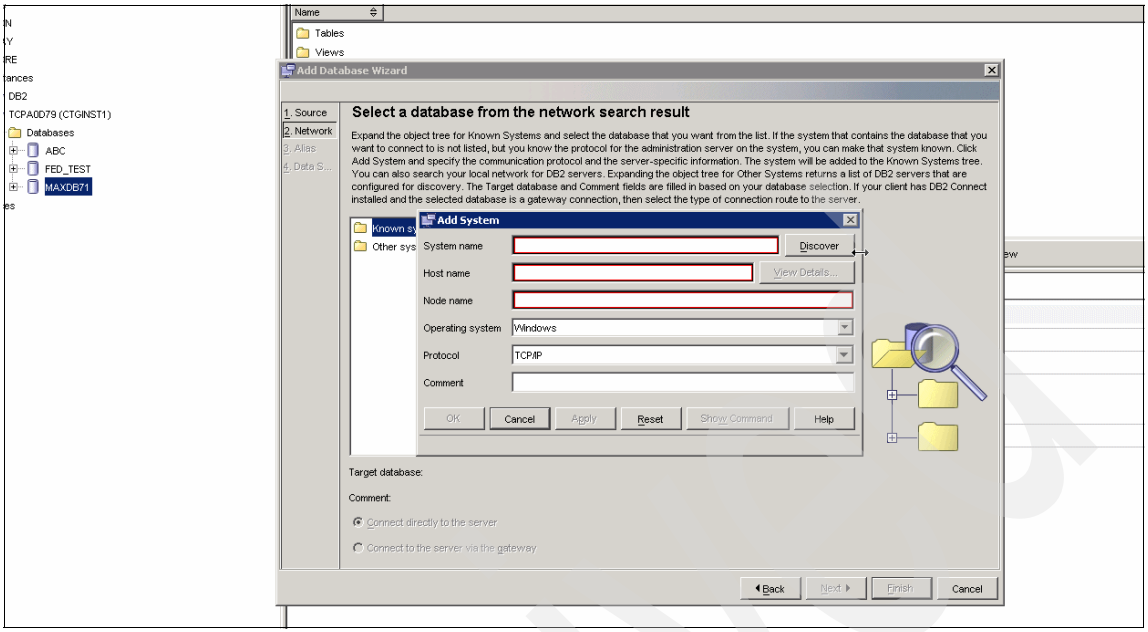


Figure 6-10 Add system dialog

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 6-11.

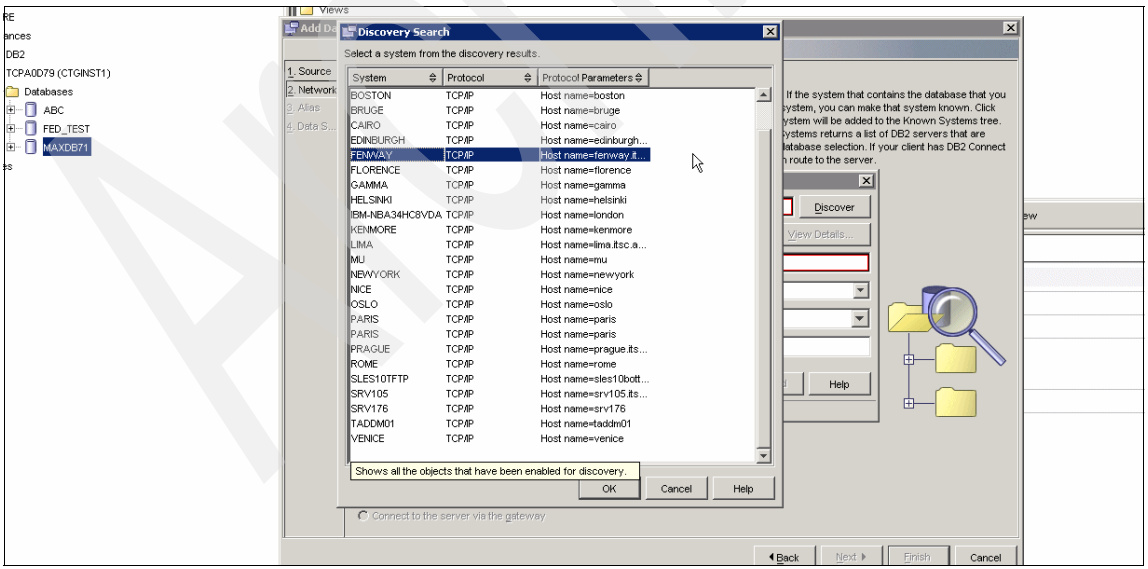


Figure 6-11 List of discovered systems

Select the system that you want to catalog; in our case, it is fenway. Click **OK** and then **Next**. The dialog shown in Figure 6-12 will appear.

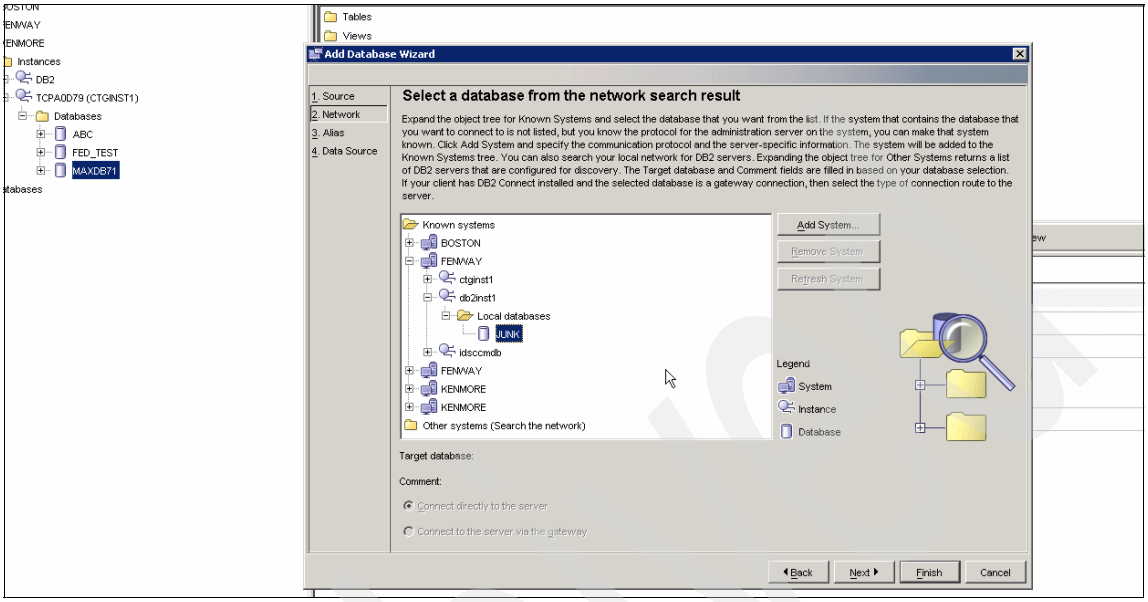


Figure 6-12 Selecting a database from the discovered system

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.

Once you make your selection, click **Next**. The dialog shown in Figure 6-13 will appear.

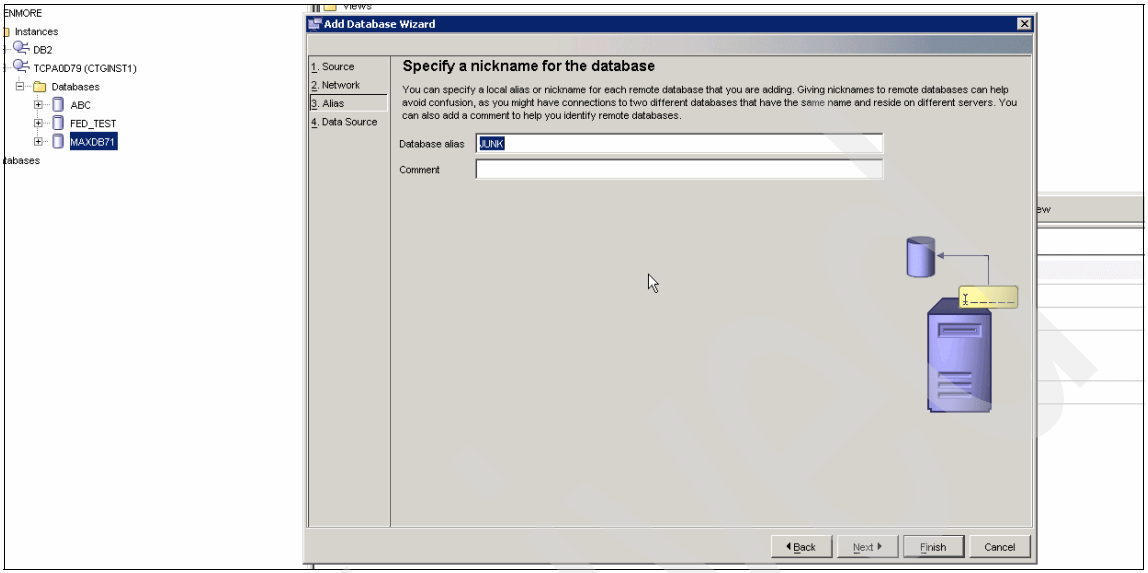


Figure 6-13 Specifying an alias for the remote database

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 6-14 will appear.

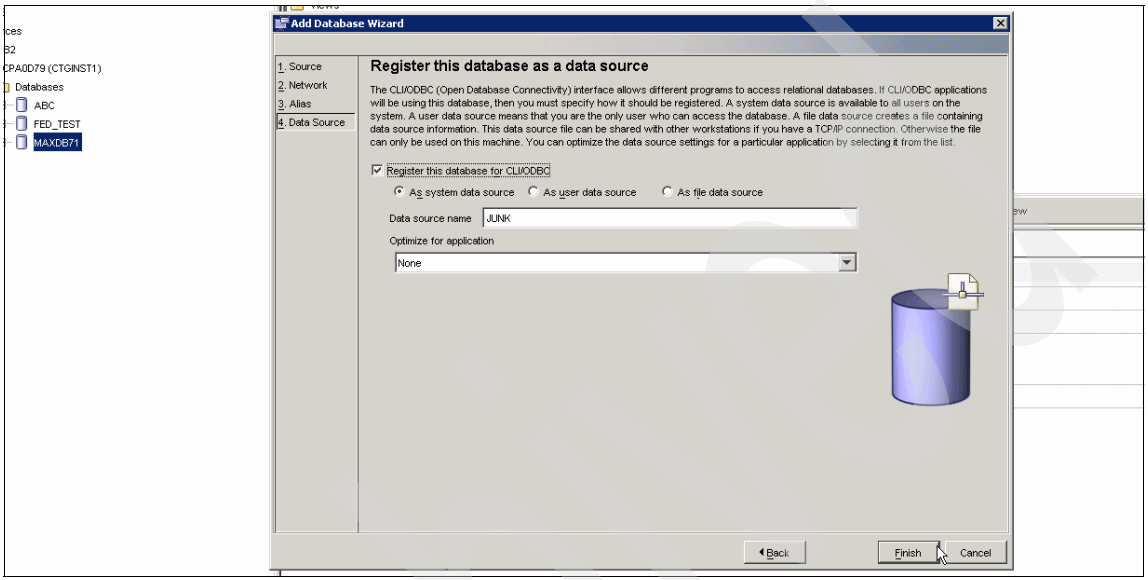


Figure 6-14 Register the database as a data source

You can click **Finish**; registering this database for CLI/ODBC is optional.

You have now concluded the steps of cataloging the remote node and database.

6.2.2 Create 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 a different wrapper.

Note: Once 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.

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 6-15.

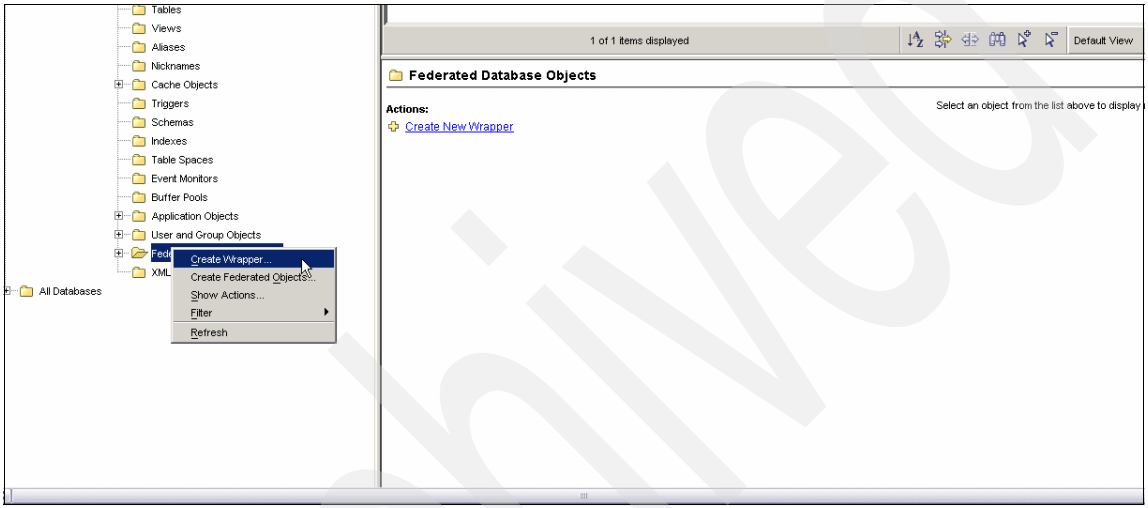


Figure 6-15 Create Wrapper option in DB2 Control Center

In the Data Source drop-down menu, select **DB2** and click **OK**, as shown in Figure 6-16.

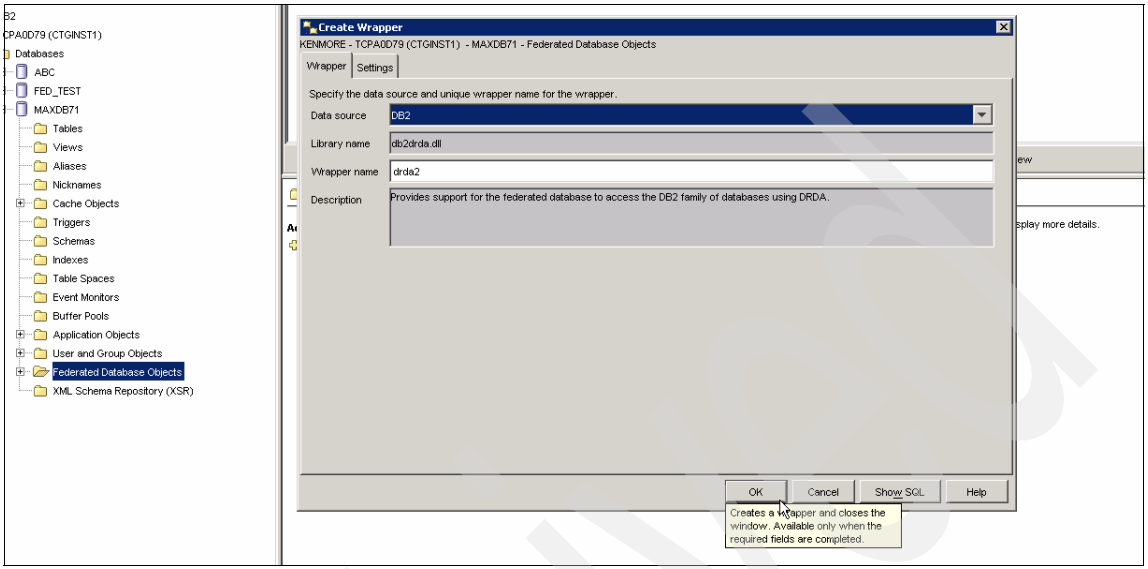


Figure 6-16 Create Wrapper dialog

This completes the creation of the wrapper. You see the wrapper defined as **DRDA®** under the Federated Database Objects folder, as shown in Figure 6-17.

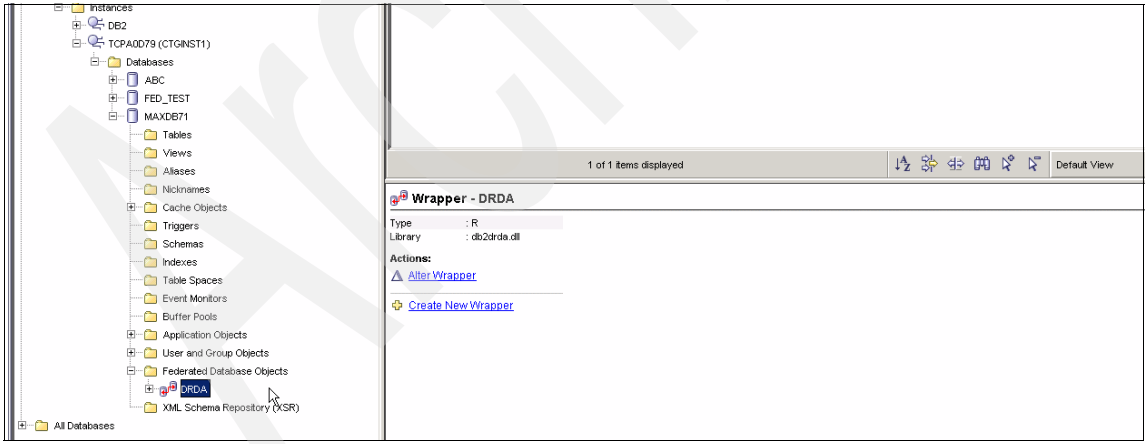


Figure 6-17 Viewing created wrapper

6.2.3 Register server

The next step is create a server definition. The register server operation defines a data source specifically to a federated database.

In DB2 Control Center, expand the DRDA wrapper entry, right-click it, and select **Create**, as shown in Figure 6-18.

Figure 6-18 Menu to create a server definition

The **Create Server Definitions** dialog is shown in Figure 6-19. Click **Add...**.

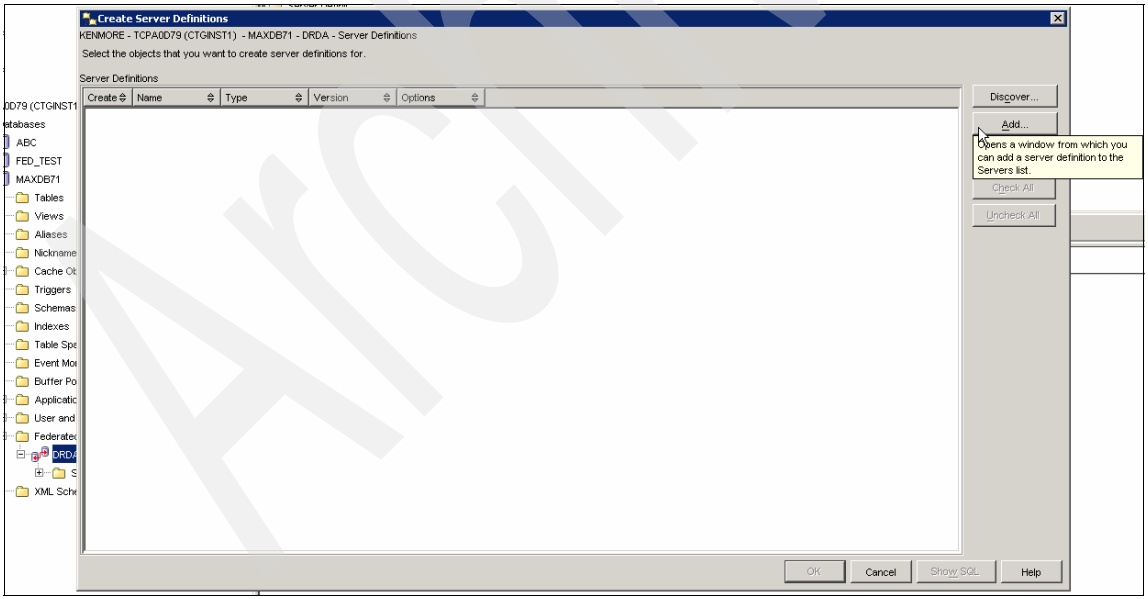


Figure 6-19 Create Server Definitions dialog

The Create Server Definition window is shown. There are two tabs, Server Definition and Settings, that you need to consider.

In the **Server Definition** tab window, provide the appropriate data entries according to your environment (Figure 6-20). 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 that will differ in your environment.

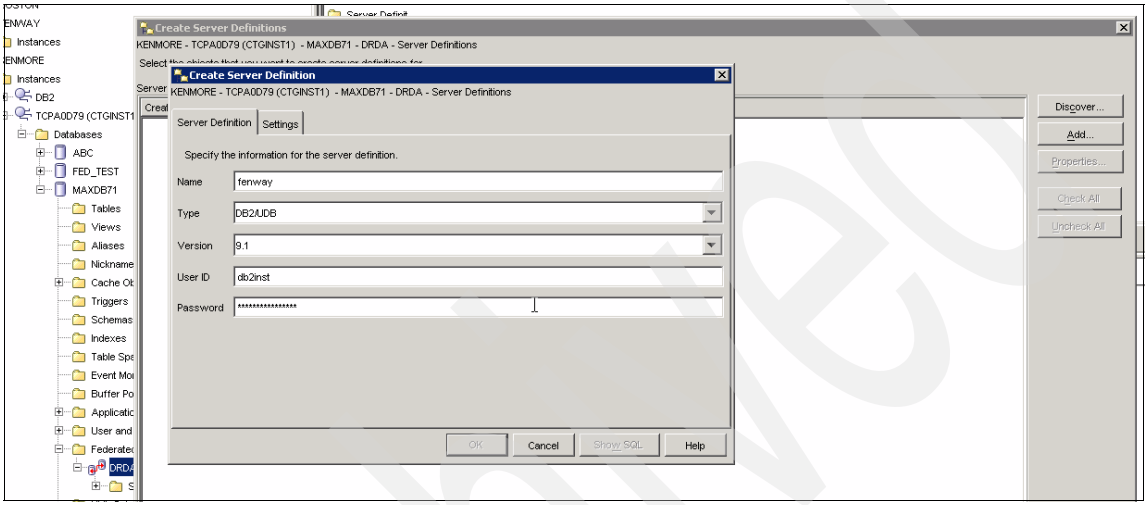


Figure 6-20 Server Definition dialog

Next, 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.

Click **OK** and you find a new server definition entry in the DRDA wrapper folder, as shown in Figure 6-21 on page 159.

Figure 6-21 New server definition appears in DB2 Control Center

This concludes the definition of the server definition. The next step is to create a user mapping.

6.2.4 Create 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.

In the DB2 Control Center, expand the server definition entry you just created in the previous step, select **User Mappings**, right-click it and select **Create...**, as shown in Figure 6-22.

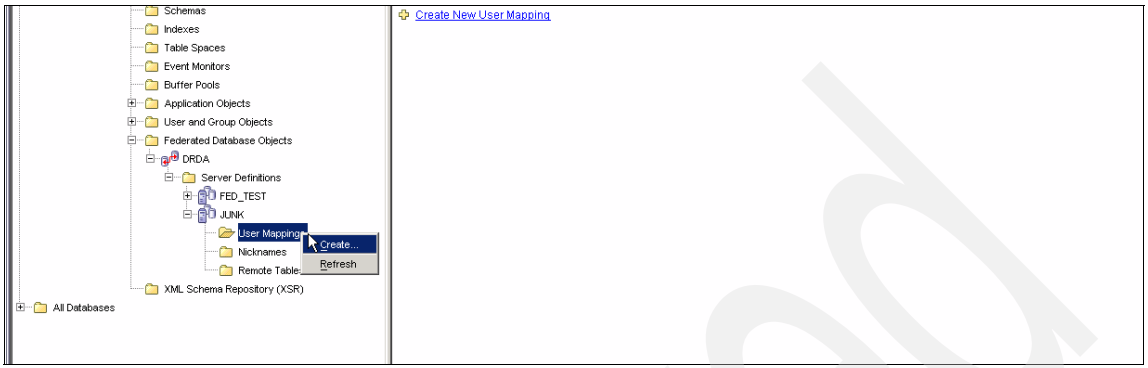


Figure 6-22 Selecting option to create User mappings

The Create User Mappings dialog window is shown. From the Available local User IDs frame, select the local user of your federation server database. In our environment, we select the user MAXIMO.

Note: Since 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.

Select the user and move it over to the Selected user IDs frame in the Create User Mappings dialog window.

In the same dialog, click the **Settings** tab and specify the values for the REMOTE_AUTHID and REMOTE_PASSWORD options, as shown in Figure 6-23.

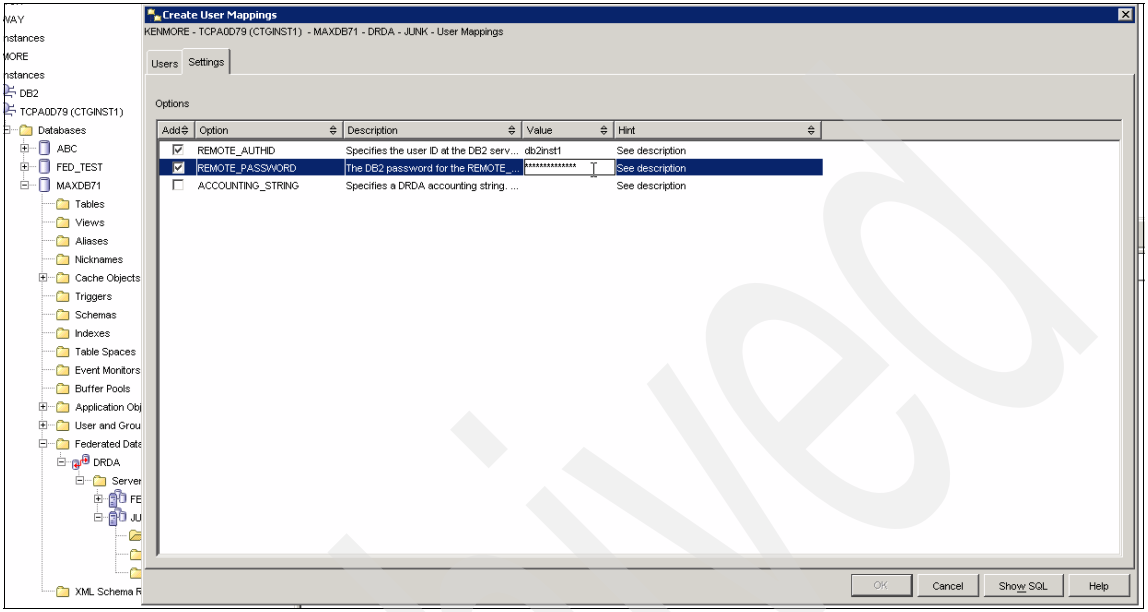


Figure 6-23 User Mapping settings

Click **OK** and you will find an new entry in the User Mappings section of the Server Definitions folder, as shown in Figure 6-24.

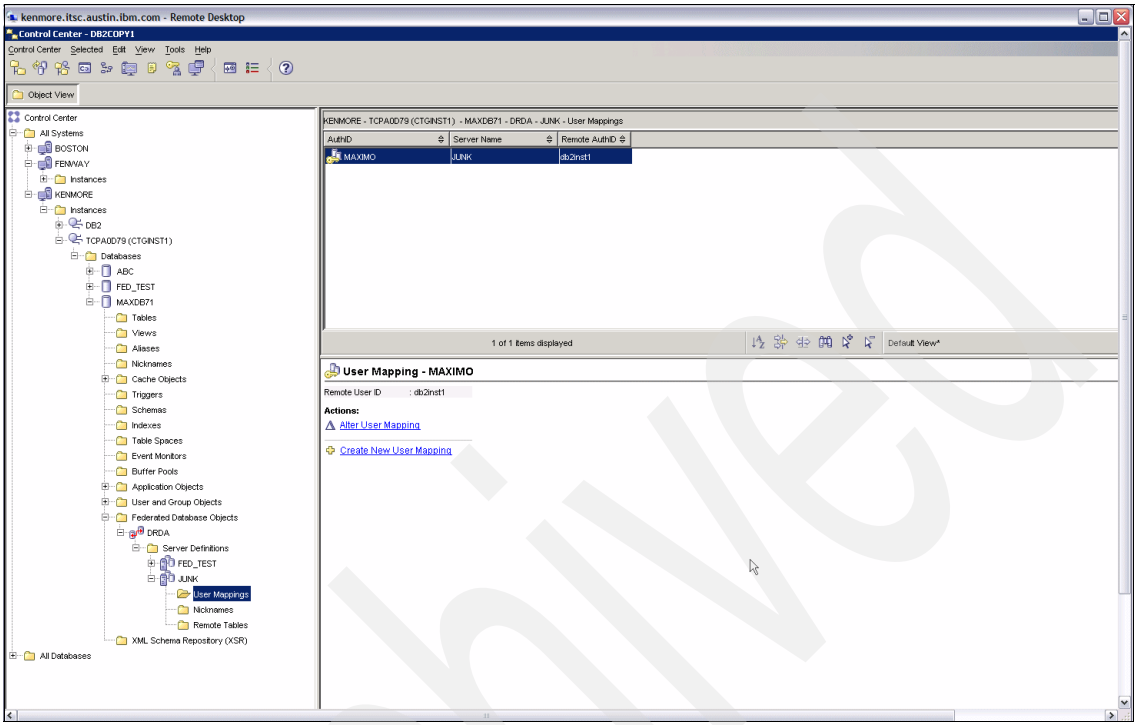


Figure 6-24 New User Mapping entry

This concludes the User Mapping definitions process.

6.2.5 Create 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 a different 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).

In the DB2 Control Center, from the Server Definition that you created, select **Nicknames**, right-click it, and select **Create...**, as shown in Figure 6-25 on page 163.

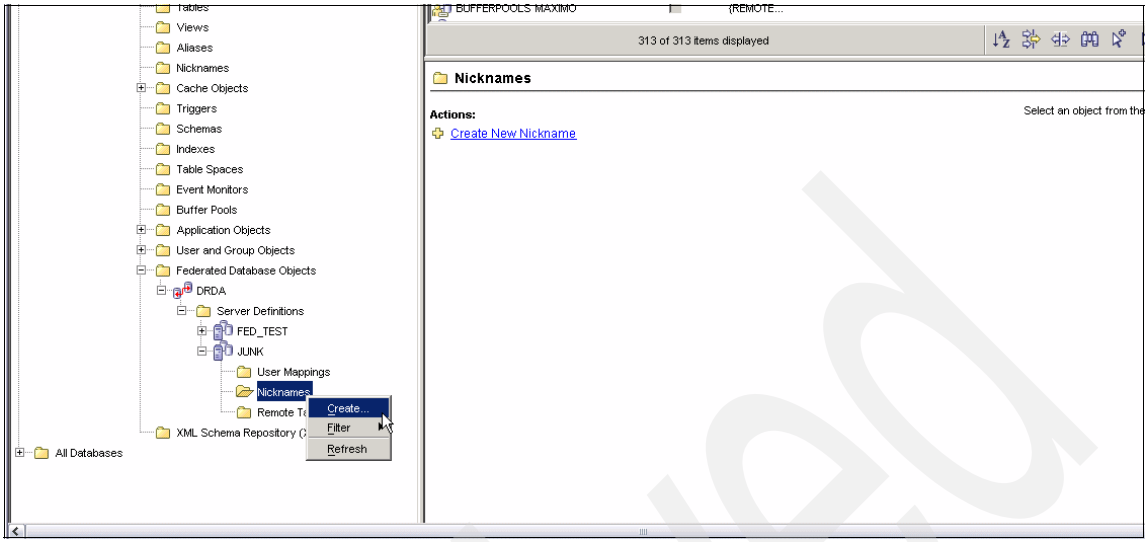


Figure 6-25 Option to create nicknames

The **Create Nicknames** dialog window will appear. Click **Add...**. This will bring up the **Create Nickname** dialog shown in Figure 6-26.

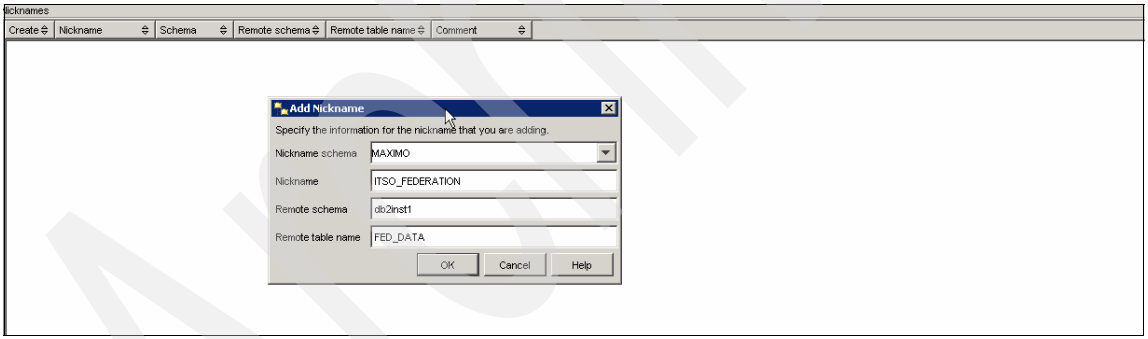


Figure 6-26 Create Nicknames dialog

You have to 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 6-27, 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.

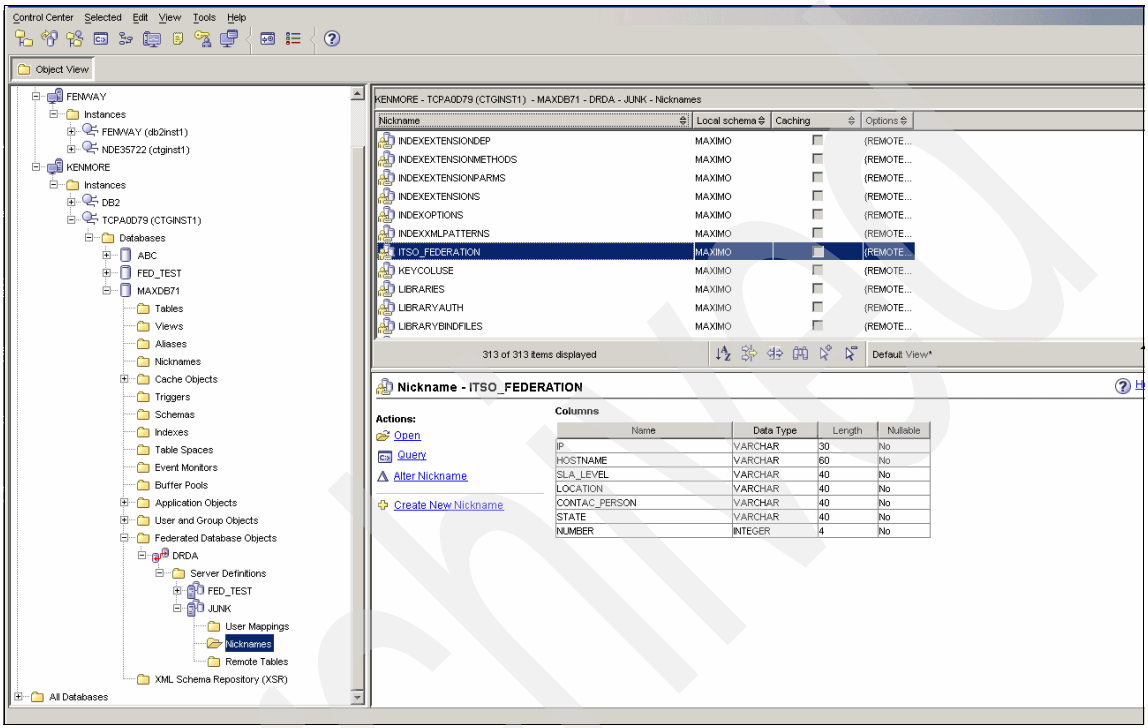


Figure 6-27 Nickname has been created

After you set up the federation at the data source layer, you are now ready to make use of this setup in the CCMDB environment itself.

6.3 Create 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.

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 6-28.

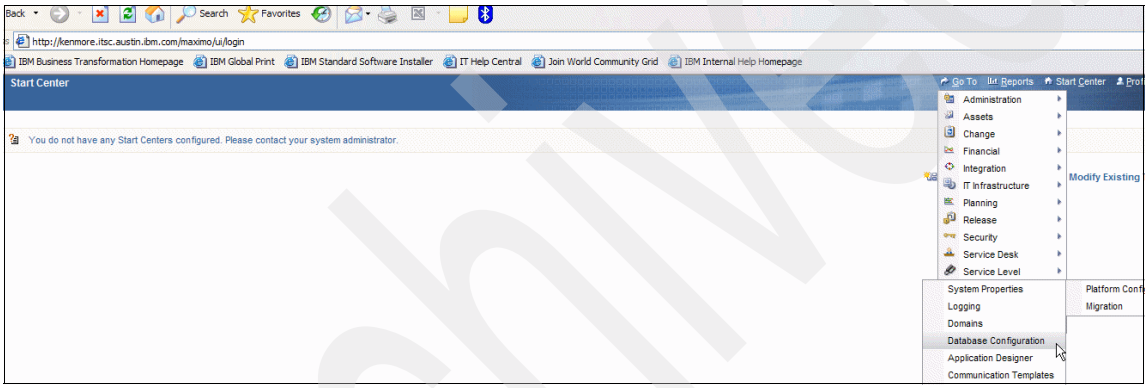


Figure 6-28 Database Configuration window

Next, in the menu bar, click the icon to select a **New Object**, as shown in Figure 6-29.

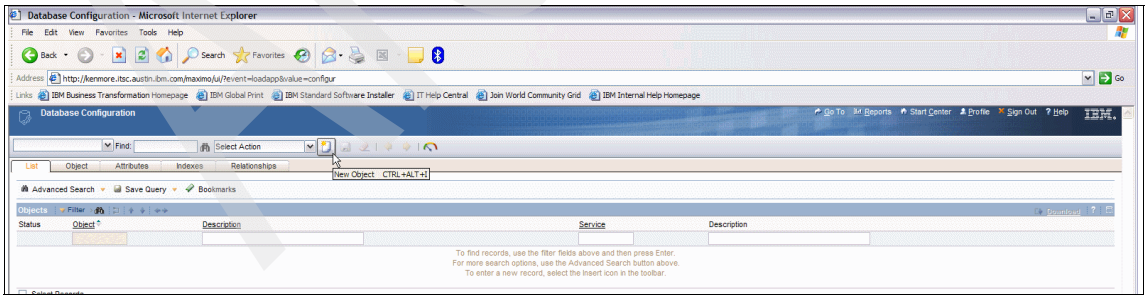


Figure 6-29 Select New Object

This will bring up 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 6-30.

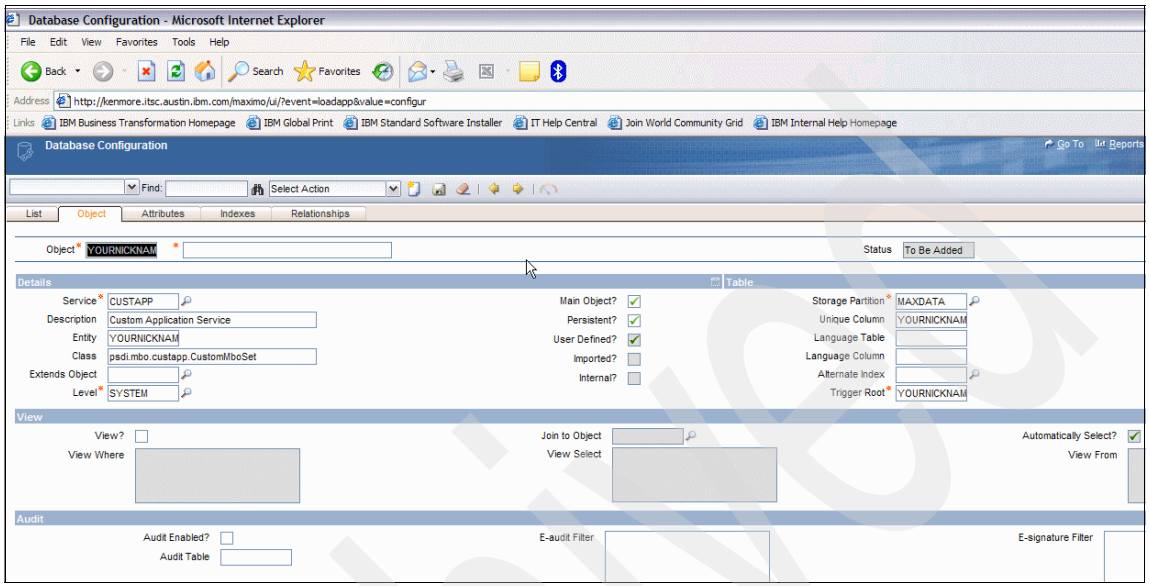


Figure 6-30 Specifying the nickname

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 since this is the nickname that we use to connect to the federated database table FED_DATA. Once 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.

You also need to 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.

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 is 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.

We also select the **Main Object** check box and add the Number column of our FED_DATA table into the Unique Column field of the MBO definition window.

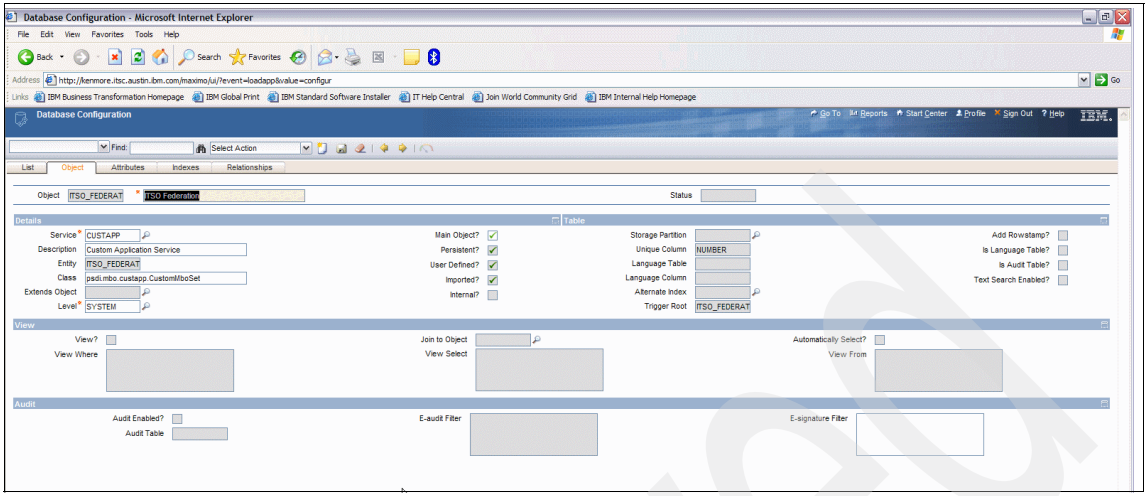


Figure 6-31 MBO Definition window

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.

If you click the **Attributes** tab, you should 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.

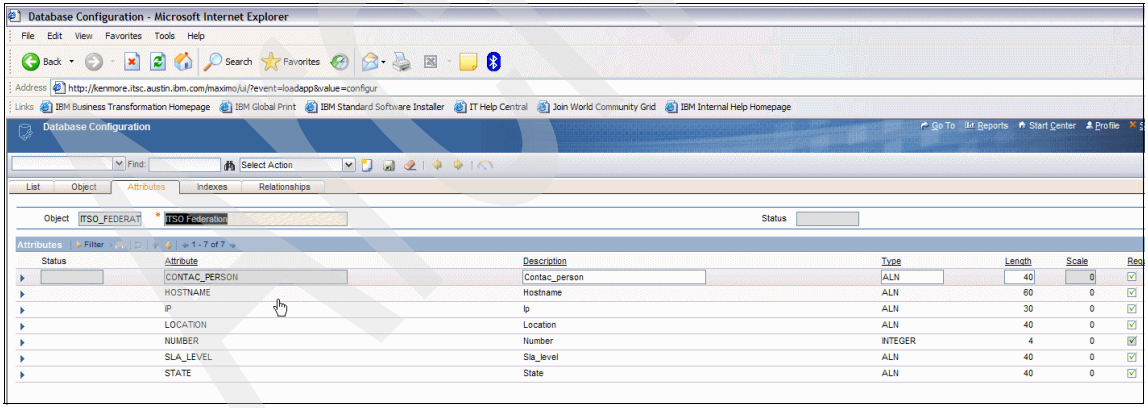


Figure 6-32 Attributes tab

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 6.4, “Generate the object in the CCMDB database” on page 168.

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.

6.4 Generate the object in the CCMDB database

Now that you have defined the object, you must generate it in the database. This is accomplished by running the **configdb** program that is located on the CCMDB process runtime database system.

In order to run the **configdb** program, you need to stop the application server first. We also recommend that you 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 6-33.

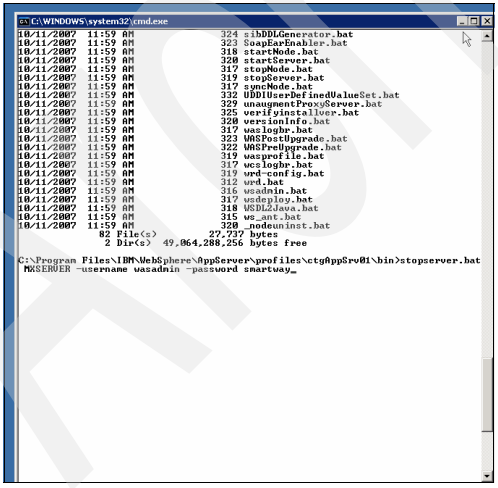


Figure 6-33 Stopping the application server

If you kept all the defaults at installation time, you should find the same directory structure on your application server system, which is kenmore in our case.

Use the **stopserver.bat** program shown in Figure 6-33 on page 168 in order to stop the application server from the command line.

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`

You have to authenticate; we use the wasadmin account in our environment, as shown in Figure 6-34.

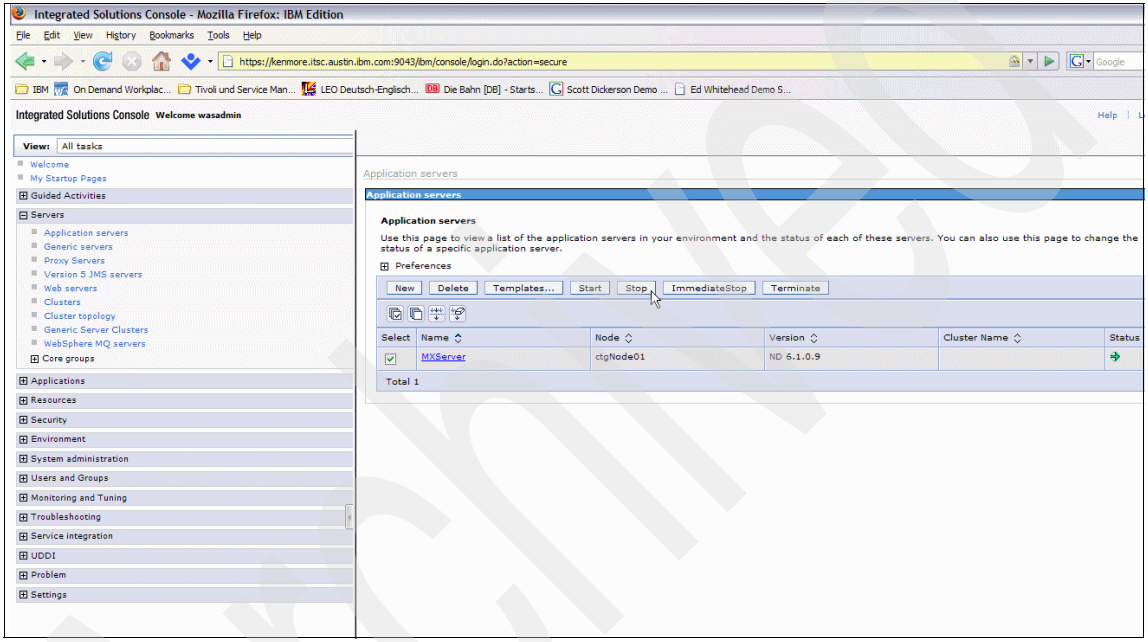


Figure 6-34 WebSphere administration console

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 6-35.

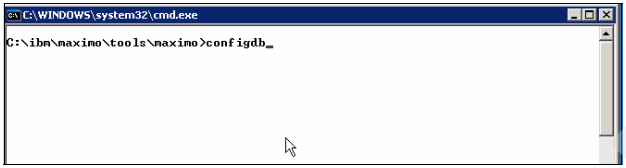


Figure 6-35 Run the configdb command

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 6-36. You can sort it by date.

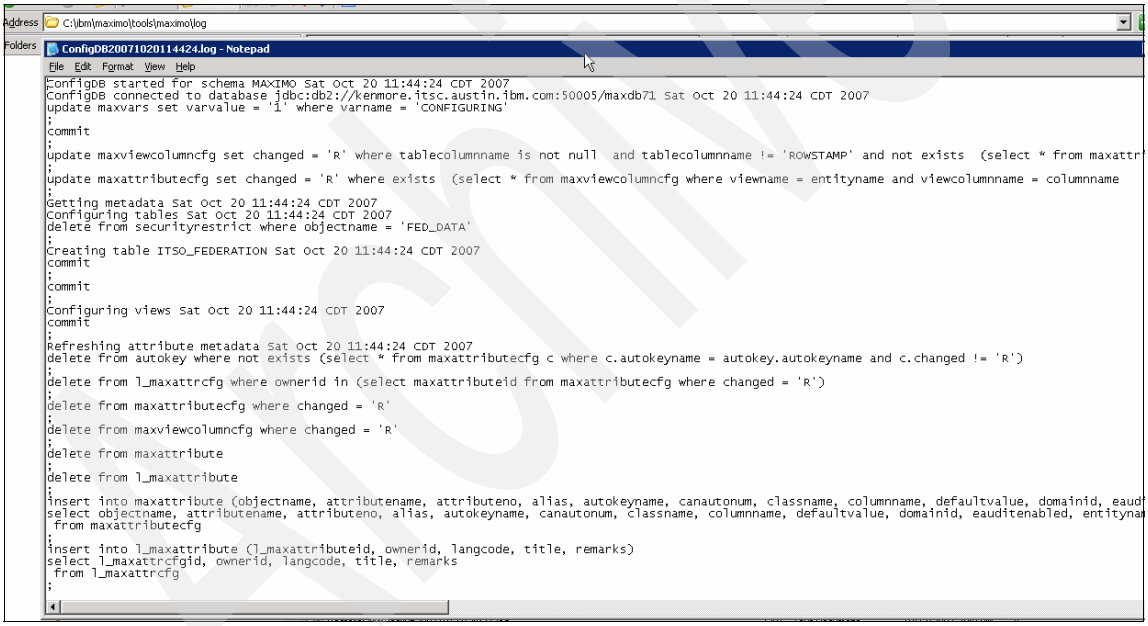


Figure 6-36 Results of configdb command

You can see from the log shown in Figure 6-36 that updates to the local MAXDB71 database are created in order to point to the remote FED_DATA table using the nickname ITSO_FEDERATION.

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.

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 6-37.

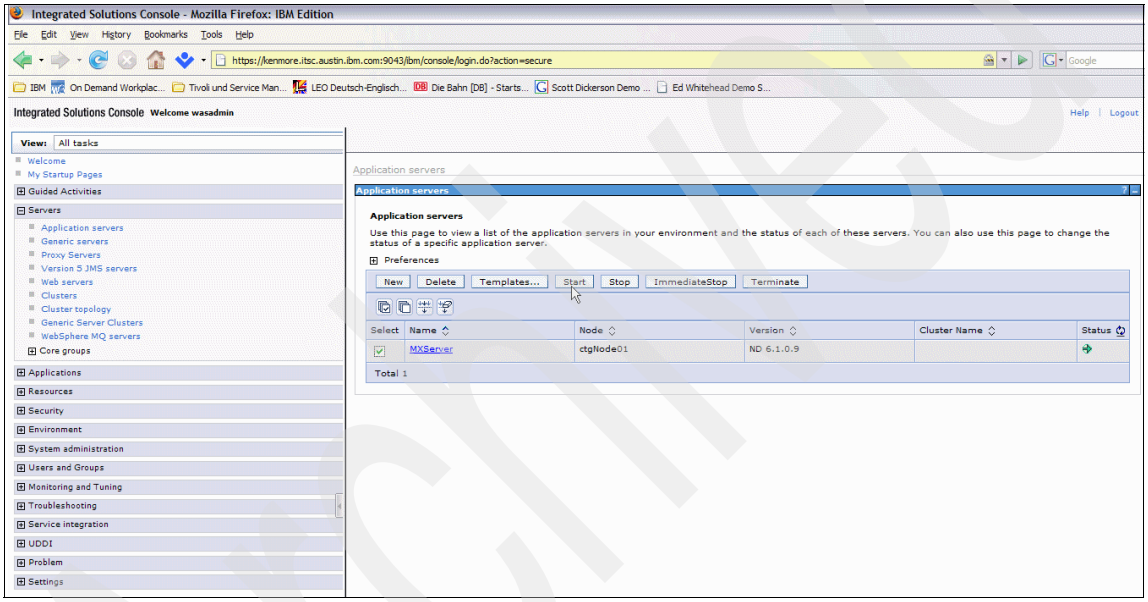


Figure 6-37 Restart the application server through the WebSphere administration console

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 6-37.

6.5 Define a relationship

In order to display additional data provided by a federated database table in an existing application like 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 that is, for example, not discoverable or should be kept and maintained separately.

You can use the new MBO pointing to the federated table in isolation, but we recommend 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 6-38 shows an overview of our relationship setup. We explain how to perform this setup in this section.

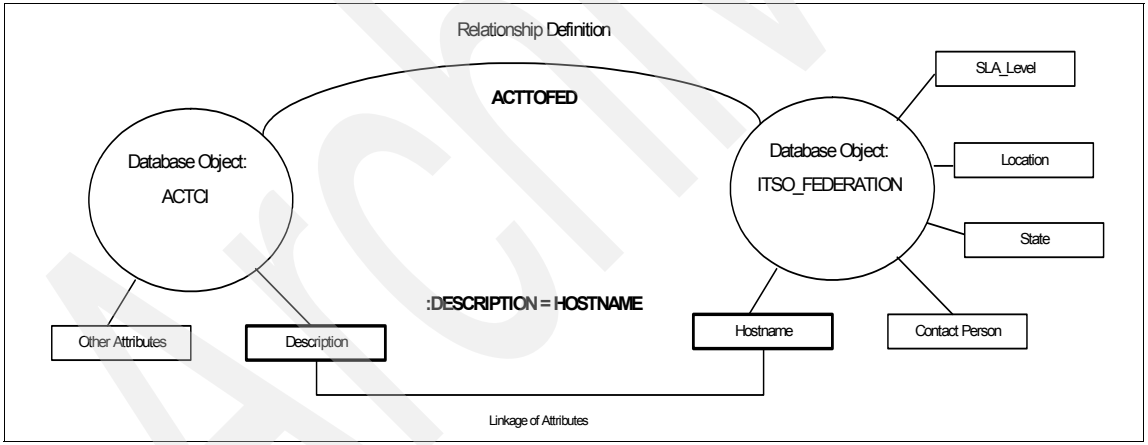


Figure 6-38 Relationship definition overview

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. 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.

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 will bring up the database definition for this field. Figure 6-39 shows this procedure for the Description field in the Actual Configuration Items application.

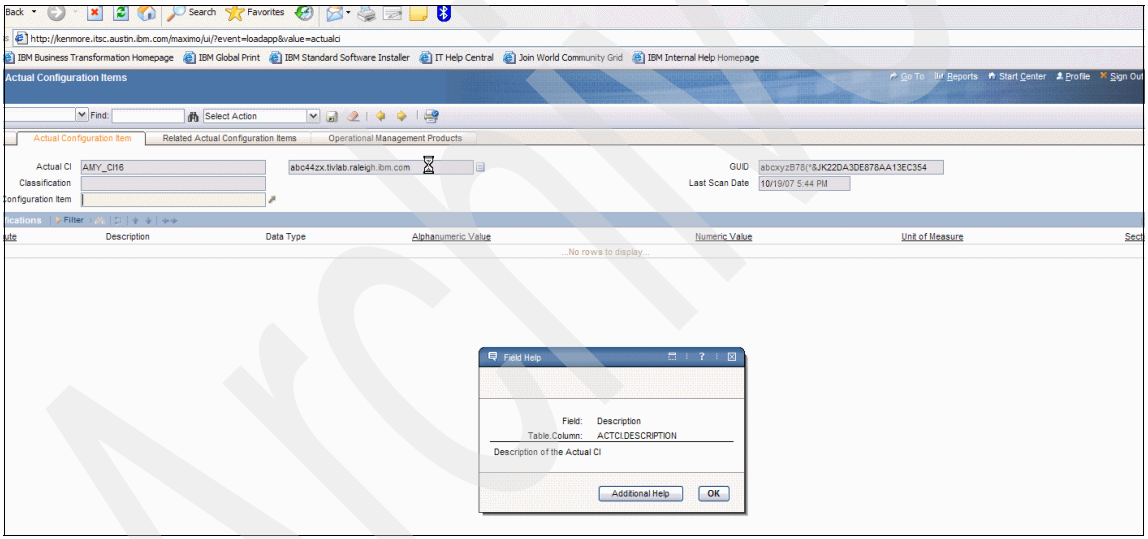


Figure 6-39 Database definition for the Description field

The pop-up dialog shows that the field reflecting the host name of the Actual CI is actually point to the ACTCI.DESCRIPTION attribute in the database. This is the attribute that we link to the HOSTNAME column of our federated database table.

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 6-40).

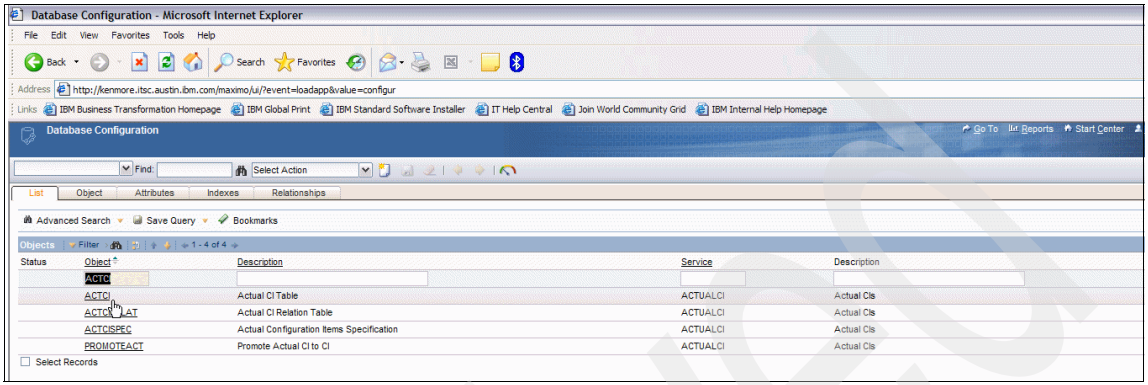


Figure 6-40 Searching the Database Configuration for the ACTCI object

Select the **Relationships** tab and you can see all existing relationship definitions (Figure 6-41).

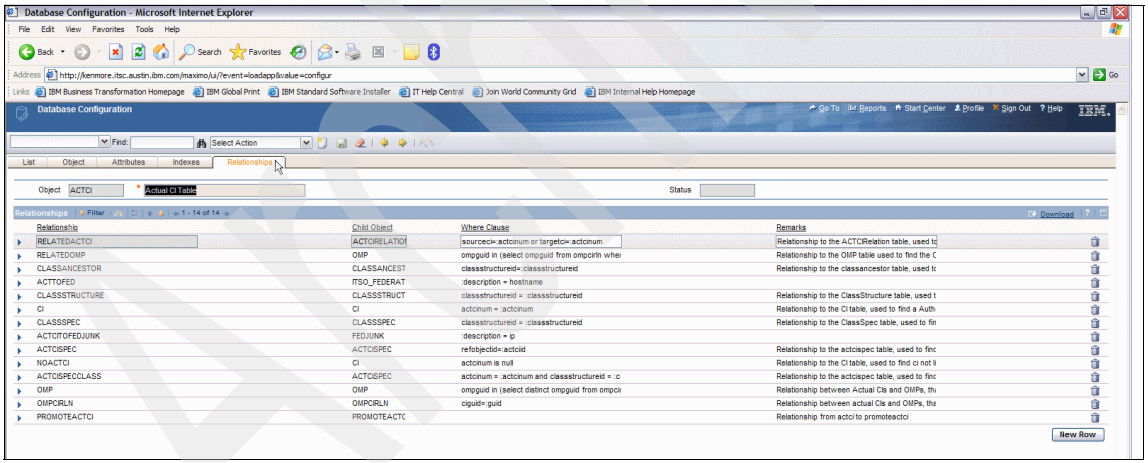


Figure 6-41 Relationships tab

Click the **New Row** button in order to create a new relationship definition (Figure 6-42 on page 175).

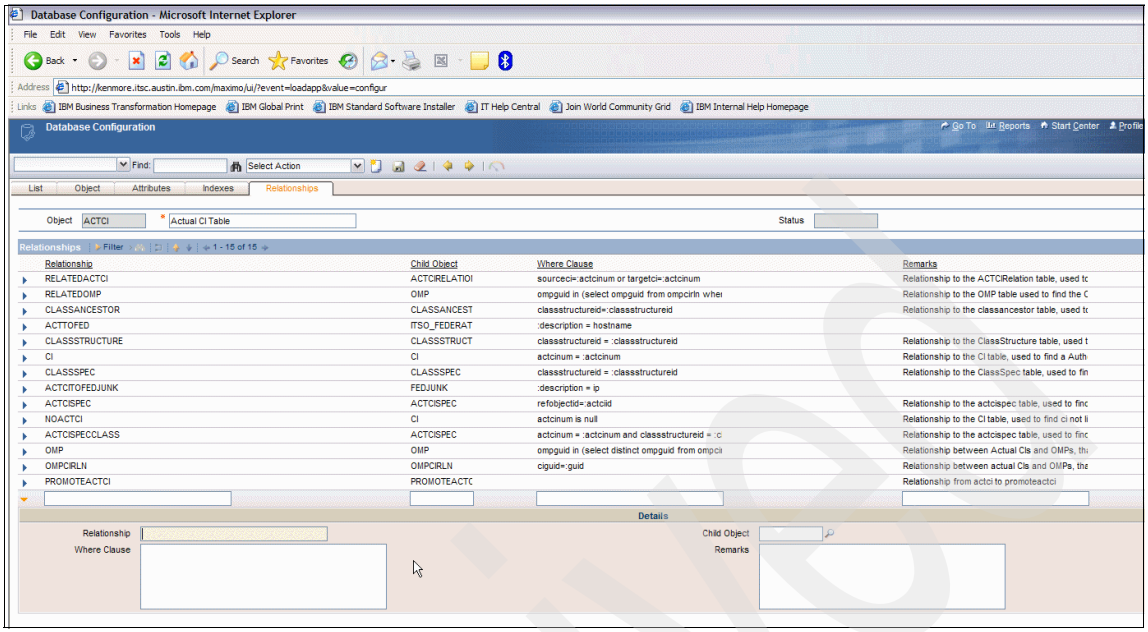


Figure 6-42 Creating a new row in the relationships definition

You now have to create the relationship definition by linking the DESCRIPTION attribute of the ACTCI MBO to the HOSTNAME attribute of the ITSO_FEDERATION MBO (Figure 6-43).

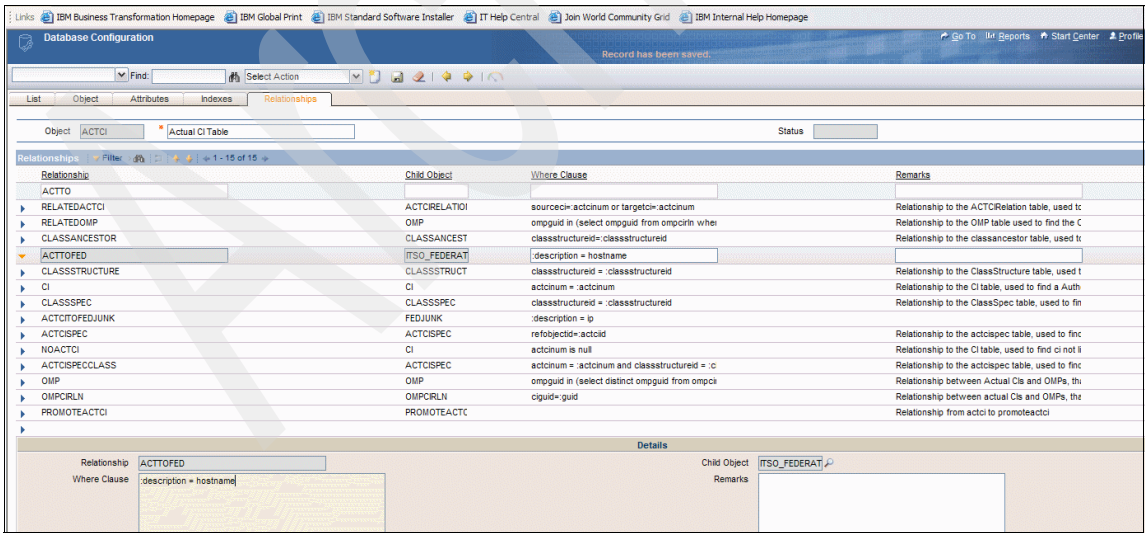


Figure 6-43 Creating the relationship

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.

6.6 Create 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. 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 6-44).

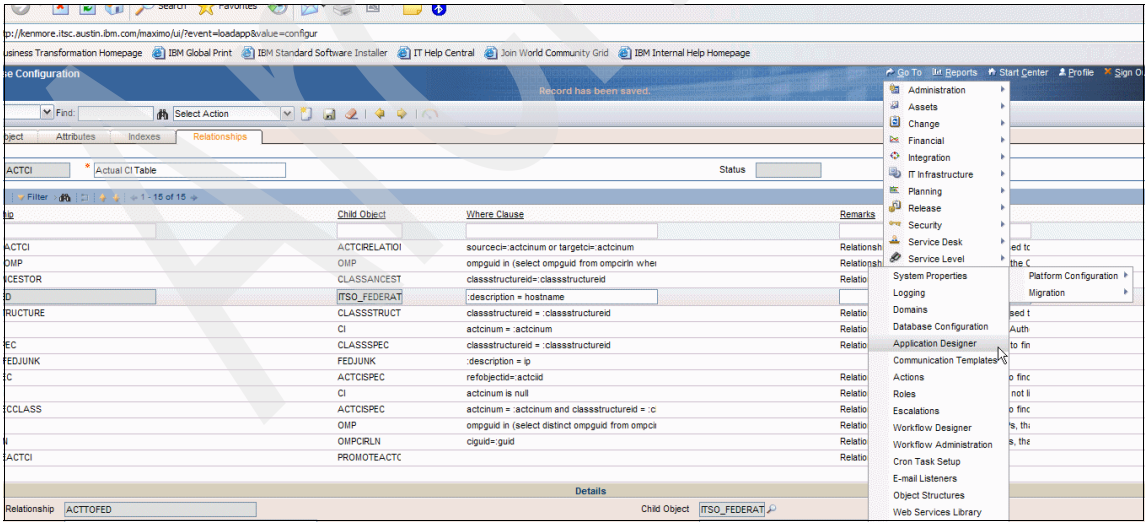


Figure 6-44 Launching the Application Designer

Search for the Actual Configuration Items application (Figure 6-45).

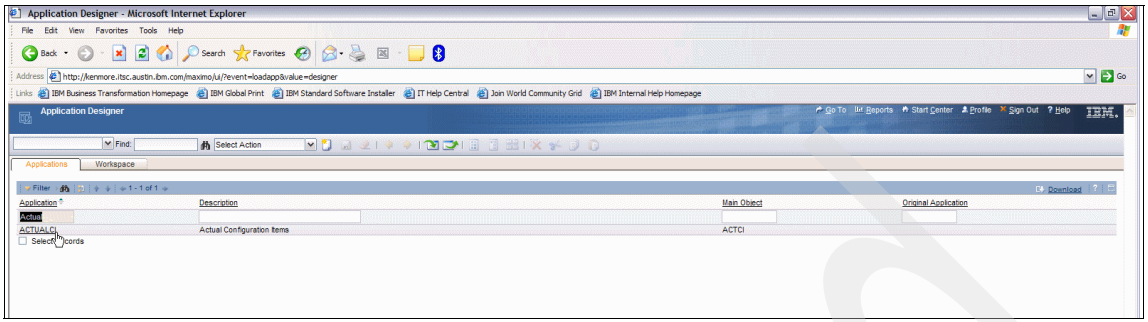


Figure 6-45 Searching for Actual CI application

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 6-46).

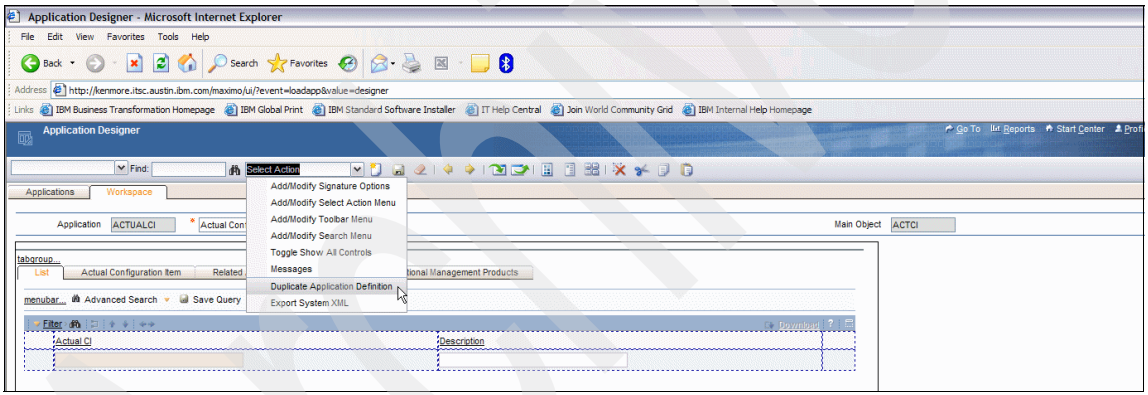


Figure 6-46 Selecting Duplicate Application Definition

This will bring up the **Duplicate Application** dialog (Figure 6-47).

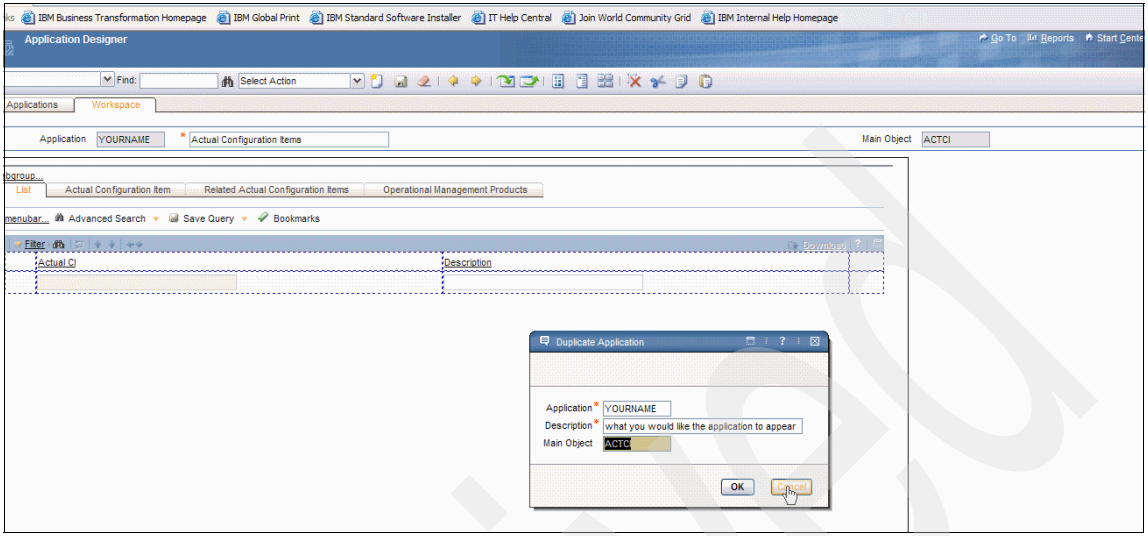


Figure 6-47 Duplicate Application dialog

Give your application a name and description. The description is what actually appears 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 6-48 on page 179).

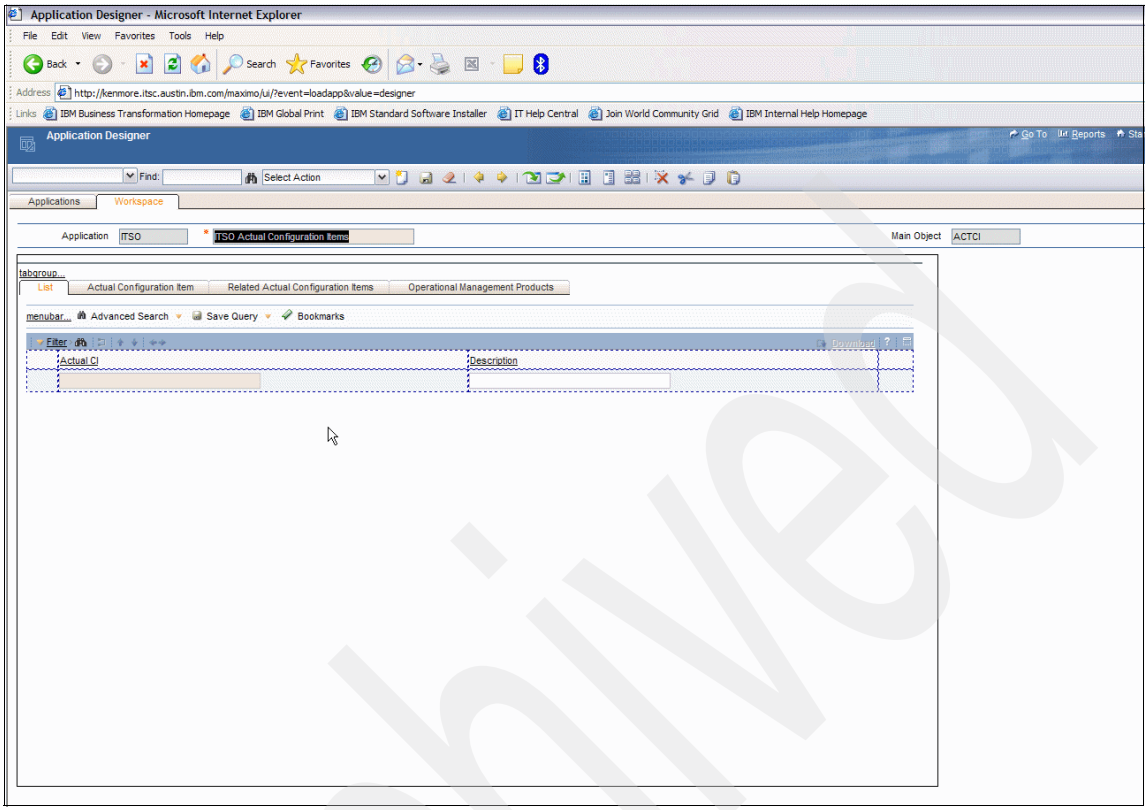


Figure 6-48 Modifying the user interface of the application

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.

Bring up 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. This will bring up 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.

Right-click the **Textbox** label, this will bring up a properties dialog (Figure 6-49).

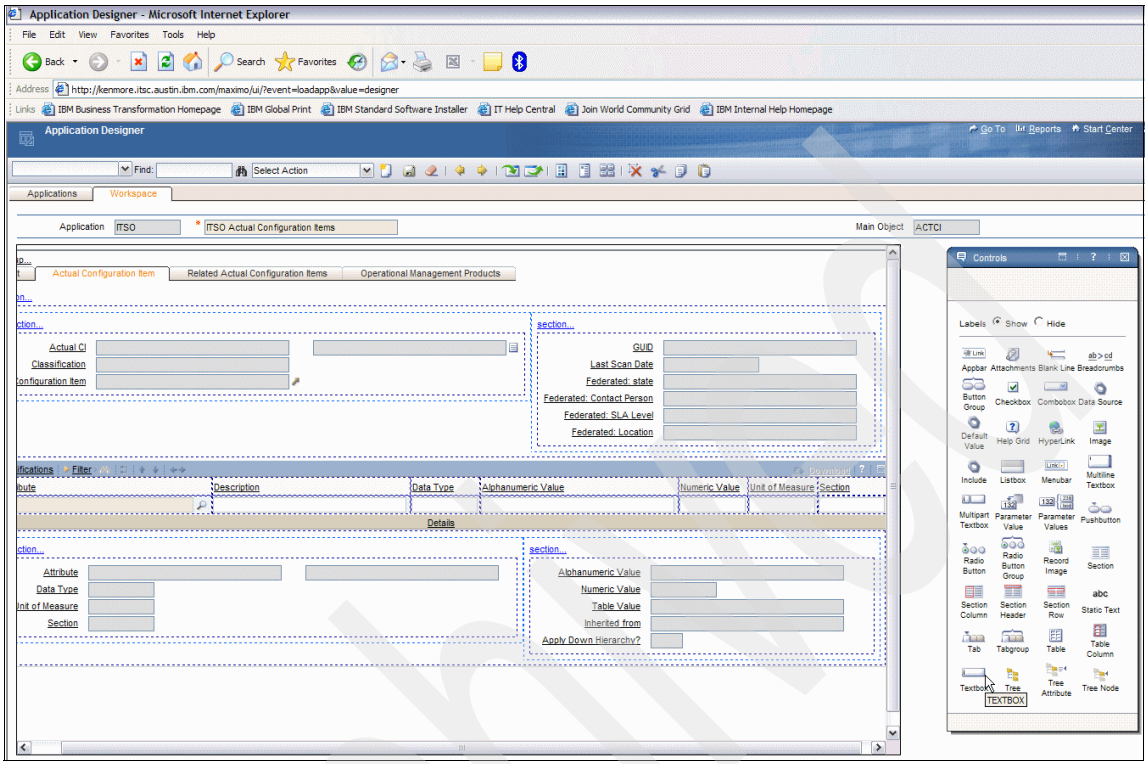


Figure 6-49 Properties dialog selection

Selecting **Properties** will bring up the dialog where you can define specifications for this field, for example, to resolve the binding to the database (Figure 6-50 on page 181).

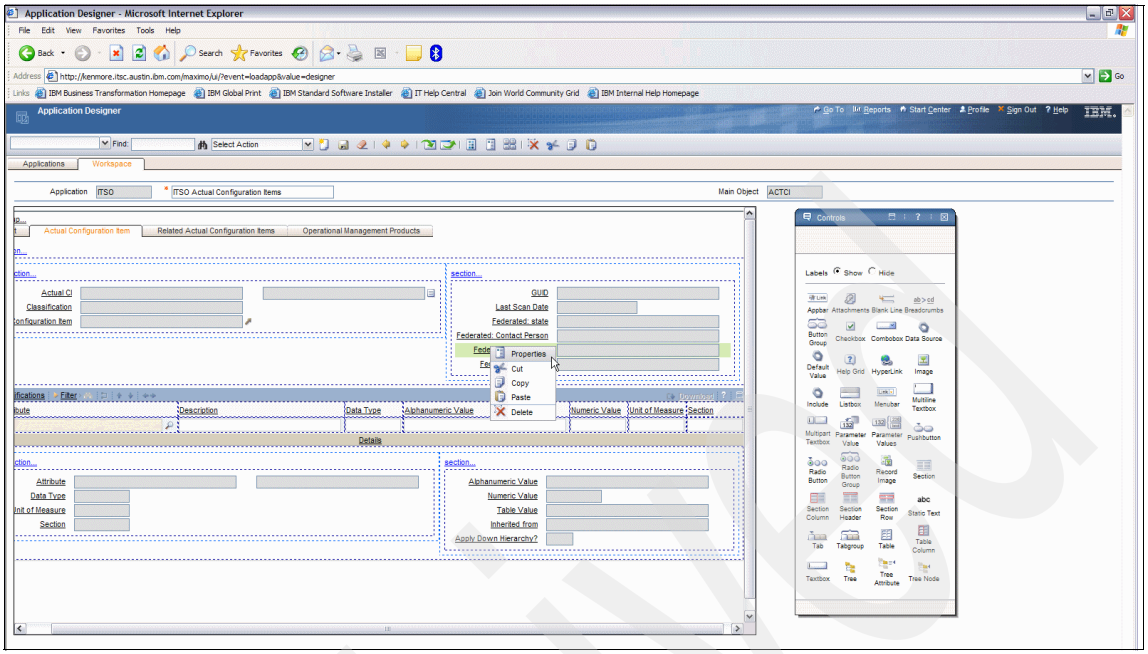


Figure 6-50 Properties dialog

In the Label specification, specify a name for the field as you want to present it to the users of the application. In Figure 6-50, 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.

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 you application. We added four attributes in total using the same approach as described above (Figure 6-51).

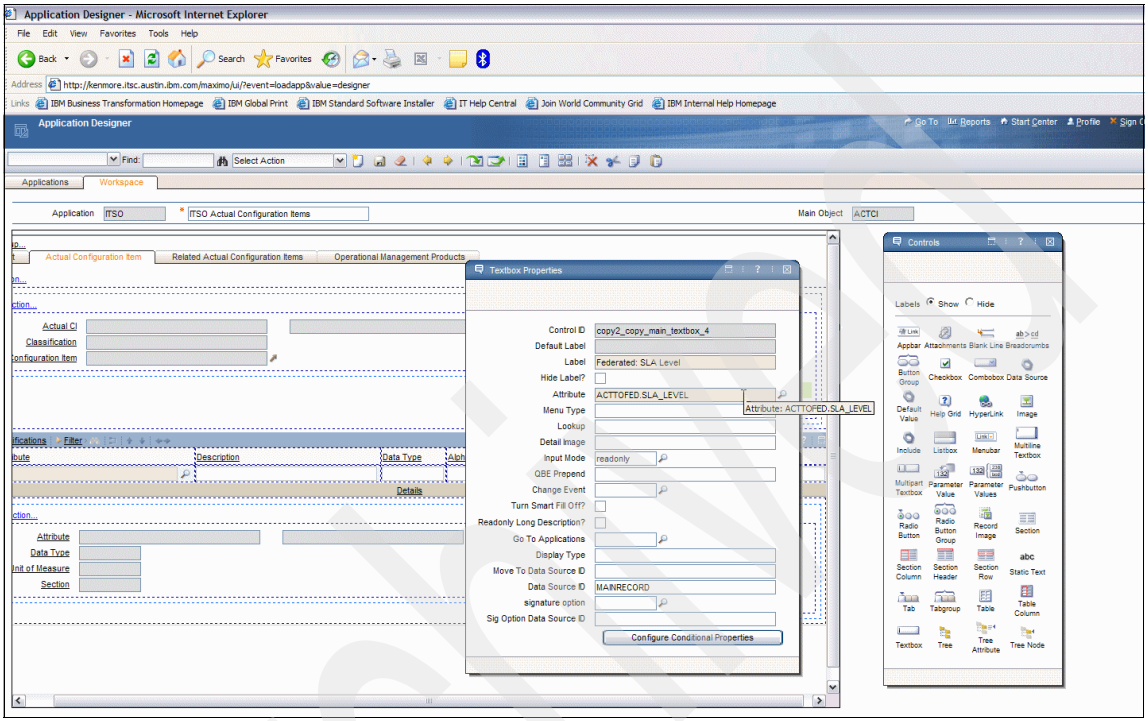


Figure 6-51 Added attributes

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 definition. Check your relationship definition.

You are now ready to use your new application. Your configuration setup is completed.

6.7 Use the new application

We launch our application by using the CCMDB Web User Interface Go To menu. We select **IT Infrastructure** → **ITSO Actual Configuration Items application** to launch the ITSO application (Figure 6-52).

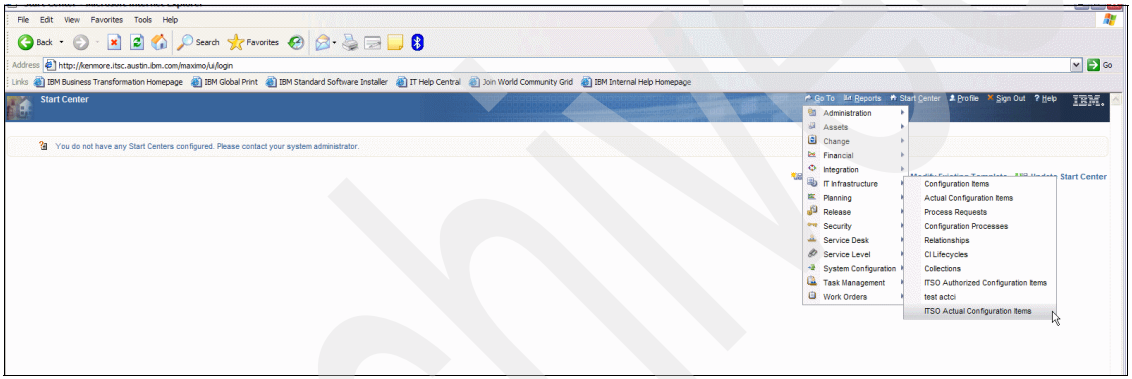


Figure 6-52 Starting the application

Since 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 6-53 on page 184).

Note: 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.

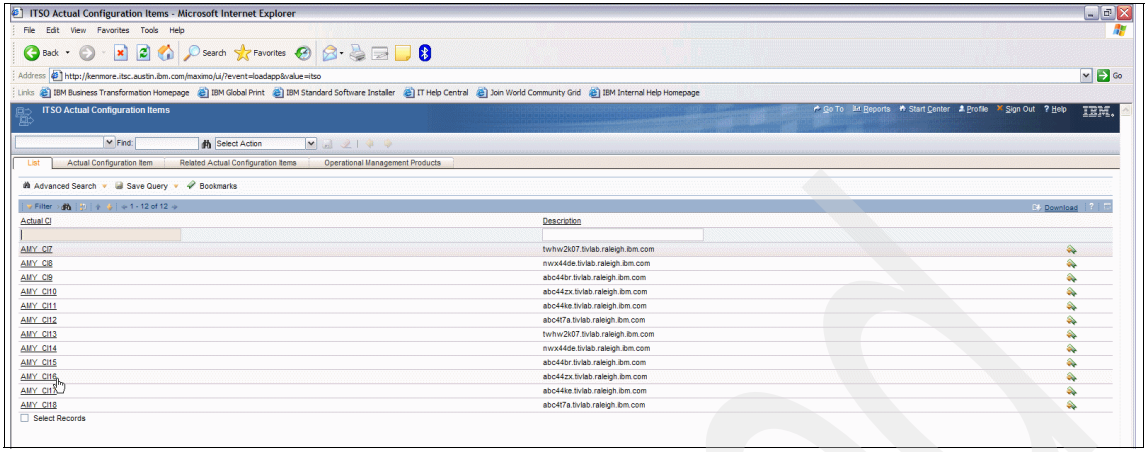


Figure 6-53 Viewing list of Actual CI data

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.

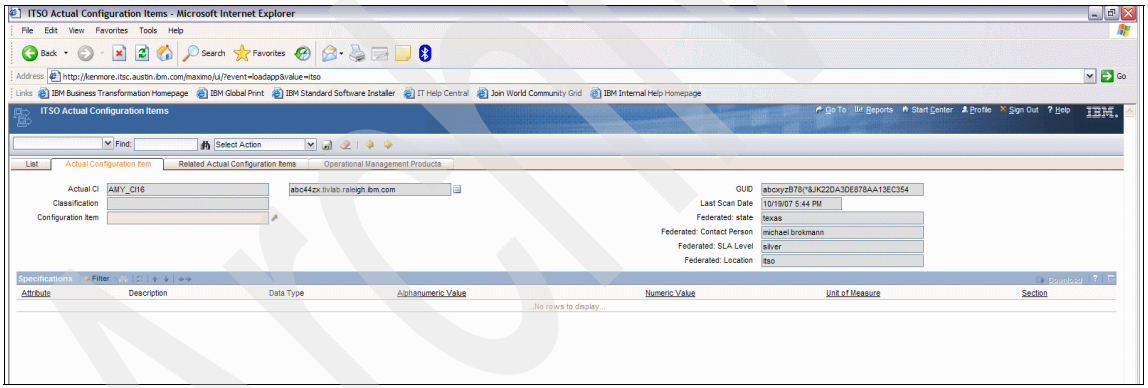


Figure 6-54 Viewing details contained in a federated data source

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 CCMDB database.

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 6-55 on page 185).

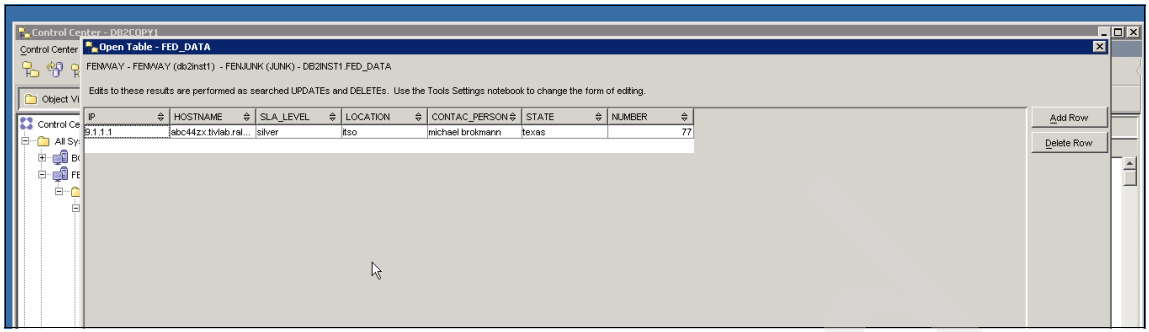


Figure 6-55 Changing data in DB2 Control Center

After changing the data in the federated database, the modified data shows up in the application user interface.

Note: 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.

6.8 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 new and different data items is quite simple and straightforward.

Part 3

CCMDB Process Engine and PMPS

7

Process flow technology

The main purpose of any CMDB implementation is to support IT service management processes. Processes require data in order to accomplish their work. The IBM CCMDB solution is a combination of data and process layers. Change and Configuration Management process templates are delivered as part of the CCMDB V7.1 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.

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

Recently, a set of best practice guidelines for IT service management processes has been documented in the ITIL library of documents. IBM has taken this best practice documentation and extended it based on its own 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.

For a deeper insight into ITUP and the ITUP Composer tooling, please see Chapter 3, “IBM Tivoli Unified Process Composer process mapping and design” on page 23.

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.

Figure 7-1 is 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.

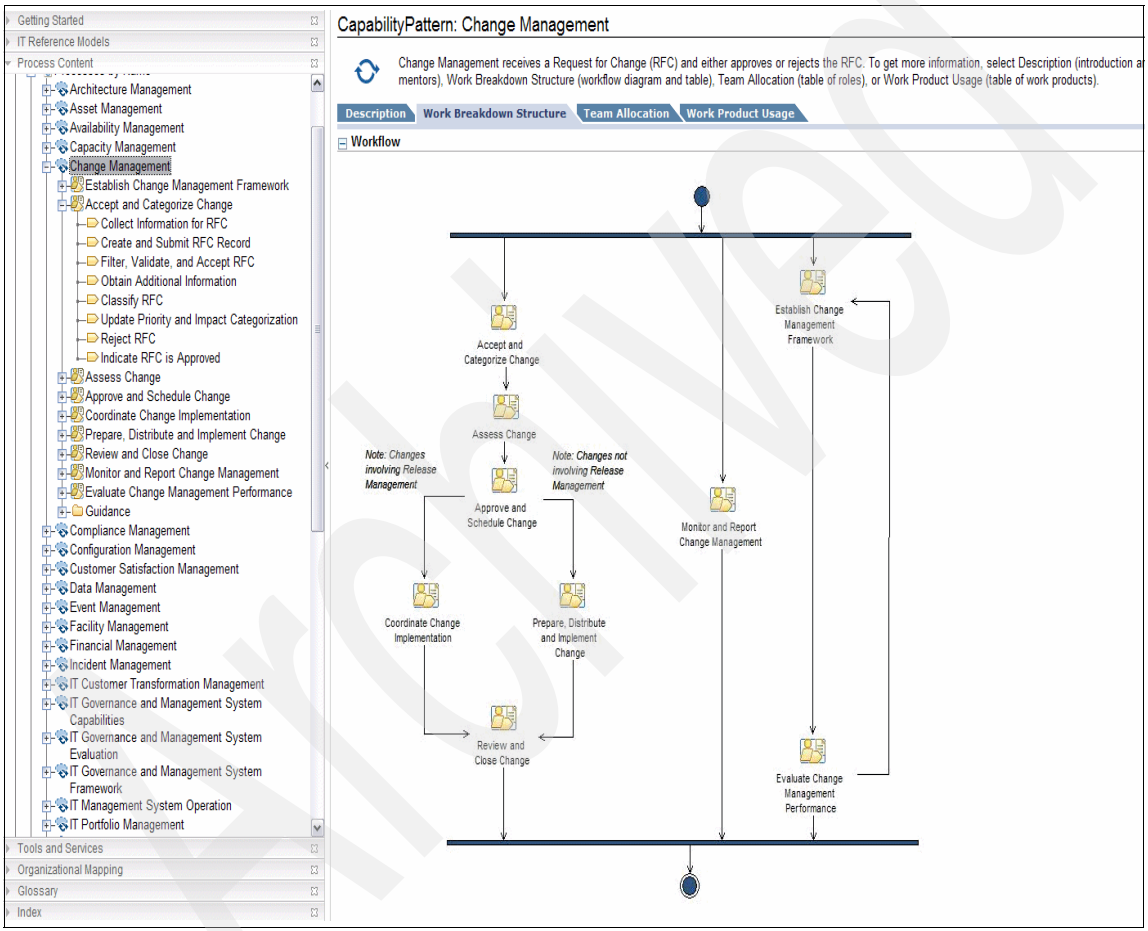


Figure 7-1 ITUP Change Management activities

Each of the steps or activities is described in further detail. Figure 7-2 on page 191 is a drill-down view into the Accept and Categorize Change activity of the overall Change Management flow.

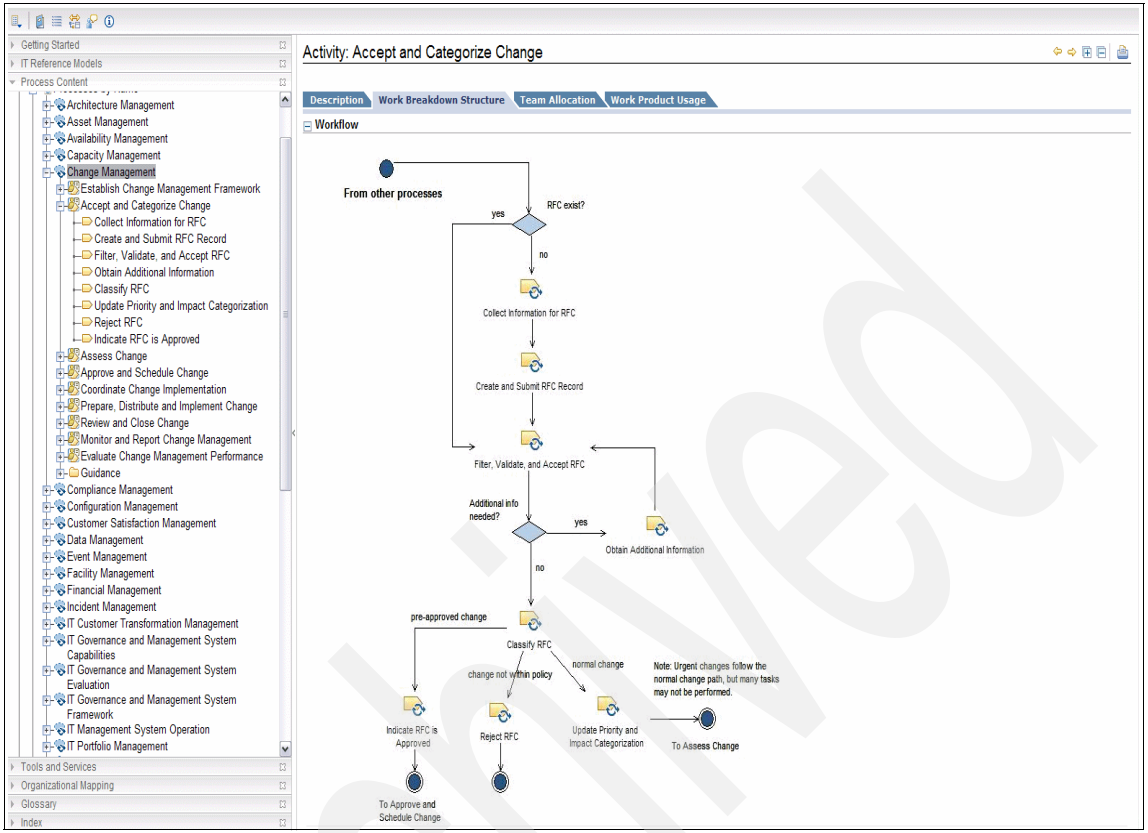


Figure 7-2 ITUP Change Management process request

The tasks of this activity are aligned to support the generation of a process request into Change Management.

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 different default process templates for Change and Configuration Management. Nevertheless, theory usually is different 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 are different in each IT environment and depend very much on the specification or classification of the process request.

A Change Management process will differ depending on its nature. Different people will be involved, different steps have to be taken for assessment, for approval, 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 different people depending on the classification of a change request.

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

The workflow technology must be flexible enough to adapt to different 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 different 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 different 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 different process permutations in order to satisfy real live requirements. Figure 7-3 reflects the necessity of a flexible and adoptable approach of workflow technology.

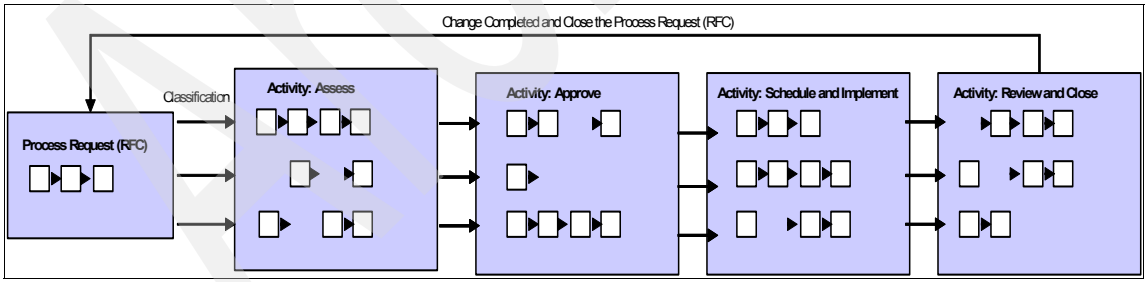


Figure 7-3 Flexibility in process design

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 different process 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 is different from a process template for password change or deploying a complex J2EE application. As reflected in Figure 7-3 on page 192, the tasks inside the activities usually differ depending on the classification of the request. They can be modified before the template gets applied to the request in case no predefined template matches the need. The end-to-end flow can even be changed while the process is already in progress in case an unexpected situation occurs. In some cases, a whole activity might need to be skipped or automated transparently for the user. An example of this would be the approval activity of a pre-approved change type. A process request is closed once the last task of the final activity is completed.

In this chapter, we describe the process flow technology components of the IBM CCMDB solution. We highlight how the solution addresses the flexibility and adoptability requirements mentioned.

Although there is one process runtime and workflow technology inside the solution, a combination of different administrator facing applications are used to model and customize end-to-end process flows of different IT service management solutions.

In 7.1, “Technology overview” on page 194, we describe the different technology areas and applications that can, but do not have to, be used in modeling and executing a process flow, while in 7.2, “An end-to-end example” on page 205, we use an example of a default process template for Change Management in order to explain how they are jointly used in the CCMDB solution.

Please be aware that the intention of this chapter is not to provide a detailed user manual of how to use and apply the technology; its goal is to explain how the technologies are used together to represent an end-to-end process so that administrators of the solution know where and how to modify the solution to fit their environment.

7.1 Technology overview

A combination of different 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 Job Plan, a Work Order, a Workflow, as well as an Action or Action Group, and will touch on Process Requests as related to the overall process flow.

Figure 7-4 relates an example of an overall abstracted end-to-end process model of Change Management to the technologies used for the definition and Figure 7-4 implementation of the flow model. As there are patterns of which technology to use in which use case, there are definitely exceptions to the rules.

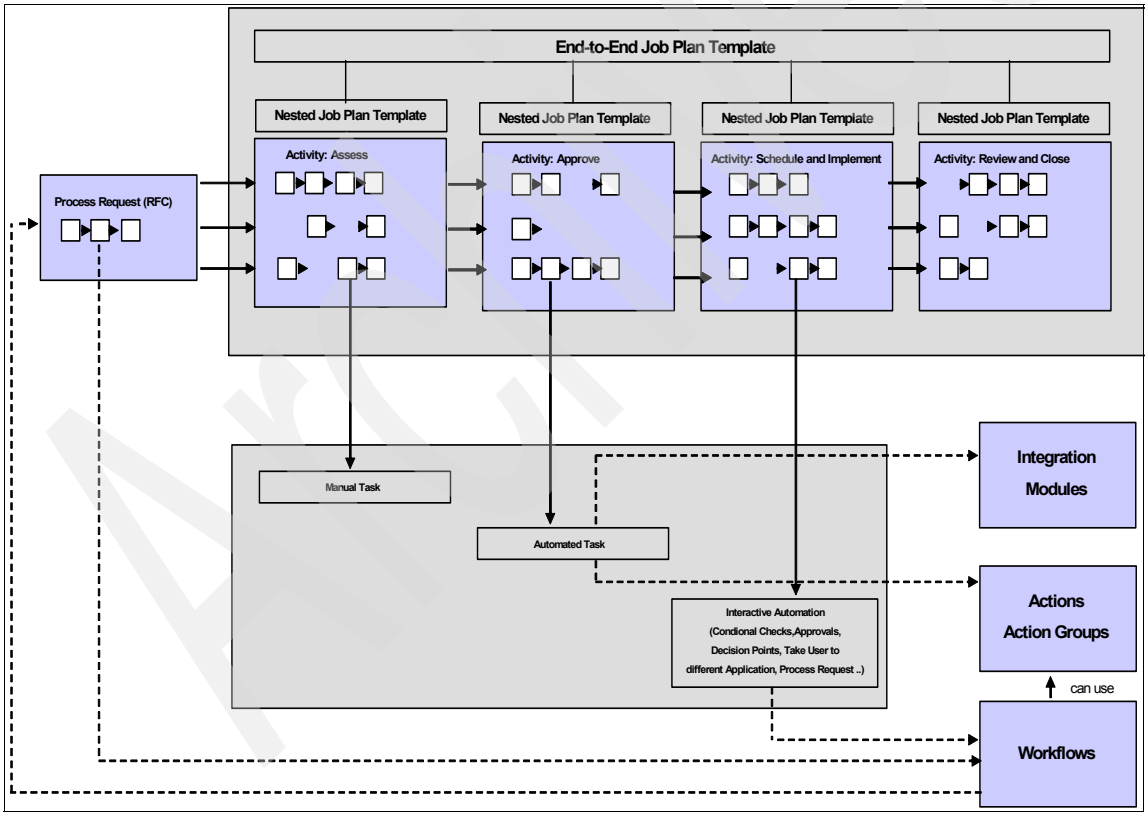


Figure 7-4 Process flow technology interaction

In this section, we explain each of the technologies and major roles in the overall implementation, while in 7.2, “An end-to-end example” on page 205, we walk through a concrete example provided in the CCMDB default implementation.

7.1.1 Process request and work order

Each process flow has its origin in a process request. A process request is submitted and classified by a requester. The Process Request application is used for creating a process request. A process request is always necessary as the starting point for a process flow instantiation. Although a process request can be considered as being part of the overall process flow, technically there is a separation between a request and the activities of the respective process.

In Figure 7-5 on page 196, the process request is classified. Once the process request is defined, it has to be submitted by the requester and be approved or rejected by someone in a responsible role, such as the Change Manager.

Once 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. In both cases, the classification of the process request is a key factor to determine the right Job Plan process template to be applied.

Once 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. Once 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.

For a step by step example of a Change Management end-to-end example, from creating to closing a process request, please see 8.2, “Change Management Process Manager” on page 243.

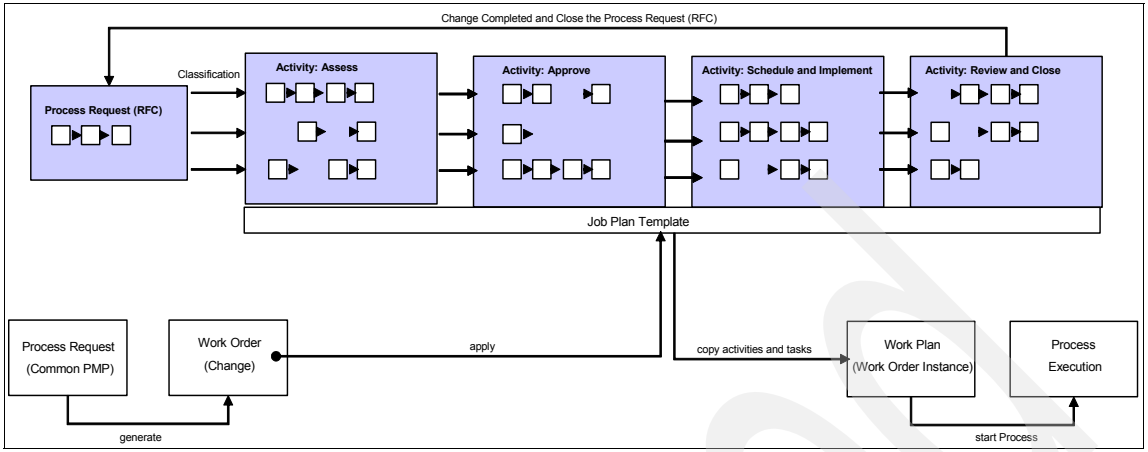


Figure 7-5 Technical flow from process request to Work Order Plan

The process request classification makes use of 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 Item(s) targeted for this change can be specified in the process request (Figure 7-6).

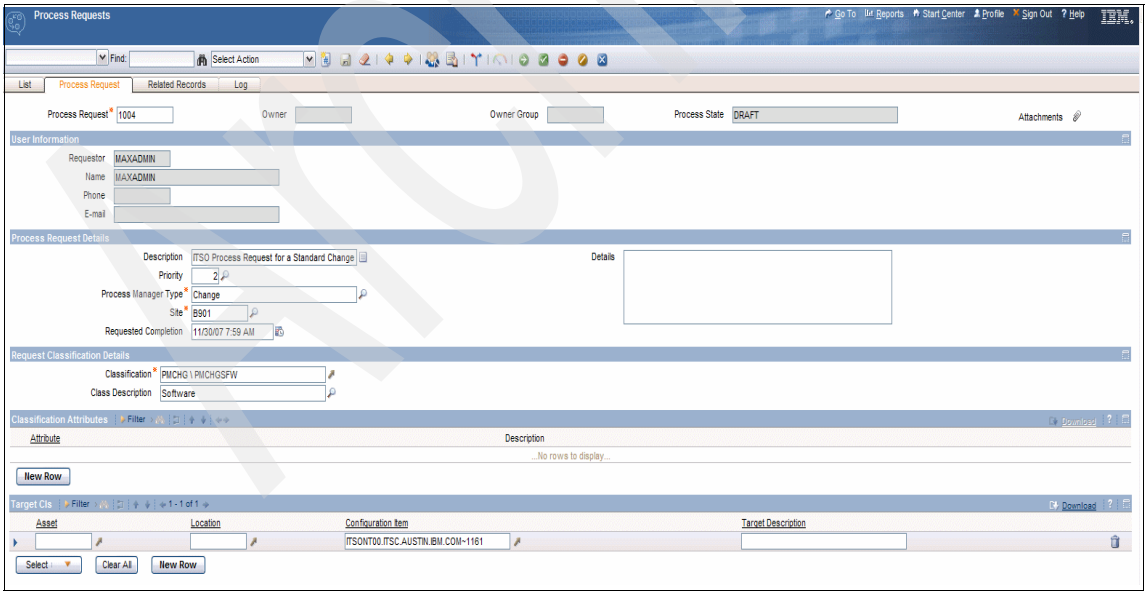


Figure 7-6 Process request

Figure 7-6 on page 196 shows an example of a process request to Change Management, including its classifications. A configuration item has been specified as well. Remember that a process request has to be submitted and approved before a Work Order object gets generated.

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.

7.1.2 Job Plan

Job Plans are used to represent the end-to-end definition of a process flow. It is a detailed description of the work to be performed once 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. Once 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. One of the main differences 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: We are referring to the specific workflow facility or application of the CCMDB rather then the general capability to provide a workflow engine to host end-to-end process workflows. The overall workflow engine leverages the Job Plan as well as the Workflow application to define and run automated service management processes.

In Figure 7-1 on page 190 and Figure 7-2 on page 191, we show an example of default Change Management activities and tasks.

Job Plans are used to define the activities and tasks of a specific end-to-end process flow. They are used to provide a description of the overall process. Please 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 an end-to-end 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 like 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 while 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 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 get executed by a workflow engine.

In Figure 7-7, the last three rows show the default top level process definitions. They contain the top-level activities of the process flow. Drilling into the example of the Standard Change Job Plan, you see the activities shown in Figure 7-8 on page 199 defined.

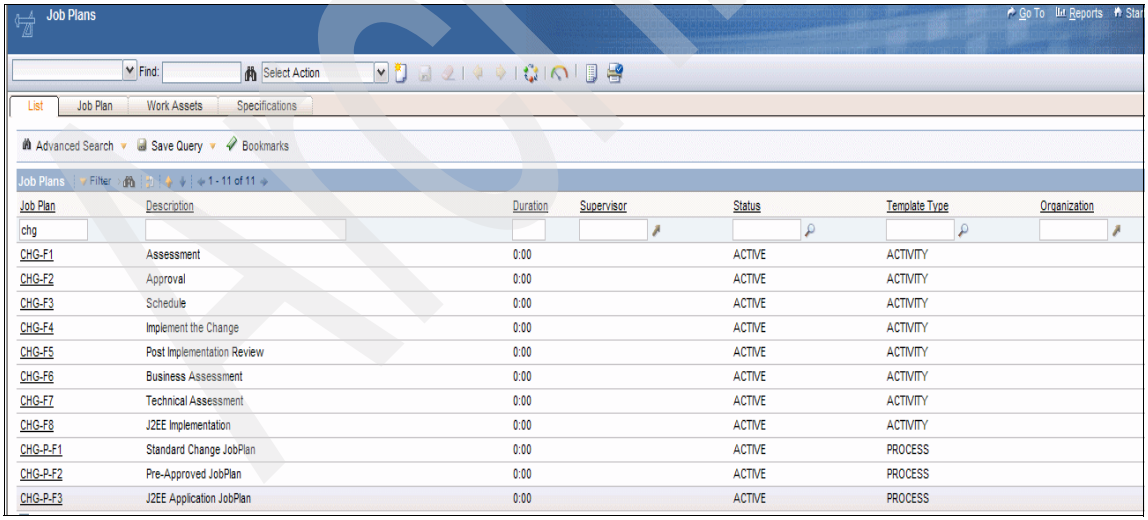


Figure 7-7 Change Management default Job Plans

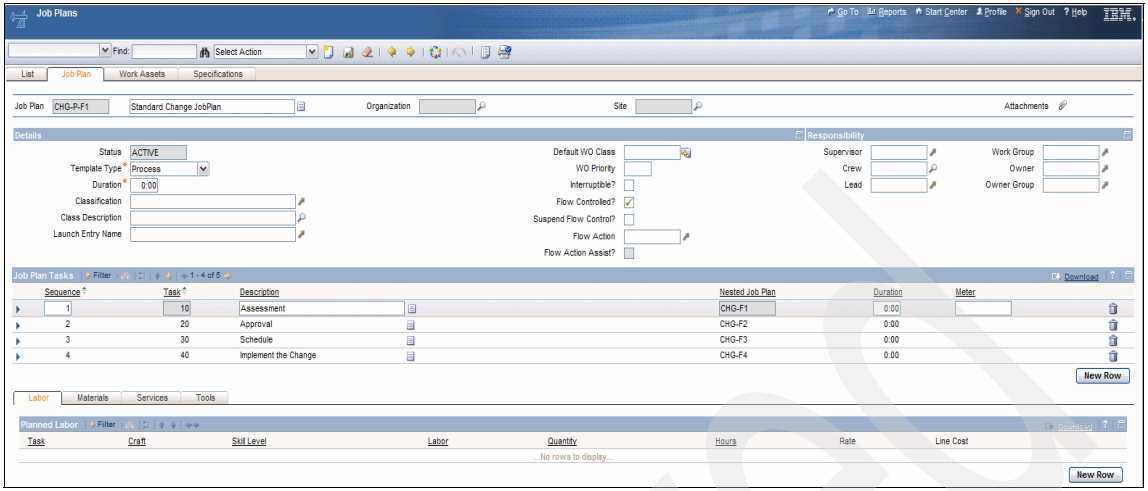


Figure 7-8 Standard change Job Plan activity definitions

Assessment, *Approval*, *Schedule*, *Implement the Change* and *Post Implementation Review* (you need to scroll to the next page in order to see the activity definitions) are the major activities defined inside the top level Job Plan definition for the Standard Change Job Plan.

They are defined as tasks of the top level Job Plan. Each of the activities is defined as a nested Job Plan. The nested Job Plans are of type *Activity*.

Figure 7-9 shows the nested Job Plan for the *Post Implementation Review* activity.

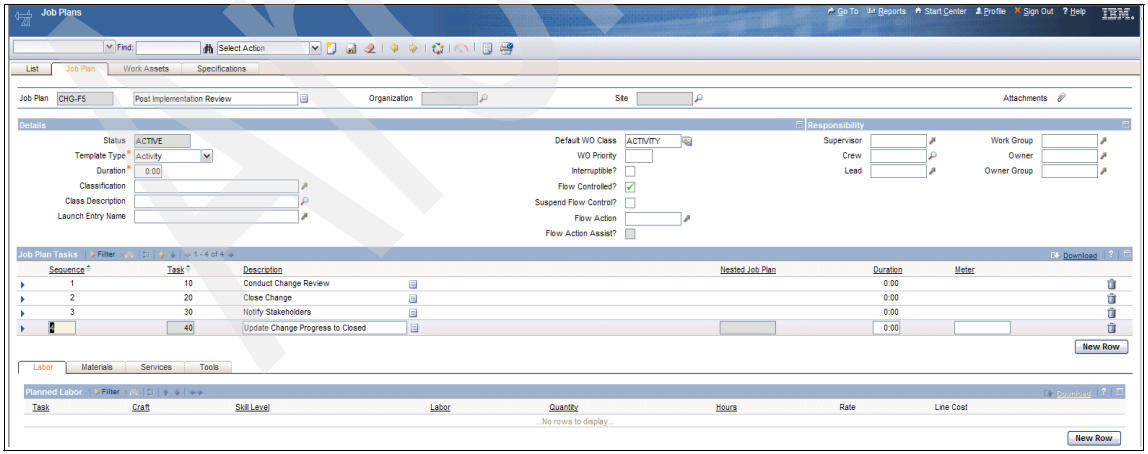


Figure 7-9 Nested Job Plan for Post Implementation Review activity

In this case, the task definitions are not referring to nested Job Plans but define the steps to be done inside the activity.

These tasks or steps can be manual tasks, automated tasks by running an action, automated tasks by running a program that is executing a process on an external system (for example, an operation management product like Tivoli Provisioning Manager or Tivoli Configuration Manager), or can be tasks that interactively guide you through a series of steps. These different types of task definitions are reflected in Figure 7-4 on page 194. In order to define tasks as manual, automated, or interactive steps, the Job Plan templates leverage the Action / Action Group and Workflow facilities of the CCMDB solution. In 7.2.2, “Process flow definition” on page 212, we explain in more detail with some examples of how you define, inside a Job Plan task definition, a call out to an action or workflow definition so that at the time of the runtime of the Work Order Plan, the tasks get automated or semi-automated.

In Figure 7-22 on page 212, there are four tasks defined in a sequence on which a responsible owner of the task has to work. You see that the final task in the *Post Implementation Review* Job Plan is to update the overall Change progress to close. This also closes the initial process request that has opened the Change Work Order object.

A Job Plan represents a high level description of the process as well as a detailed definition of the steps to be performed inside each of the activities. It is flexible enough to adopt to organizational requirements while defining a process template as well as changing the steps at runtime.

A Job Plan is a hierarchal structure using nested structures. The depth of the structure is not limited by the technology. A sequence of steps is defined with the ability to define dependencies between the steps. Each task defines the tasks that need to be finalized before it can start itself. A task can define one or multiple predecessor tasks that have to be finalized before it starts executing. We show how to define the predecessors of a task in 7.2.2, “Process flow definition” on page 212.

Tasks of a nested Job Plan hierarchy can run in parallel or sequentially, depending on the predecessor definitions. This allows you to define a parent child hierarchy as well as a way to define sibling relationships between the different 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 7-10 on page 201 illustrates an example of a dependency tree of nested Job Plans and a dependant sequence of task definitions.

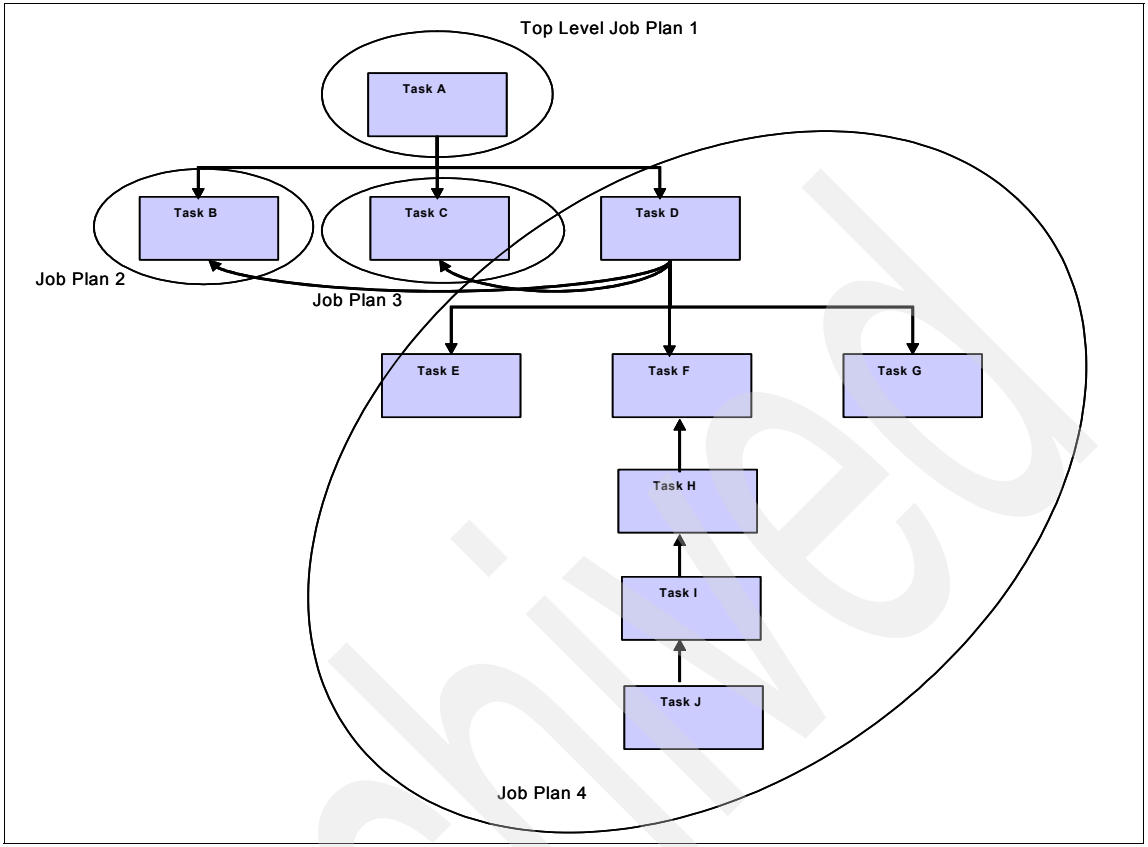


Figure 7-10 Nested Job Plan hierarchy and task dependencies

Figure 7-10 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 the end-to-end 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 predecessor(s). Tasks can run in parallel or sequentially.

There are different 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. Please refer to 7.2.2, “Process flow definition” on page 212 for a more detailed description of how to link the various technology components together.

7.1.3 Workflow

In 7.1.2, “Job Plan” on page 197, we explain that Job Plans are used to model the end-to-end process flow definition. The CCMDB V7.1 solution provides its out of the box best practice examples of Change and Configuration Management process definitions using Job Plan templates.

The workflow facility is another way to define and represent process flow definitions inside the system. Nevertheless, the overall end-to-end flow definition is defined in Job Plans while workflow definitions are leveraged from within tasks of the overall Job Plan. Workflows can be initiated from a task in progress.

The following list provides the main reasons and requirements to call out from a Job Plan task to a workflow:

- Interactive automation with a user: If you need to provide some interaction with the person in charge of the respective process step, for example, to make a decision between different choices to be taken, an approval decision, or the forwarding to a different 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 some data before a decision can be taken 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 so called *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 approvals: Workflows are the preferred way to obtain a positive or negative approval 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 a different 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.

Workflows can call actions or action groups for automating some of its execution steps.

As a general rule of thumb, you call out from a Job Plan template task to a workflow if you have the need to guide the user through some interactive steps. While you use actions from within task definitions when running something in the background, you usually call out to workflows when you must run something in the foreground, such as an interactive dialog with a user.

Although there are some overlaps in the capabilities of what Job Plans and workflows can provide, there are some key differences. Workflows cannot be changed while in progress, so you must think about all the possible exceptions during the design phase. Job Plans (work order plans) can be even changed at runtime if needed for any reason. Job Plans are better suited to adopt to very detailed, type dependant use cases of concrete process scenarios. We do not discuss in detail the differences between the two technologies. but rather provide you with the main reasons when to use workflows within a task definition of a Job Plan template.

Workflows are modeled using a graphical designer called the workflow designer. Various example are given in 7.2.2, “Process flow definition” on page 212.

7.1.4 Action and action groups

If a step in a process flow is required to be automated, you can call out to an action or an action group by specifying the action inside the Job Plan template definition. Actions are usually a codified sequence of short logic. You use actions from within task definitions if you must run automation in the background.

An action can change the status of a task, 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. 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 7-11).

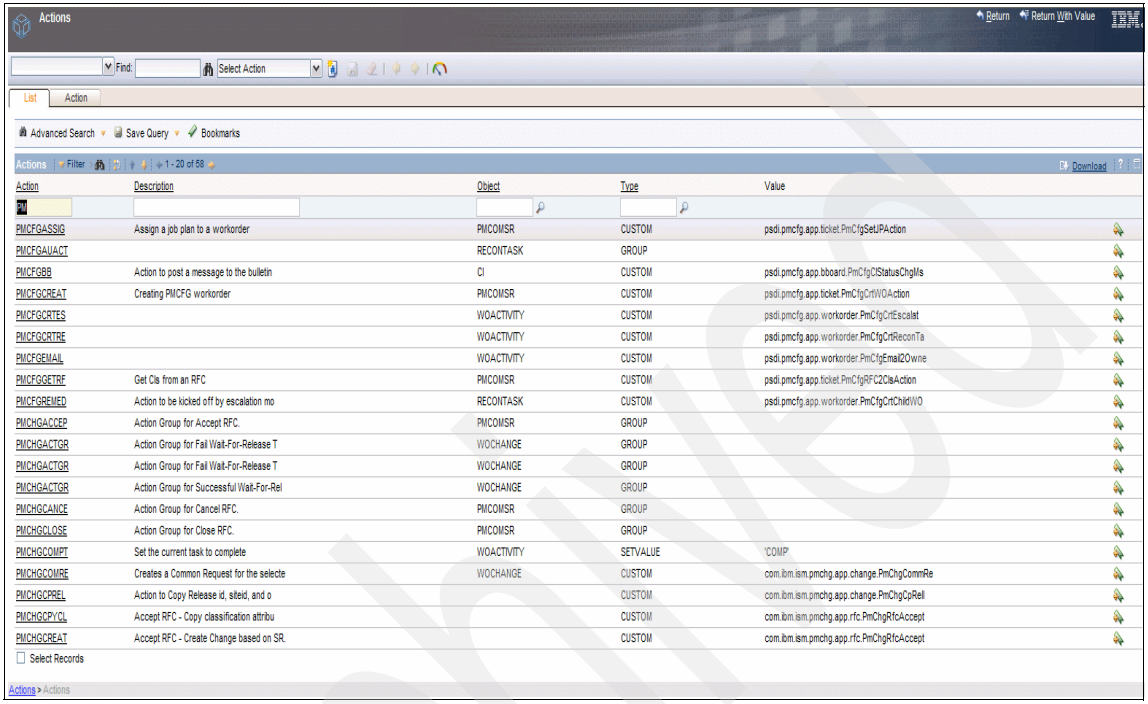


Figure 7-11 Listing of all actions provided by Process Manager Products

The PMCHGPROGTOASSESSGRP action group shows that two SETVALUE and one CUSTOM actions are used inside this action group. The custom action is using a Java class for its implementation. The SETVALUE actions set the value of the task and the status of the overall change process flow to a new status.

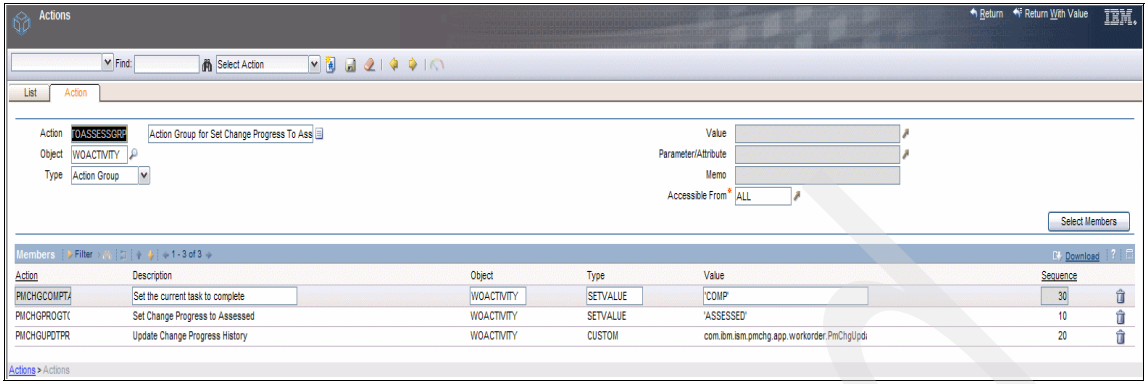


Figure 7-12 Action group example for Change Management

Compared to previous versions of the Maximo technology (a key technology in the CCMDB V7.1 solution), there are some changes in how end-to-end process flows are modeled. While in previous versions the Workflow technology has been primarily used to model end-to-end process flows, in Version 7.1, Job Plans are the primary entity to model the flows. Workflows are called out to in specific cases, for example, when an interactive dialog is required with a person taking a specific role in the process. The workflow technology can still be used as it has 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.

We have attempted to gain more flexibility in the process design of process variations of a specific process type in order to better adopt to real life requirements. Another reason is higher flexibility in changing the process model even at runtime. While you cannot change a workflow while it is in progress, you can change the definition of a Job Plan or Work Plan (in case the Job Plan has been instantiated). When designing a workflow in the Workflow Designer application, you therefore have to think of all possible exceptions that can happen at runtime. Using Job Plans or Work Order Plans, you are able to react and change the process flow if needed.

7.2 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 now provide concrete examples of default Change Management process flow definitions provided by the CCMDB V7.1 solution and explain where the interaction of the technologies are applied.

7.2.1 Process request and work order

In 7.1.1, “Process request and work order” on page 195, we show an example of a process request for Change Management, also known as a Request for Change. Once the request is submitted, the status of the process request is changed from *Draft* to *Queued*.

The submit request is using workflow technology in order to validate the request and apply the new status. The interaction of a process request and the workflow technology is highlighted in Figure 7-4 on page 194.

The process request submit action gets the name of the workflow to call from a system property called pmp.submit.workflow. This is the submit master workflow for all kinds of process request submissions, regardless of the process request type. You can find the system property in the System Properties application. If you search for “submit”, you will find the submit master workflow called ISMSUBMIT (Figure 7-13).

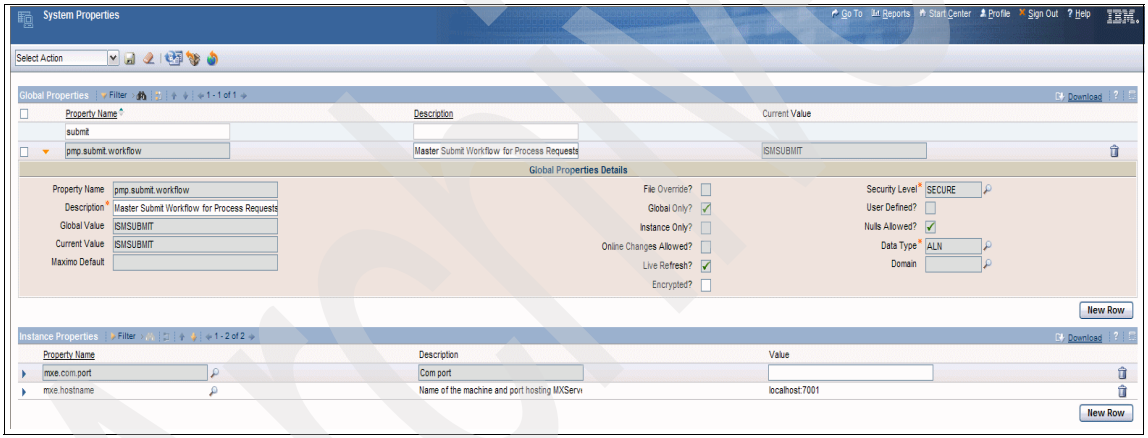


Figure 7-13 Submit Master Workflow System Properties

Using the workflow designer application, the ISMSUBMIT master workflow looks like the window shown in Figure 7-14 on page 207.

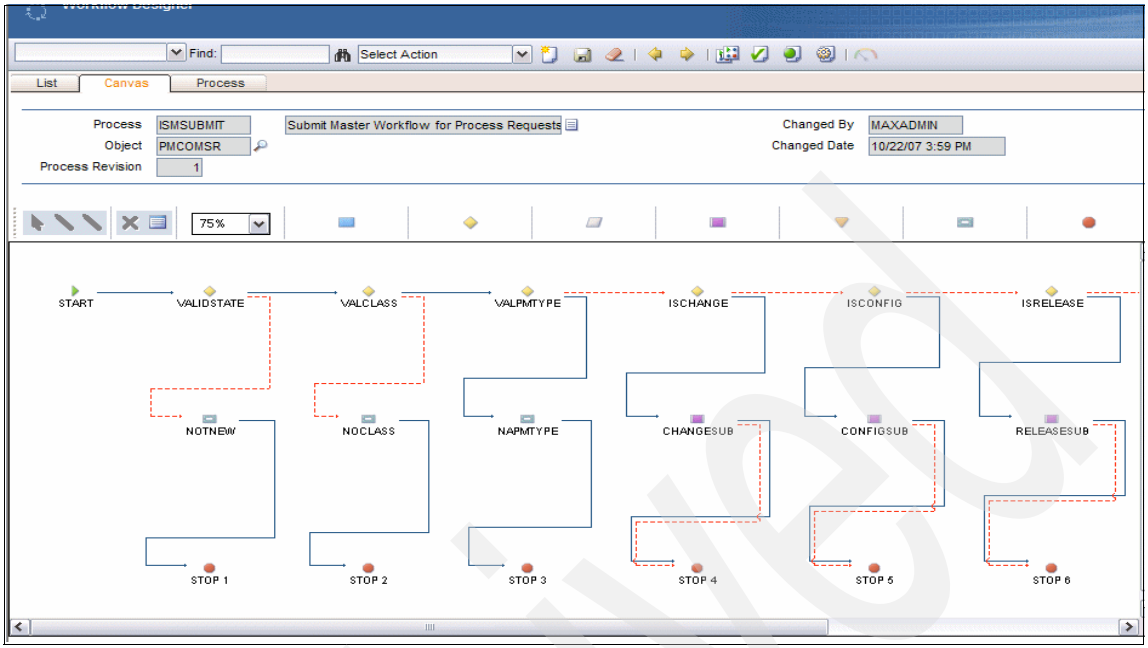


Figure 7-14 ISMSUBMIT master workflow

The top row of the workflow designer canvas window shows various condition nodes to validate various input parameters that have been specified in the process request window. It then figures out, based on the input parameters, which process type the process request is addressing.

The condition node ISCHANGE, for example, is checking if the process request is of type change (Figure 7-15).

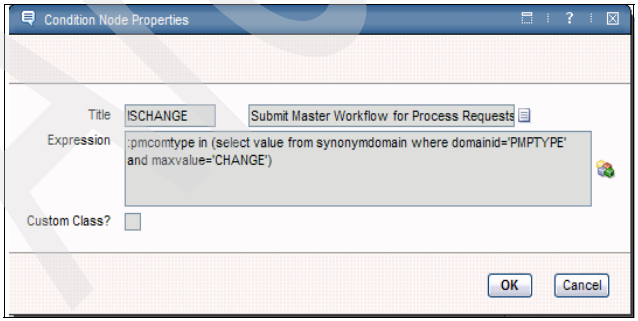


Figure 7-15 Workflow condition to check for process manager type

A simple expression is used to check if the Process Manager type equals CHANGE.

If the condition is true, a subprocess node called CHANGESUB is initiated. This is the sub-workflow that is submitting the request as a type change (RFC) (Figure 7-16).

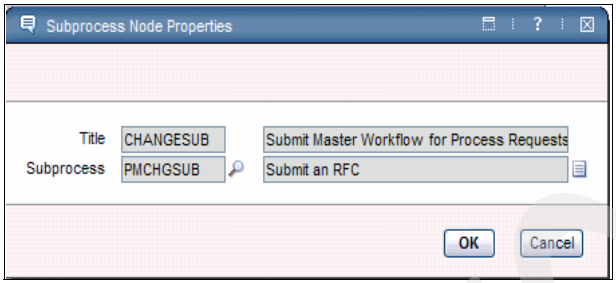


Figure 7-16 Subprocess workflow node

After the subprocess is initiated, the main workflow ISMSUBMIT is stopped. This is an example of a generic workflow making use of conditional checks to call more specified workflow routines.

Looking at the definition of the PMCHGSUB workflow in the workflow designer application, we can see the definition shown in Figure 7-17.

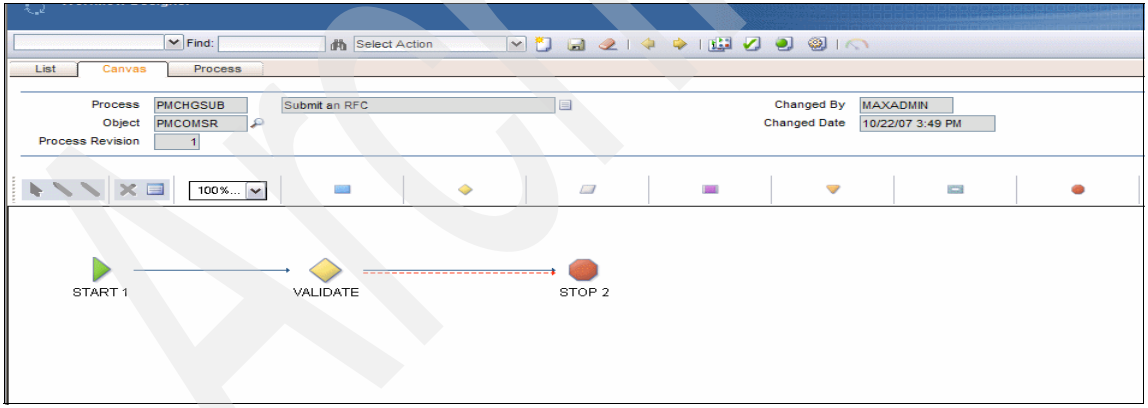


Figure 7-17 PMCHGSUB workflow

After an initial validation phase, the connector line between the VALIDATE and the STOP 2 node causes the action properties to execute (Figure 7-18 on page 209).

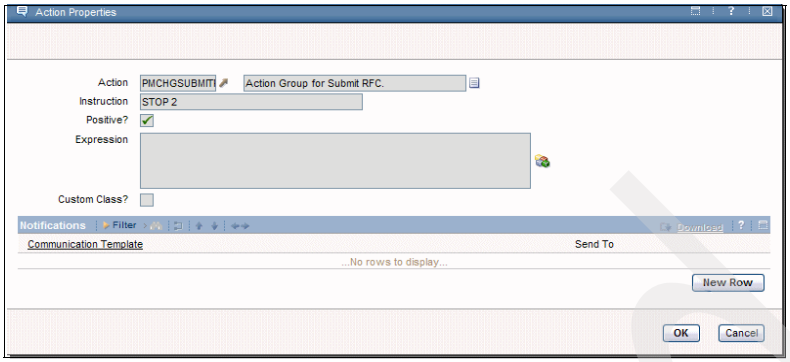


Figure 7-18 Action properties of PMCHGSUB workflow

The action property definition dialog reveals that the workflow is using an action for automating one of its steps. The action is called PMCHGSUBMITRFCGRP. You can see that the action is only triggered if the validation is positive by looking at the Positive? check box.

In a final step, we investigate the Action definition using the Actions application (Figure 7-19).

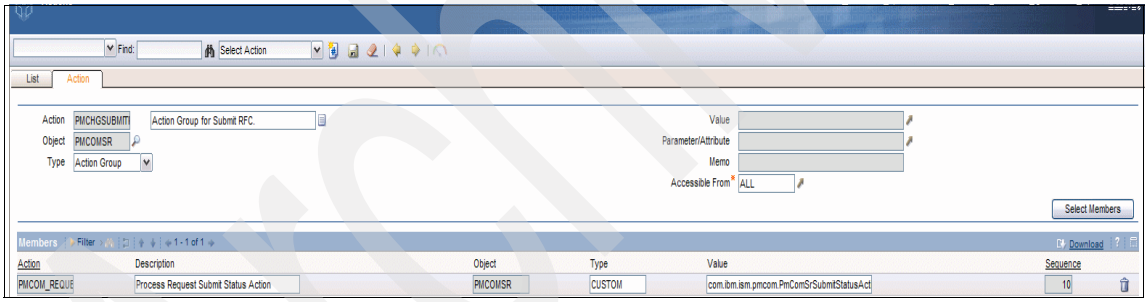


Figure 7-19 PMCHGSUBMITRFCGRP action definition

This is the action that finally sets the status of the process request to QUEUED. The action is a custom action and is implemented by a custom Java class.

Once the class is successfully submitted, a person who has the authority to accept the request reviews the request in the Process Request application and either rejects or accepts the request.

Similar to the ISMSUBMIT master workflow, there are master workflows for accepting and rejecting a process request. The master workflow for the acceptance of a request is ISMACCEPT, while the master workflow for the reject operation is ISMREJECT. The master workflows are defined in the System Properties application.

The ISMACCEPT master workflow is calling a specific subprocess workflow for accepting a process request of type change called PMCHGACC. The PMCHGACC workflow is using an action group called PMCHGACCEPT in order to automate various activities (Figure 7-20).

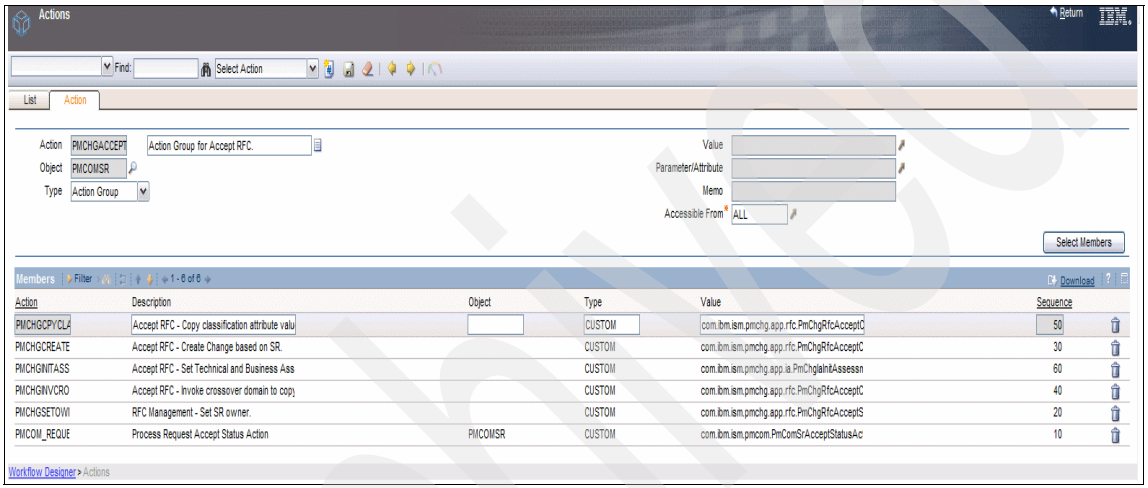


Figure 7-20 PMCHGACCEPT action group

All actions listed in the action group are of type custom and are implemented by Java classes.

For example, the second row of the action group member in Figure 7-20 reveals that a Change Work Order object is created. The first row reveals that all the classifications that have been specified in the process request are copied over to the Change Work Order object. The last row reveals an action that sets the status of the request itself to ACCEPTED.

Once you have accepted a process request, a work order object is generated. You can either see the work order object in the Work View or the Change application (Figure 7-21 on page 211).

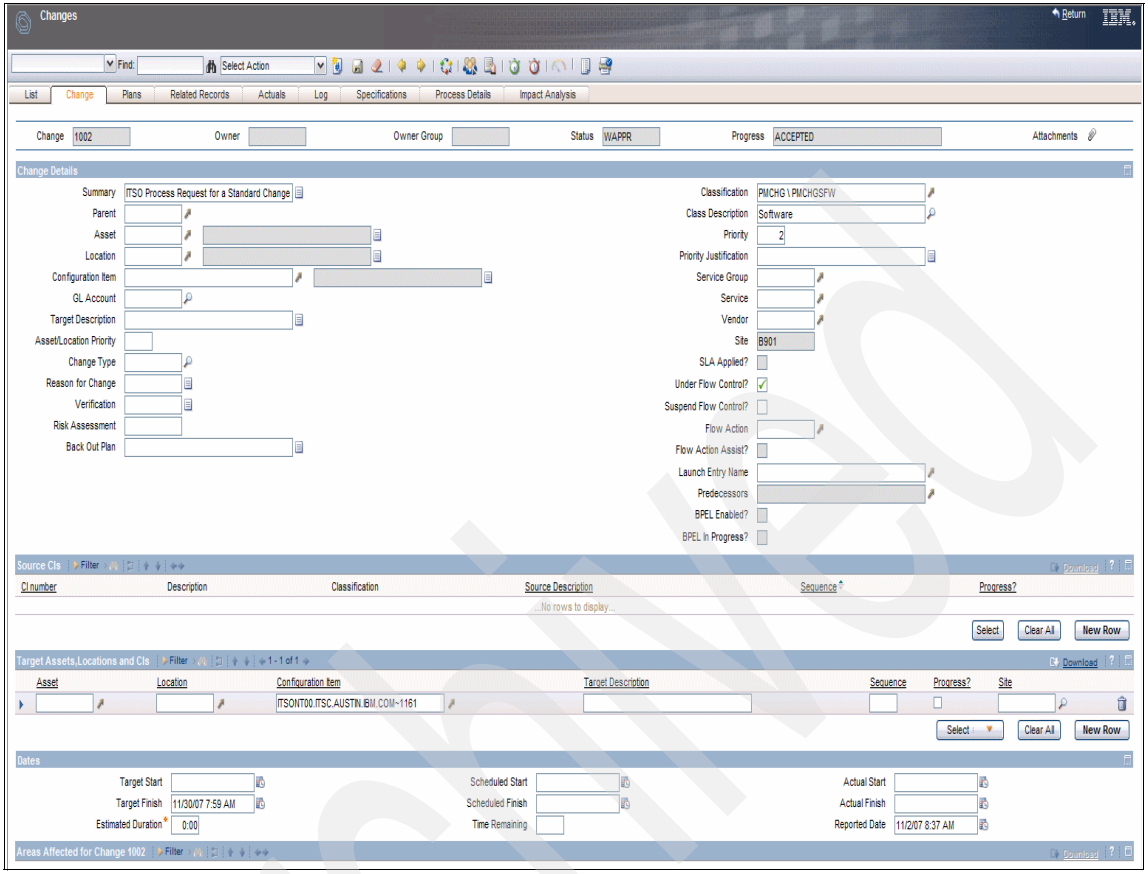


Figure 7-21 Change Work Order object created from process request

The reason we explained the process request background behavior in some detail is that you probably have a requirement to modify the behavior of the submit or accept operations.

Given that you have the requirement of setting some attributes of the Change Work Order object based on the input parameters of the process request, you can, for example, do so by modifying the workflow that is triggered when accepting the RFC process request (PMCHGACC). Given that a change request is opened to patch the operating system of the CEO's mobile computer, you probably want to treat the request differently from a normal change request.

In this case, you can, for example, add an action to the action group PMCHGACCEPT used within the PMCHGACC workflow to set the value of some specific parameters using the Action Type SetValue. You would use this action from within the workflow after using a conditional node in the workflow to check for a specific condition (Computer System owned by CEO).

Using actions of type SetValue is a frequently used concept used in the product. Figure 7-22 shows an action of type SetValue that sets the change progress state to IMPLEMENTED.

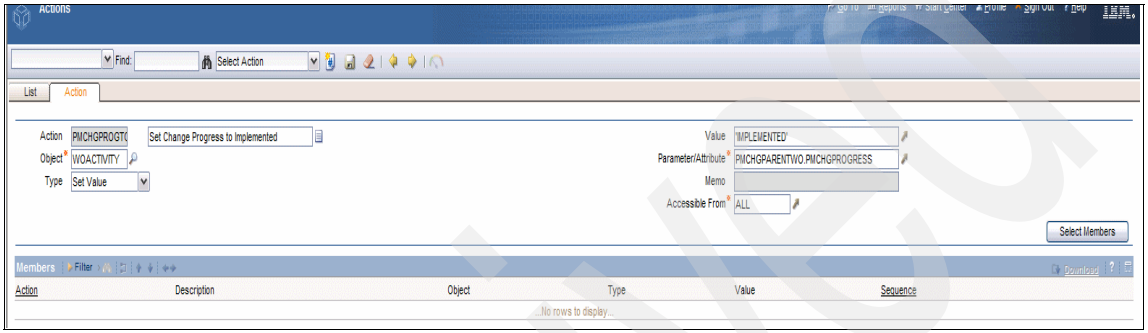


Figure 7-22 SetValue action definition

The Parameter/Attribute field holds the field to be changed to the value specified in the Value field.

The same model of modifying the submit or accept workflow can be considered when you want to apply a Job Plan template automatically to the work order object based on some specific classification attributes of the process request.

7.2.2 Process flow definition

After you have gone through the process of submitting and accepting a process request, a work order object is created. In order to generate a Work Order Plan from the Work Order, a Job Plan template is applied. In our example, the Work Order object is a Change Work Order object.

After you have created a Change Work Order Object, you can find the record in the Change application. You can see that all attributes of the process request have been copied over to the Work Order object (Figure 7-23).

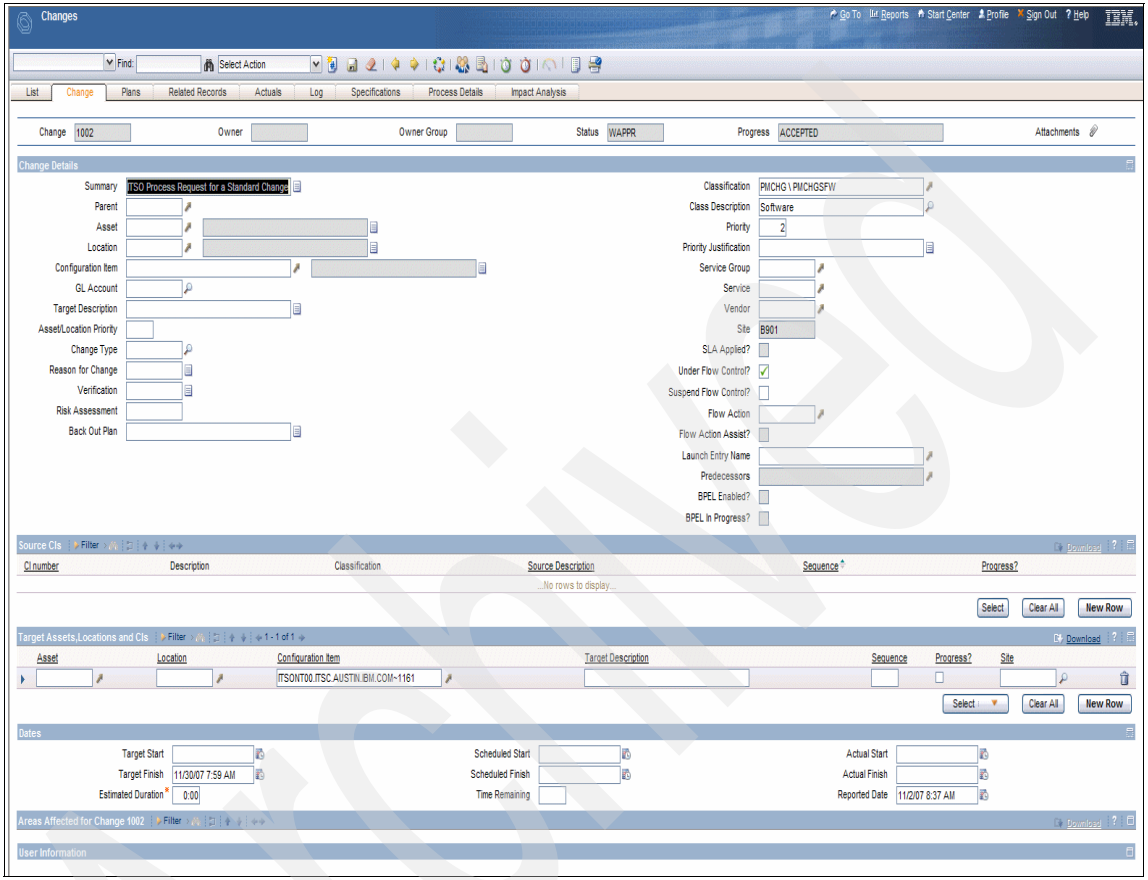


Figure 7-23 Applying a Job Plan to a Change - the Change Work Order Object

In order to apply a predefined Job Plan template to the Change object, select the **Plans** tab and click the arrow symbol next to the Job Plan field. A list of predefined Job Plan templates aligned to the type of Work Order object, in this case of type Change, is presented in a new dialog. Click the template you want to apply and the Job Plan activities are listed as child objects in the Change record dialog (Figure 7-24).

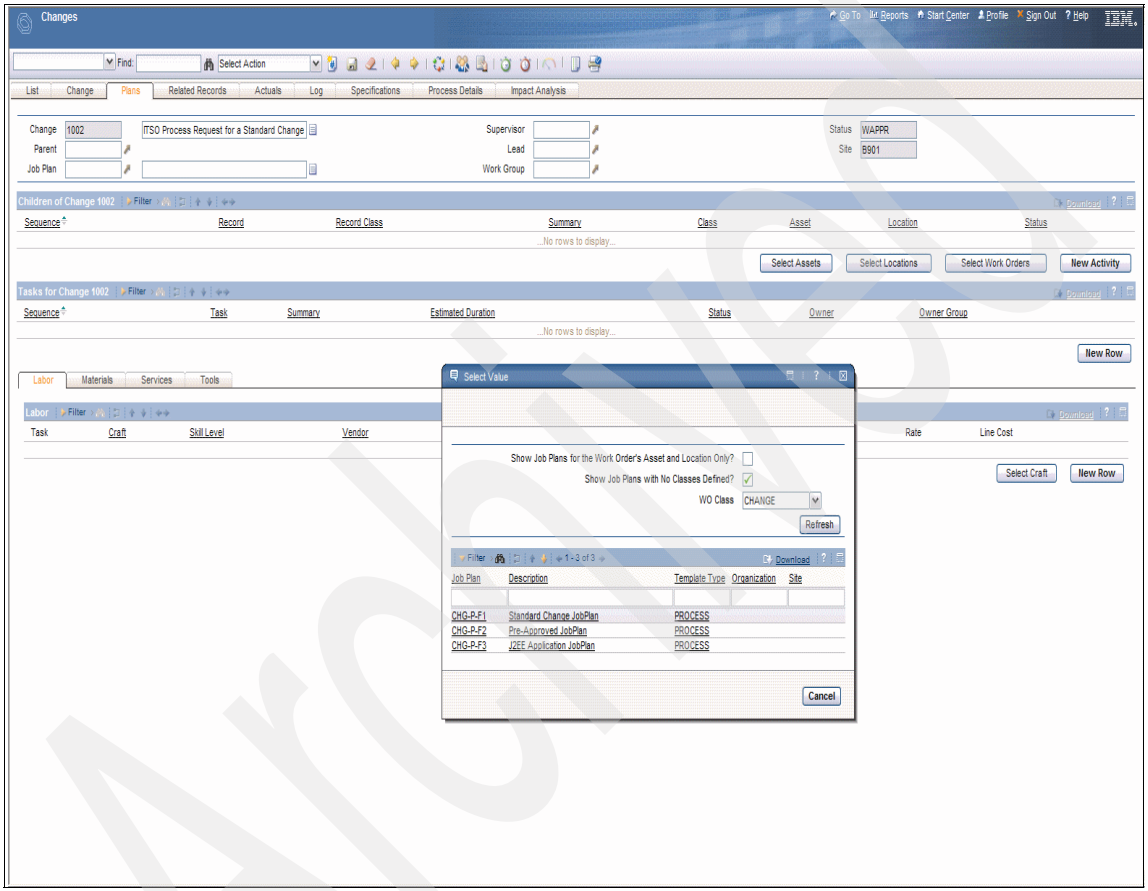


Figure 7-24 Applying a Job Plan to a Change Work Order object

Please refer to Chapter 8, “Process Managers” on page 233 where we explain the end-to-end life cycle of a Change Management process from its origination inside a process request until the request gets closed after the change process itself is completed.

The rest of this chapter explains the major configuration fields of a Job Plan template and explains the various possibilities for task definitions.

Figure 7-25 shows the top level definition of the *Standard Change Job Plan*. It shows the major activities (four out of five listed in this window). Each of the activities is defined as a task in a defined sequence. Each task has a task number that is foremost relevant for defining predecessor dependencies between the different tasks.

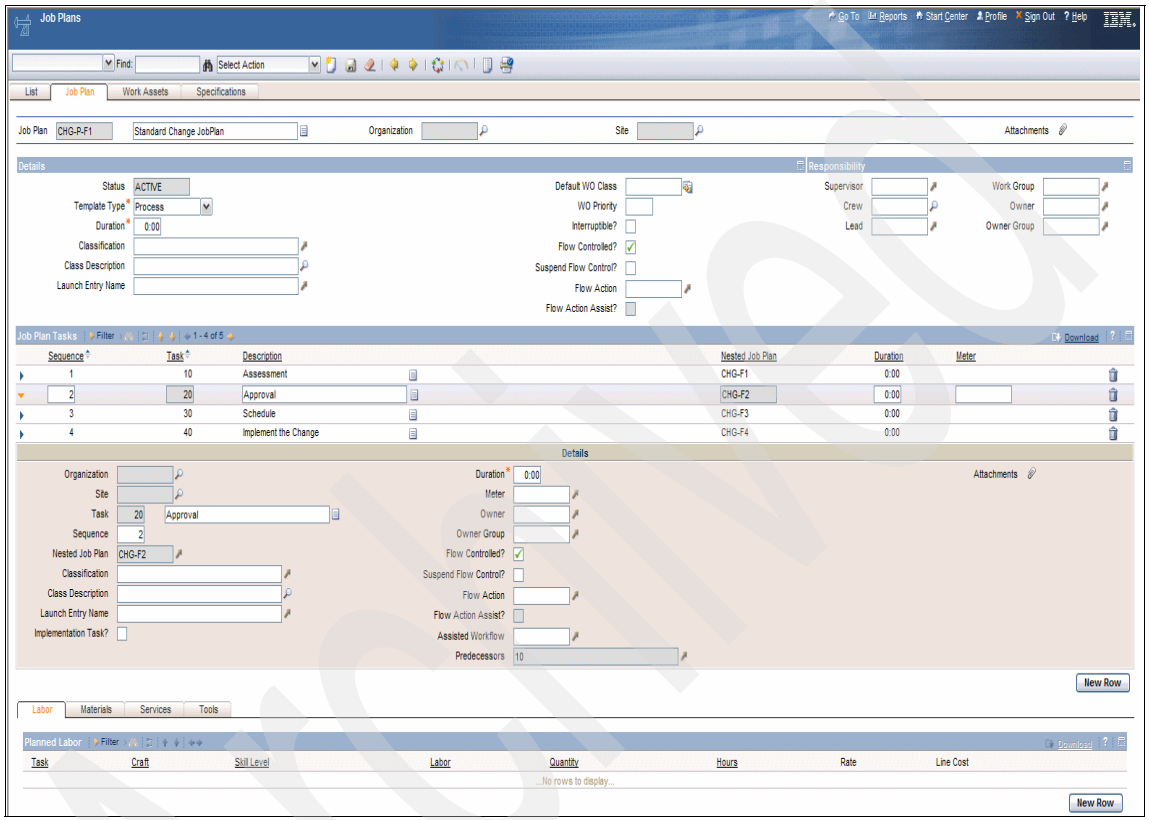


Figure 7-25 Job Plan top level task definition of activities

Each Job Plan template has a header section that holds attributes that are targeted to the parent work order object, while each task has its own detailed definition. The task definition is valid for the child work order object. Remember, a child work order object is created after a Job Plan template has been applied to the work order object.

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 would 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.

Since 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 Approval phase is referring to the nested Job Plan CHG-F2. The task has a predefined predecessor, which is task 10. Since task 10 is referring to an activity, all the tasks in the activity have to be completed before the first task inside the Approval activity can start.

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

In the lower section of the Job Plan template window are additional tabs like 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.

While Figure 7-25 on page 215 shows the task definitions for the high level activities of the process flow, Figure 7-26 on page 217 shows an example of task definitions of a process flow template specifying concrete steps to take.

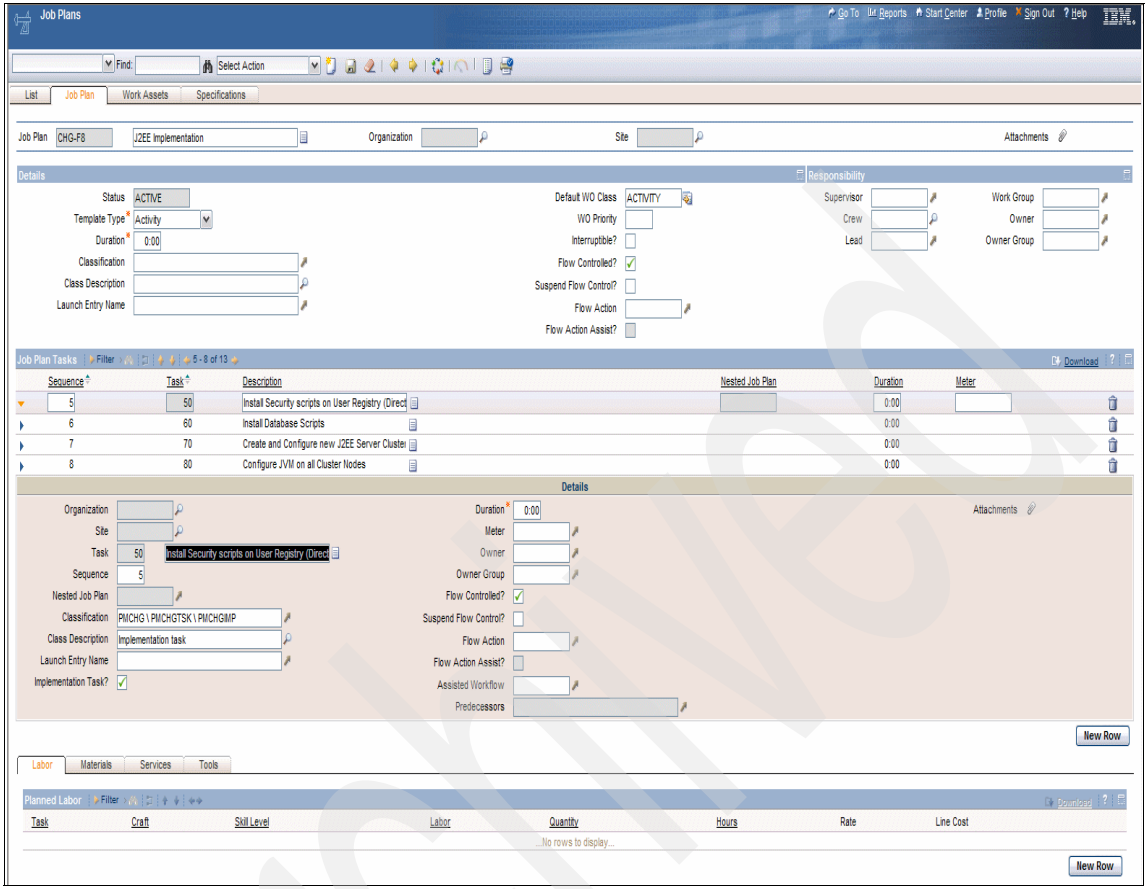


Figure 7-26 Job Plan task definition of J2EE Implementation Job Plan

The J2EE Implementation Job Plan template defines the steps required to deploy a complex J2EE application. Tasks like Install Database Scripts or Configure JVM™ on all Cluster Nodes define type dependant implementation steps. You can customize or further detail them if needed.

Figure 7-26 shows an example of how classifications can be used to further classify the nature of the task. Classifications can be used in validations or decision points of the workflow logic if necessary.

The task definition does not have any predecessor definition, which means that it can run in parallel to all other tasks defined inside this Job Plan, given the other task definitions do not have a predecessor definition either.

The following example of a task definition inside the Assessment activity Job Plan template reveals that the Assess Security Impact task merely requires task 10 to be completed (Figure 7-27). It does not rely on any other task in the Job Plan to be completed.

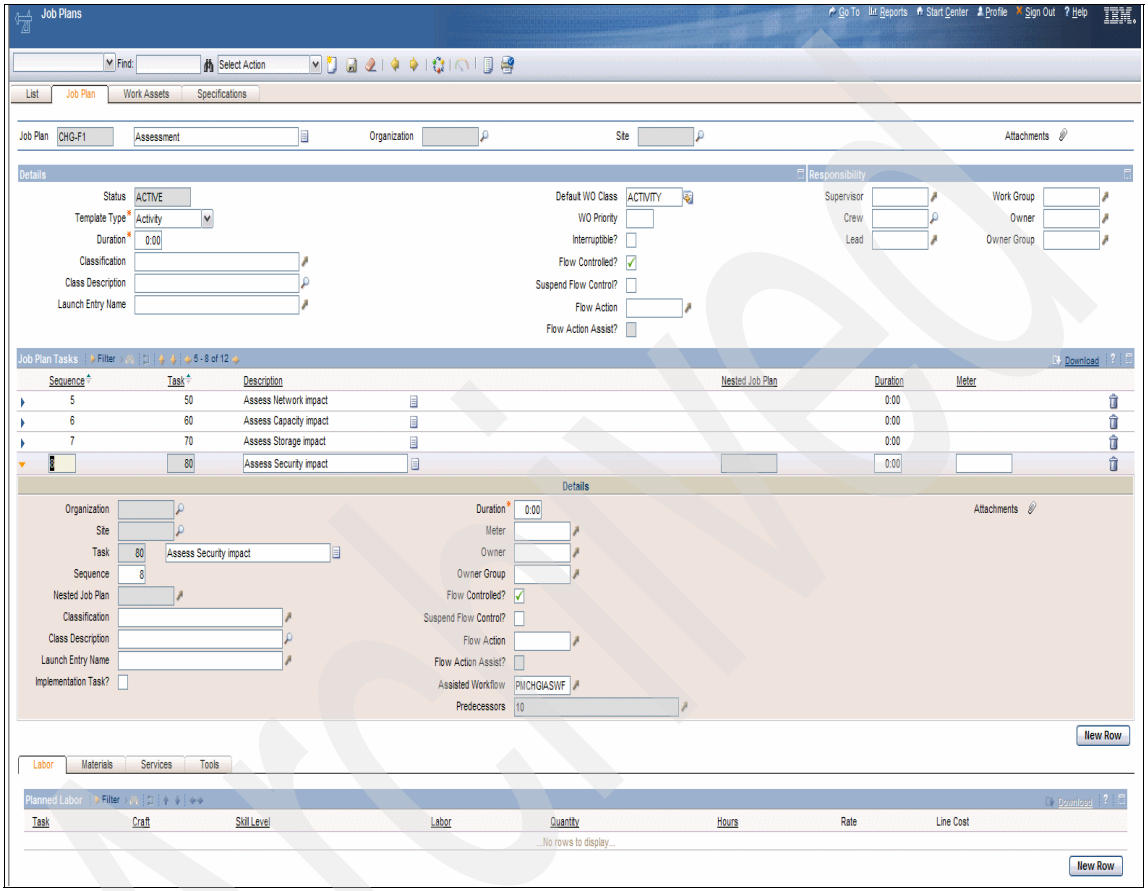


Figure 7-27 Assessment Job Plan template

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 some interactive automation, for example, requiring a decision or routing him to another application. Another reason to use the workflow technology is if some conditional checks are required. Workflows are usually triggered if some 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 7-28, the workflow PMCHGIASWF is specified in the Assisted Workflow field.

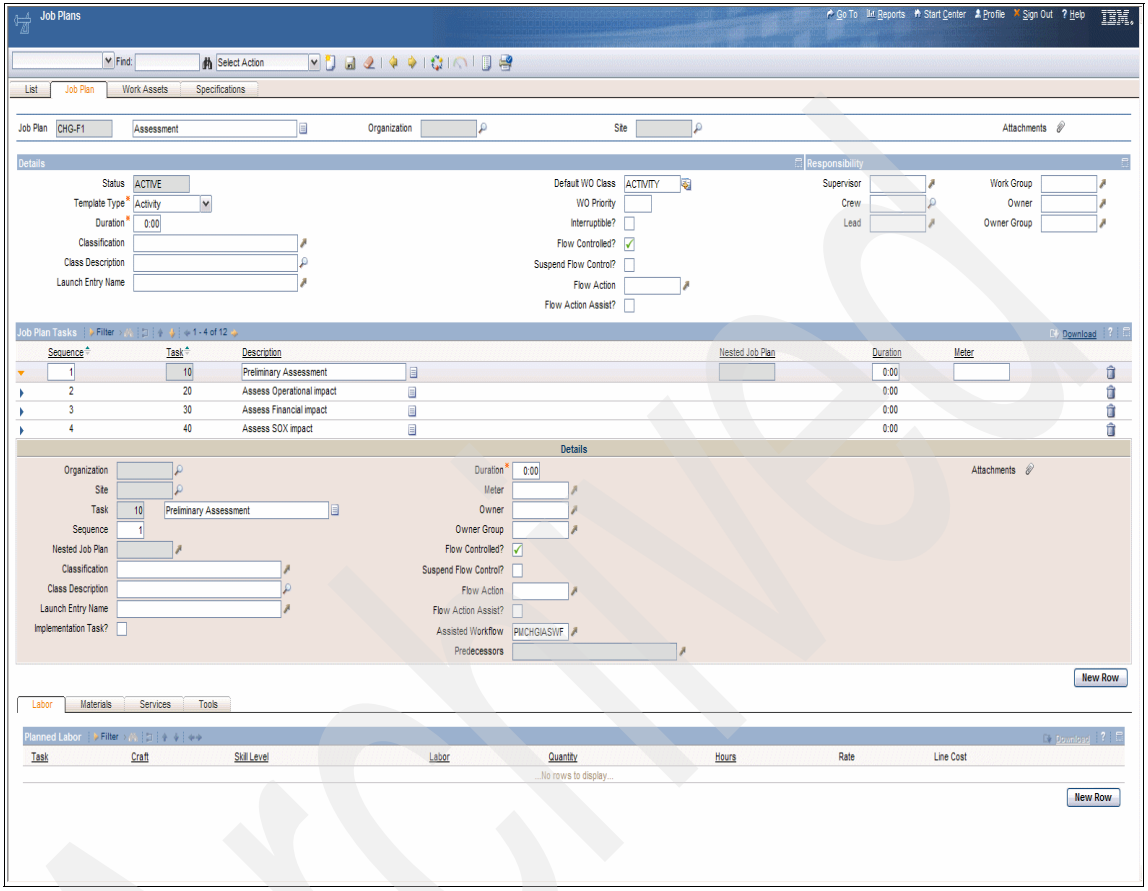


Figure 7-28 Assisted Workflow task definition

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 REDIRECT (Figure 7-29). An interaction node is used to route a user to another application in the system.

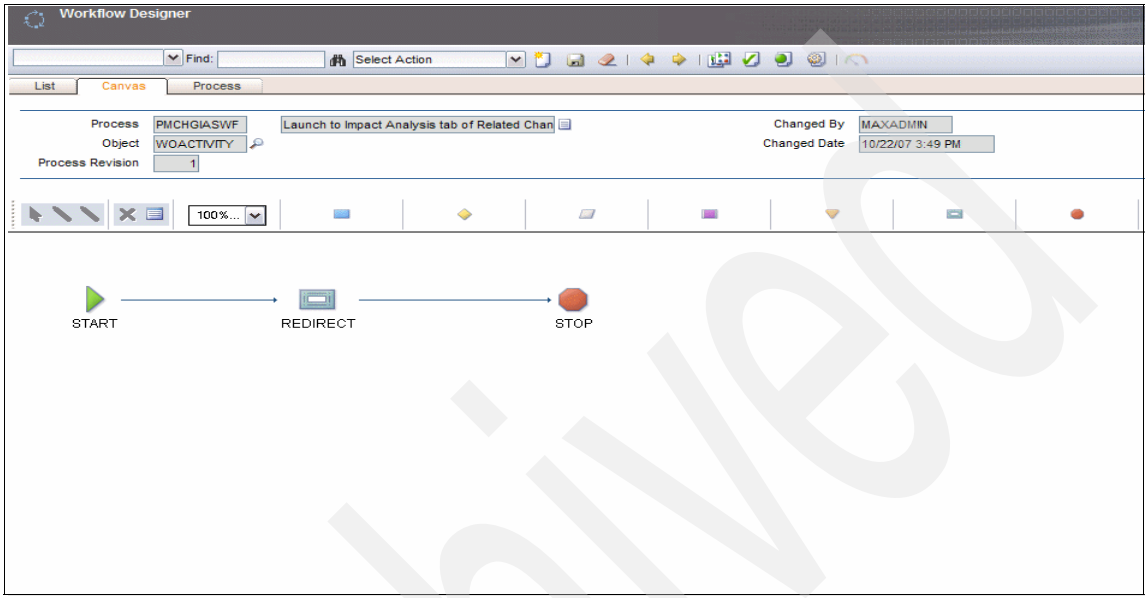


Figure 7-29 Assisted Workflow Canvas in Workflow Designer Application

The properties of the interaction node (right-click the icon) show the definition shown in Figure 7-30.

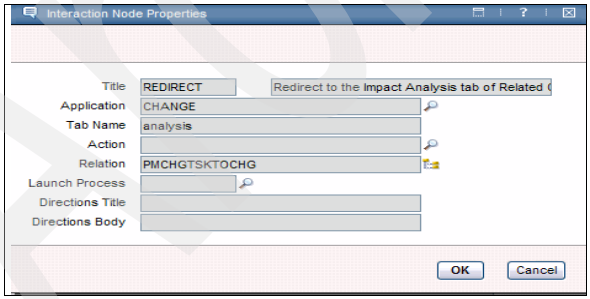


Figure 7-30 Workflow Interaction Node Properties

The property definition shows that the user is guided to the Impact Analysis tab of the Change Application when this task of the workflow is initiated. Remember that the task definition is inside the Assessment Job Plan template definition.

Impact analysis is a key concept of Change Management to analyze the potential technical and business impact of a change procedure.

Note: A workflow can use an action or action groups inside its execution model. We prefer to call an action from a workflow rather then calling the task directly from the task definition of the Job Plan if, for example, we require a conditional check before a decision can be made as to which action to run.

Next, we explain an example of a task definition that calls out to a predefined action. An action or action group is specified if you want to run a self-sufficient program in the background. Another reason to specify an action in the Flow Action field of the task definition is if you want to call out to an action that itself calls out to a workflow.

This is the example that we explain on the following pages. The action called APPACTWOA is specified in the Flow Action field (Figure 7-31).

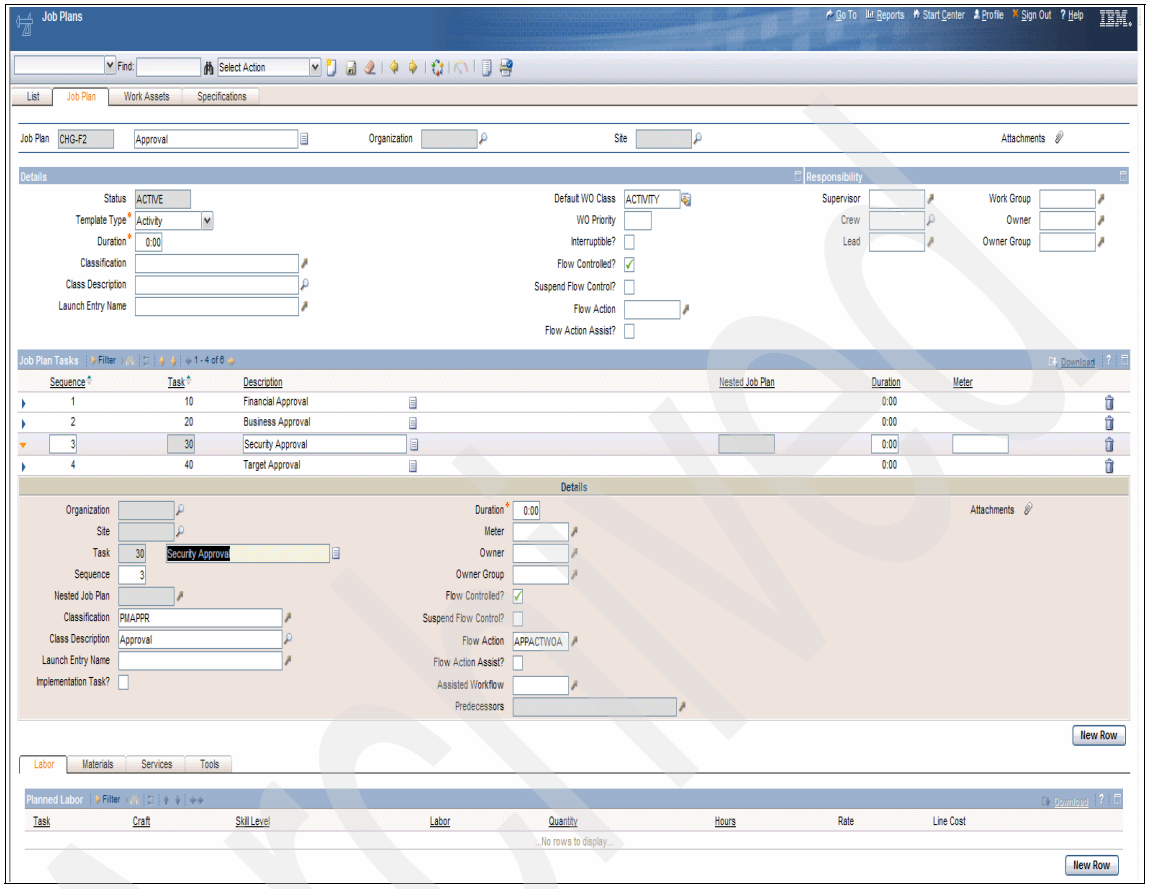


Figure 7-31 Flow Action definition from Security Approval task

If the Flow Action Assist? check box is checked, the action will not run automatically when the task is set into progress. An example would be if you are required to manually start the action 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.

In the Actions application, we look at the definition of the APPACTWOA action (Figure 7-32 on page 223).

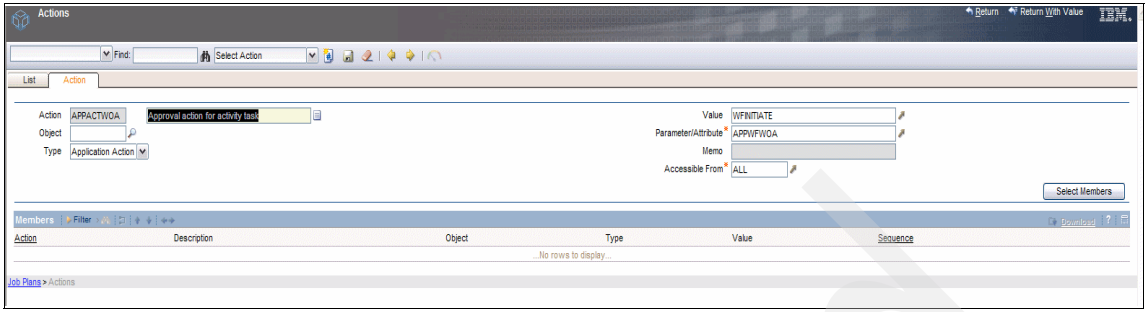


Figure 7-32 Action definition in action application

The action is of type Application Action with a value of WFINITIATE. This tells us that the action initiates a workflow. The name of the workflow is specified in the Parameter/Attribute field. The name of the workflow in Figure 7-32 is APPWFWOA. Remember that the task definition is inside the Job Plan template for the approval activity phase.

Figure 7-33 shows the APPWFWOA workflow in the workflow designer application.

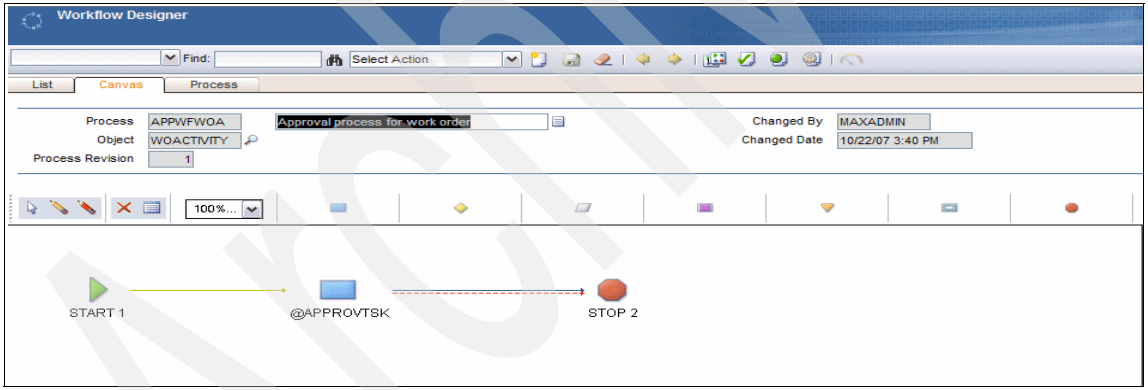


Figure 7-33 Workflow definition of workflow called by action

The @APPROVTSK is a task node definition. Task Nodes in a workflow layout definition is used to assign a record to a role that is resolved to a group of responsible people. The assignment of work at runtime of the process is shown in the Start Center in-box of the respective people.

Looking at the @APPROVTSK task node properties, you can see that the record will be shown in the in-box of the work order approval owners. Those are specified by using the role APPROLEWOA (Figure 7-34).

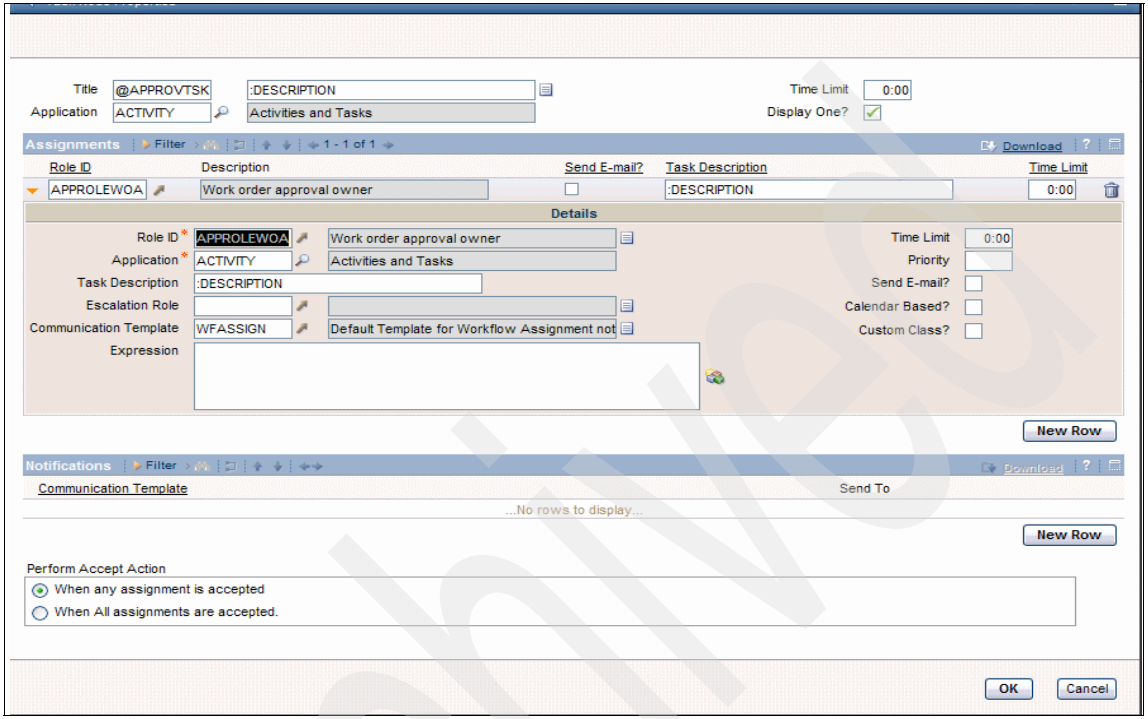


Figure 7-34 Approval workflow task properties

The definition in Figure 7-34 requires at least one of the approval owners to approve. This is specified by the radio button **When any assignment is accepted**. The other option would be to require all users of the approval owner group to approve.

Once a user approves the record while the process is in progress, the overall process is taken to the next step.

The approver or group of approvers (depending on the resolution of the role definition) is shown in the approvers in-box with a message that is using the following communication template (WFASSIGN) (Figure 7-35 on page 225).

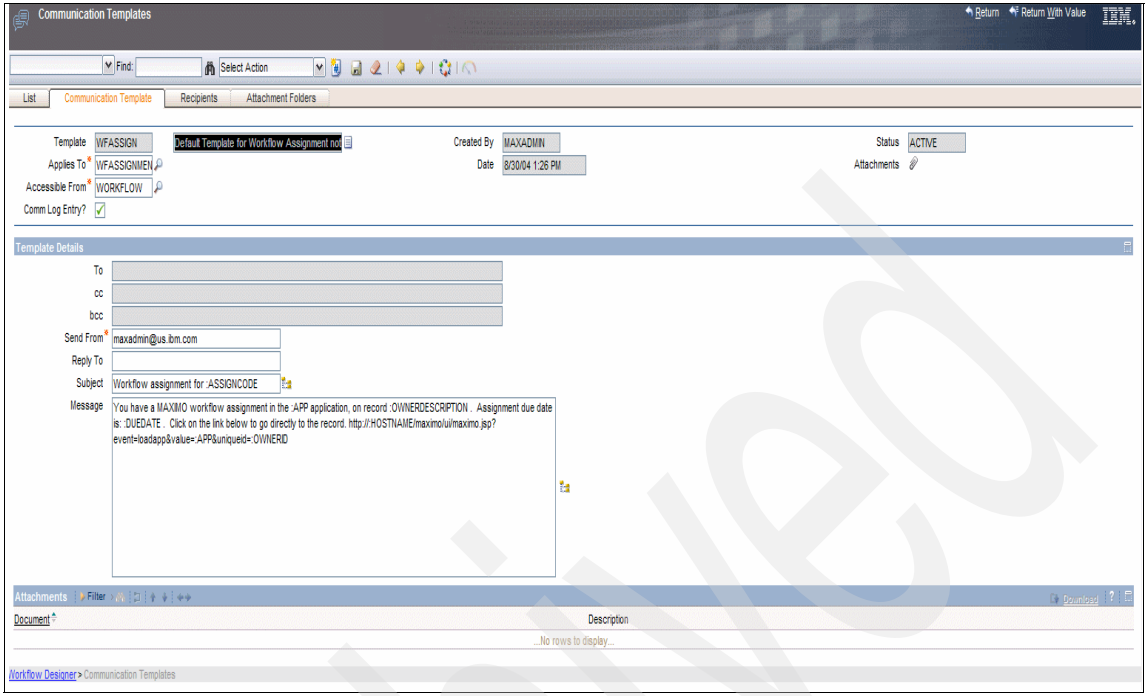


Figure 7-35 Communication template used by approval task node

A communication template defines a predefined message text with variables that gets posted to either the users start center in-box or sent by e-mail.

Depending on a positive or negative approval decision, the APPWFWOA workflow defines what needs to be done next. This is defined as a property on the connection lines between the @APPROVTSK and the STOP 2 node.

Figure 7-36 shows the properties of the connection line for a positive approval decision.

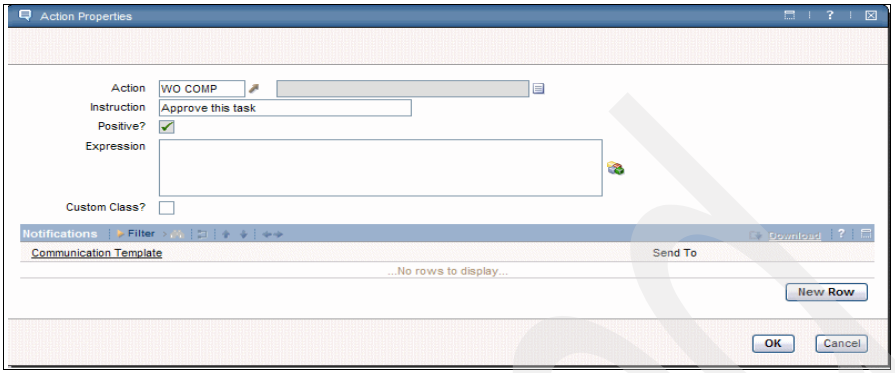


Figure 7-36 Positive decision point for approval task node

In this case, the WO COMP action is called to set the status of the defined approval task to completed. Remember that this task is just one of many approval tasks in the Job Plan template. The same mechanism is used for all approval tasks in the Job Plan template for the approval activity.

Our final example is another example of how to define an assisted workflow as part of the task definition (Figure 7-37). This time our intention is to highlight the interaction between the Change and Configuration Management process flows. 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 CCMDB 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:

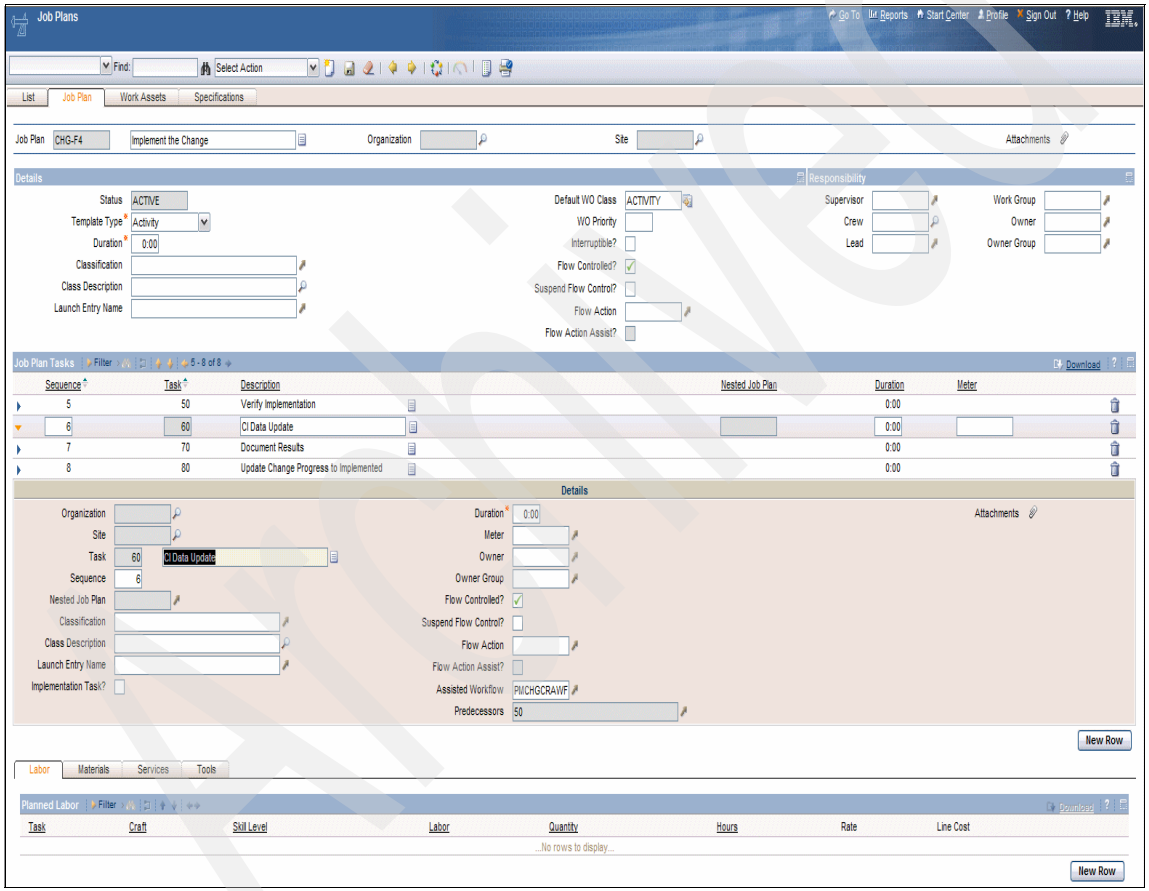


Figure 7-37 Change Management generates a Process Request to Configuration Management

The definition of the PMCHGCRAWF workflow in the workflow designer application is shown in Figure 7-38.

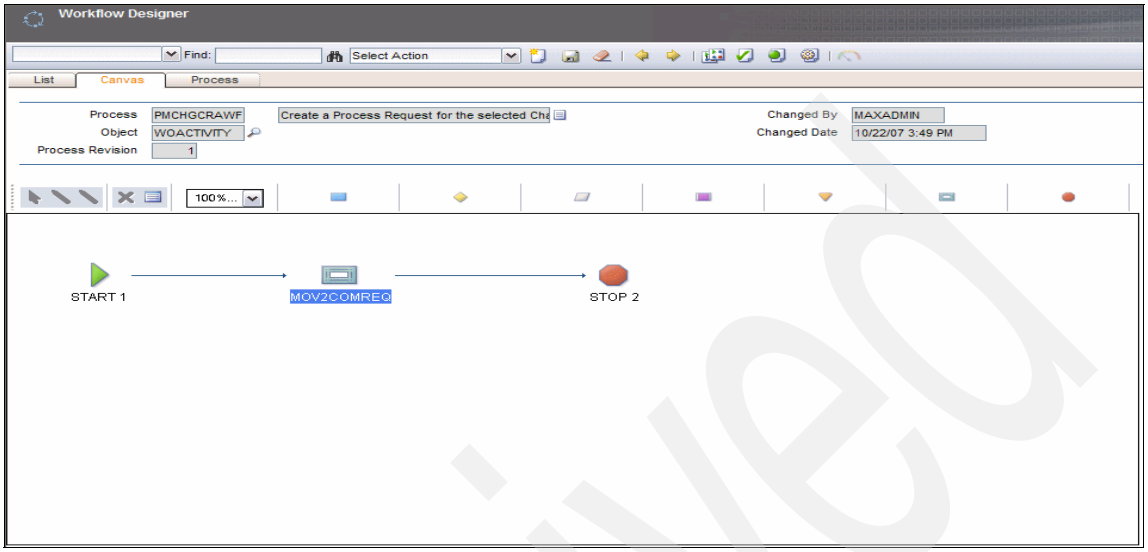


Figure 7-38 Definition to generate Process Request Configuration Management

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 different types. Remember that a process interaction always requires a process request before a work order object can be generated.

A another good example for a predecessor definition is shown in Figure 7-39 on page 229.

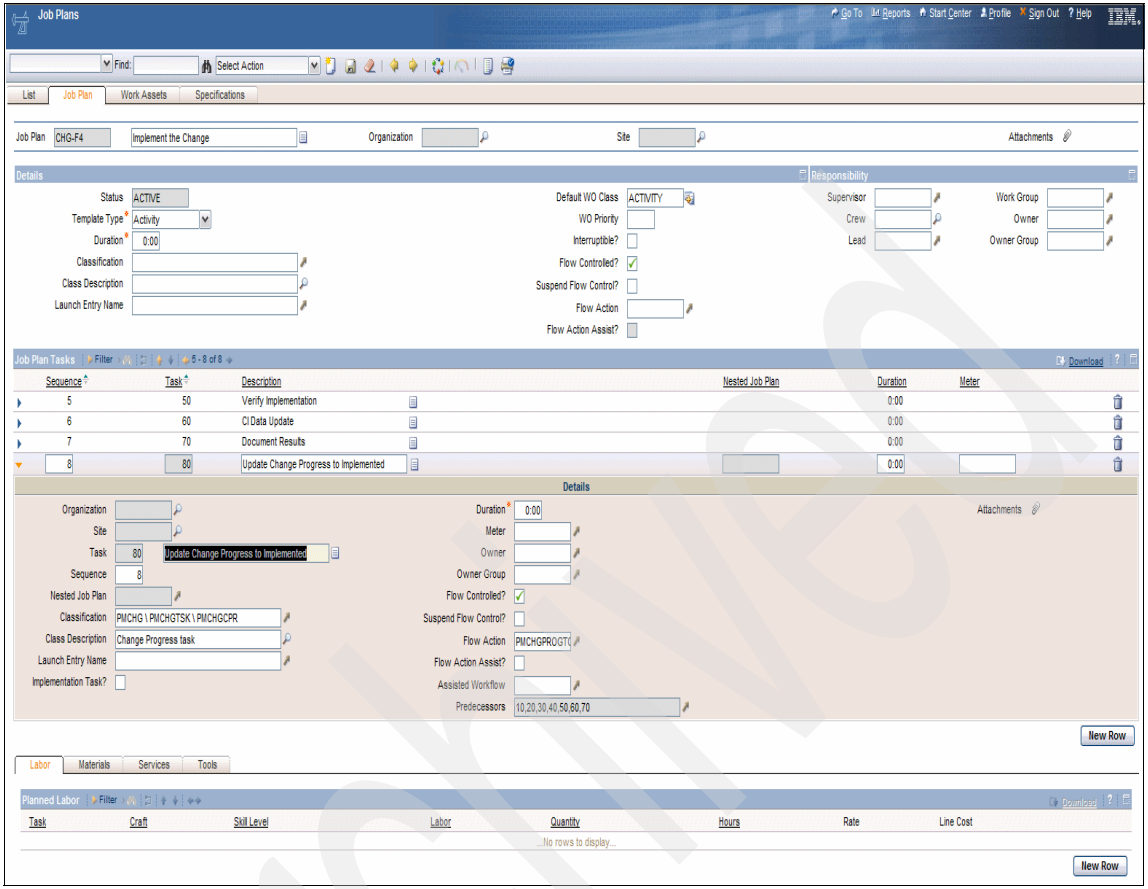


Figure 7-39 Predecessor definitions in task definition

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 may not start in parallel to any other task of this Job Plan.

So far we have explained how a task can define different 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.

Nevertheless, there will be steps in the overall process flow that are manual since there is no way to completely automate every task. An example of such a manual task is the Document Results task of the Implement the Change Job Plan template (Figure 7-40).

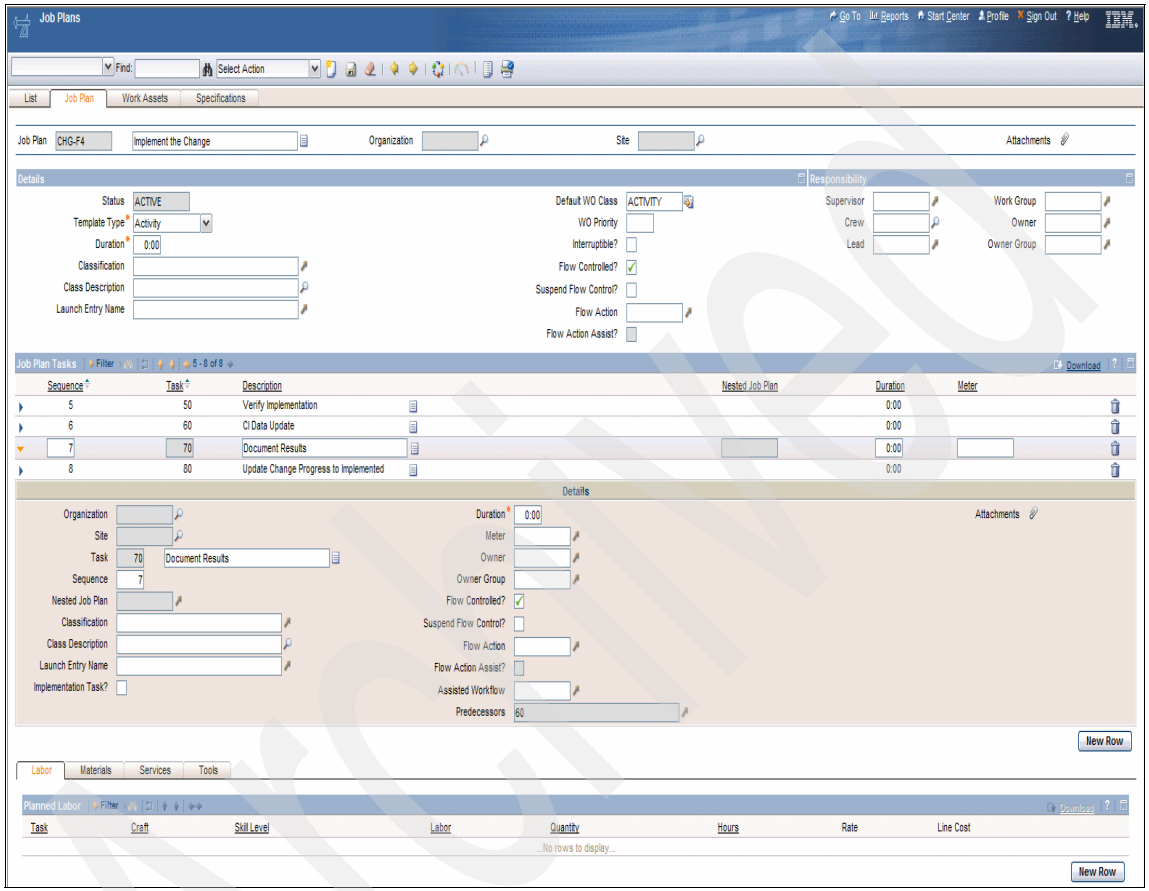


Figure 7-40 Definition for a manual task

The user has to document the steps that have been taken during the change implementation manually. There is no way to automate this kind of documentation.

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 like 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 7-41 reveals the values of definitions that have been defined in the Launch in Context application.

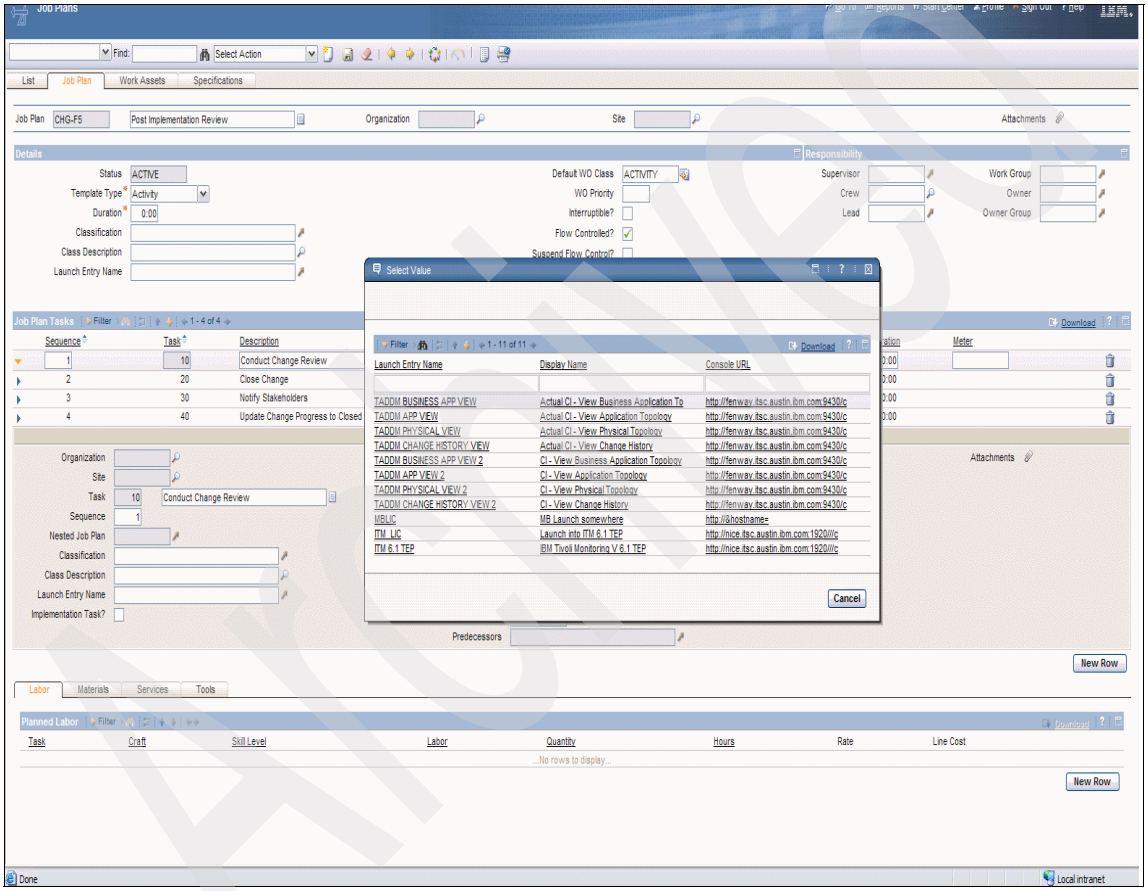


Figure 7-41 Task Launch in Context Definition

7.3 Summary

In this chapter, we explain the major capabilities of a Job Plan template task definition. We explain how to define manual and automated tasks, tasks to Launch in Context to an external system, how to interact with other process types, and define the sequence of tasks inside a process by defining predecessor relationships between the tasks.

8

Process Managers

This chapter describes the basics of both the Change Management and Configuration Management PMPs and how they may be used and integrated with other Process Managers.

8.1 Overview of Process Managers

Process Managers (PMs) are Web applications that permit integration, automation, and implementation of processes. It is composed of flexible workflows that can be customized as necessary. PMs enable the creation of executable process flows, provide a user interface to allow users to perform process procedures, gather information from different sources, interact with external tools, leverage 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.

Process Managers leverage industry best practices like IT Infrastructure Library® (ITIL), Control Objectives for Information and Related Technology (COBIT), and enhanced Telecom Operations Map (eTOM). They enable implementations of IT service management process because they already have default processes aligned to the best practices, so it is not necessary to build a process from scratch.

Note: 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 should be controlled through key performance indicators.

Process Managers are essential components of the IBM Service Management architecture. Figure 8-1 on page 235 shows how Process Managers relate to the IBM Service Management architecture.

.

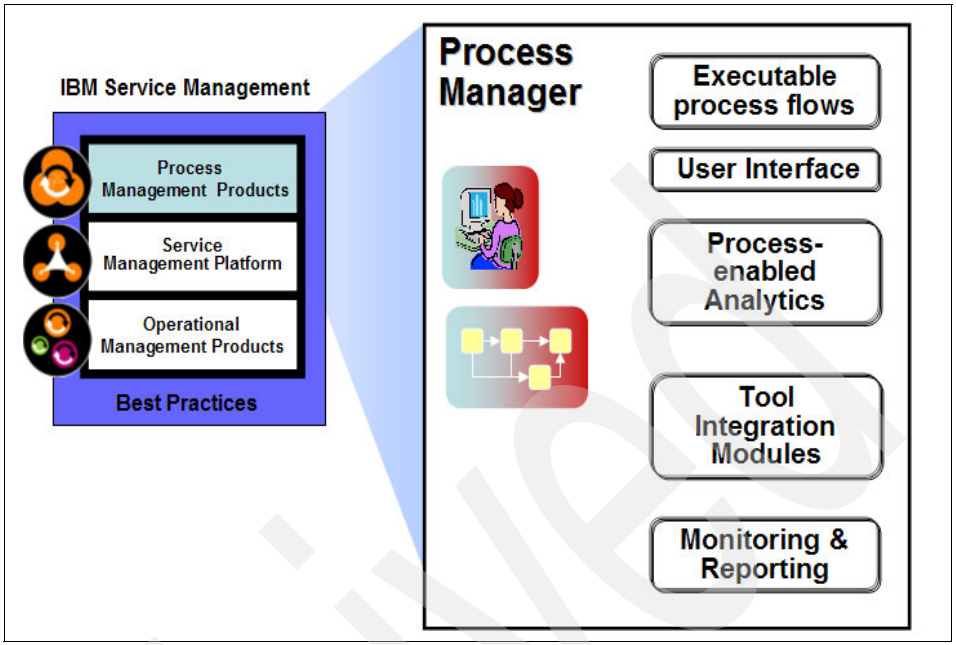


Figure 8-1 Architectural context of IBM Service Management

8.1.1 Process Manager role

Once a best practice process is understood and selected, organizations then have to determine the best way of implementing that process. For organizations that need to automate their processes, it is often necessary to hire developers to build workflow-based applications from scratch. This requires organizations to go through a full development cycle to model, simulate, develop, deploy, and execute these processes. This is often an expensive proposition, requiring advanced development tools to build workflow-based applications.

An alternate option is to use 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.

Once 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 different 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 bottlenecks and can be used to re-engineer processes to improve organizational efficiency.

Figure 8-2 gives an overview of the Process Manager role.

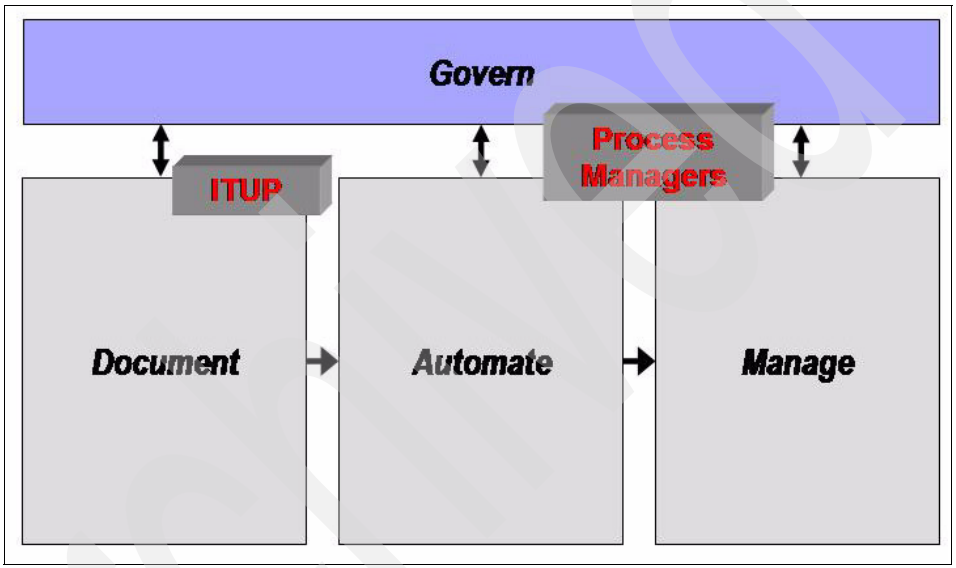


Figure 8-2 Process Manager role

8.1.2 How Process Managers work

Process Managers 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 different sources (including external tools) through tool integration modules.

A Job Plan is a template, with a detailed description of work to be performed on an asset, configuration item, or location. When using Job Plans, it is not necessary to enter the same information every time that a work order is created for a similar request. A Job Plan can be applied to an unlimited number of work orders. After a Job Plan is applied to a work order, its resource estimates and tasks are copied into a work plan for the work order. The work plan can be

modified so that the procedures, labor, materials, services, and tools are more specific to the work order, without affecting the original Job Plan template.

Job Plans can be applied to different types of changes:

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

The Job Plans application should be used 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.

The starting point for the Process Manager is when it receives a process request from the Process Request Application that is responsible for administering the process requests that were submitted. Within it, 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 8-3 illustrates the process request flow and how it interacts with the Process Manager.

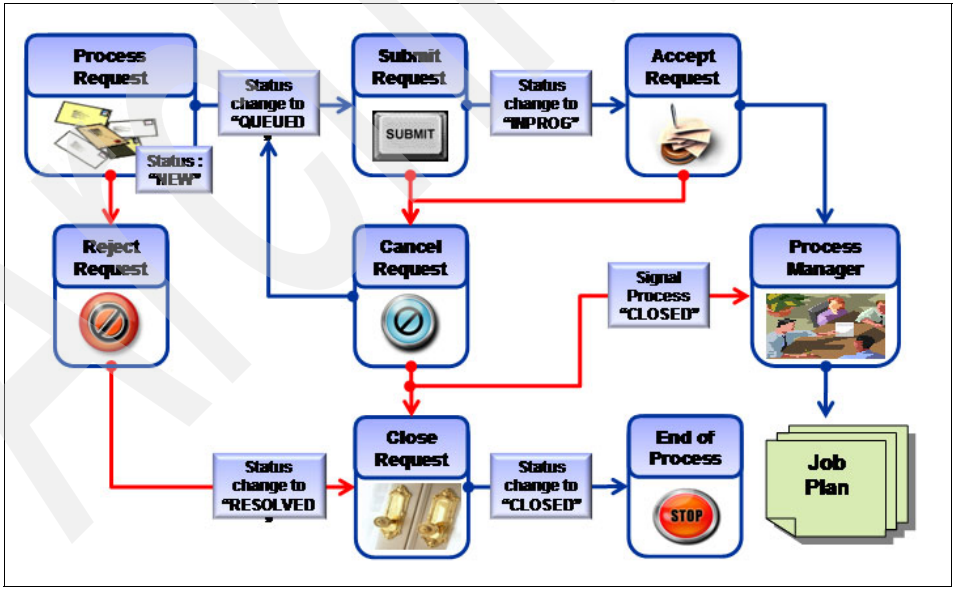


Figure 8-3 Process request and Process Manager

Nested Job Plan

A Nested Job Plan is a Job Plan that is used to create work order hierarchies. It also supports the concept of creating small Job Plans that are intended to be re-usable components of large work projects.

Each Job Task will be able to define a nested Job Plan. When applied to a work order, these nested Job Plans will create child Work Orders instead of Tasks.

More information can be found in Chapter 7, “Process flow technology” on page 189.

8.1.3 Job Plan

This section explains how to work with Job Plans.

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

Create 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.

Defining a Job Plan consists of the following steps:

- Defining the tasks by breaking the job down into steps in the Job Plan Tasks table window
- Defining the labor craft/skills and hours on the Labor sub-tab
- Defining the materials needed on the Materials sub-tab
- Defining the services needed on the Services sub-tab
- Defining the tools needed on the Tools sub-tab

These steps are applied according to the purpose of each Job Plan. For example, some Job Plans may not require materials.

A required field is the template type. It defines the type of template that should be applied to the Job Plan. It can be:

- Process: Should be used to classify a Job Plan as a process. It is the top level of the hierarchy.
- Activity: Should be used to classify a Job Plan as an activity. It is the lowest level of the hierarchy.

- Maintenance: Should be used to classify a Job Plan as maintenance. It is mostly used for enterprise asset management situations.
- Configuration PMP: Should be used to classify a Job Plan as a Job Plan for configuration PMP purposes.

Figure 8-4 shows the Job Plan form.

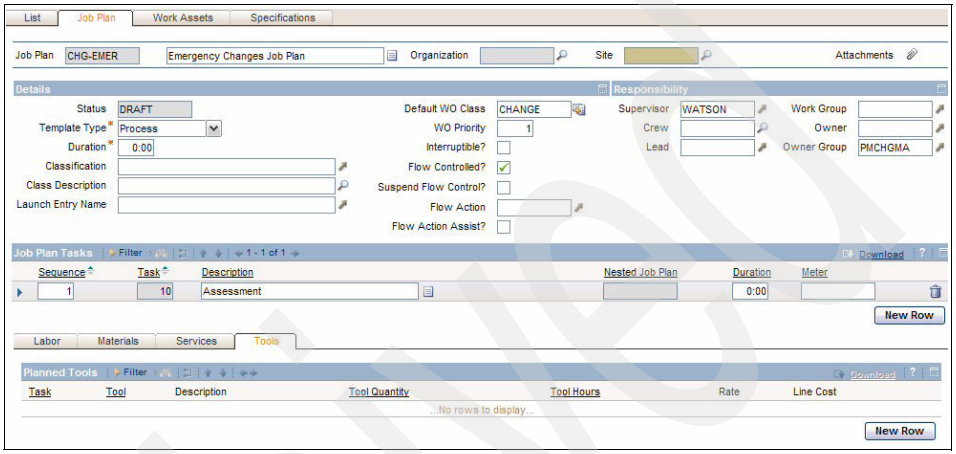


Figure 8-4 Create Job Plan

Define 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 should be performed. Tasks can have the same sequence number if they should or could be performed simultaneously. After you apply a Job Plan to a work order, CCMDB copies the sequence numbers to the work order's tasks.
- Task ID: Unique identifier of the task. The default is for CCMDB 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 8-5 shows the Job Plan Tasks window.

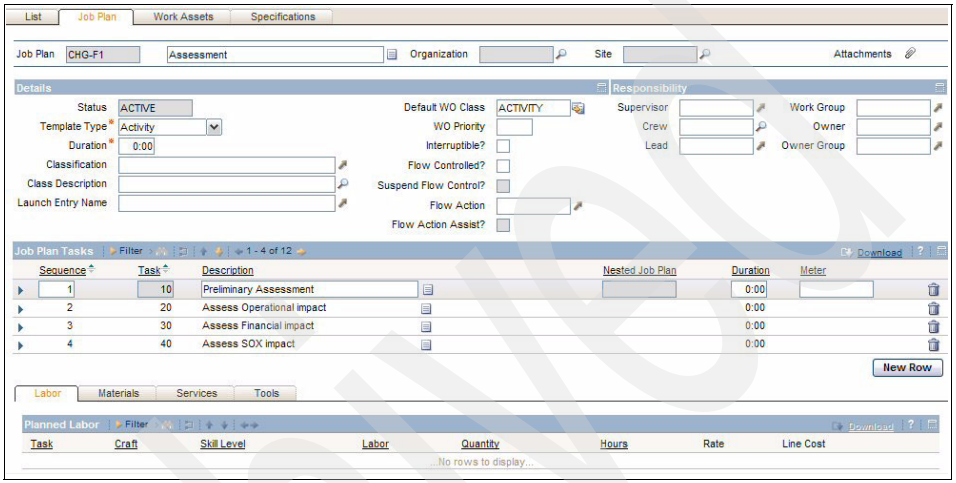


Figure 8-5 Job Plan Tasks

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 predecessors tasks.

Figure 8-6 on page 241 illustrates the Predecessor tasks.

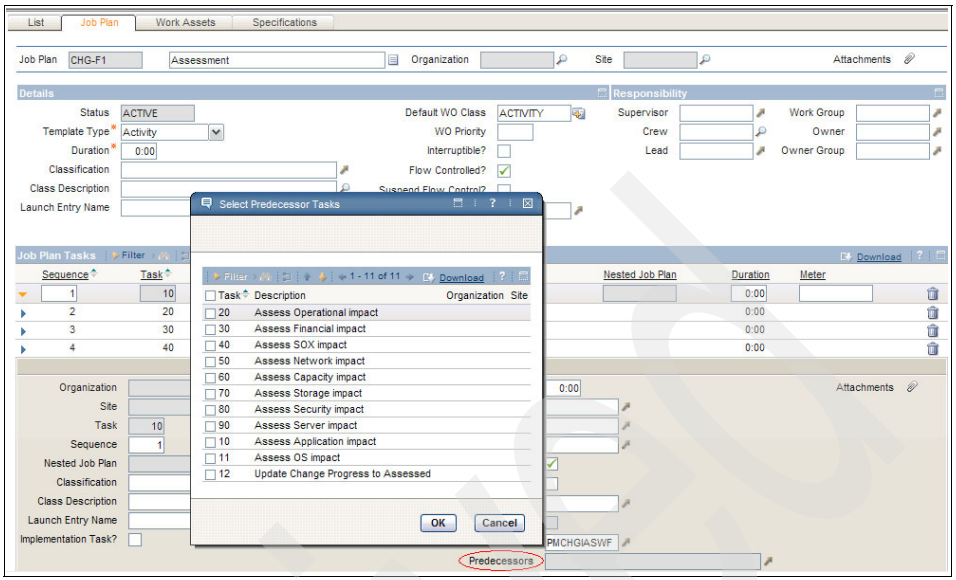


Figure 8-6 Selecting Predecessor Task

After defining all Nested Job Plans, they should be associated to the main Job Plan; Figure 8-7 illustrates a Job Plan with Nested Job Plans.

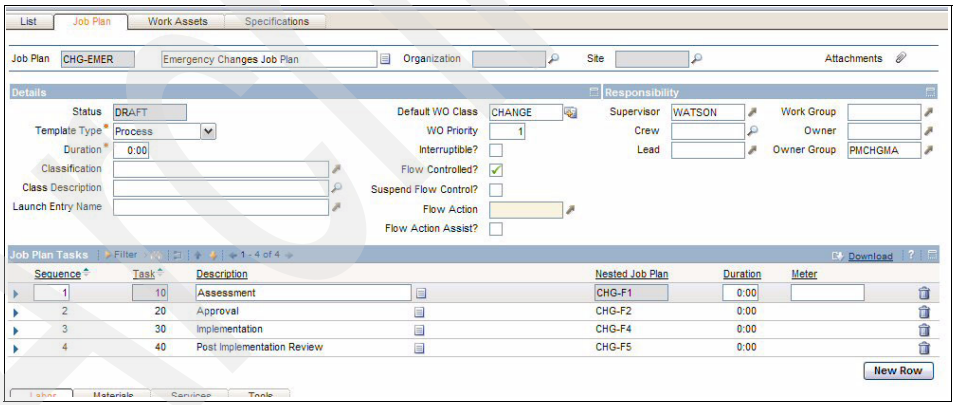


Figure 8-7 Job Plan with Nested Job Plans

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 8-8 illustrates the association of a Nested Job Plan.

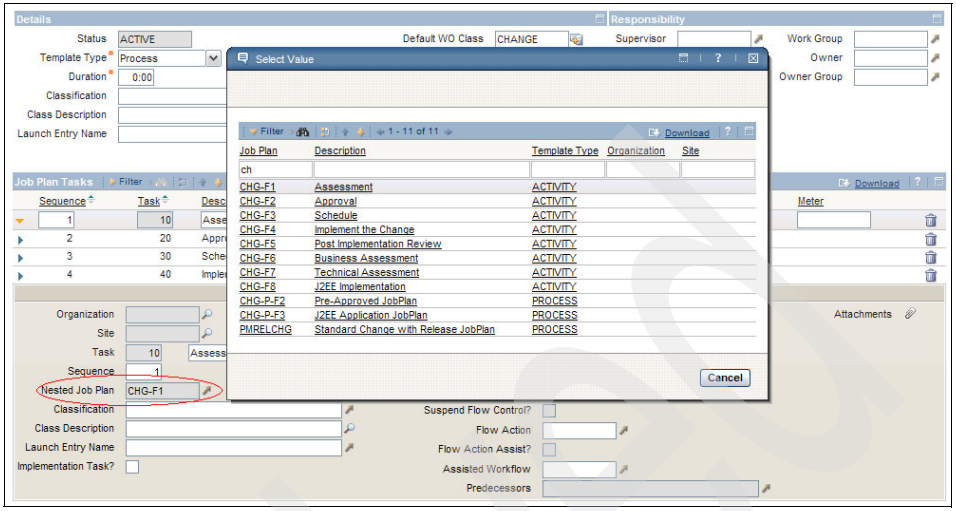


Figure 8-8 Associating a Nested Job Plan

Note: 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.

Delete 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.

Duplicate Job Plans

The Duplicate Job Plan action should be used to create a copy of an existing Job Plan, for example, if you want to create a similar Job Plan for two different Sites. Once 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 a different 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 should be deactivated by having their status changed to INACTIVE rather than deleted.

8.2 Change Management Process Manager

This section discusses the Change Management Process Manager.

8.2.1 Change Management overview

Change Management is a process responsible for protecting the environment from unauthorized changes. Through it, standardized methods and procedures are defined for efficient and prompt handling of all change in order to minimize or avoid the impact of change-related incidents on service quality and consequently on business.

Information Technology has become a critical success factor for business, meaning that interruptions in IT service and quality issues may cause significant impact to 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 are some of the most common occurrences in the IT environment that require a RFC:

- Business requirements have changed.
- An incident or problem resolution that 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 business unit has changed locations, requiring the wholesale relocation of office equipment and computing resources.
- Vendors or contractors have changed their products or services.

Note: ITIL defines the objective of Change Management as to ensure that changes requests are recorded and then evaluated, authorized, prioritized, planned, tested, implemented, documented, and reviewed in a controlled manner.

Change life cycle flow

The Change Management process begins when a Change Requester submits a Request for Change with minimum information. The person responsible for receiving the requests analyzes the request, 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.

Some suggestions of what information a change record should contain are:

- Description
- Configuration items impacted (can be changed during the assessment step)
- Reason for 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 should 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 approval.

Note: Preapproved or standard changes are potential changes that have already been analyzed and approved by the Change Manager/Change Advisory Board. Standard changes 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 may 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 some changes are implemented using an assignment by the Change Manager (within Change Management). 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 8-9 illustrates the Change Management process flow just described.

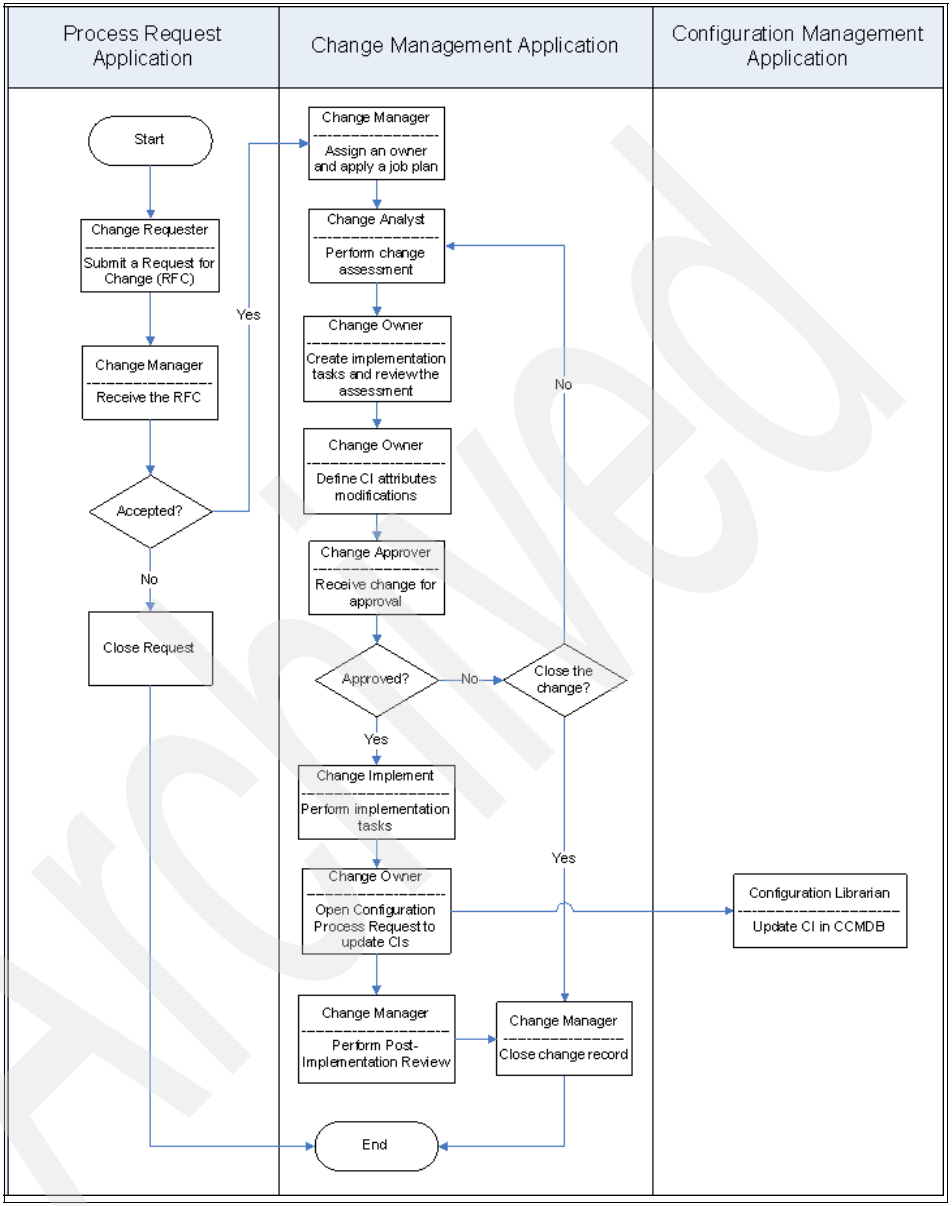


Figure 8-9 Change Management process overview

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 may fulfill many different roles. Roles are not individuals, nor are they necessarily equivalent to job titles; instead, they describe how individuals assigned to the roles will behave.

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:

- Change manager

  The Change Manager is primarily responsible for the overall quality of the Change Management process. He 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 him in his 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 a RFC and assigning it to one of the approval 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 a RFC. A 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 different 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 8-10 illustrates Change Content/Layout.

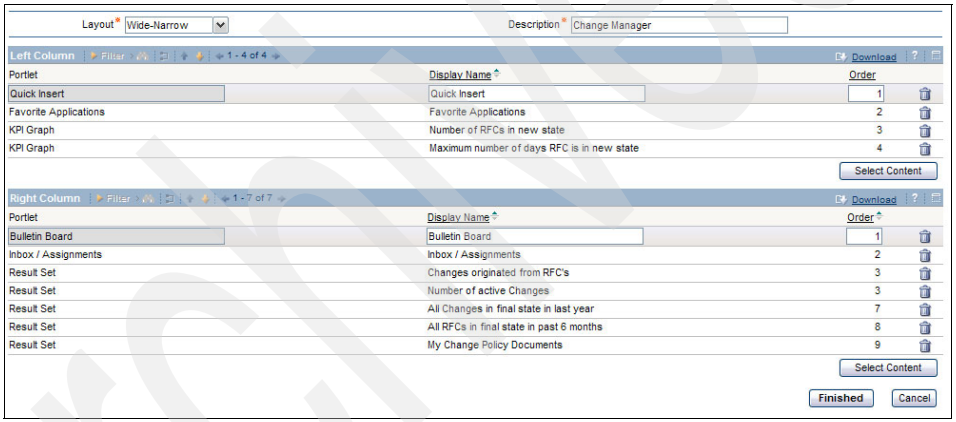


Figure 8-10 Customize Start Center

Some Change Management roles are illustrated in the following figures.

Figure 8-11 illustrates a Start Center for Change Manager role.

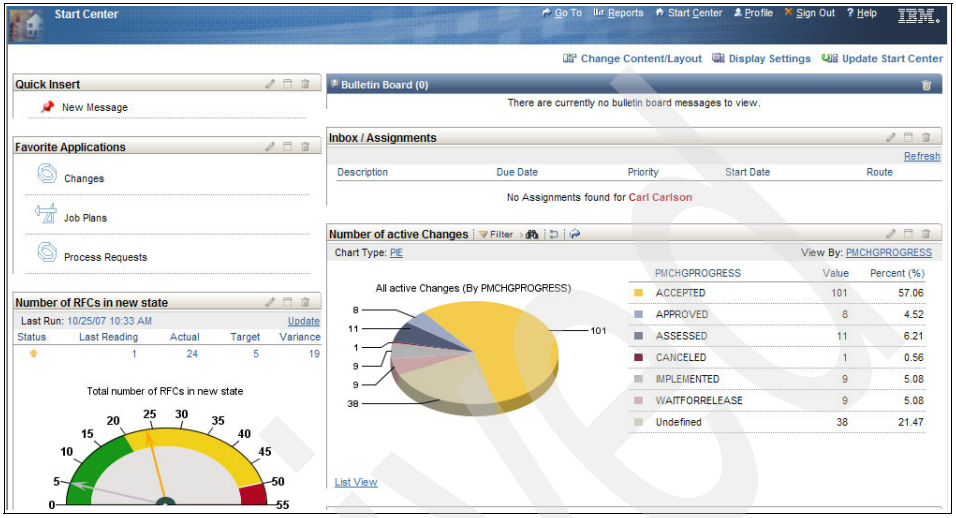


Figure 8-11 Change Manager Start Center

Figure 8-12 illustrates a Start Center for Change Administrator role.

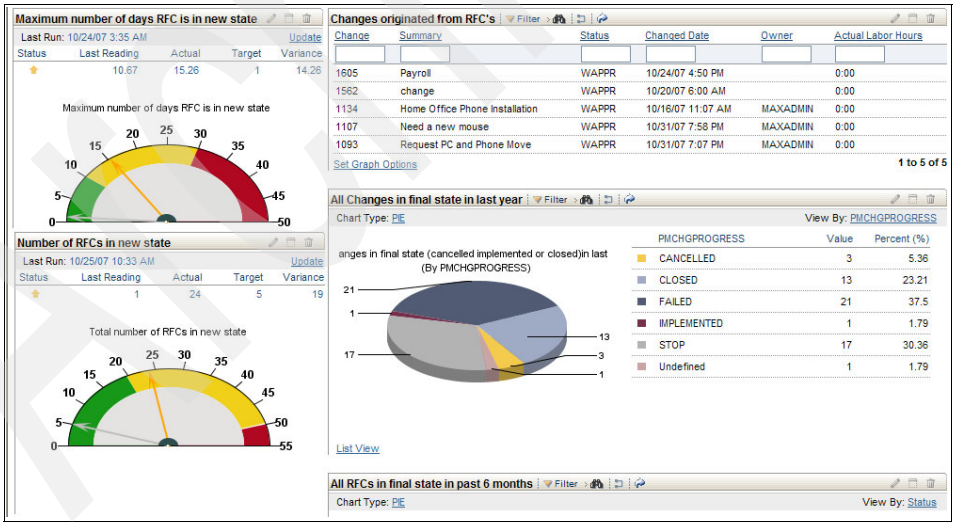


Figure 8-12 Change Administrator Start Center

Figure 8-13 on page 251 illustrates a Start Center for Change Owner role.

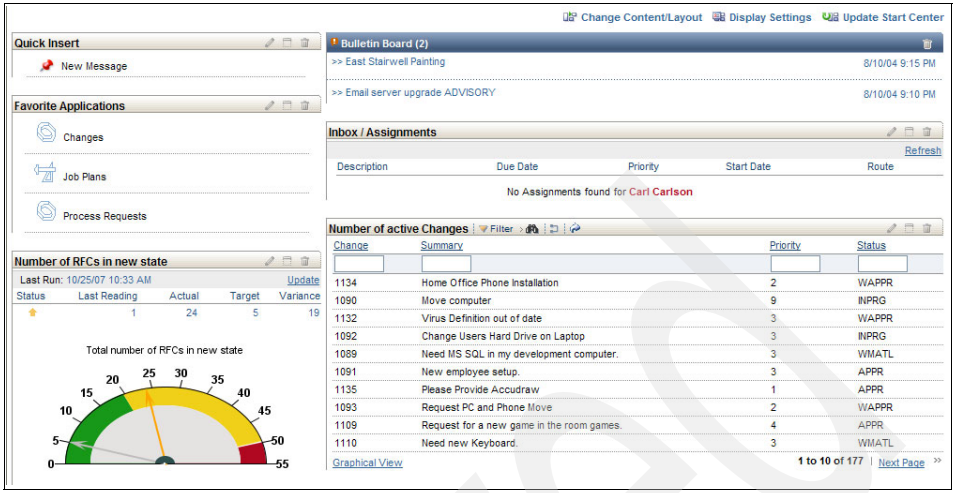


Figure 8-13 Change Owner Start Center

Figure 8-14 illustrates a Start Center for Change Analyst role.

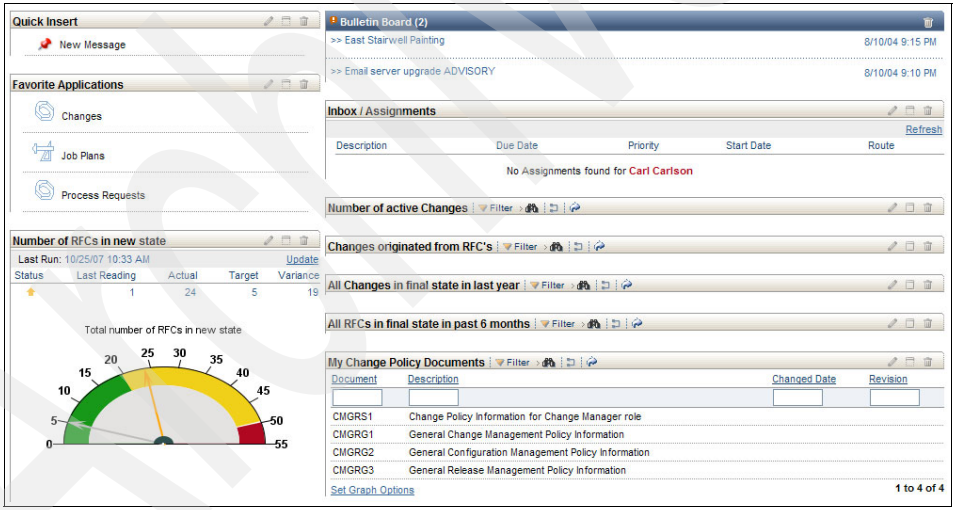


Figure 8-14 Change Analyst Start Center

8.2.2 Change Process Step-By-Step Within CCMDB

This section provides a step-by-step explanation about how the Change Management process works in CCMDB.

Some functions are more complex and will be further explained in detail.

Submitting a 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 8-15 on page 253 illustrates a Submit Process Request.

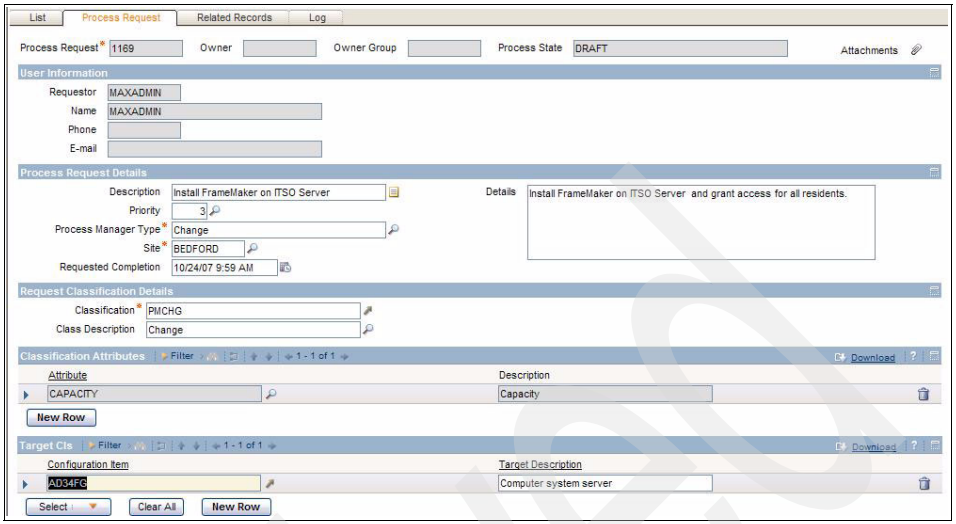


Figure 8-15 Submit Process Request

Accepting or rejecting the RFC

Responsible role: Change Manager

The Change Manager receives the request for change in the queue displayed in his Start Center. The Change Owner opens the request and reviews it, sees that it meets the basic requirements, accepts the request, and assigns a 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 a RFC is accepted, it becomes a change record.

It is important to 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 8-16 illustrates a RFC queue.

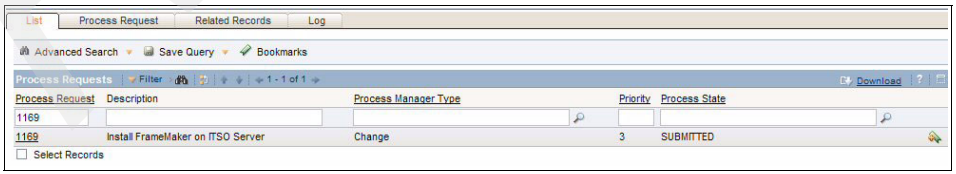


Figure 8-16 RFC Queue

Assigning a Change Owner

Responsible role: Change Manager

The change record will be displayed to the Change Manager in his 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 different 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 8-17 illustrates this option.

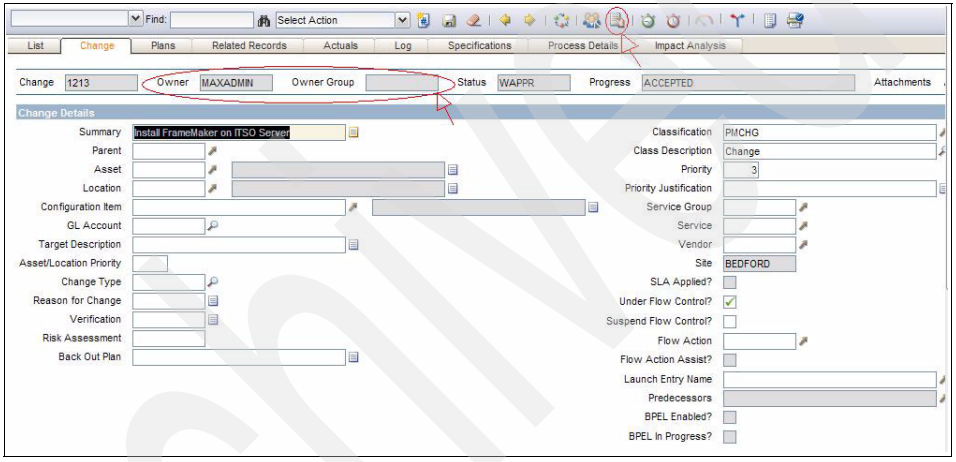


Figure 8-17 Take Ownership

- The Change Manager assigns the Change Owner.

  Click the **Select Owner** button and then select the Change Owner. The change will be assigned to the selected name.

  Figure 8-18 on page 255 illustrates the Select Owner window.

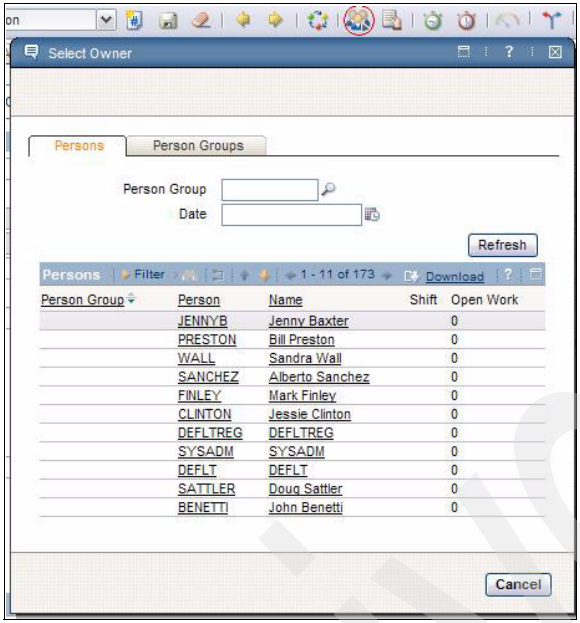


Figure 8-18 Select Owner

Selecting an appropriate Job Plan

Responsible role: Change Owner

The Change Owner opens the change in the Changes application. He 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.

Select the Change Owner, go to Plans tab, and select a Job Plan (Figure 8-19).

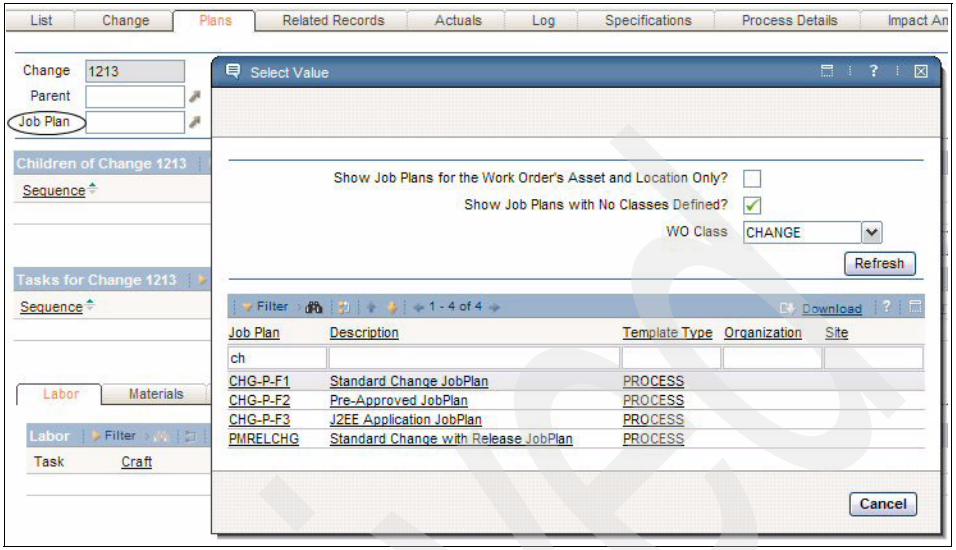


Figure 8-19 Select Job Plan

Note: We have explained how to manually apply a Job Plan to a change object. 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.

To add a new Activity, click the **New Activity** button.

To add a new Task, click the **New Row** button.

Figure 8-20 on page 257 shows the Job Plan applied.

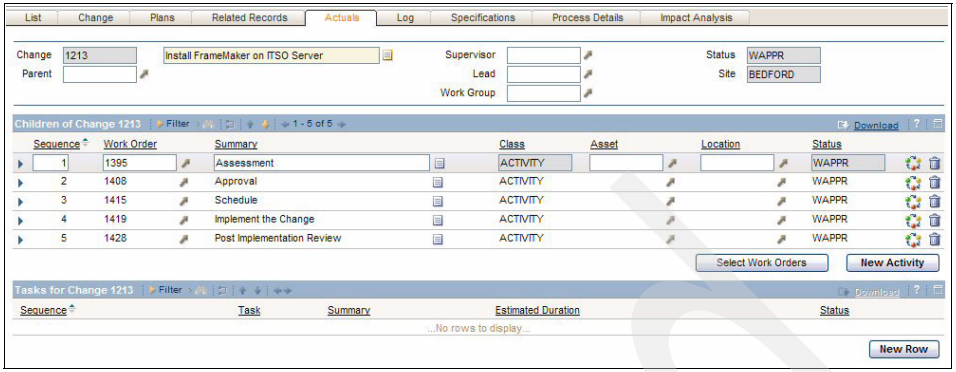


Figure 8-20 Job Plan Applied

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 8-21.

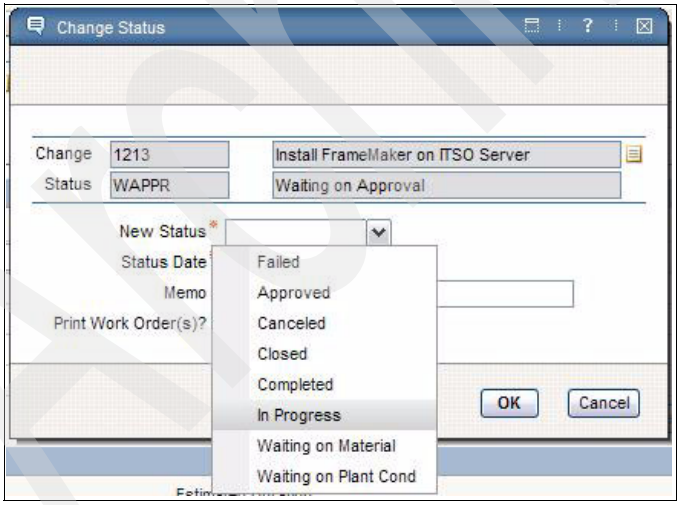


Figure 8-21 Change Status

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 Impact Analysis tab and select the **Technical Assessment Results** tab.

Each Change Analyst has to click the **New Row** button to add an assessment analysis (Figure 8-22).

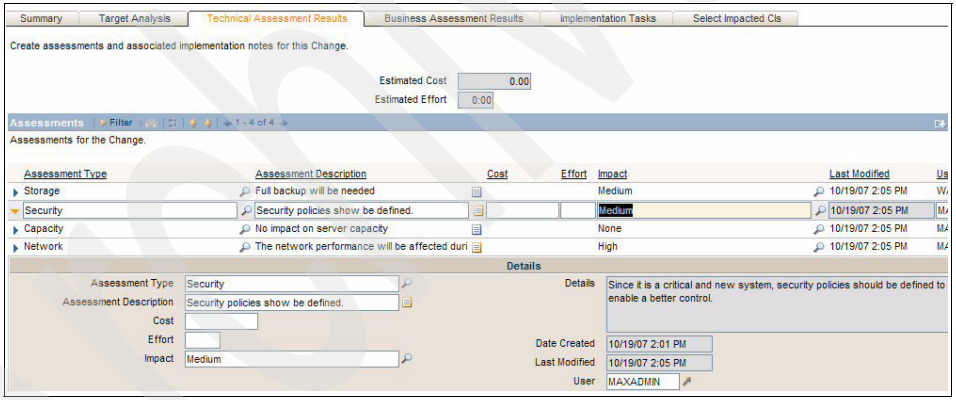


Figure 8-22 Technical assessment

Each technical assessment row has an implementation note associated with it. It should 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 **New Row** button in the Implementation Section (Figure 8-23 on page 259).

Figure 8-23 Technical assessment implementation notes

Route 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 complete in the Plan tab or an automatic task can be defined in the Job Plan (Figure 8-24).

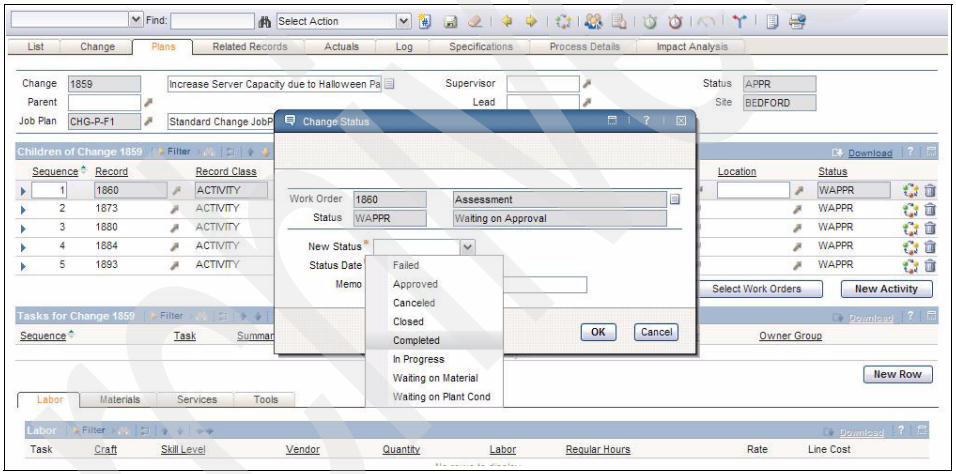


Figure 8-24 Route tasks

Performing 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 Impact Analysis tab and select the **Business Assessment Results** tab (Figure 8-25 on page 260).

Each Change Analyst has to click the New Row button to add an assessment analysis.

Note that Business Assessment does not have implementation notes since it is a business analysis and will not have implementation tasks associated.

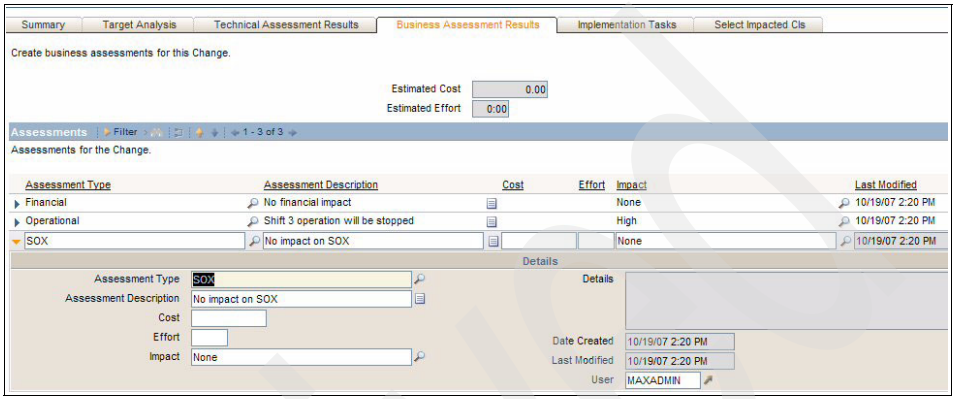


Figure 8-25 Business assessment

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 (Figure 8-26 on page 261).

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.

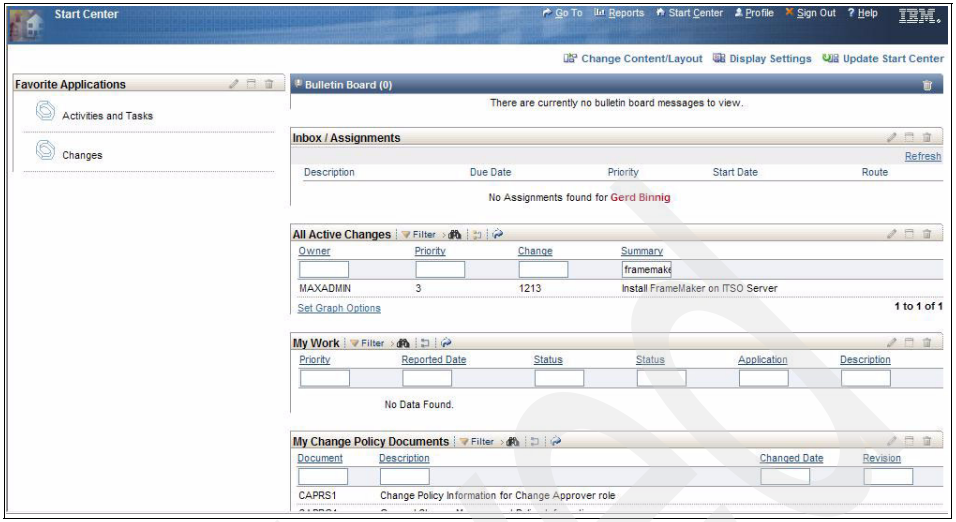


Figure 8-26 Change Approver Start Center

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** (Figure 8-27 on page 262).

Another way to request more information is by adding a new task for the change record and assigning it to the responsible person.

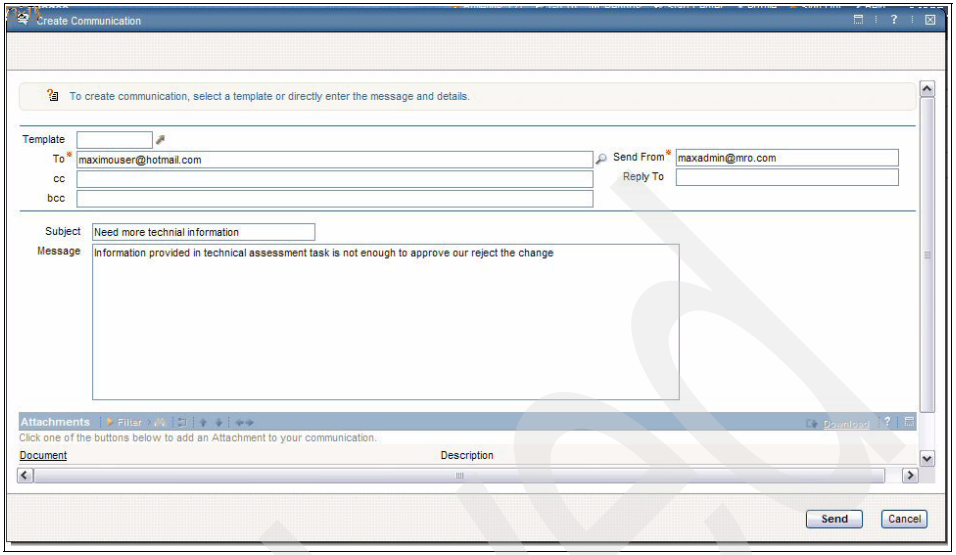


Figure 8-27 Send communication

To approve or reject a RFC, select the **Change Status** option from the Select Action menu (Figure 8-28).

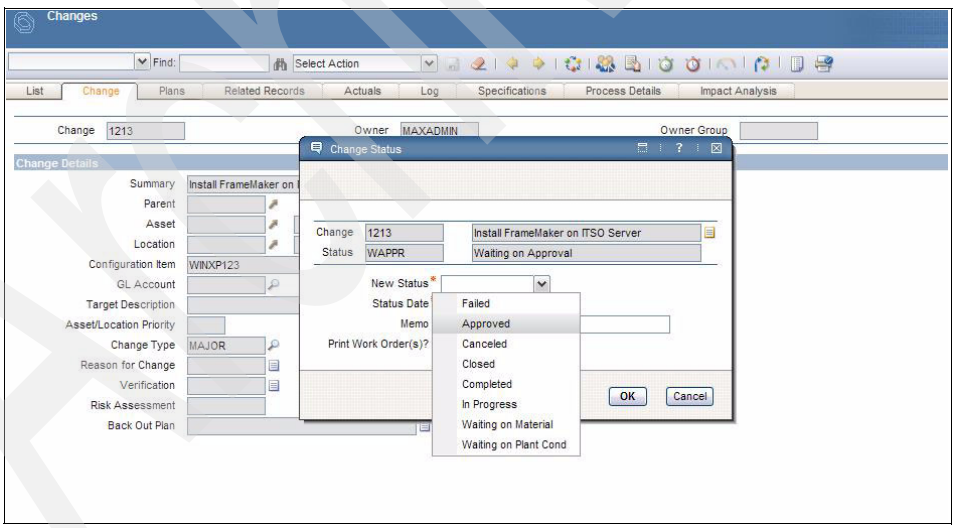


Figure 8-28 Approving a change

Creating implementation tasks

Responsible role: Change Owner

After the assessments and approvals 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, he chooses target CIs from the list of CIs identified in the original request. He then identifies impacted CIs (those that are affected even though they are not direct targets of the change).

To create implementation tasks, go to the Implementation Tasks tab and click **Create Tasks** (Figure 8-29).

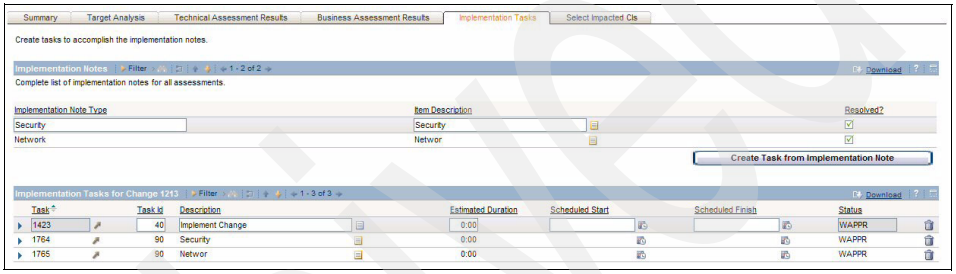


Figure 8-29 Create task from implementation notes

When scheduling implementation tasks, the Change Owner consults the Change Window application to determine whether some of the affected CIs have defined change windows, which specify the times when the CI can be taken out of service in order to make changes. He 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 should 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** (Figure 8-30).

Saving as plan means that the modifications will be done when the change is complete.

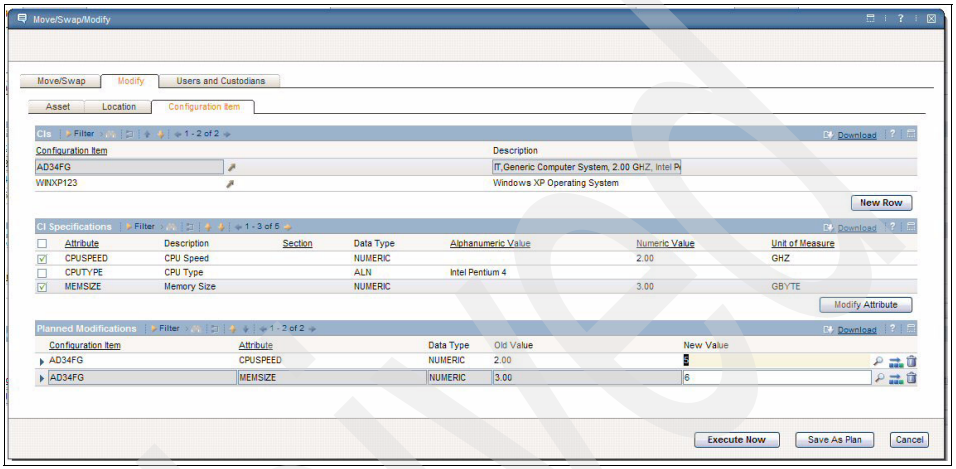


Figure 8-30 Defining CI attributes modifications

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 a owner. This information and others should be configured in the Activities and Tasks application (Figure 8-31 on page 265).

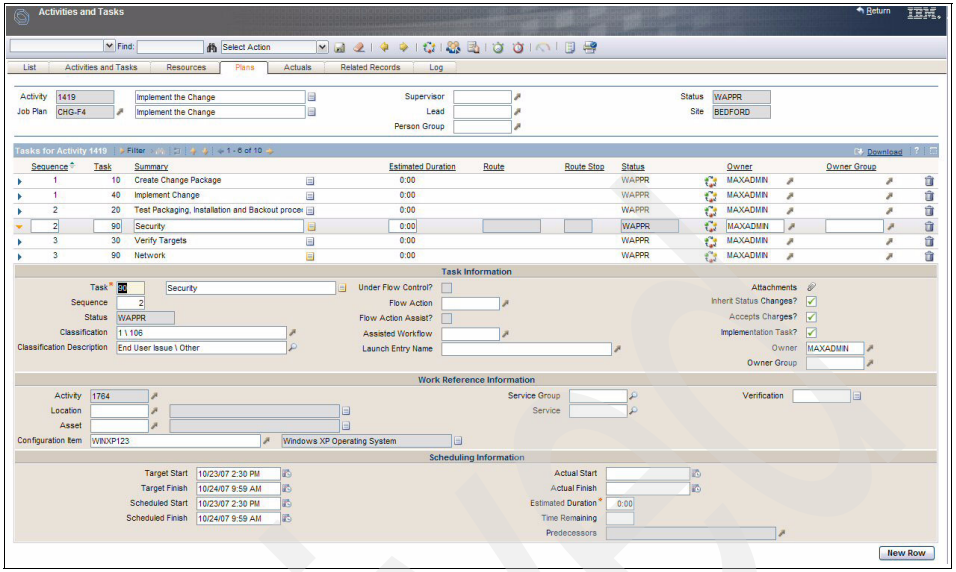


Figure 8-31 Activities and Tasks application

Note: In our examples, the tasks owners are always maxadmin, but in practice, these would be different persons.

There are three ways to access the Activities and Tasks application.

The first one is from the GO TO menu (Figure 8-32).

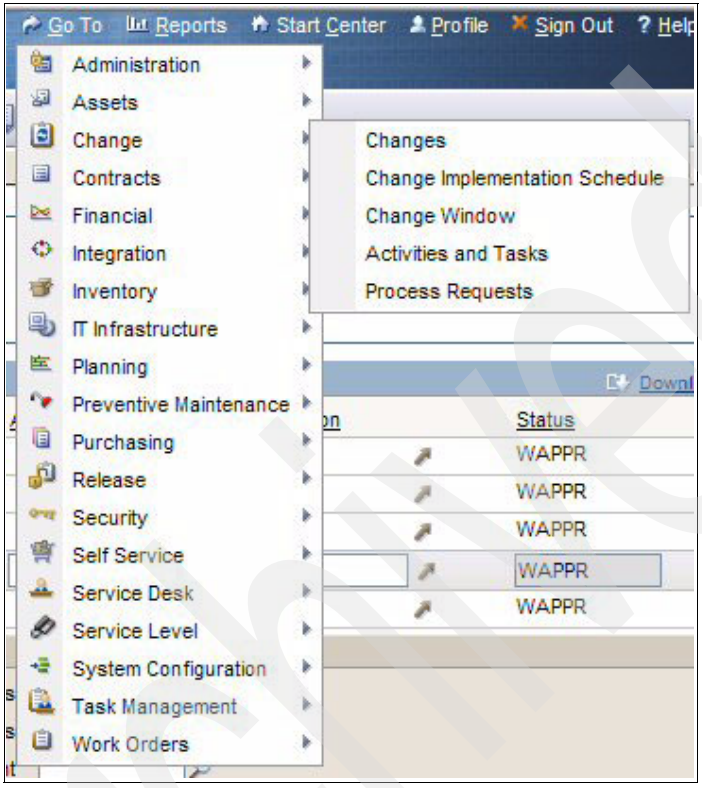


Figure 8-32 Activities and Tasks - option 1

The second one is from within the change in Plans tab. Choose the activity, click the arrow near Record field, and then click **Go To Activities and Tasks** (Figure 8-33 on page 267).

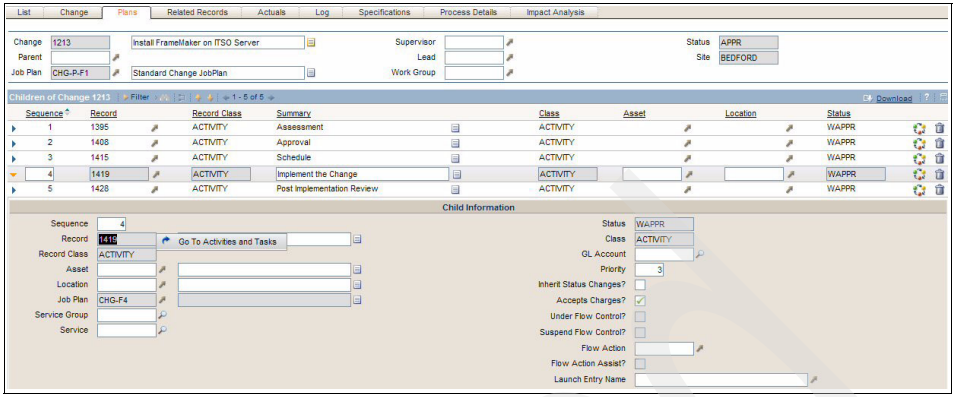


Figure 8-33 Activities and Tasks - option 2

The third one is from within the Implementation Tasks tab. Choose one task, click the arrow near the Reference WO field, and then click **Go To Activities and Tasks** (Figure 8-34).

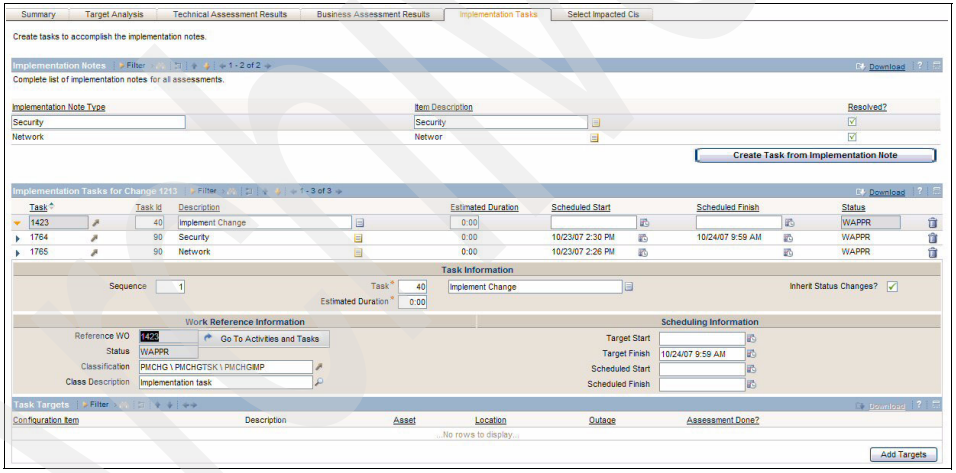


Figure 8-34 Activities and Tasks - option 3

Implementing the tasks

Responsible role: Change Owner

After the Change Owner has defined and approved the implementation tasks, the Change Implementer(s) should 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.

Some suggestions for tasks status are:

- 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** (Figure 8-35).

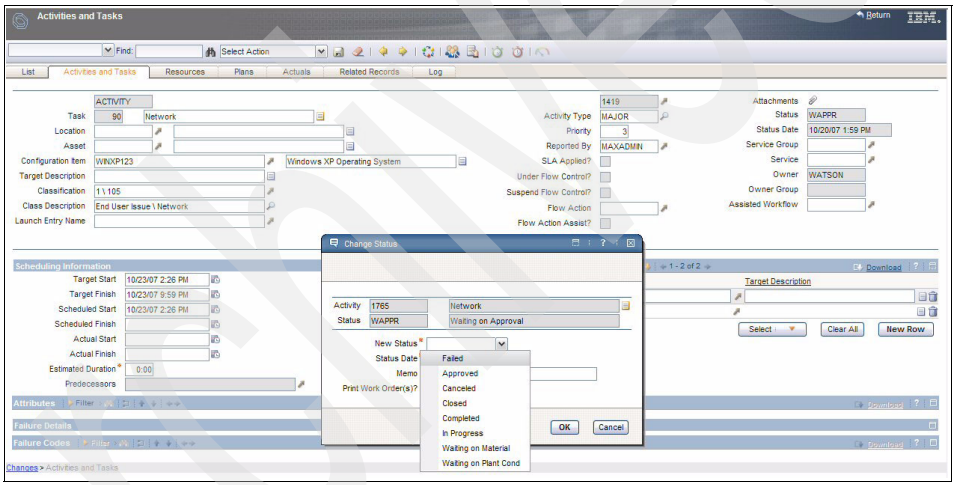


Figure 8-35 Update task status

The Change Implementer(s) should also update the details of each task implemented in the Log tab (Figure 8-36 on page 269).

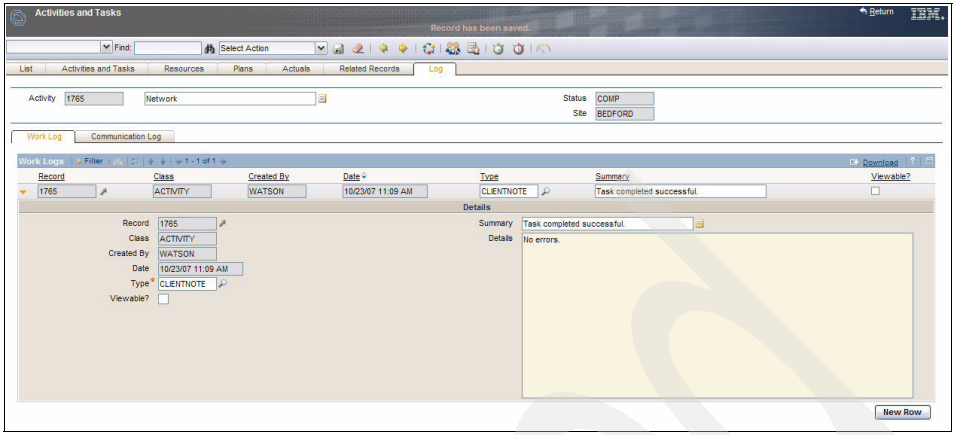


Figure 8-36 Work log task implemented

Submitting a Configuration Process Request

Responsible role: Change Owner

As discussed in “Defining CI attributes modifications” on page 263, a Configuration Process Request should 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** (Figure 8-37 on page 270).

For a detailed explanation of this function, refer to 8.4, “Interaction with other processes” on page 295.

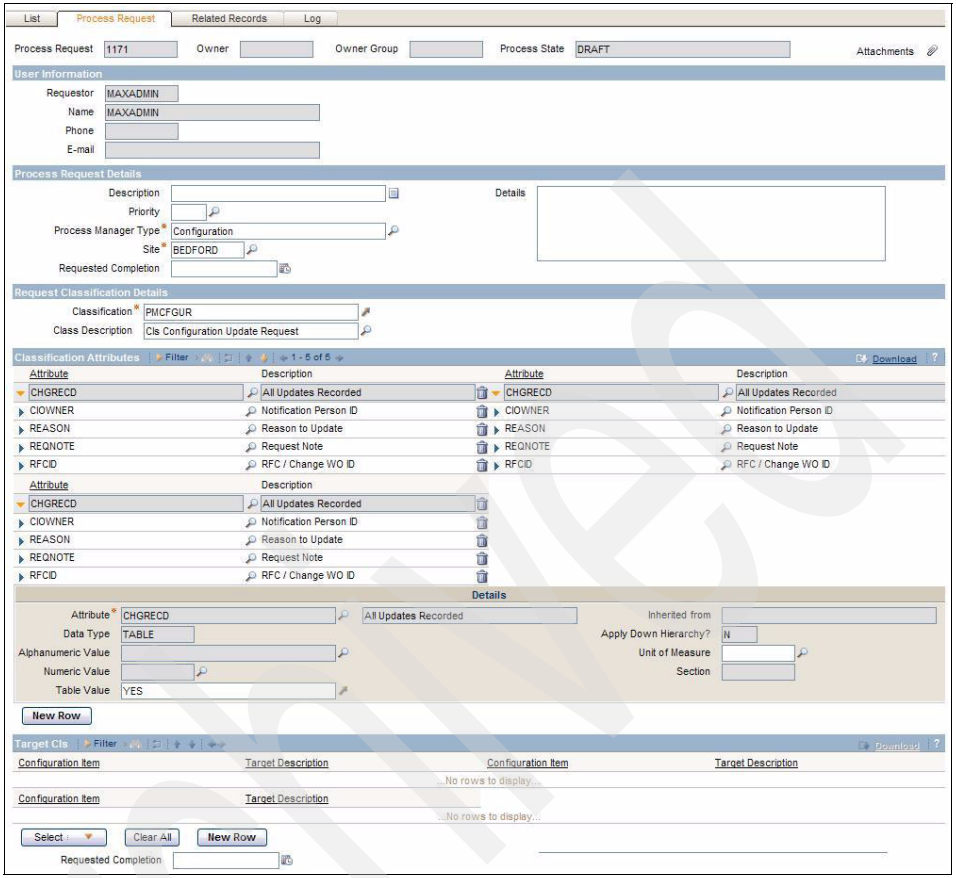


Figure 8-37 Create a configuration process request

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 should 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 (Figure 8-38).

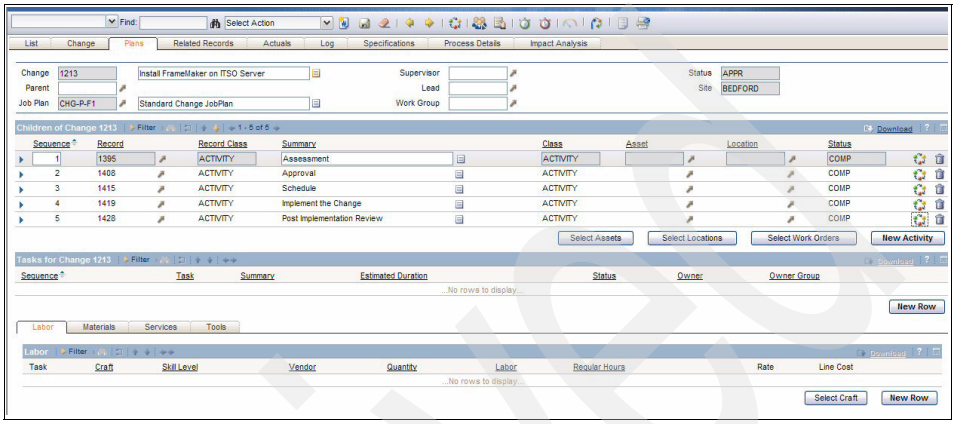


Figure 8-38 Post implementation review

Closing a 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 (Figure 8-39).

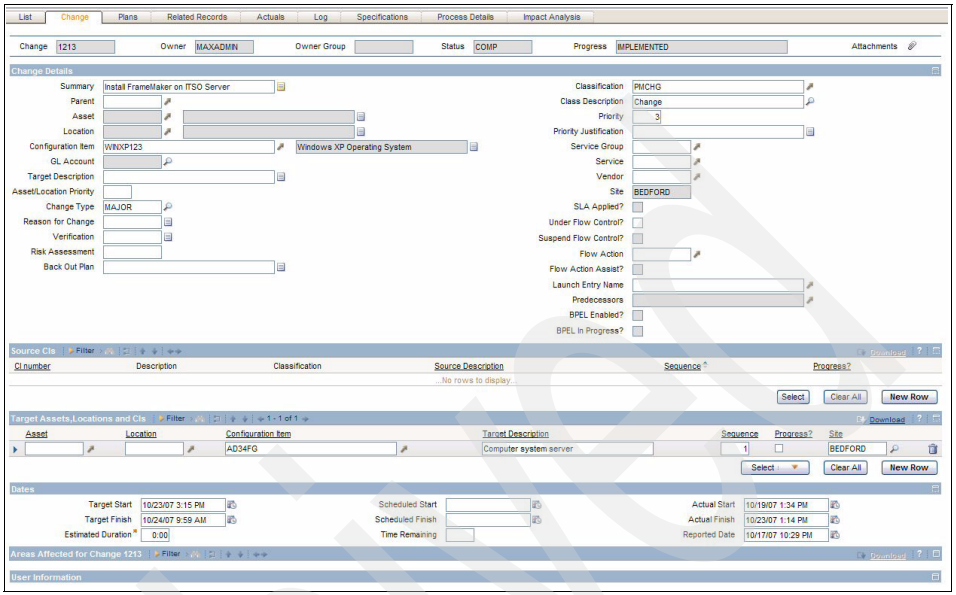


Figure 8-39 Closing a RFC

When the change status gets the status COMPLETE or CLOSE, the related Request For Change record in the Process Request application automatically gets the status COMPLETE (Figure 8-40).

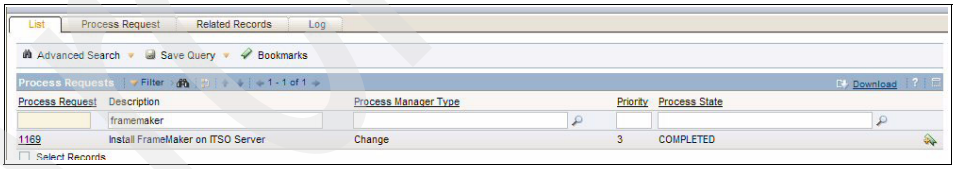


Figure 8-40 RFC Closed

8.3 Functions applicable to Change Management

This section provides detailed information about some of the most important functions that can be used with an RFC and change records.

8.3.1 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 8-41 illustrates a list of changes requests.

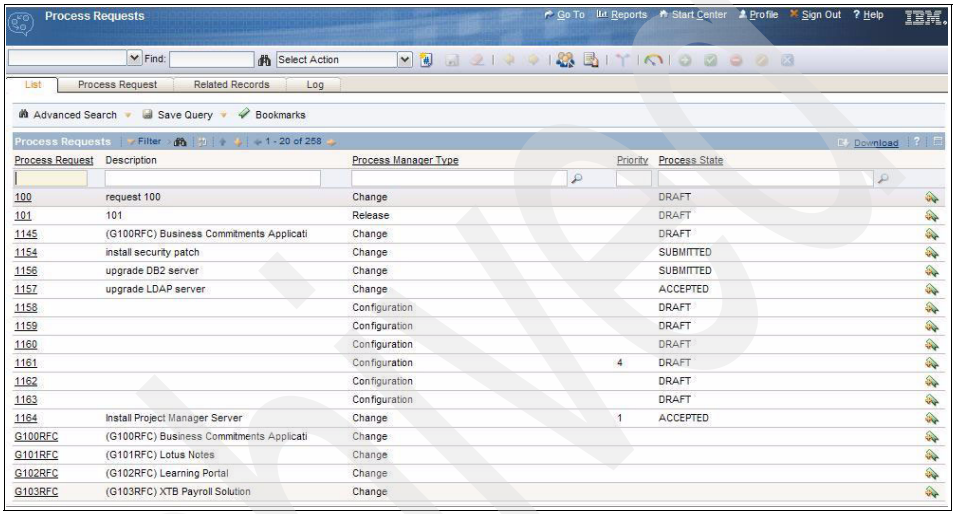


Figure 8-41 Process Request list

The Change Owner or the role whose responsibilities include making the initial response to requests for change should 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:

- Submit should be used to submit a Request for Change. The status of the request is changed to Submitted.
- Accept should be used to accept a Request for Change. The status of the request is changed to Accepted.
- Reject should be used to reject a Request for Change. The status of the request is changed to Rejected.
- Cancel should be used to cancel a Request for Change. The status of the request is changed to Canceled.

- Close should be used to close a Request for Change. The status of the request is changed to Completed.

Figure 8-42 shows the list of actions.

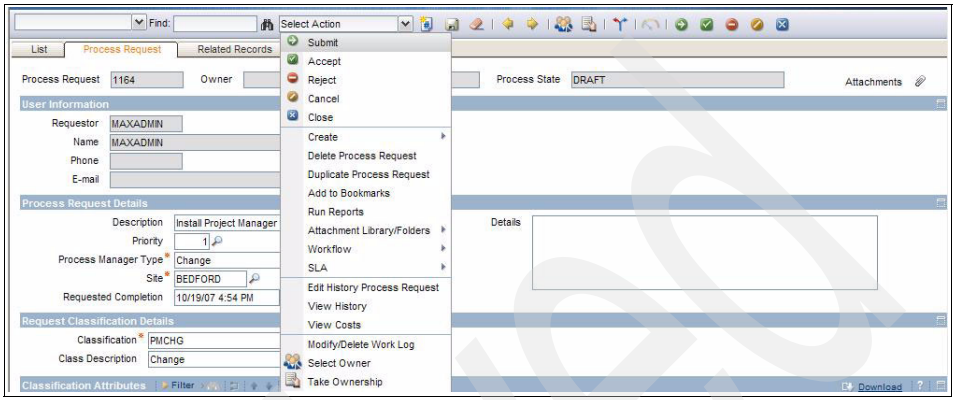


Figure 8-42 Process Request details

Each company should have rules that help decide how to respond to each request. These are some questions that could 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 should not 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 appears on lists of requests that require a response. If it was accepted, it appears on lists of changes that are in progress.

Depending on each company's process, it might need to:

- 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.

8.3.2 Change impact analysis

Impact analysis is an essential activity of the Change Management process because it is responsible for ensuring that a 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 may suffer some degradation of service due to that implementation (see Figure 8-43).

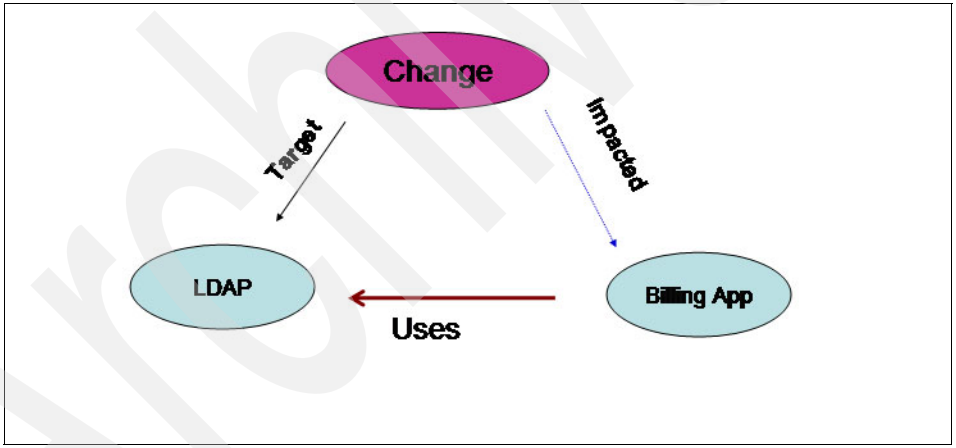


Figure 8-43 CIs target and impacted concept

Once all the consequences are documented, the subsequent steps of the process will use this information. For example, approvals and implementation scheduling will depend on accurate impact analysis 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; you should 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.

Impact Analysis tab

The Impact Analysis tab is shown when a change record is selected. It contains six sub-tabs to organize the impact analysis information and should be used by Change Analysts to both document the results of their assessments and identify CIs impacted by the change.

The sub-tabs are:

- Summary
- Target Analysis
- Technical Assessment Results
- Business Assessment Results
- Implementation Tasks
- Select Impacted CIs

Figure 8-44 illustrates the Impact Analysis tab; the use of each sub-tab is described later.

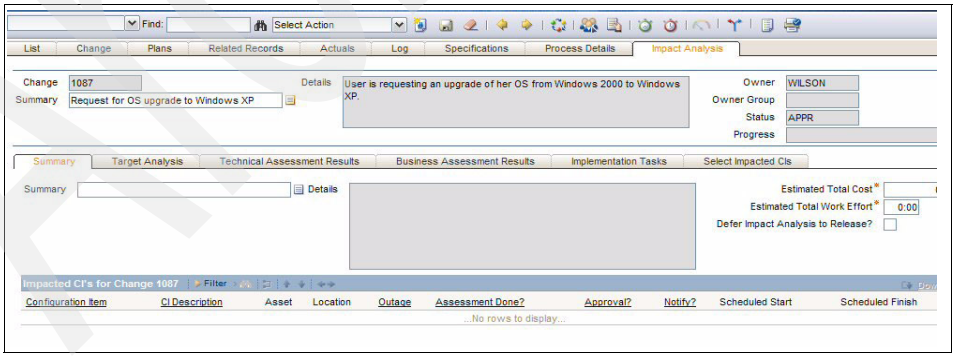


Figure 8-44 Impact Analysis tab

Summary sub-tab

The Summary sub-tab provides a high level view of the impact assessment results. It contains two sections: a summary of the impact, and a roll-up of the CIs identified as impacted by this Change (Figure 8-45).

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 may be all the impact assessment data required. This section also contains a roll-up of the Estimated Cost and Estimated Work Effort specified in the Technical and Business impact assessment results.

The Impacted CIs section contains a table listing all the impacted CIs associated with any of the implementation tasks in this Change.

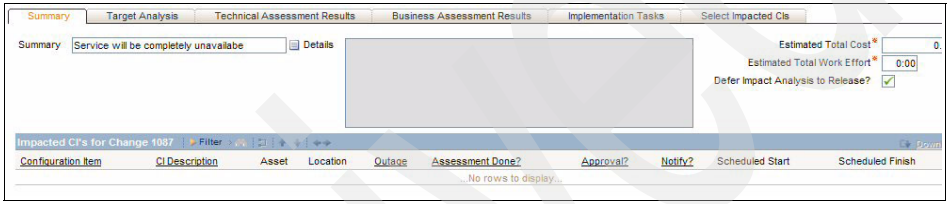


Figure 8-45 Impact Analysis - Summary sub-tab

Target Analysis sub-tab

The Target Analysis sub-tab is used to manage and analyze the targets of a Change. It contains three sections: the Change targets, the attributes of a selected target, and the relationships of a selected target (Figure 8-46 on page 278).

The Change Targets section allows the user to manage the set of target CIs for this Change. The user can view, add, and delete Change targets in this section. Note that managing target CIs of the Change can also be done in the Change tab; this capability is provided here as a convenience to those doing impact analysis.

The Target Attributes section allows the user to examine the attributes of the selected target. The user will select a target in the Targets section, and the CMDB attributes of that target will be displayed in the Attributes section. This is convenient, for example, if the analyst needs to check whether a target has the required amount of memory.

The Target Relationships section allows the user to examine the CMDB relationships that exist for a selected target. The user will select a target in the Targets section, and the CMDB relationships of that target will be displayed in the

Relationships section. All the relationships where the selected target is either the source or the target will be displayed.

Note: A very important field in Change Target tab is the OUTAGE field, where the user should inform the projected outage for each CI. Some examples of outage are:

- None: No outage.
- Degradation: Degraded performance.
- Offline: The configuration item needs to be offline.

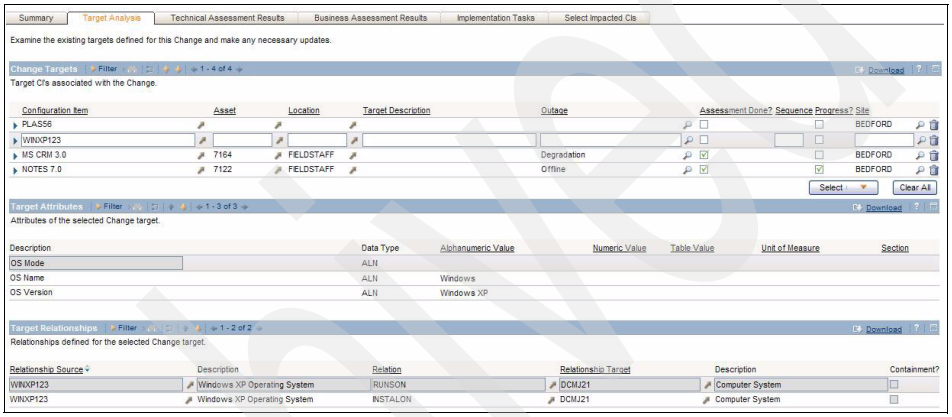


Figure 8-46 Impact Analysis - Target sub-tab

Technical Assessment Results sub-tab

The Technical Assessment Result sub-tab is used to capture assessment analysis performed by technical subject matter experts (SME), as well as implementation-related information the SME needs to pass along to those doing the implementation planning (Figure 8-47 on page 279).

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.

- User

  The user responsible for this assessment

While performing the technical assessment, each expert should use the Implementation Note feature to record items that will need to be considered during the implementation phase. These notes should 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 would need 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 may 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.

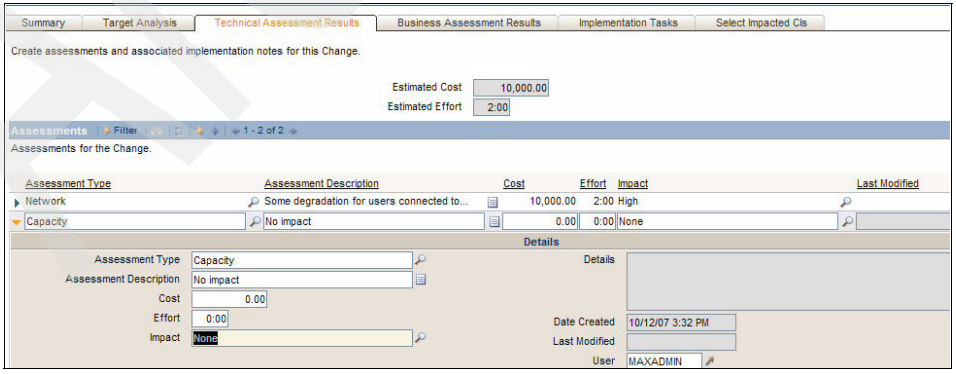


Figure 8-47 Impact Analysis - Technical Assessment Results sub-tab

Business Assessment Results sub-tab

The Business Assessment Results sub-tab is used to analyze and record the business impacts of the change. Business subject matter experts should 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 8-48 on page 281).

Each expert should complete these 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.

- User

  The user responsible for this assessment.

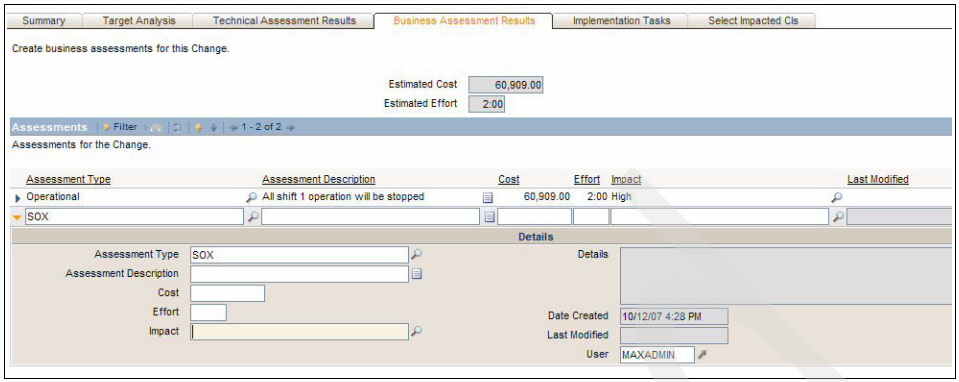


Figure 8-48 Impact Analysis - Business Assessment Results sub-tab

Implementation Tasks sub-tab

The main purpose of the Implementation Tasks sub tab 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.

Note:

- 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 Plans tab using the New Activity button. Creating the activities in this manner will result in them automatically being created as children of this Change.

At the top of the sub-tab, all the implementation notes created during technical assessment are displayed. Implementation tasks can be created from the Implementation Notes by highlighting a note, then clicking the **Create Task From Implementation Note** button. When the button is clicked, a dialog displays a list of the Change's child activities. The user selects an activity and a task associated with the implementation note is created in that activity.

Once implementation tasks are created, 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 must be opened to display the target list and Add Targets button. When the **Add Targets** 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.

As target CIs are added to the implementation tasks, the outage specification should 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 should have identified the outage level caused by the implementation work. This would have been 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 would need to be taken down during the update. Also, the current IT configuration would have been 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 may 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.

A Resolved check box is provided in the Implementation Notes table so the Change Analyst can indicate 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.

Figure 8-49 on page 283 shows the Implementation Tasks sub-tab.

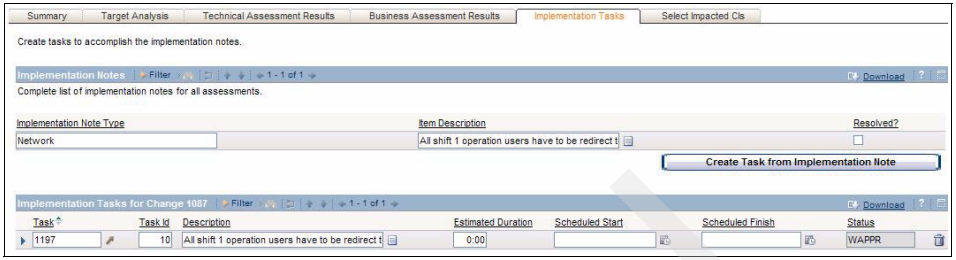


Figure 8-49 Impact Analysis - Implementation Tasks sub-tab

Select Impacted CIs sub-tab

After the implementation tasks and targets are defined, the CIs impacted by this Change will be identified on this tab. The identification of impacted CIs is a manual process. The Change Analyst will use information identified during the previous assessment steps to determine which CIs will be impacted.

Impacted CIs are associated with implementation tasks. There are a couple of reasons the impacted CIs are scoped to the implementation task rather than just the Change. First, it is the implementation work done when completing an implementation task that may 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 tis relationship to a target of the implementation task:

1. Select an implementation task from the list.
2. Select a target of that implementation task.
3. Use the **Selected Impacted CI from Task Target Relationship** button on the task target table.

This button will bring up 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 may be that even though a CI does not have a relationship defined in the CMDB with one of the implementation targets, a Change Analyst may know it will be impacted. In this situation, use the **Select from CIs** button on the impacted CI table to identify any CI from the CMDB as an impacted CI. It will bring up 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.

Once an impacted CI is identified, it may be that due to that impact, other CIs related to it may also be impacted by this implementation task. To see what CIs have a relationship to an impacted CI, use the **Select CI from Impacted Relationships** button on the impacted CI table to identify additional impacted CIs based on their relationship to the highlighted impacted CI. This will bring up a dialog listing all the CIs that have a relationship defined in the CMDB with the selected impacted CI. One or more CIs can be selected from this list. The CIs selected from the dialog will be added to the impacted CI list for the highlighted implementation task.

Figure 8-50 shows the Selected Impacted CIs sub-tab.

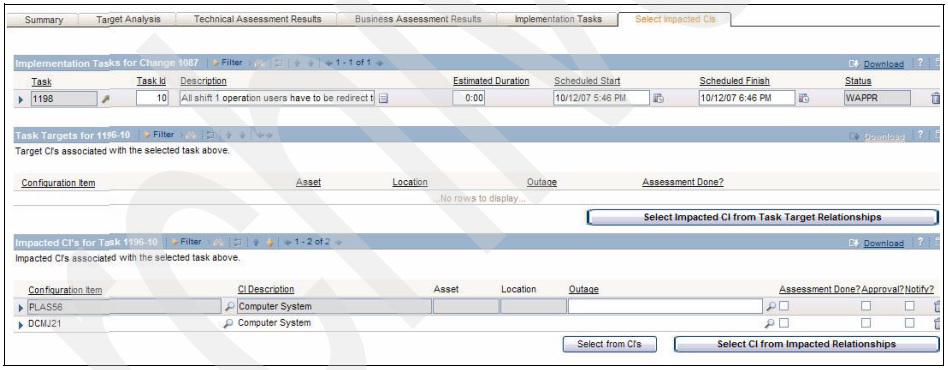


Figure 8-50 Impact Analysis - Selected Impacted CIs sub-tab

8.3.3 Change Management Schedule

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.

Why use 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 could 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 changes in four different ways:

1. Calendar view
2. Time window view
3. CI view
4. CI collection 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 will appear showing the scheduled start and end times, the change number and description, the owner of the change, and a description of the task. Several 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 8-51).

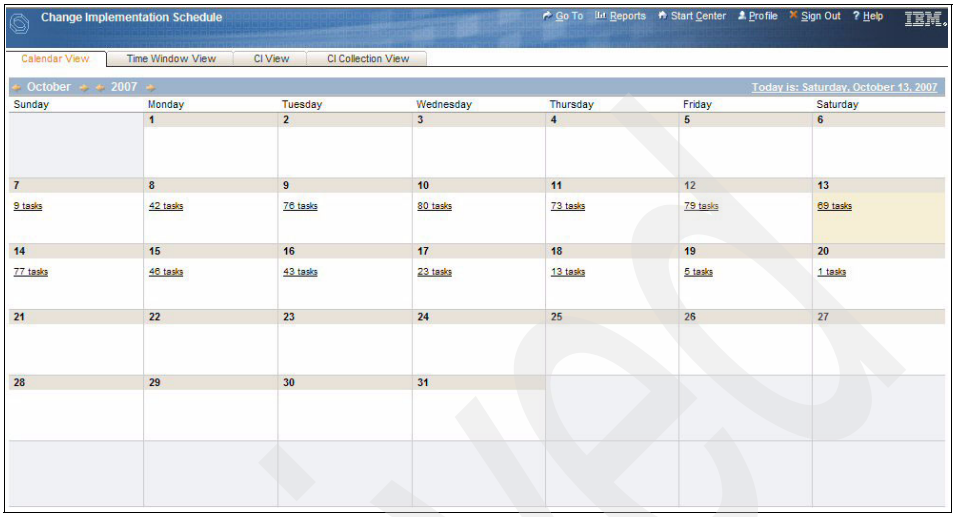


Figure 8-51 Change Implementation Schedule: Calendar View

The tasks scheduled for a day are shown in Figure 8-52.

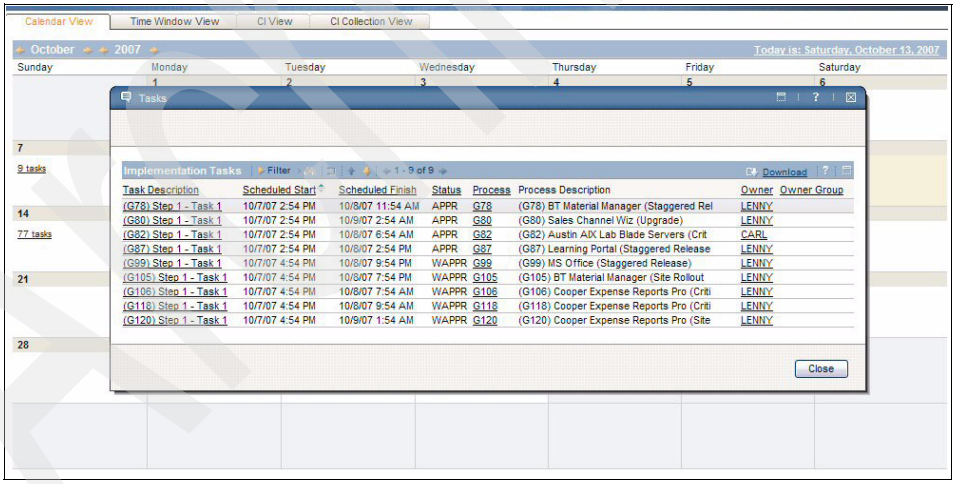


Figure 8-52 Change Implementation Schedule: Tasks Scheduled

Time Window View

The Time Window View shows the number of scheduled implementation tasks each day during a period specified.

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 8-53).

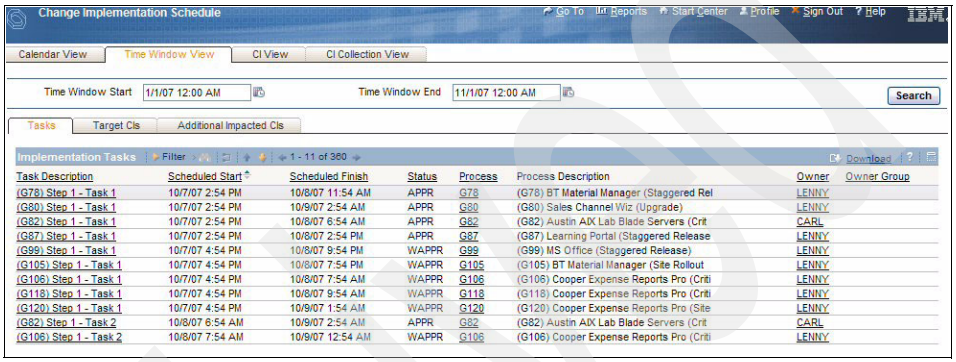


Figure 8-53 Change Implementation Schedule: Time Window View by Tasks

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 8-54 and Figure 8-55 on page 288).

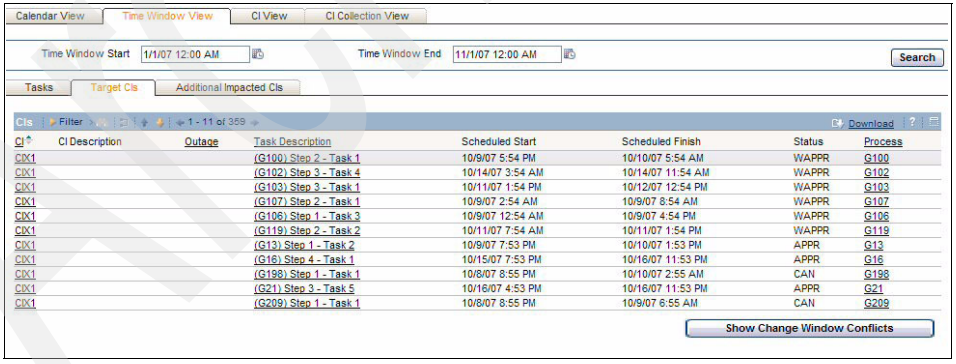


Figure 8-54 Change Implementation Schedule: Time Window View by Target CIs

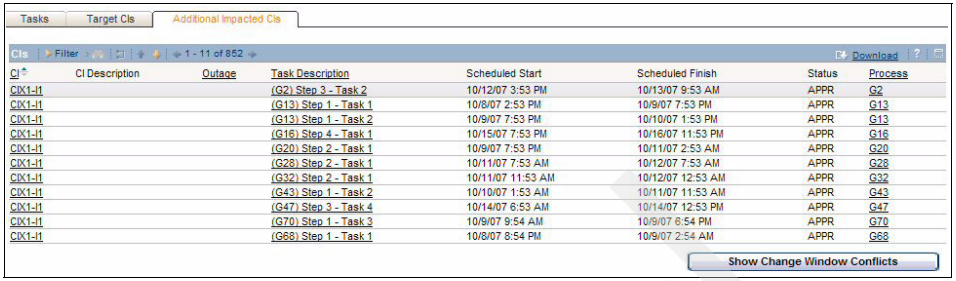


Figure 8-55 Change Implementation Schedule: Time Window View by Additional Impacted CIs

CI View

The CI View shows the scheduled implementation tasks for a configuration item.

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 8-56).

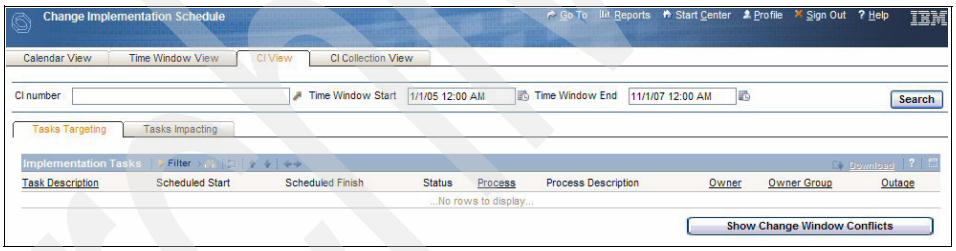


Figure 8-56 Change Implementation Schedule: CI View by Tasks Targeting

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 8-57).

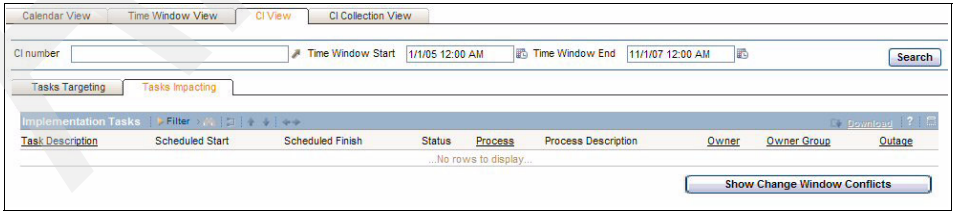


Figure 8-57 Change Implementation Schedule: CI View by Tasks Impacting

CI Collection View

The CI Collection View shows the scheduled implementation tasks for a collection of configuration items.

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 8-58).

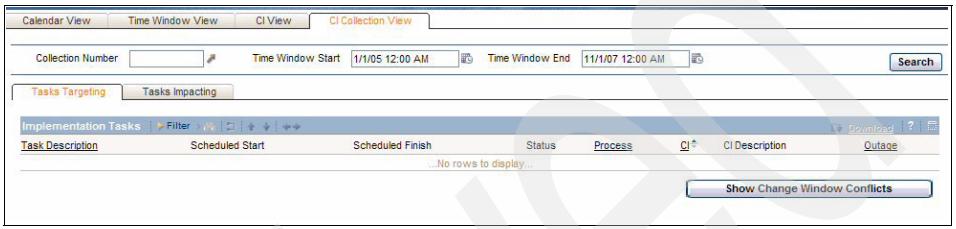


Figure 8-58 Change Implementation Schedule: CI Collection View by Tasks Impacting a CI

8.3.4 Change Window

Change Window is a central repository for negotiated maintenance windows, and should 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 may 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 8-59 illustrates the Change Window concept.

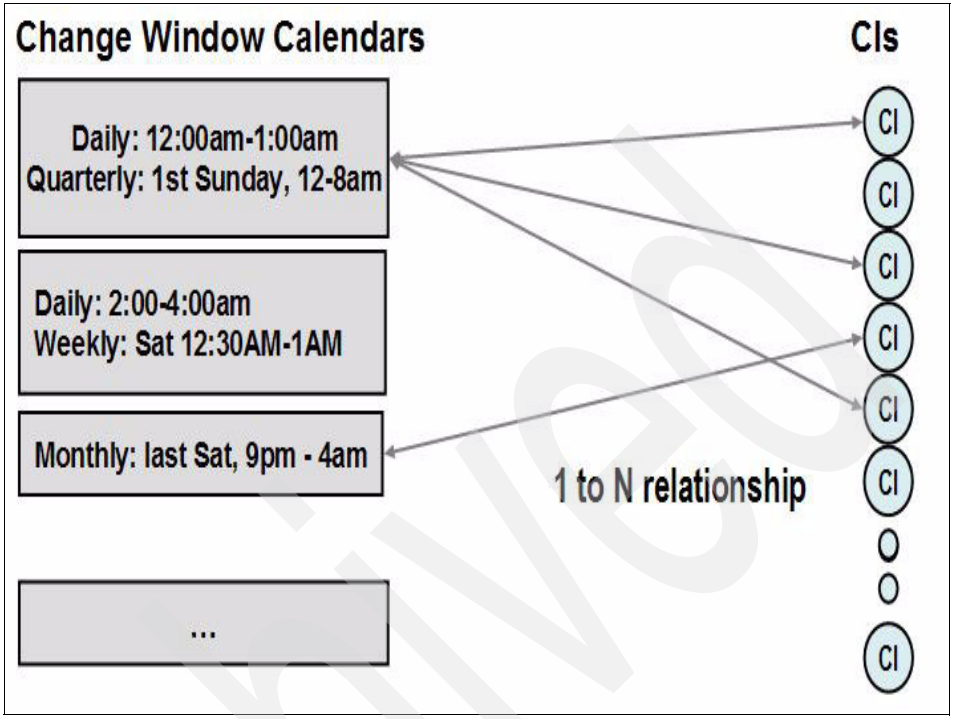


Figure 8-59 Change Window concept

Follow these instructions to work with Change Window:

1. Select **Change Application** → **Change Window** to open the Change Window application.
2. Click the **Change Window** 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 will appear in which you can specify the type, date, start, and stop times, and any notes you want to enter.
4. While using either of these views, you can also open a change window to modify it, or delete a change window.

Figure 8-60 on page 291 gives an overview of the Change Window fuctionality.

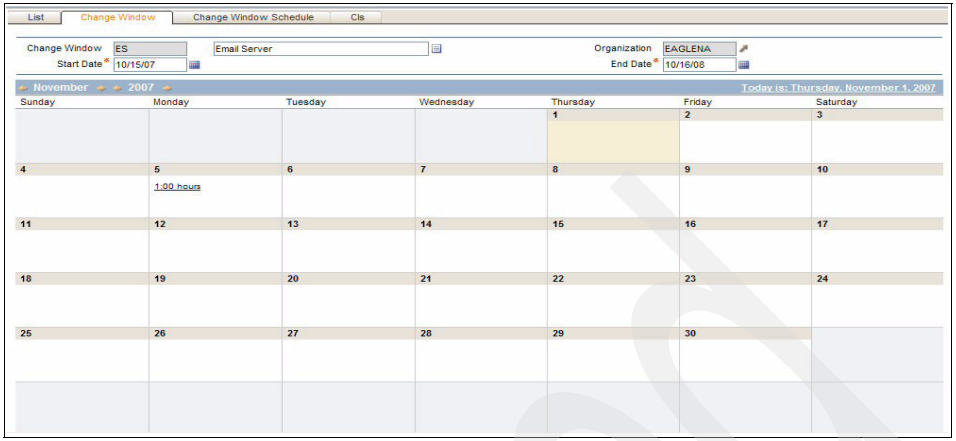


Figure 8-60 Change Window functionality

Change Window Conflicts

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** (Figure 8-61).

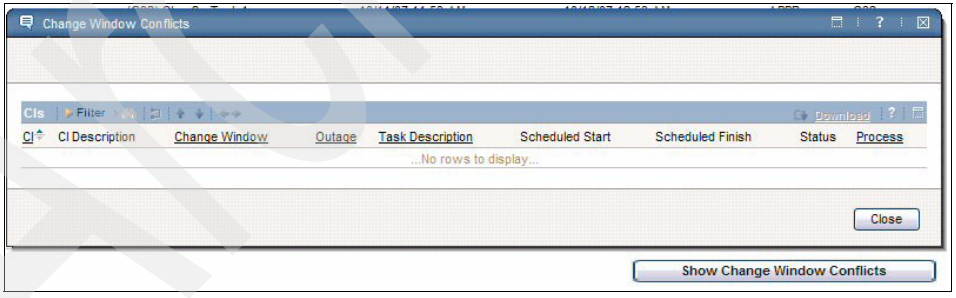


Figure 8-61 Change Implementation Schedule - Window Conflicts

Another way to verify change window conflicts is while viewing a change in the Changes application, which can be done by selecting **Schedule Conflicts** → **Target CIs not in Change Window or Schedule Conflicts** → **Impacted CIs not in Change Window**. A dialog box will appear showing the details for each task whose start or end time falls outside the change window for a target or impacted CI:

- Scheduled start and end times of the task
- CI identifier and description
- Task description

The fields appears as links that can be clicked to find more details.

Figure 8-62 and Figure 8-63 illustrate the change window conflicts.

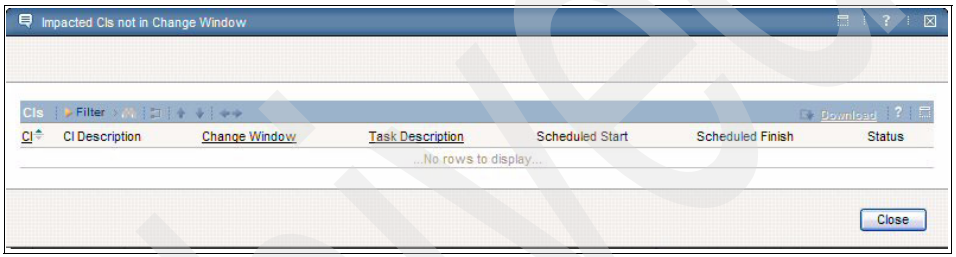


Figure 8-62 Impacted CIs not in Change Window

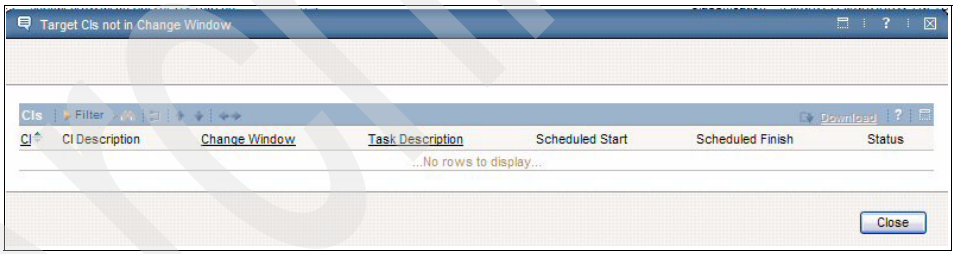


Figure 8-63 Target CIs not in Change Window

8.3.5 Tracking the progress of a change

The progress of a change is tracked through the following fields:

- Progress: It shows what phase of the cycle is the record is in and its overall progress.
- Status: It shows the sequence of the status.

The progress and status of a change should be updated after each completed phase (Figure 8-64 and Figure 8-65).

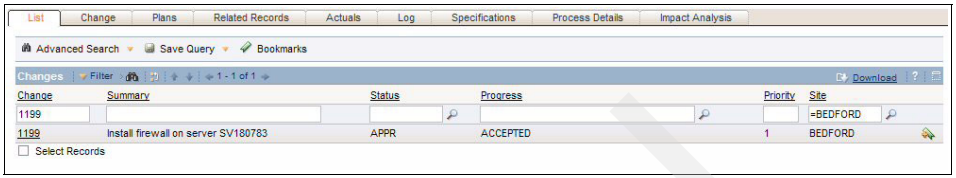


Figure 8-64 Change Progress Attribute - Change List

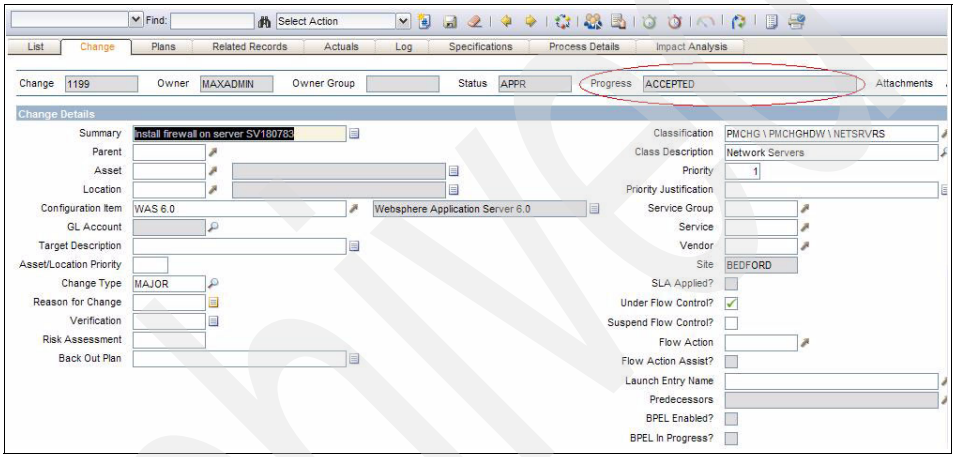


Figure 8-65 Change Progress Attribute - Change Detail

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. Refer to 8.1.3, “Job Plan” on page 238 to see 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 (Figure 8-66 on page 294).
   d. If necessary, enter a comment related to the modification
   e. Click **OK**.

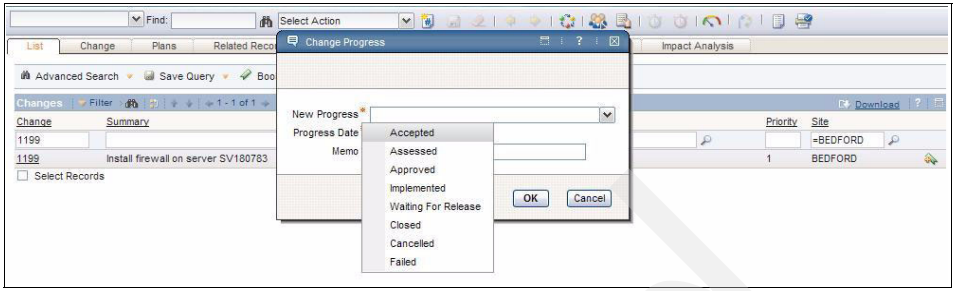


Figure 8-66 Change Progress list

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 8-67).

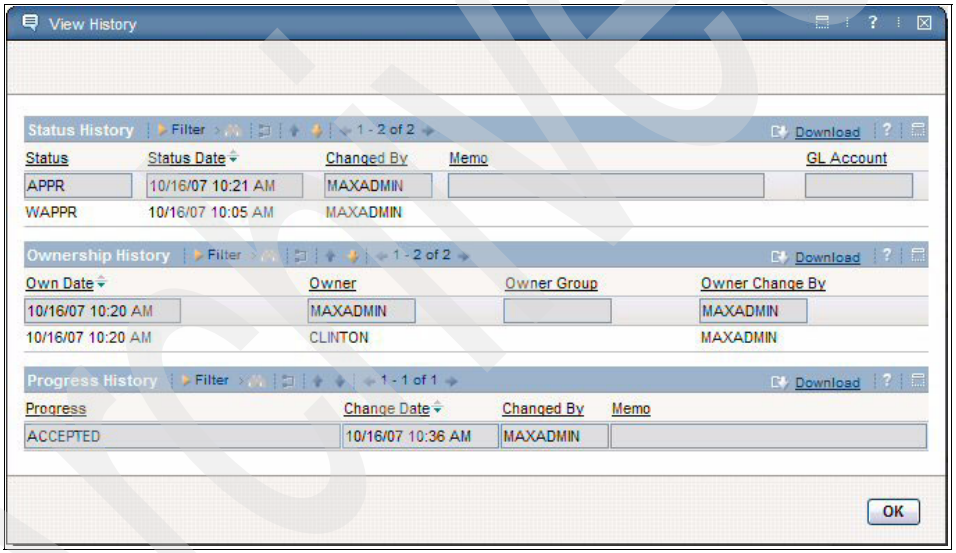


Figure 8-67 Change Progress History

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 8-68).

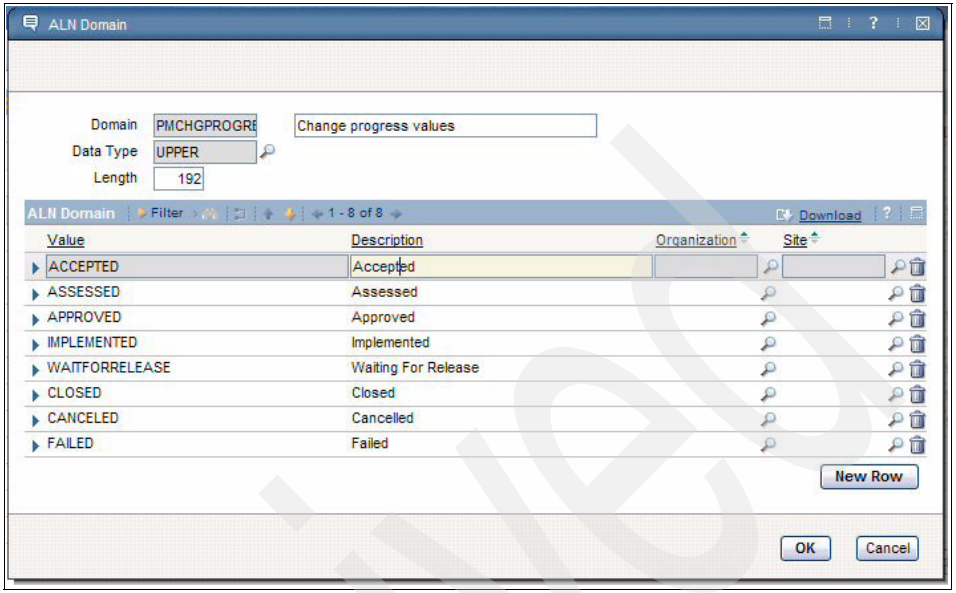


Figure 8-68 Modifying change progress list in domain application

8.4 Interaction with other processes

This section describes some of the 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.

Update CI attributes After a change implementation

CI attributes modifications should be specified in the Move/Swap/Modify option in the Select Action menu, and then a request should 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 would not be associated with the change record since the CI attributes update will occur in a different 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.

Note: If a change were 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.

The following steps should 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 (Figure 8-69 on page 297).
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.

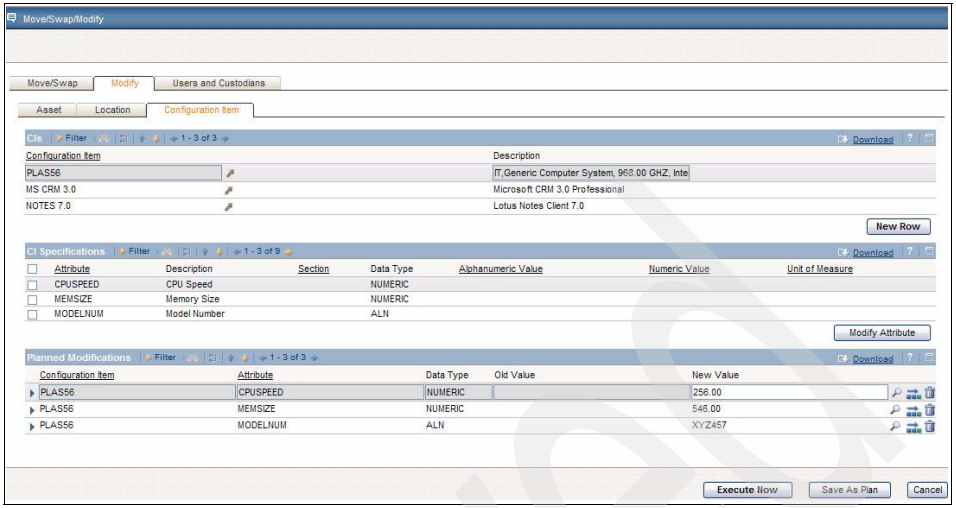


Figure 8-69 Move/Swap/Modify

5. From within the change, create a process request to Configuration Management to update the CI. In the new process request (Figure 8-70 on page 298), do the following:

   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.

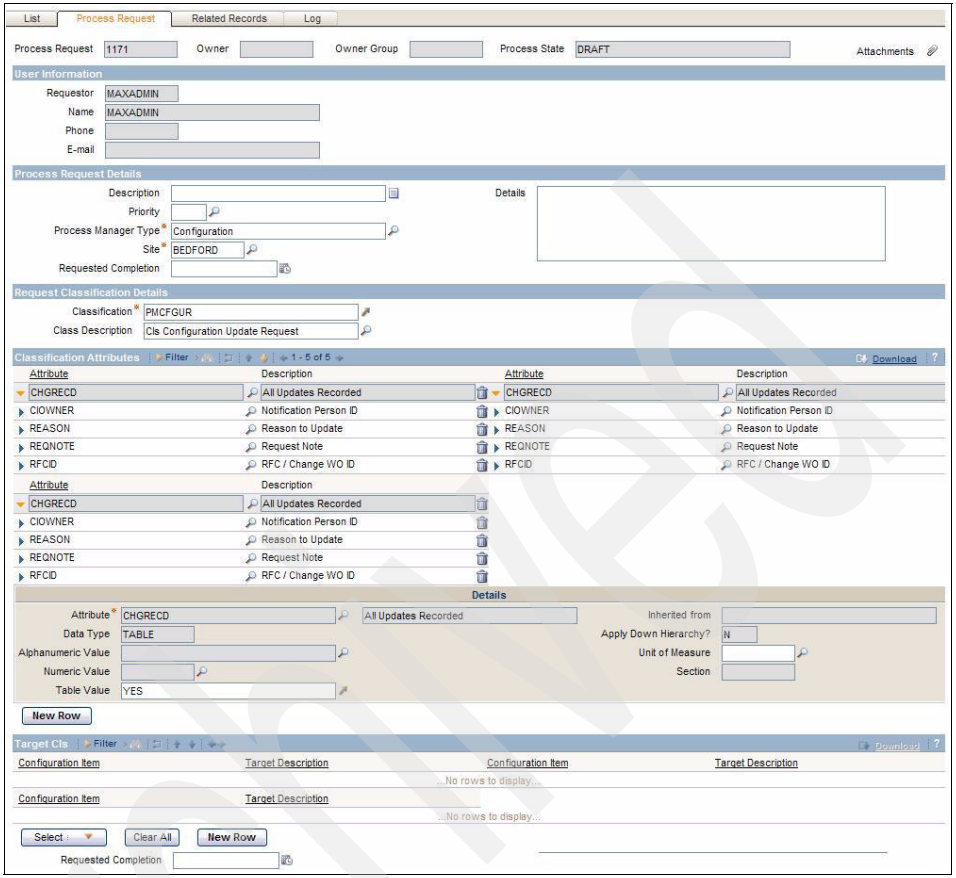


Figure 8-70 CI Update Process Request

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 could 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 CCMDB, 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 basics actions that can be performed to integrate changes and releases. These actions can be accessed by selecting **Select Action** → **Release Requests**.

Add a Change to a Specific Release

The Add Change to Release dialog box should be used to request that a specific Release handle a Change (Figure 8-71 on page 300).

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.

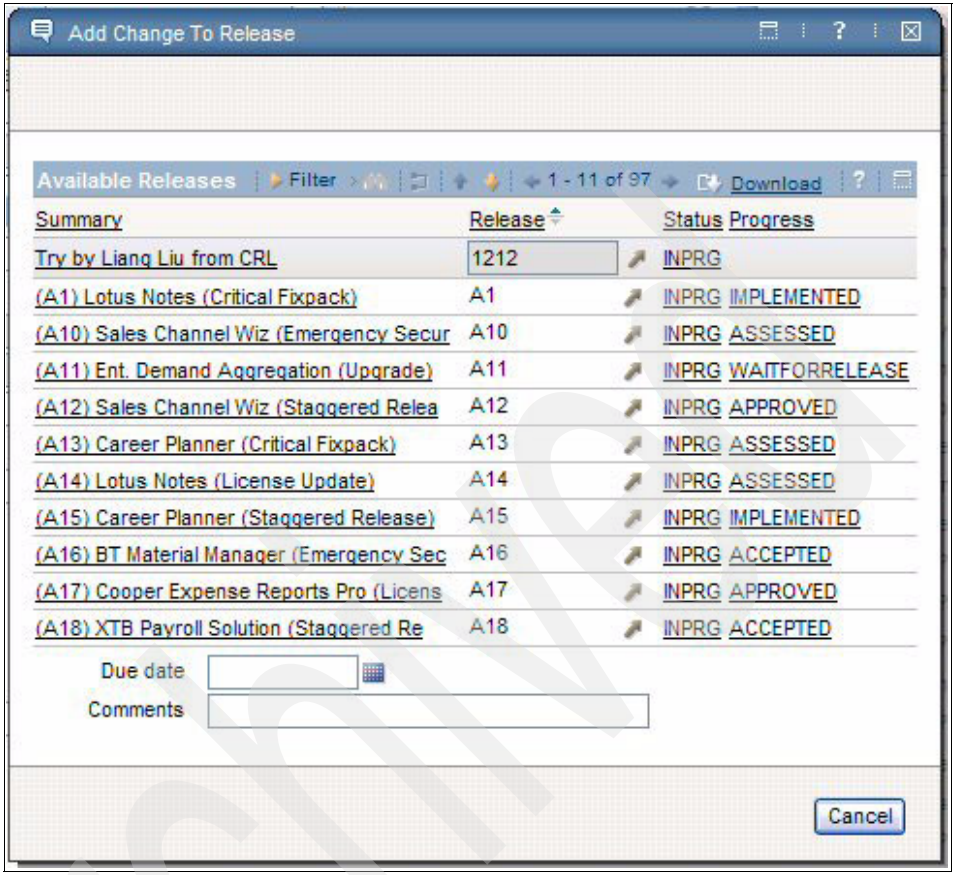


Figure 8-71 Add Change To Release

Remove a Change from a Release

The Remove Change From Release dialog box should be used to request that a Change be removed from a Release to which it is currently assigned (Figure 8-72 on page 301). 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**.

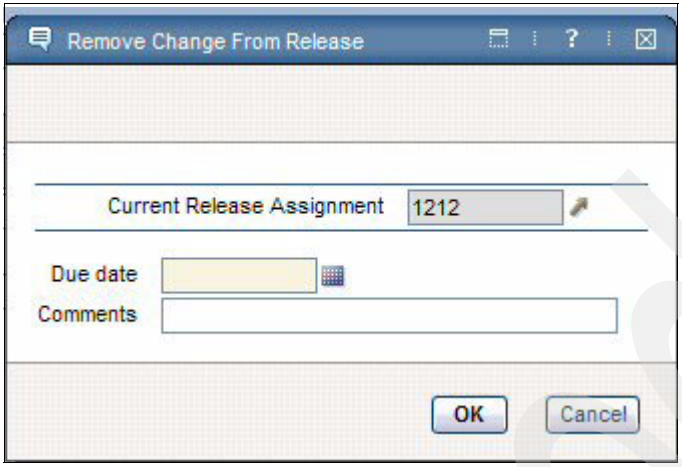


Figure 8-72 Remove Change From Release

Make a Change Available for Any Release

The Make a Change Available for Any Release dialog box should 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.

Note: This functionality is useful when you have different groups working in the environment. For example, a team responsible for developing new releases of a software or improvements may 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 (Figure 8-73). Both of these fields are optional.

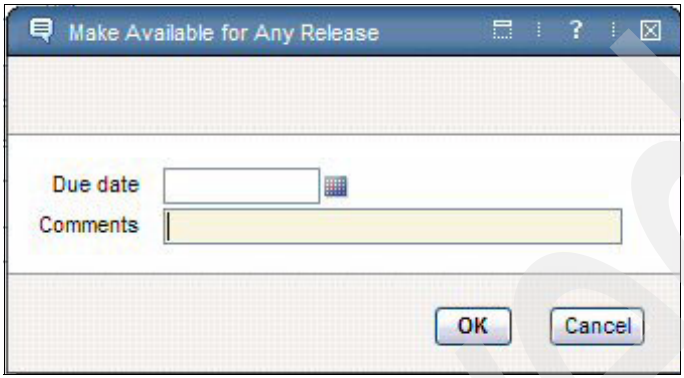


Figure 8-73 Make a Change Available for Any Release

Cancel an Outstanding Request

The Cancel Outstanding Request dialog box should 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 should be canceled and, from the Select Action Menu, select **Release Requests → Cancel Outstanding Request** (Figure 8-74 on page 303).

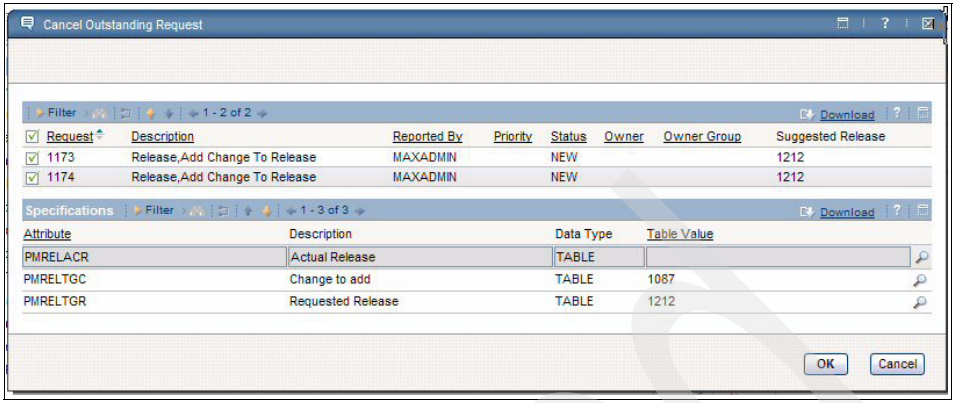


Figure 8-74 Cancel an Outstanding Request

8.5 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. It can be individually managed, and they are usually treated as self contained units for the purposes of identification and change control. Configuration items may be a service, hardware, software, support staff, or documentation. They are uniquely identified by names and other attributes.

Note: Relationship is 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 sub processes:

- 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 life cycle.
- 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 Audit CI Process in Configuration Processes Application.

8.5.1 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 some 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 8-75 illustrates the default relationship list.

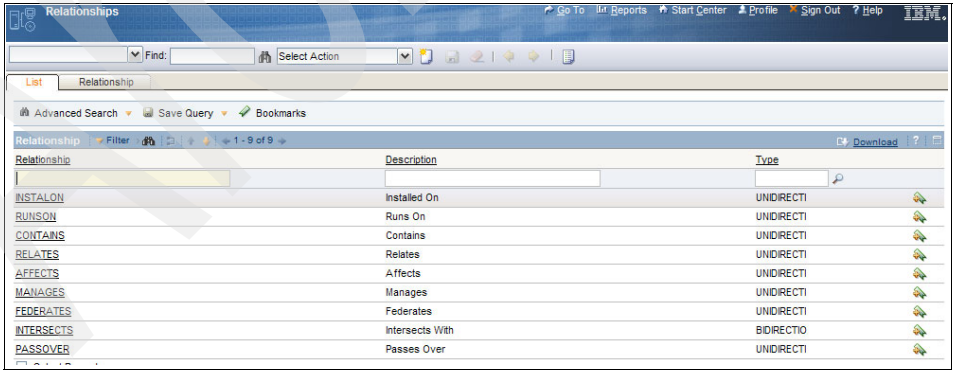


Figure 8-75 Relationship list

8.5.2 Configuration Management roles

The following roles are defined for Configuration Management. If you enable the installation process to configure your directory server, it will create these 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 Center by role

Each role in CCMDB can have a different 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 8-76).

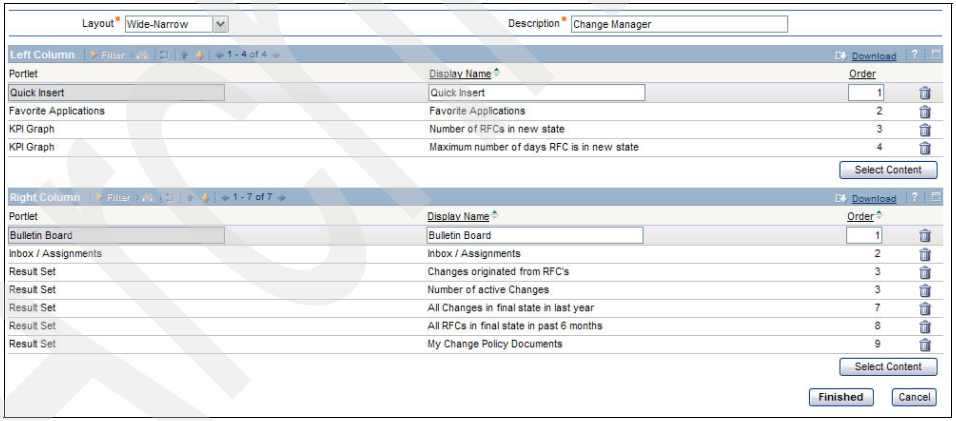


Figure 8-76 Customize Start Center

The Configuration Management roles are illustrated in the following figures.

Figure 8-77 illustrates a Start Center for Configuration Administrator role.

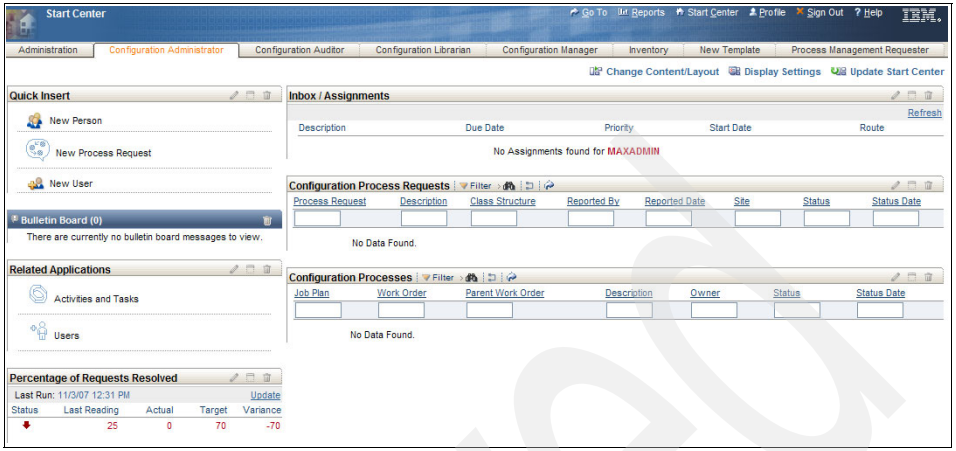


Figure 8-77 Configuration Administrator Start Center

Figure 8-78 illustrates a Start Center for Configuration Auditor role.

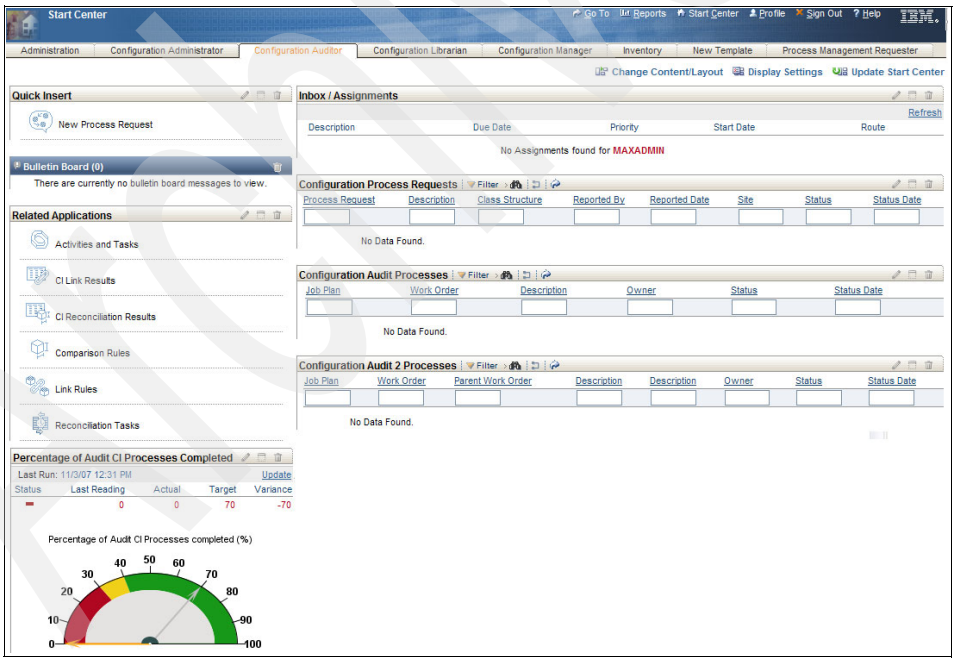


Figure 8-78 Configuration Auditor Start Center

Figure 8-79 illustrates a Start Center for Configuration Librarian role.

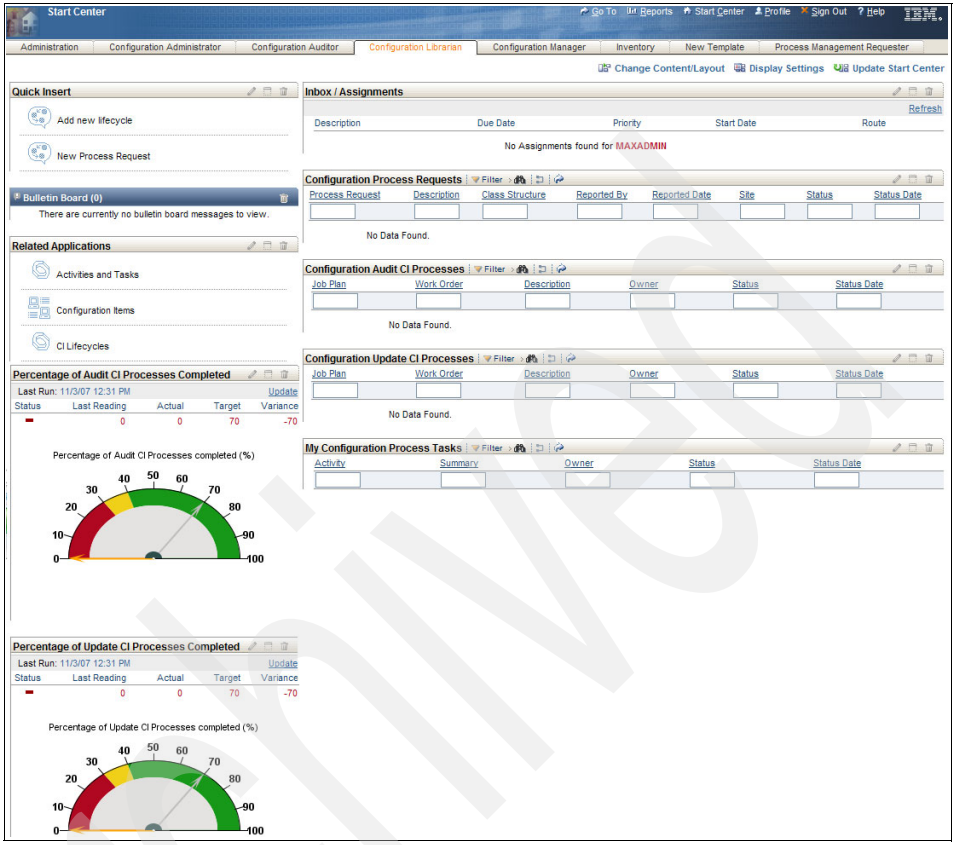


Figure 8-79 Configuration Librarian Start Center

Figure 8-81 on page 309 illustrates a Start Center for Configuration Manager role.

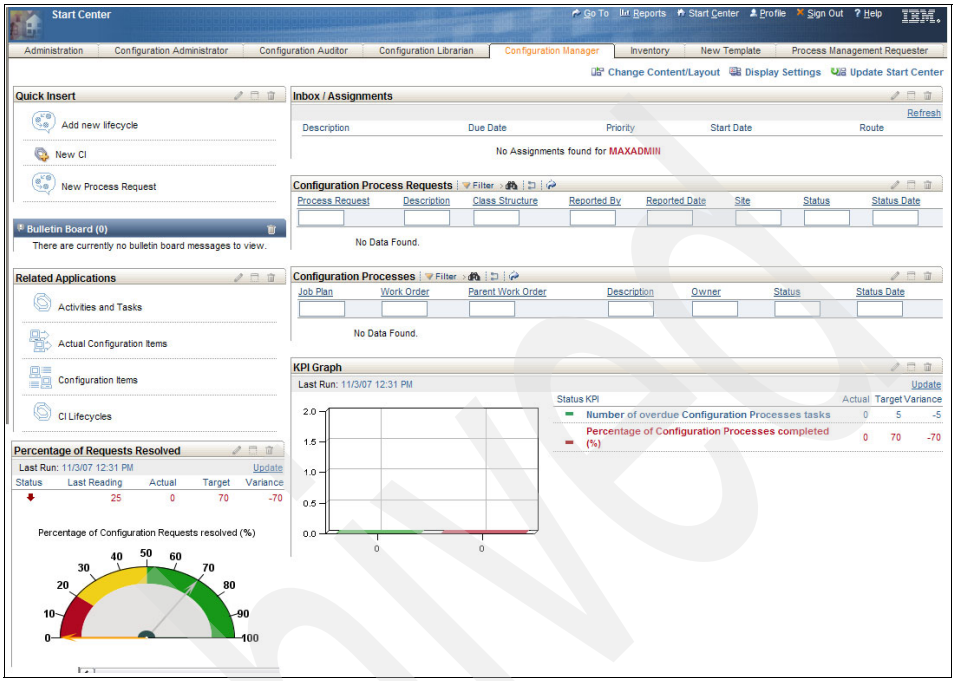


Figure 8-80 Configuration Manager Start Center

8.5.3 CI Lifecycle

A life cycle can be described as the various stages in the life of a configuration item. Life cycle states are customizable, but some of the more common ones are represented in Figure 8-81 on page 309.

The life cycle defines a set of states and permitted transitions that occur between them. Every life cycle must have one state designated as the default state. When a new CI is created, it automatically takes the default state of the life cycle that has been applied to that type of CI. If there is no life cycle explicitly associated with that CI type, then it takes the default state from the system default life cycle.

States may be protected or unprotected. A protected state is one that requires a higher level change process such as a Request For Change (RFC) in order to move CIs to or from it.

A protected state should be used 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. They correspond to a movement of a configuration item from one life cycle status to the next. If a transition occurs to or from a protected state, an RFC must be initiated and approved before that transition can take place.

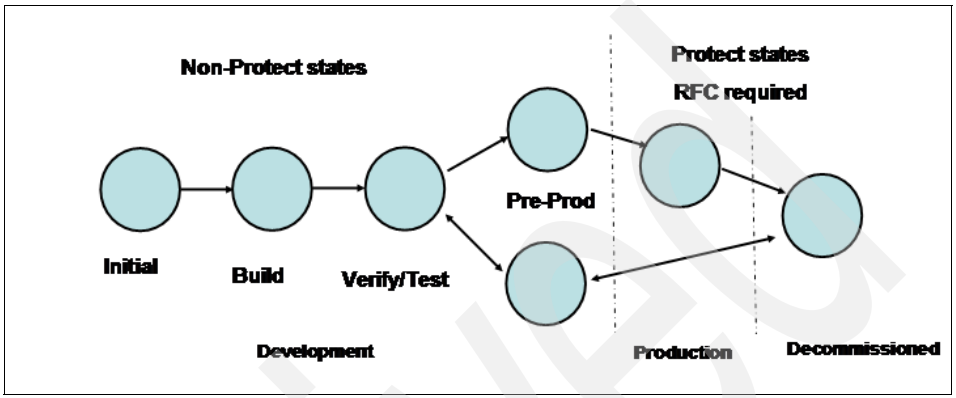


Figure 8-81 CI Lifecycle

Some terms are specific to the CCMDB Lifecycle application and will be used through this section.

- Life cycle: The various stages in the life of a configuration item (CI). The life cycle defines a set of states and the permitted transitions between them.
- Default life cycle: The life cycle that is applied to a CI type that has not had a life cycle applied to it. You can designate any defined life cycle as the default life cycle.
- Status: The name of a required field in many types of records that shows the current stage in the life cycle of the associated configuration item.
- Transition: A change in state, corresponding to a movement of a configuration item from one life cycle status to the next.
- Default State: The state within a life cycle to which a new CI is assigned.
- Protected state: A life cycle 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 life cycle states as they are processed by CCMDB logic.

8.5.4 CI Lifecycle management

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

- Create new life cycle
- Add a life cycle state
- Define life cycle transitions
- Apply the life cycle to CI types
- Modify an existing life cycle
- Delete a life cycle state

Create new life cycle

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

Perform the following set of actions to add a new life cycle:

1. Select **Go To** → **IT infrastructure** → **CI Lifecycle** (Figure 8-82).

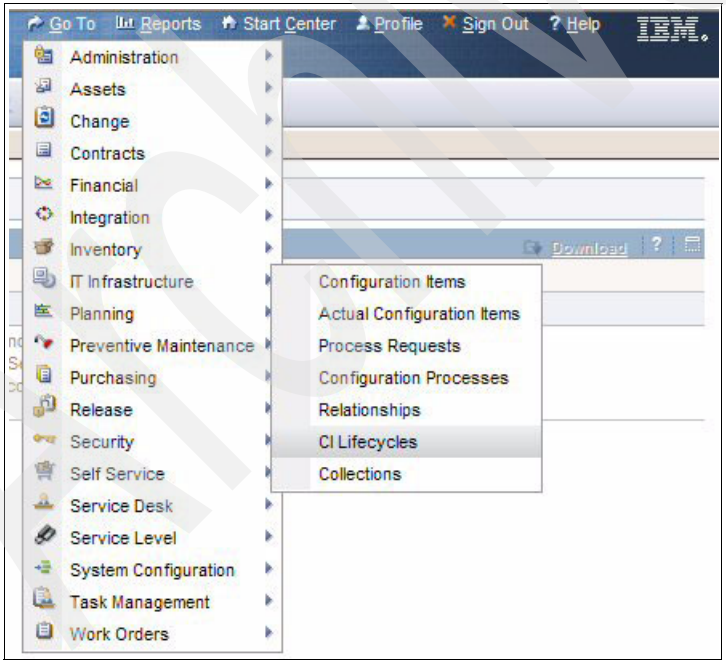


Figure 8-82 CI Lifecycle menu

2. You can choose to either **Add a new Lifecycle** from the Select Action menu or you can click the **Add a new Lifecycle** icon to the right of the Select Action drop-down menu (Figure 8-83). Selecting **Add a new Lifecycle** opens the Lifecycle tab. An ID designation for the new life cycle is automatically generated and cannot be altered.

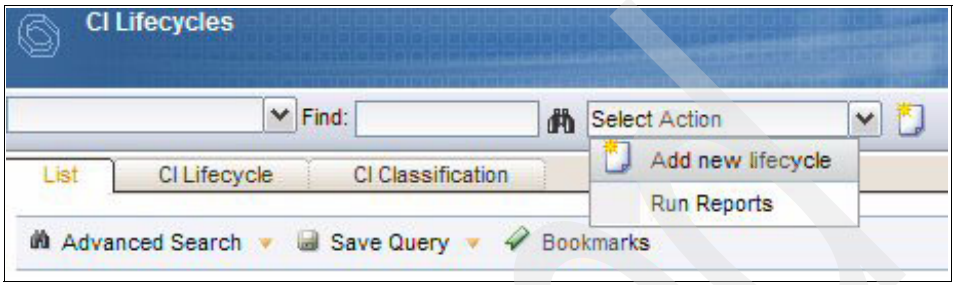


Figure 8-83 Add new CI Lifecycle

3. Type the name of the new life cycle you want to create in the Lifecycle Name field. If you want this life cycle to function as the default life cycle, select the **Default** check box (Figure 8-84). Optionally, add a description of the life cycle to the Description field.

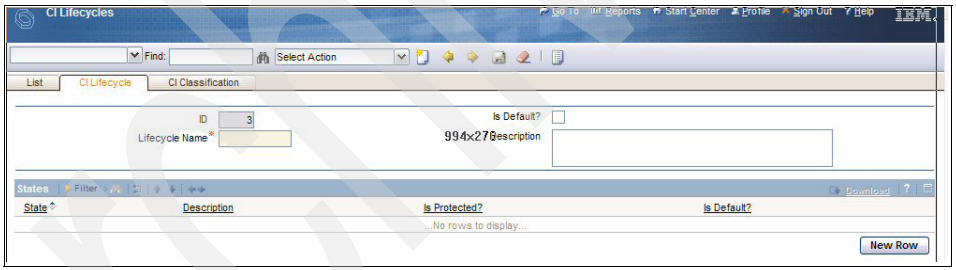


Figure 8-84 New CI Lifecycle

Add a life cycle state

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

Ensure you have finished the Adding a New Lifecycle Task by doing the following steps:

1. Click **New Lifecycle State**. The Lifecycle State Details table appears underneath the States table (Figure 8-85 on page 312).
2. Select **New Row** → **Select a state** from the State drop-down menu.
3. Select the **Is protected?** check box if you want to create a protected state. You cannot delete a protected state without having a Request for Change (RFC) associated with the life cycle.

4. Select the **Is Default?** check box if you want this state to be the default.
5. Optional: Add a description to the Description field for the state.
6. Repeat this task until you have defined all the desired states.

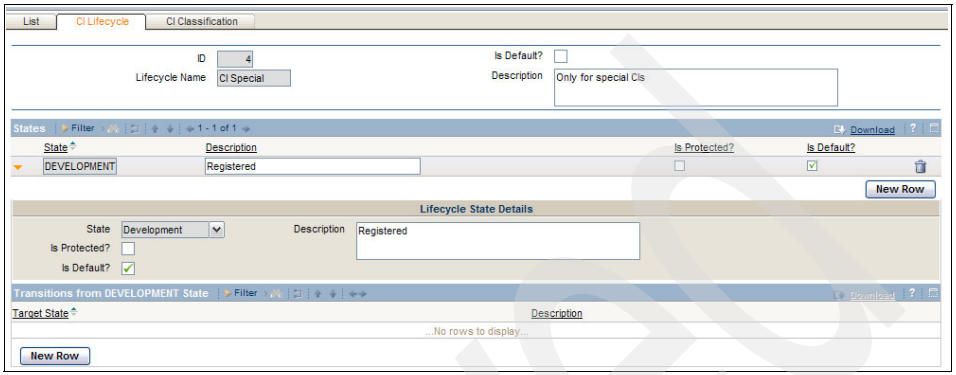


Figure 8-85 Add new state

Define life cycle transitions

The third step in creating a life cycle is the processing of defining the transitions to each life cycle state. After you have added your life cycle and created a life cycle state, you define the transition or transitions to each state (Figure 8-86).

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

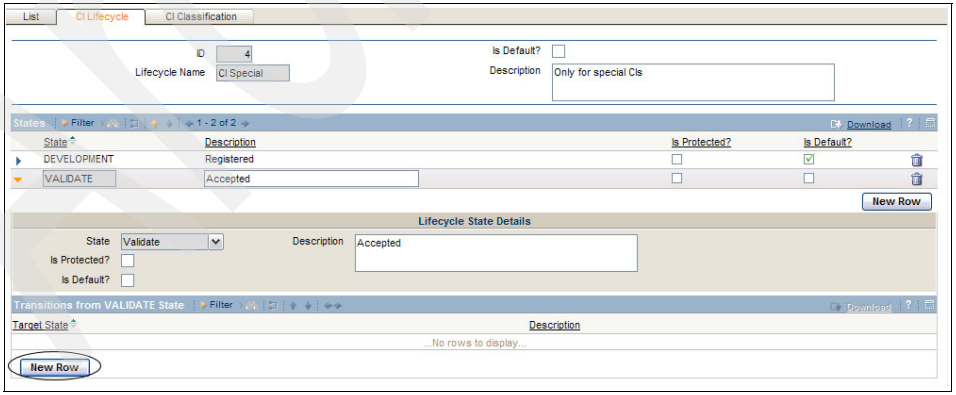


Figure 8-86 Define life cycle transitions

Apply the life cycle to CI types

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

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

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

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

Some examples of life cycle classifications are:

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

To apply the life cycle to CI types, click the CI Classification tab, then click **New Row**. If you know the name of the classification you want to associate with your life cycle, type it into the Classification field. If you do not know the name, click the **Detailed Menu** icon next to the Classification field and select the appropriate classification from the list by clicking the blue box beside the listing (Figure 8-87).

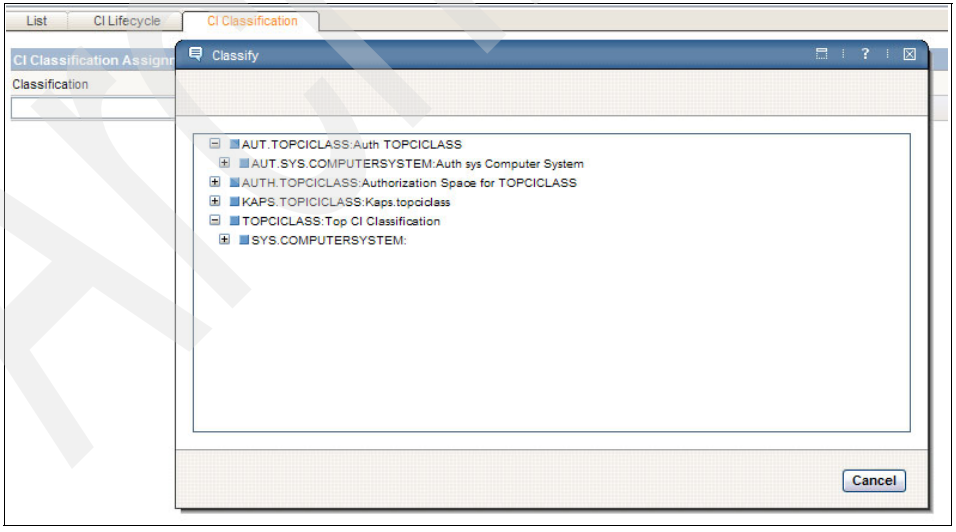


Figure 8-87 Classify life cycle

Modify an existing life cycle

It might become necessary to modify some of the existing attributes of a life cycle after they have already been applied. To add a new state name to the existing states of a life cycle or to change existing transitions between states, choose the life cycle you want to modify, make the changes, and save it (Figure 8-88).

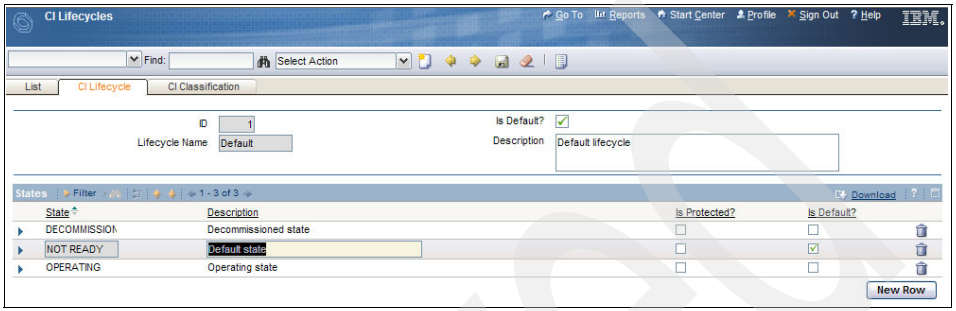


Figure 8-88 Modifying CI Lifecycle

Delete a life cycle state

A life cycle can be deleted, but it is necessary make sure that no configuration items are in the state of being planned to be deleted.

When a state from a life cycle is deleted, any CIs that were in that state will be in an undetermined state. These CIs must be assigned to a new state within the life cycle that applies to its CI type before any other function can be performed with them.

For example, you cannot modify any attribute of a CI that does not have a valid life cycle state. You can assign the CI to the default state, or to any state to which a transition is defined from the default state. It is better to move all CIs out of the state before you delete it.

To find CIs in a certain state, open the Configuration Items application, enter the state in the Status field, and press Enter (Figure 8-89 on page 315). If you have states with the same name in more than one life cycle, some of the CIs in the list might be in a different life cycle 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 life cycle from which you plan to delete the state.

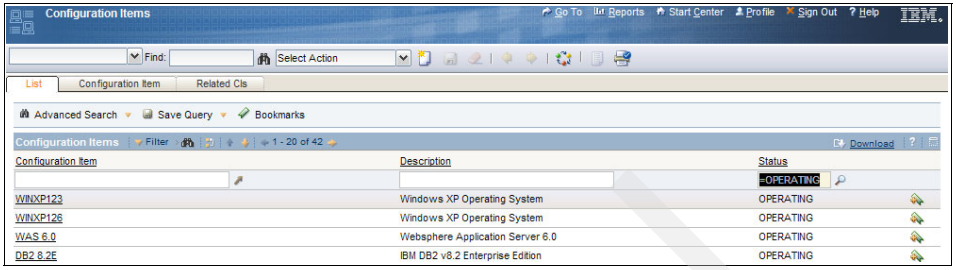


Figure 8-89 Find CIs

To delete a state from a life cycle, follow these steps:

- Open the life cycle in the CI Lifecycles application.
- 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.
- Click the trash icon on the line of the state you want to delete. The name of the state will be struck through in the list.
- Click **Save** to save the updated life cycle.

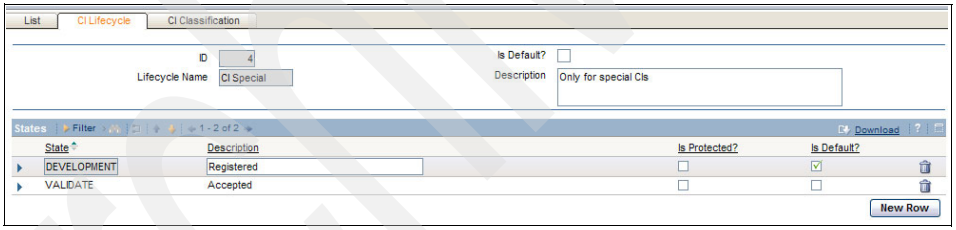


Figure 8-90 Delete state from life cycle

The following cases may cause a configuration item's status becomes invalid:

- A life cycle state is deleted from the CI Lifecycle.
- A CI Lifecycle is deleted.
- A new life cycle is assigned to a CI classification.
- The life cycle assigned to the CI classification is changed.
- The CI classification assignment in the CI Lifecycle is deleted.
- The CI is re classified.
- Other cases.

The current design to handle the invalid configuration item's status is:

- Users should try to avoid modifying LC states, LCs, or CI classifications with existing CIs.
- Warnings have been added to the UIs.
- Warnings have been added to the Docs.
- If a CI's status becomes invalid, it is the user's responsibility to reassign the valid status to it.
- The status change menu shows the new available states in the current LC.
- CI attribute modification is not allowed if a CI's status becomes invalid.

8.5.5 Discover configuration item

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.

Refer to *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565 for more details about the common data model and about TADDM and its role in the discovery process.

8.5.6 Authorized configuration item

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. Table 8-1 shows the characteristics of each type.

Table 8-1 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.

Audits and reconciliation 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; for more details about how to create Authorized CIs using the promotion, refer to Chapter 5, “CI promotion” on page 113.

Create an Authorized Configuration Item

To create an Authorized Configuration Item, select **Go To Menu** → **IT Infrastructure** → **Configuration Items**, and then click the **New CI** button (Figure 8-91).

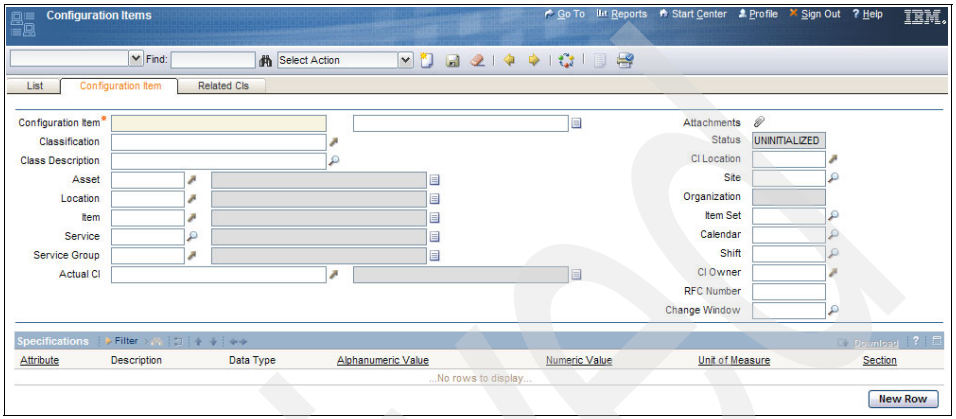


Figure 8-91 Create New CI

The information required to create a new CI is:

- Configuration item: Unique name for the CI.
- Configuration item description: Description for the CI.
- Actual CI: 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.
- The following fields should be entered 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 should be applied to the new CI.
  - Asset: Related asset to associate the CI.
  - In the Classification field, you can associate the CI with a classification. Using the Detail Menu, you can select **Classify** to select a classification, or you can select **Go To Classifications** to open the Classifications application. The Specification table window displays the attributes associated with the classification
  - RFC number that originated the CI.
  - Using other fields, you can associate the CI with an asset, location, item, service, or service group. Some 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.

In the same tab, the attributes for the new CI should be defined. It only can be defined if a Classification is associated with it.

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 8-92).

The fields below should be filled as needed:

- Data Type
- Unit of Measure
- Section
- Alphanumeric Value
- Numeric Value
- Table Value
- Inherited from
- Apply Down Hierarchy?

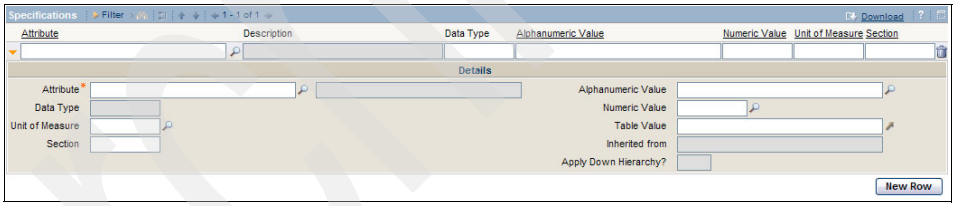


Figure 8-92 New CI attributes

Define relationships

In the Related CIs tab, you can define the relationships that the New CI has with other CIs, as shown in Figure 8-93.

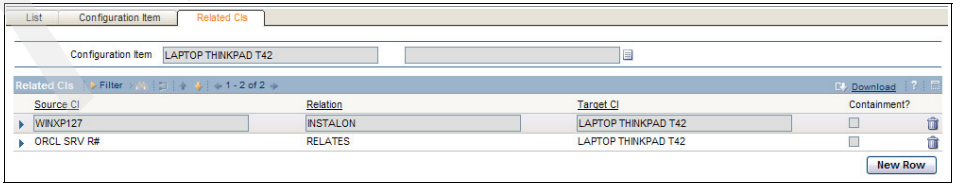


Figure 8-93 Related CIs

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.

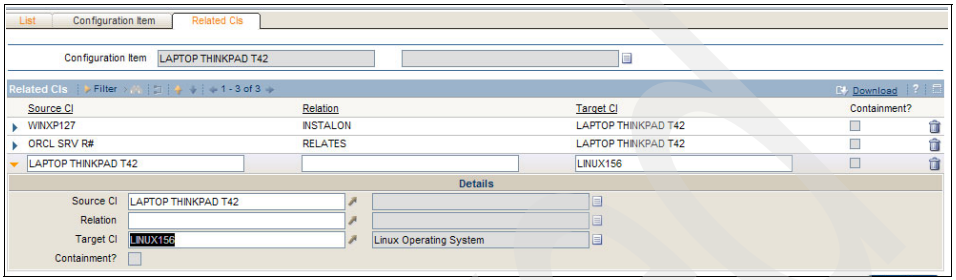


Figure 8-94 Create New Relation

Update Authorized configuration item

Select a configuration item from the list, edit it, define the modifications, and save it (Figure 8-95). 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. If the CI is in a protected state, you might not be able to change the value of an attribute.

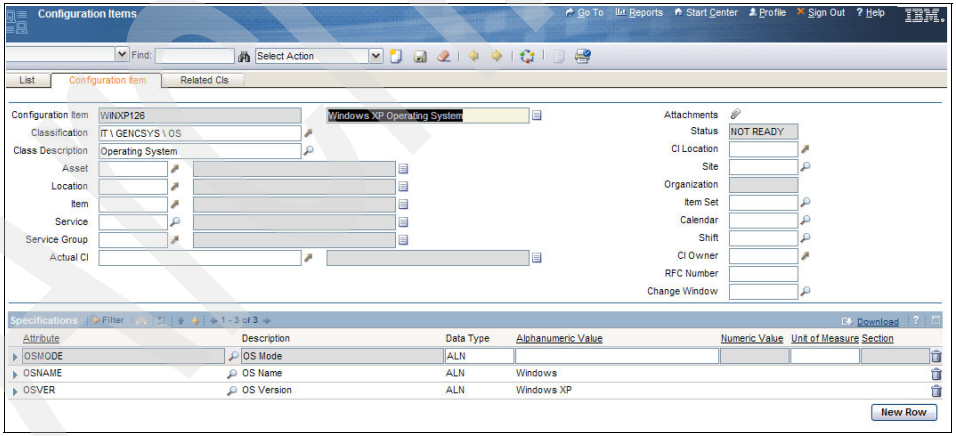


Figure 8-95 Update Authorized configuration items

Delete Authorized configuration item

Select a configuration item from the list, select **Select Action** → **Delete CI**, and click the **OK** button in the confirmation message (Figure 8-96 on page 321).

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 appears on an incident, problem, or change record.

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

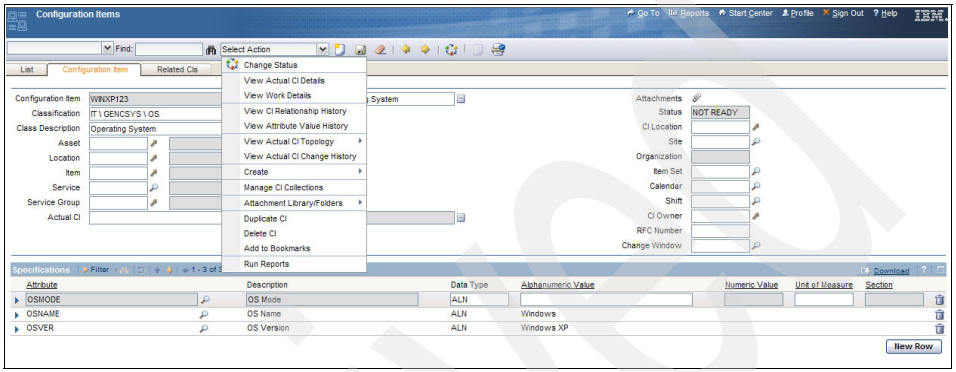


Figure 8-96 Delete Authorized configuration item

Duplicate Authorized configuration item

Select a configuration item from the list and then select **Select Action** → **Duplicate CI** (Figure 8-97).

All fields of the CI record will be duplicated, except the Configuration Item field and information in the Related CIs tab.

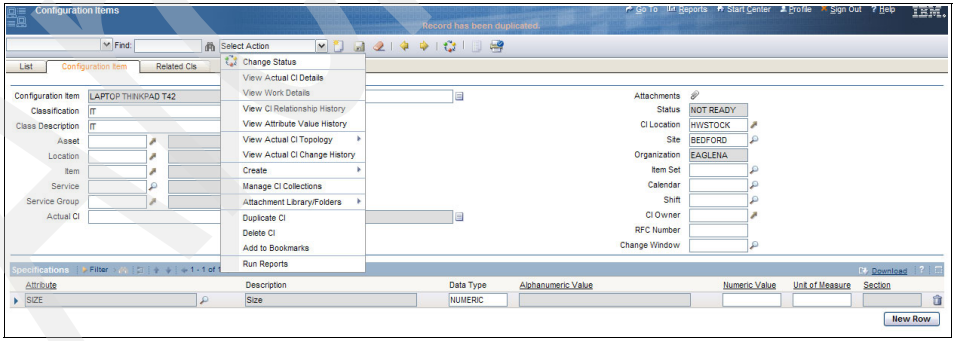


Figure 8-97 Duplicate Authorized configuration item

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.

Manage CI Collection

Manage CI Collection should be used 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 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.

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. At this point, the following actions can be performed:

- Associate a Collection with CI.

  Click **New Row** and in the Collection field enter a collection, or click **Detail Menu** to select an option and retrieve a value, 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 button.

- Activate or deactivate an association.

  To activate or deactivate an association, click the check box in the row that has the collection.

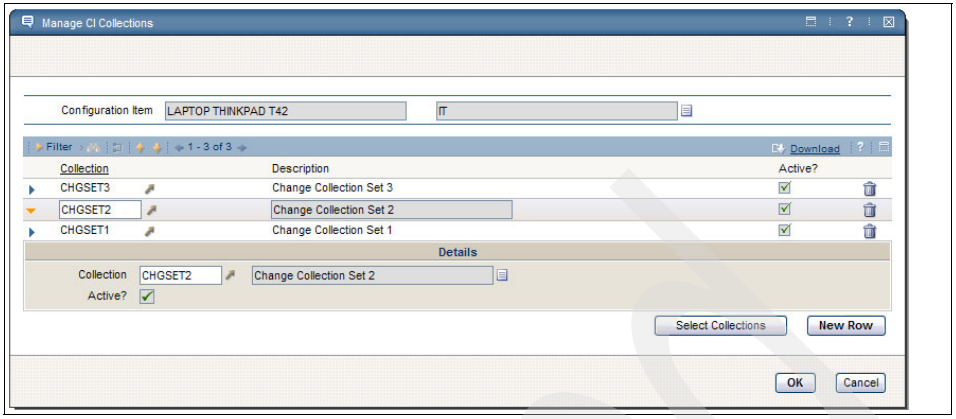


Figure 8-98 Manage CI Collection

8.5.7 Control and update CI process

The Control CIs process ensures that changes to Authorized CIs are made with the appropriate review, approval 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 8-99 gives an overview of the processes.

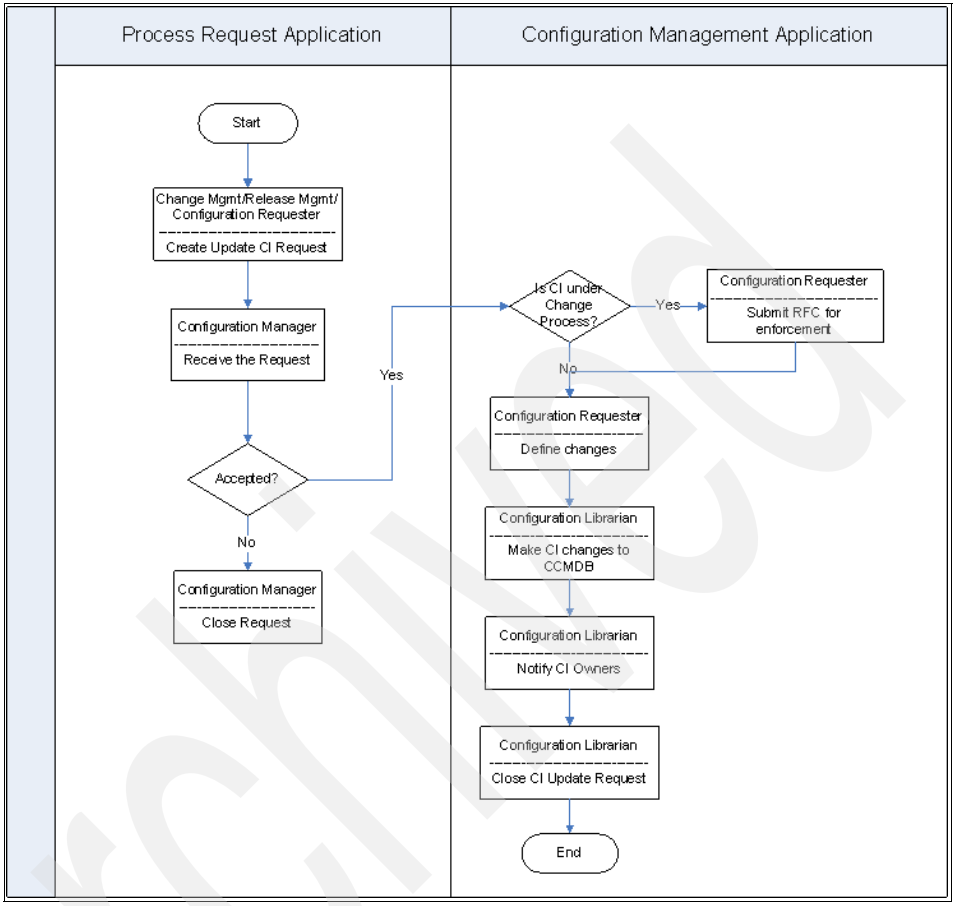


Figure 8-99 Control process flow

Create 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. The following information should be provided to characterize the process request as an Update CI Request:
   - Process Manager Type: chOose the option Configuration Management in the list.
   - Classification: PMCFGUR.
   - Class Description: CIs Configuration Update Request.
3. In the Classification Attributes table, there are two very important fields: CHGRECD and RFCID. They are used together to determine if all the updates for a RFC are recorded.
4. When **All Updates Recorded** is checked, the submit workflow verifies if the RFCID (Change Process Number) is valid. If yes, the submit workflows close the request and pops up a message informing you that the updates will be implemented when the RFC is completed.
5. After providing all the required information, click **Submit**.

Figure 8-100 shows a New Process Request form.

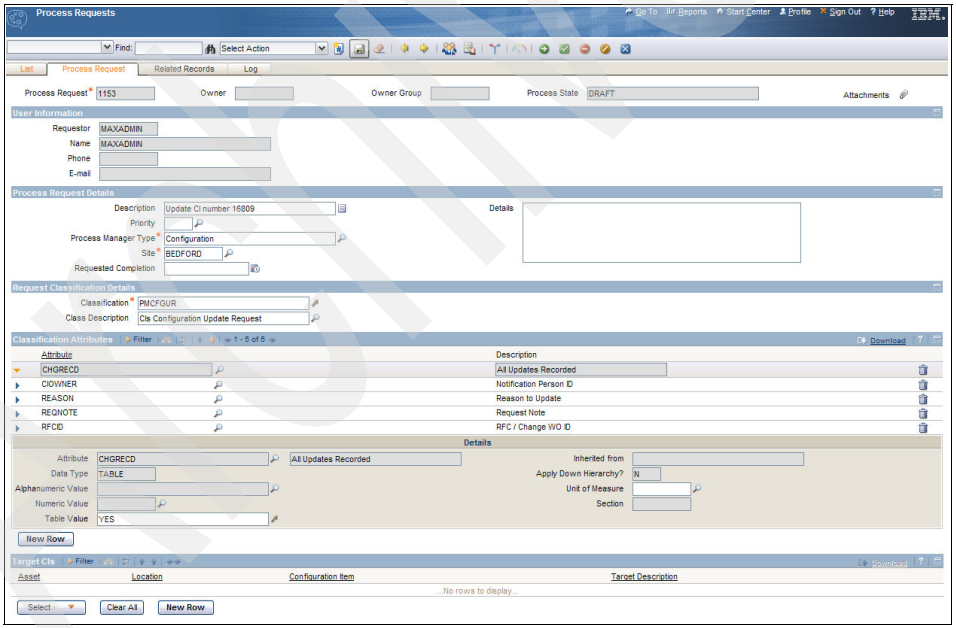


Figure 8-100 Create an Update CI Request

Accept 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**.

Note: 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 does not need to be approved again.

Figure 8-101 illustrates an Update CI Request to be approved.

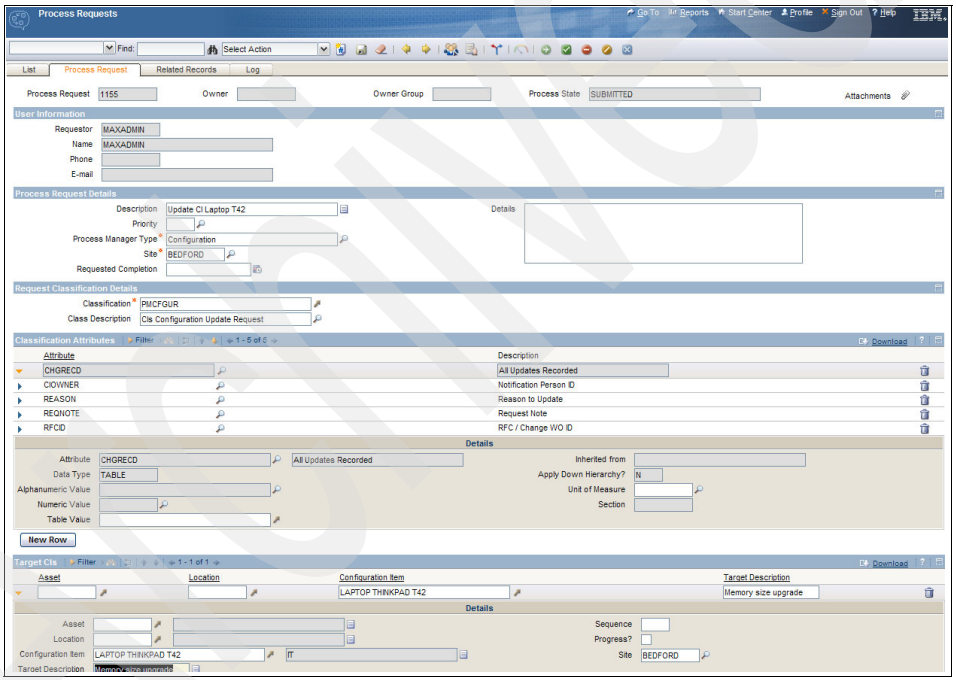


Figure 8-101 Accept CI Process Request

Create a Configuration Process Record

Responsible role: CCMDB

After the Update Process Request is accepted, it automatically becomes a Configuration Process Record (Figure 8-102 on page 327).

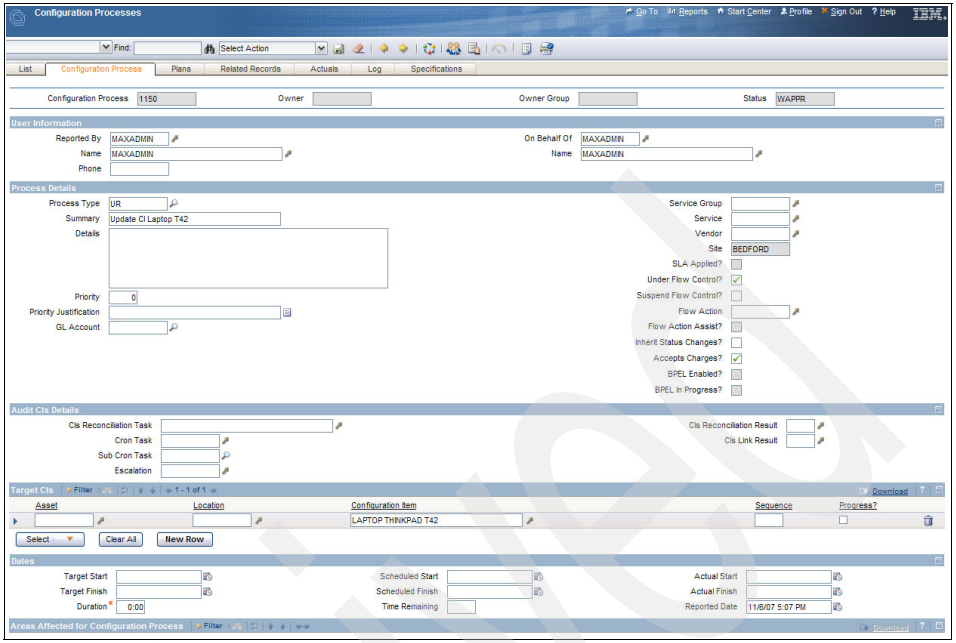


Figure 8-102 Configuration Process Record

Apply 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.

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

To delete a task, click the **Trash** button in the row to be deleted.

Figure 8-103 shows the application of a Job Plan to a Configuration Process Record.

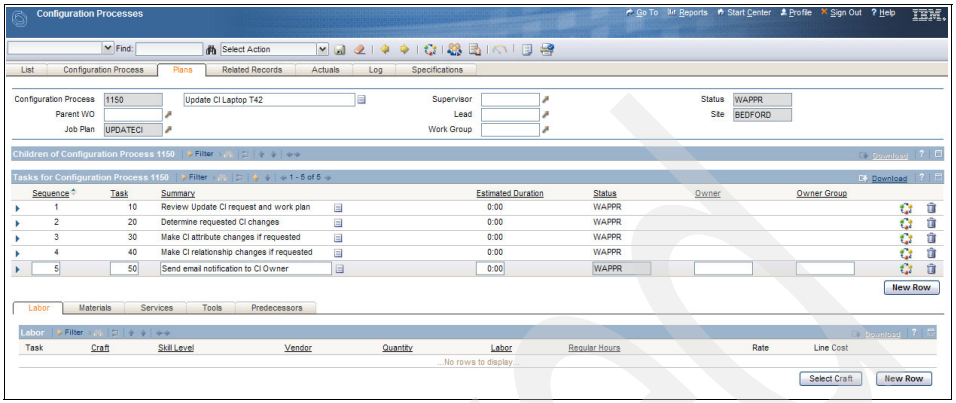


Figure 8-103 Configuration Process Job Plan

Approve 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 Figure 8-104.

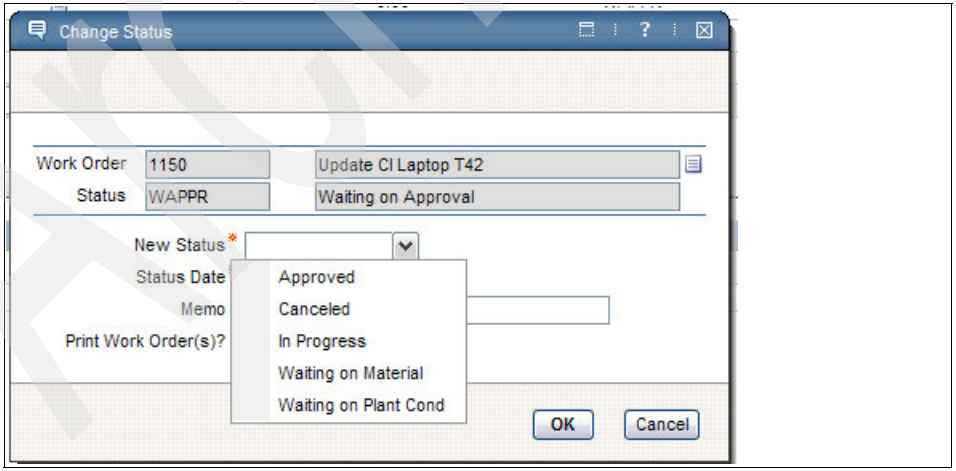


Figure 8-104 Approve Configuration Process Record

Initiate 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, as shown in Figure 8-105.

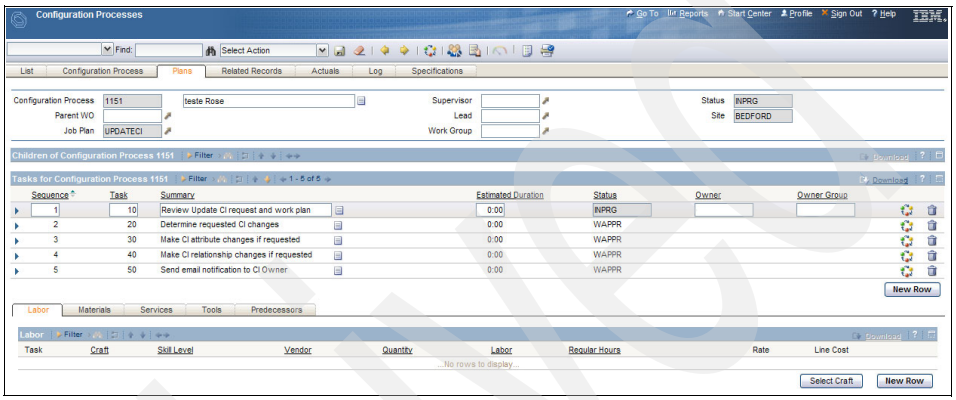


Figure 8-105 Initiate Tasks

Tasks status

When a task has its status changed to Complete, the next task status automatically gets the status In Progress, as shown in Figure 8-106.

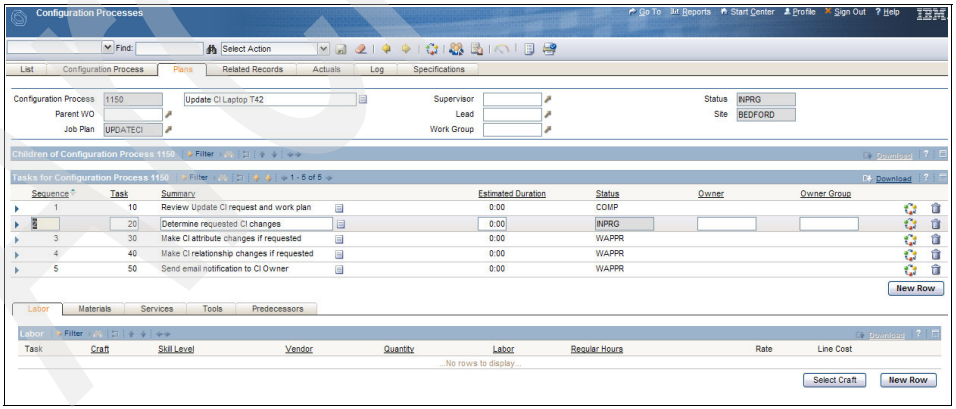


Figure 8-106 Task status change

Execute 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, should 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 should 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.

Select **Selection Action** → **Move/Swap/Modify** to make the CI changes (Figure 8-107).

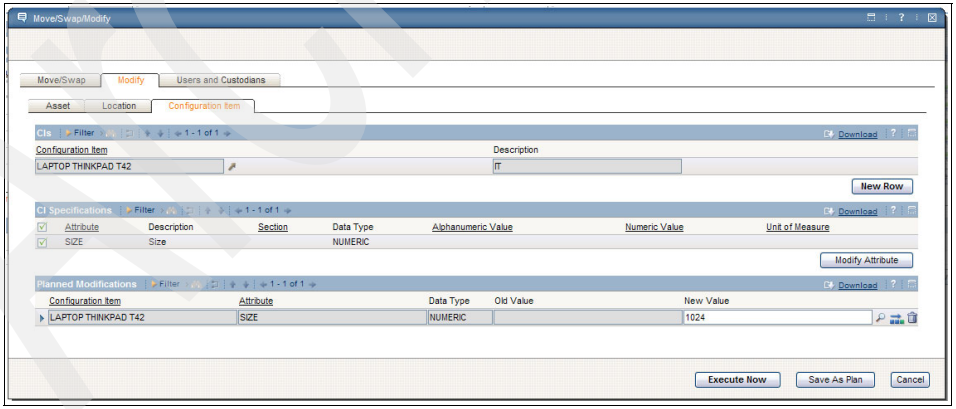


Figure 8-107 Make CI attribute changes

Task 40

The fourth task, Make CI relationship changes if requested, performs changes in CI relationships.

The changes in CI relationships should 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.

Figure 8-108 shows the Related CIs tab.

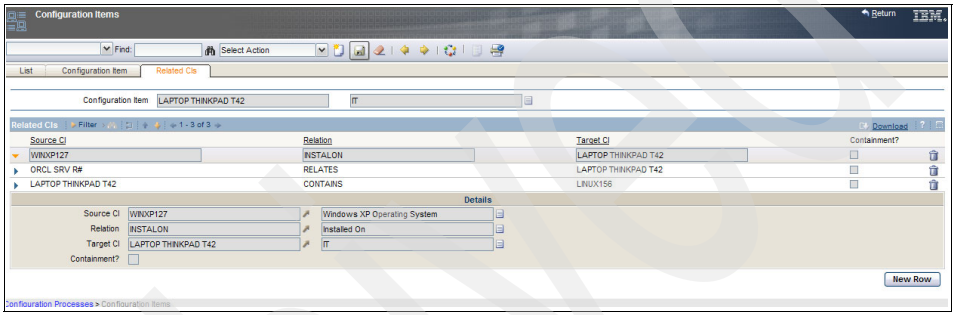


Figure 8-108 Make CI relationship changes

Task 50

The fifth and last task, Send e-mail notification to CI Owner, is to ensure that the CI owner is notified about changes in the CI.

To send an e-mail, select **Select Action** → **Create** → **Communication**.

Figure 8-109 shows a communication form.

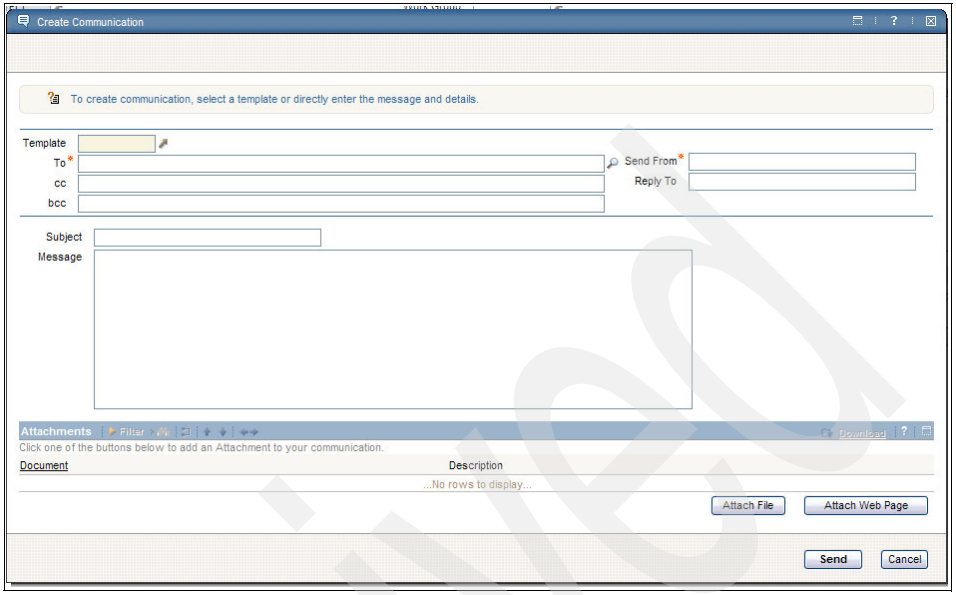


Figure 8-109 Send an e-mail notification

Close 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 8-110.

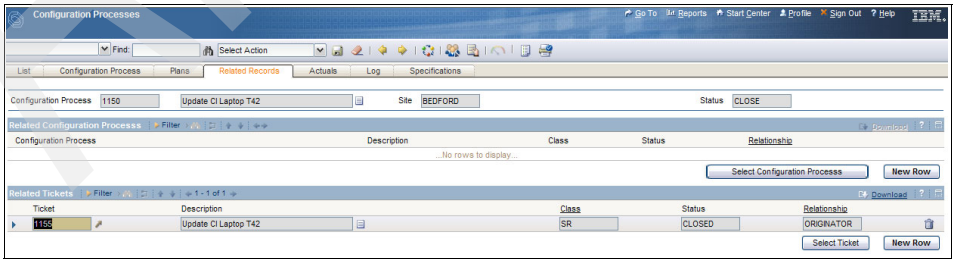


Figure 8-110 Closed Configuration Process Record

8.5.8 Verify / Audit 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:

- 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

Create, submit, and accept a CI Audit Request

This section provides a step-by-step explanation on how to request and run the Audit Process in CCMDB.

Submitting an Audit Request

To create an Audit Request, the first step is to create a Process Request.

Responsible role: Configuration Manager or Configuration Auditor

The Configuration Manager or Configuration Auditor submits an Audit Request using the Process Request application.

Select **Go To** → **IT Infrastructure** → **Process Request**.

Within the Process Requests Application, click **New Process Request**.

An automatic ID will be assigned for the new request. The following information has to be provided:

- 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.
- Classification: The request classification, in this case, Configuration Audit Request.
- Class Description: The description of classification. It will be fulfilled automatically according to the classification chosen.
- Target CIs: Select the CIs that will be audited.

Click **Submit** to submit the request

Figure 8-111 shows a sample Process Request window for the Configuration Audit Request.

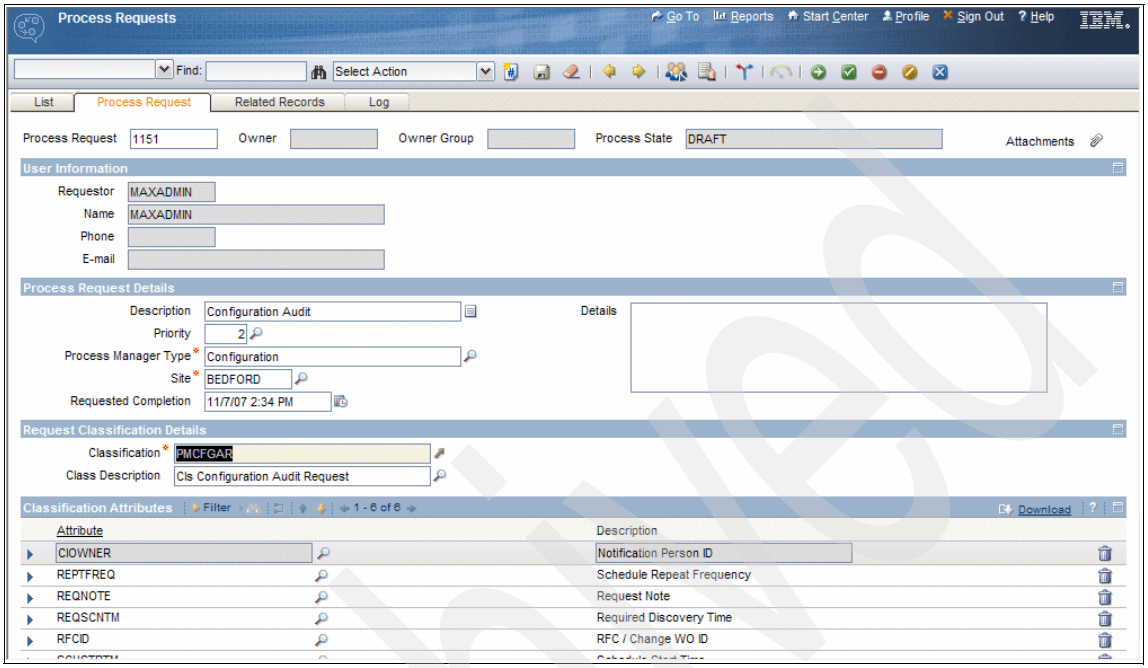


Figure 8-111 Configuration Audit Process Request

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 his Start Center. 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 a owner. Figure 8-112 shows a sample of the Configuration Process Requests window.

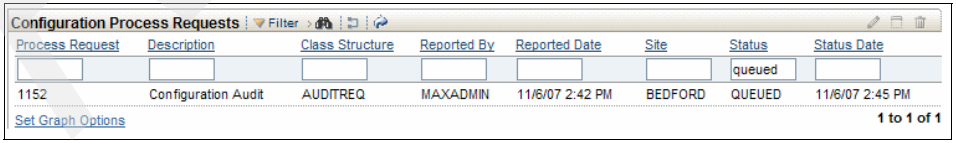


Figure 8-112 Configuration Process Requests Queue

Assigning an owner for the Process Request

Responsible: Configuration Manager or Configuration Auditor

The process request record will be displayed for the Configuration Manager (or Configuration Auditor) in his 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 to take ownership for the request. Figure 8-113 shows how to assign an owner.

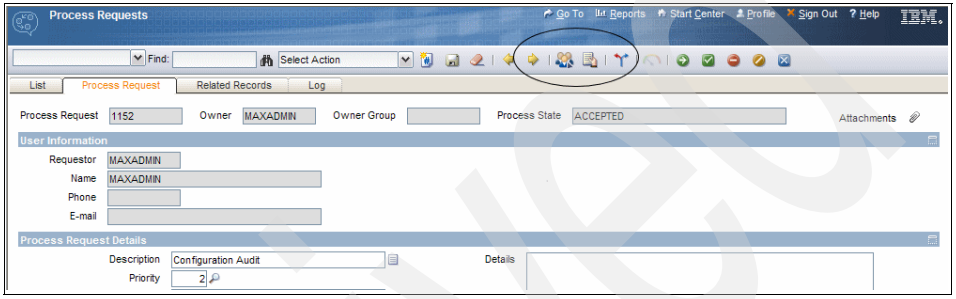


Figure 8-113 Assigning owner for configuration audit request

Review 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** → **Select the configuration process** → **Go to Plans tab** → **Review tasks** or select a new Job Plan.

Figure 8-114 shows a sample window of the configuration audit Job Plan.

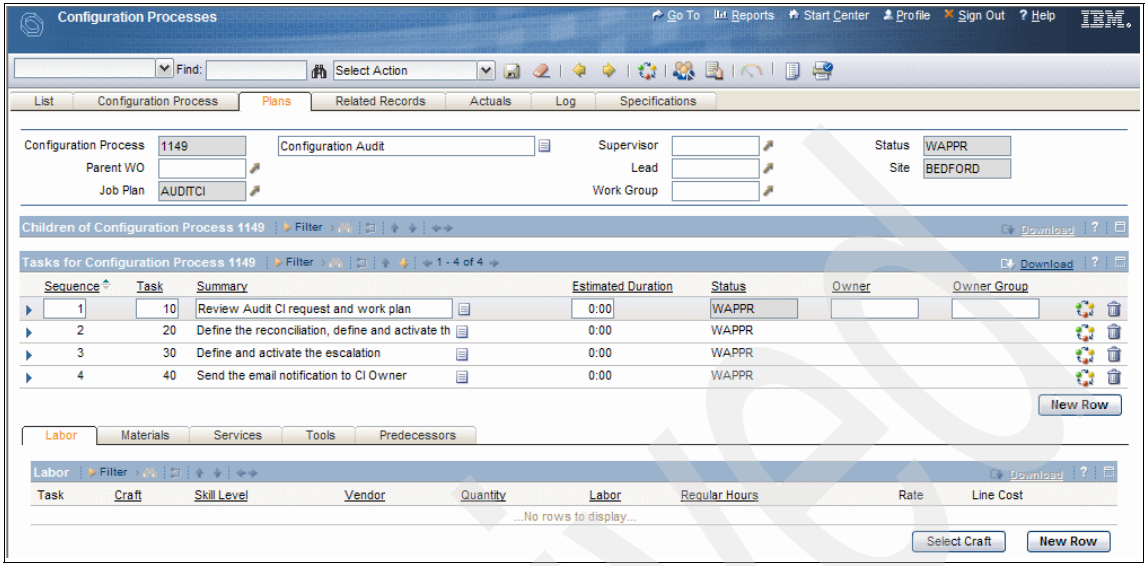


Figure 8-114 Configuration Audit Job Plan

Please note that in Figure 8-114 that the Work Order record number was used. CCMDB generates a Work Order record number that is different from the Process Request record number. For our next steps, the Work Order record number will be used (also known as a Process Request). 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 8-115 on page 337 shows how to navigate to Configuration Processes from the Process Request application.

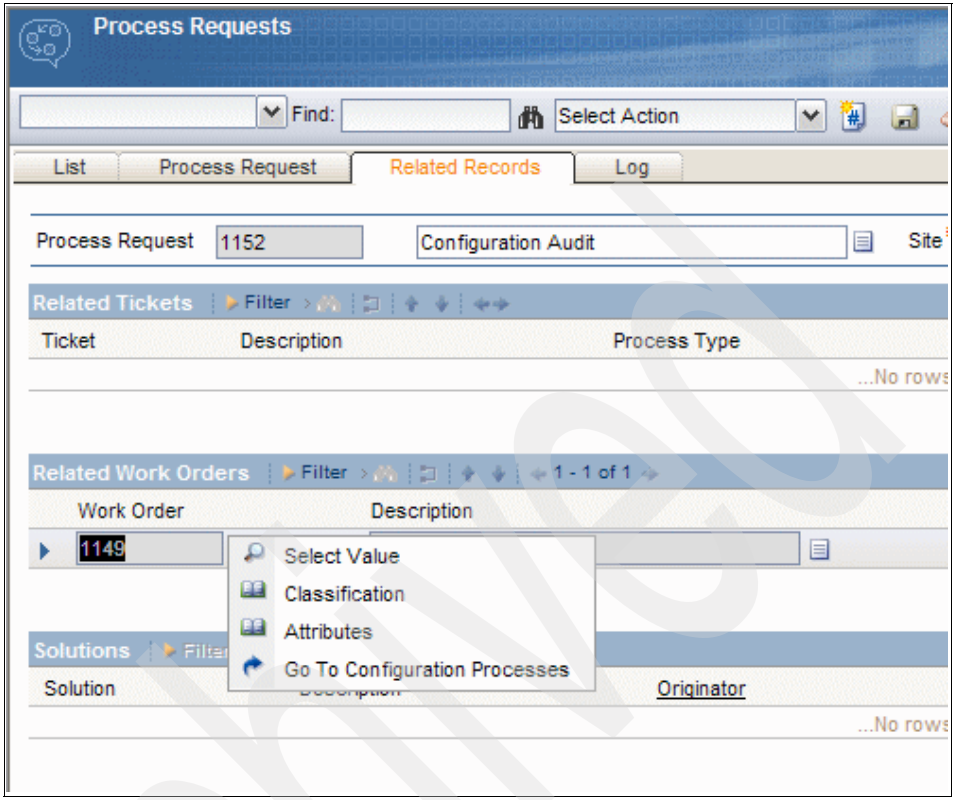


Figure 8-115 Navigating to Configuration Processes from the Process Request application

When a Job Plan is applied to a configuration process, its activities are inherited by the configuration process.

The configuration audit owner can customize the set of activities and tasks to be used to complete the audit process. In the example used here, there are only tasks configured to the Job Plan.

Audit Request is accepted

Responsible role: Configuration Audit Owner

The Configuration Audit Owner changes the status of the Configuration Process record to INPROGRESS to initiate the first activity in the Job Plan.

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

Figure 8-116 shows how to change the status of the configuration process record.

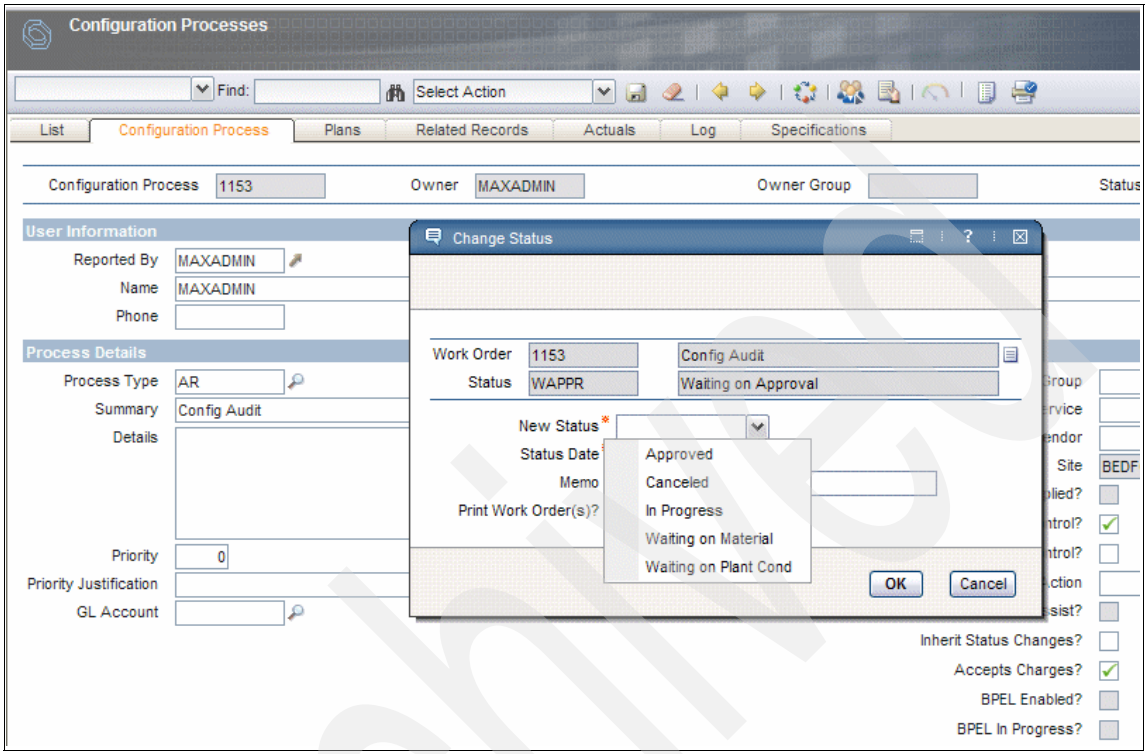


Figure 8-116 Changing the Configuration Process record status

8.5.9 Reconciliation

The Reconciliation module application allows the comparison of two kinds of CI data stored in IBM Tivoli Change and Configuration Management Database (CCMDB). CCMDB maintains two sets of CI in two different 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 should be, and what has been planned. These configuration items are in effect “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 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 IBM Tivoli Integration Composer, imports the collected data into CCMDB.

Reconciliation applications

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

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

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 reconciliation process identifies successful matches as well as discrepancies and variances between the two sets of data. The results of a reconciliation can be used to determine whether the objects actually deployed comply with corporate plans and whether the changes over the life cycle 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 a reconciliation, create a reconciliation task that combines the elements required for a reconciliation into a specific task. 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 reconciliation.

For CCMDB, there are three basic types of reconciliations that can be performed:

- 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.

  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. Do 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.

  For example, a full CI comparison can be executed and the results can determine whether the relationships in Authorized CI data between a computer system, operating system, and software component matches the relationships in Actual CI data.

Setting up a reconciliation

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 reconciliation in the CI Link Results and CI Reconciliation Results applications.

Use the following steps to set up and execute a reconciliation:

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 reconciliation.
6. View results of the reconciliation.
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. 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.

Once 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 is run. 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.

Figure 8-117 shows the Task Filters application sample window. To open the Task Filters application, select **Administration** → **Reconciliation** → **Task Filters**.

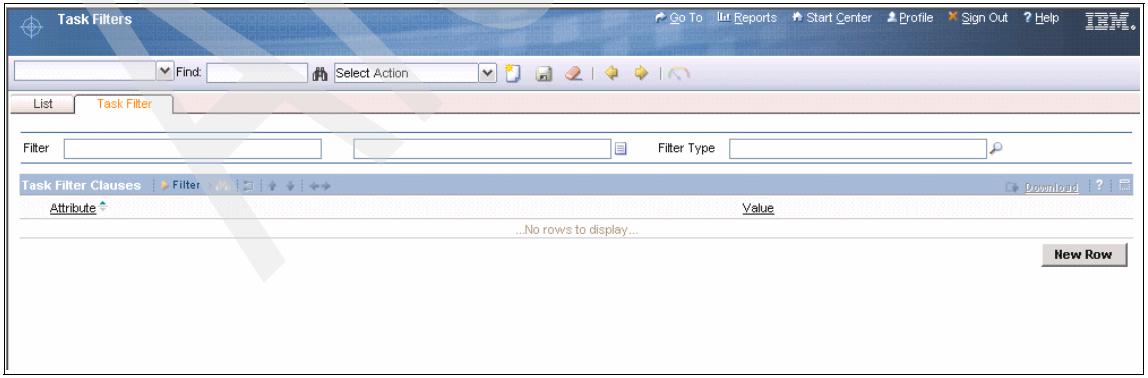


Figure 8-117 Task Filters application

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 (specified in the Filter Type field) of task filter. The type selected determines which set of objects the filter applies to. For CCMDB configuration items, either CI or Actual CI 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 should be defined. Multiple attribute clauses can be created for a task filter.

If multiple clauses that specify different 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 Site and Service Group attributes, the system selects only Actual CIs for the specified site and the specified service group; both criteria must be met.

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.

Figure 8-118 shows a Task Filters application sample window on creating a task filter.

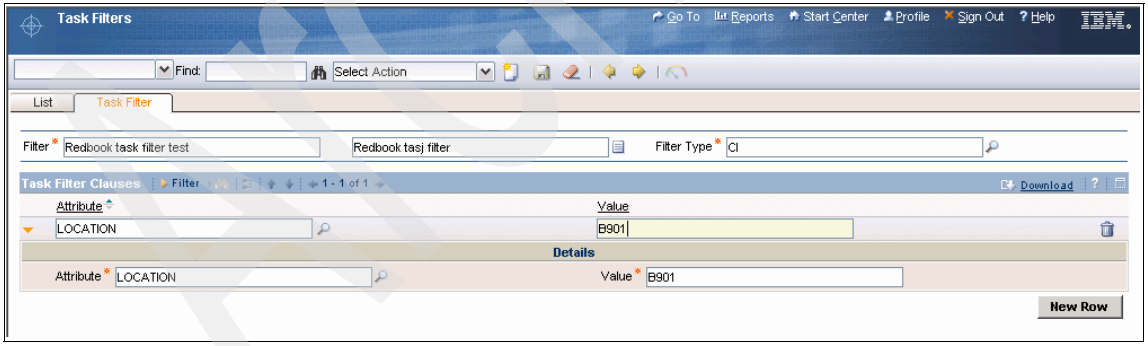


Figure 8-118 Creating a Task Filter

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 8-2 on page 343 shows the values available for each filter type.

Table 8-2 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 apply to CIs.)	Asset
	Asset Class Structure
	Collection
	Custodian
	GL Account
	Organization
	Site
	Status
	Usage
	Work Order
Deployed Asset (This filter type does apply to CIs.)	
	Asset Class
	Organization

Filter type	Attribute
CI	CI Number
	Site
	System Role

Setting up link rules

A link rule is a required component of a reconciliation task. Link rules establish the basis for reconciliation 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 configuration items (CIs) with Actual CIs is the Actual CI number (ACTCINUM).

Once a link rule is created, use the Reconciliation Tasks application to associate the link rule with a specific reconciliation task, and the system applies the link rule each time it executes the 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.

Note: 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 8-3 describes the elements of a link clause.

Table 8-3 Link clause elements

Field	Function	Rules/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.
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).

Field	Function	Rules/Requirements
Data Set 1 Class Structure	When selecting a specification as the Data Set 1 object, this field identifies a specific class structure for reconciliation.	► 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.
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).

Field	Function	Rules/Requirements
Data Set 2 Class Structure	When selecting a specification as the Data Set 2 object, this field identifies a specific class structure for reconciliation.	► Mandatory if ACTCISPEC is selected. ► Selected from a value list. Values in the list are class structure identifiers for the top-level objects.
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.

Field	Function	Rules/Requirements
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.

Note: 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.)

Figure 8-119 shows a sample of a Link Rules application. To open the Link Rulers application, select **Administration** → **Reconciliation** → **Link Rules**.

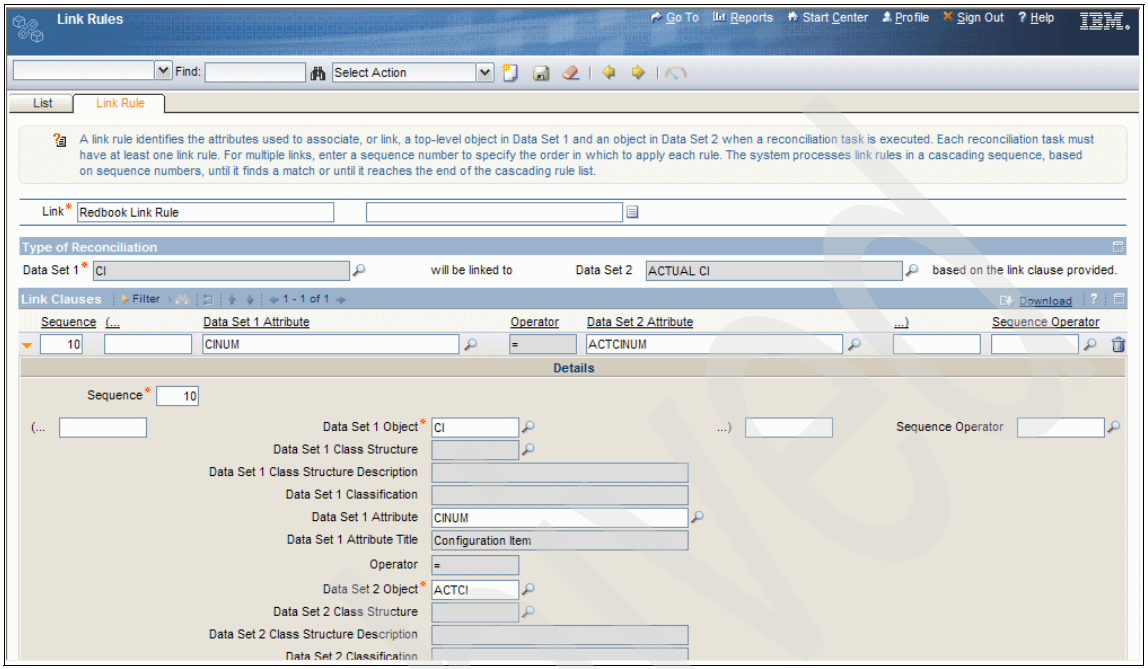


Figure 8-119 Creating link rules

Defining 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, one 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.

Once 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 8-120 on page 351 shows a sample of a Comparison Rules application. To open the Comparison Rules application, select **Administration** → **Reconciliation** → **Comparison Rules**.

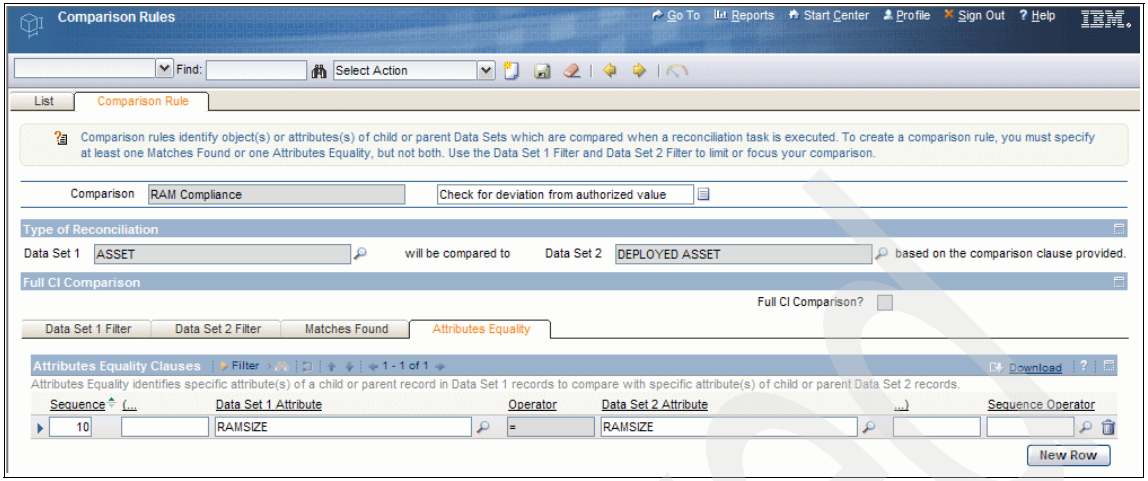


Figure 8-120 Defining Comparison Rules

Comparison Rules components

The following components can be used 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 should 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 reconciliation comparison. To designate which output objects to select, a Data Set 1 filter clause defines 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 a different 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 8-4 describes the elements of a Data Set 1 or Data Set 2 filter clause.

Table 8-4 Elements of a Data Set 1 or Data Set 2 filter clause

Field	Function	Rules/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.

Field	Function	Rules/Requirements
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).
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 reconciliation.	► 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.

Field	Function	Rules/Requirements
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.
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.

Field	Function	Rules/Requirements
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 8-5 describes the operators that can be used to define the ratio between Data Set 1 and Data Set2 object instances.

Table 8-5 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.

Setting up reconciliation tasks

Before executing a reconciliation 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.

The Reconciliation Tasks application has the following tabs:

- List: To search for tasks.
- Reconciliation Task: To define new tasks and to view, edit, duplicate, and delete existing tasks.

A reconciliation task consists of three primary components:

- Task filter (optional)
- Link rule (required)
- Comparison rule (optional)

Figure 8-121 on page 357 shows a sample of the Reconciliation Tasks application.To open the Reconciliation Tasks application, select **Administration** → **Reconciliation** → **Reconciliation Tasks**.

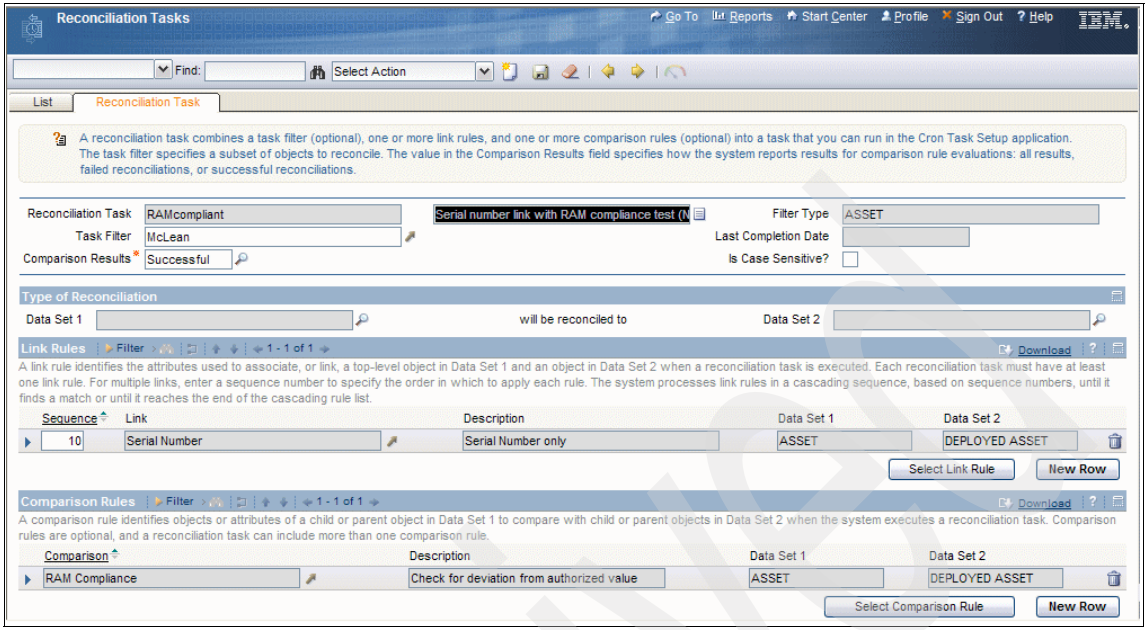


Figure 8-121 Reconciliation Tasks

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 reconciliation.
- Task filter (optional): Specify a task filter for the reconciliation task by selecting a task filter in the Task Filter field. When selecting a task filter, the application displays the type for the selected filter in the Filter Type field.
- 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. The Link Rules table window on the Reconciliation Task tab displays the following information about the link rules used in the reconciliation task:
  - Sequence: 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): In the Comparison Rules table window, specify one or more comparison rules for the reconciliation task. 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 **Select Comparison Rule** button in the Comparison Rules table window to select one or more comparison rules for a reconciliation task. Selecting this button opens a dialog box that lists comparison rules created.

  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.

From the Reconciliation Task tab, several options can be used to add task filters, link rules, and comparison rules to the task:

- Click **Select Link Rule** to open the Select Link Rule dialog box. This dialog box displays a list of existing link rules. It is possible to select one or more link rules for the task.
- Click **Select Comparison Rule** to open the Select Comparison Rule dialog box. This dialog box displays a list of existing comparison rules. It is possible to select one or more comparison rules for the task.

- Click the **Detail Menu** icon next to the Task Filter, Link, and Comparison fields to select one of the following options:
  - Open a Select Value dialog box to choose from a set of existing task filters, link rules, or comparison rules.
  - Go to the selected application. Once in the application, It is possible to create a new task filter, link rule, or comparison rule, or select an existing record and modify it.

Scheduling tasks

When scheduling a reconciliation task in the Cron Task Setup application, the name specified in the Reconciliation Task field must be used in the Reconciliation Tasks application to set up the cron task. 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.

Figure 8-122 shows a sample of a Cron Task Setup window for a reconciliation task. To open the Cron Task Application, select **System Configuration** → **Platform Configuration** → **Cron Task Setup**.

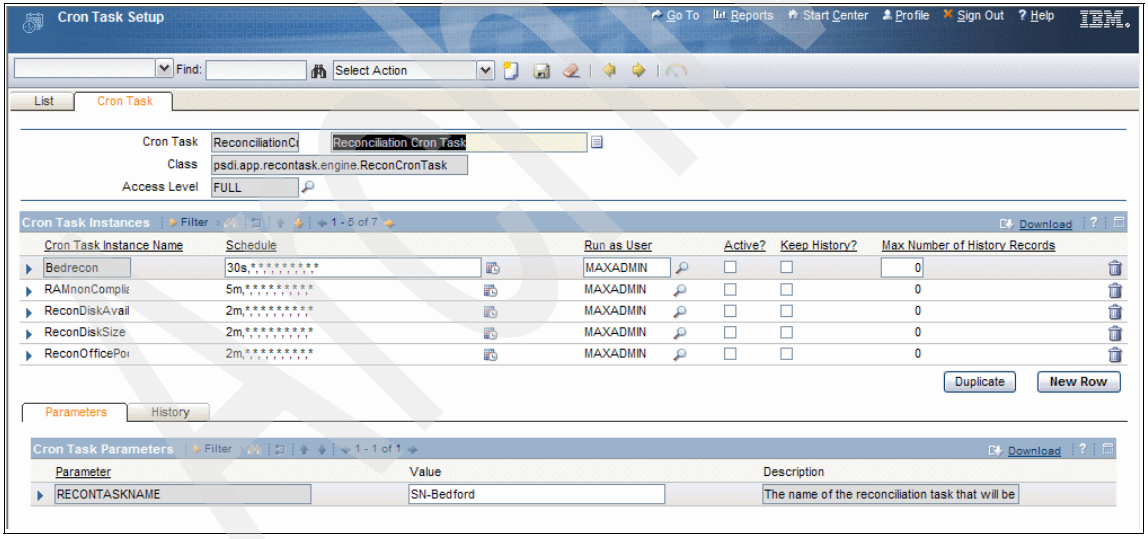


Figure 8-122 Cron Task Setup

Creating a cron task to schedule the reconciliation

Reconciliation tasks are defined in the Reconciliation module applications, but the schedule for running the task must be set up in the Cron Task Setup application. A reconciliation task record combines the components required for a reconciliation into a specific job task that the system runs based on the schedule that is set up in the Cron Task Setup application. Before running the reconciliation process, define a cron task to set up a schedule for running the reconciliation task. The name entered in the Reconciliation Task field for the reconciliation task is the parameter in the cron task that identifies which reconciliation task to process.

Figure 8-123 shows how to create a cron task and set the parameter values with the reconciliation task name.

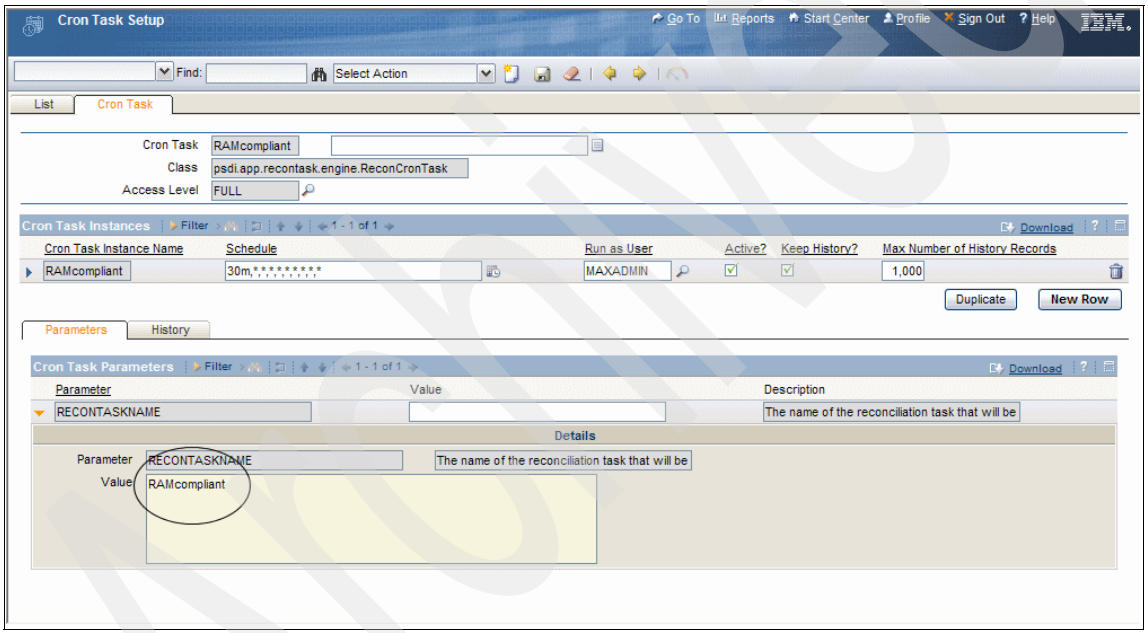


Figure 8-123 Setting up a cron task with reconciliation task name

Defining a cron task

The cron task must point to the class psdi.app.recontask.engine.ReconCronTask.

Note that the Class field contains the class file for the reconciliation process. 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

Note: The system allows the creation of cron tasks for multiple reconciliation tasks. If different 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. We recommend also that reconciliation cron tasks that import from Integration Composer and reconciliation cron tasks do not occur simultaneously.

Viewing the results of the reconciliation

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

It is also possible to use the CI Reconciliation Results application to mark reconciliation results resolved after reviewing and resolving discrepancies between Authorized CI data and Actual CI data. When a result is marked as resolved, information about how the discrepancies were resolved can also be recorded.

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.

Resolving discrepancies

To resolve discrepancies, use the CI Reconciliation Results application.

When using the CI Reconciliation Results application to view and manage result records produced after the system runs a reconciliation task, it is possible to view and manage two different kinds of 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 reconciliation process 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 allows you to select one of the following options for comparison results when setting up a reconciliation task:
  - All results, both successful and failed matches of the CI Reconciliation Results Application
  - 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.

It is also possible to use the reconciliation results application to record information about how discrepancies between Data Set 1 and Data Set 2 were resolved. If a discrepancy was evaluated and resolved, mark a result record as resolved. It is also possible to enter information about how the issue was resolved or an explanation about why the discrepancy exists in the Comments.

8.5.10 Interaction with other processes

Because Configuration Management Process is responsible for maintaining information related to CIs, it interacts with all other ITSM processes. Some examples are:

- 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 would need to be restored in the event of a major outage.

In the CCMDB context, the Configuration PMP works closely with the Change PMP.

9

Mapping IT processes with CCMDB

After the initial installation of the CCMDB product, an important step is to customize the product to meet the needs of your specific Change Management process.

9.1 Customizing the data captured by your process

Each organization has different data requirements, such as the information required for change requests. The following sections describe how you might customize the CCMDB environment to handle your specific data requirements.

9.1.1 Choose a subset of your request types to map within CCMDB

Analyze the kinds of requests that you are processing on a daily basis, and decide what kinds of RFCs that you wish to support within CCMDB. Some questions to ask are:

- What kinds of data do these RFCs need to collect at request time?
- What information needs to be provided by the requestor to the person fulfilling the request?
- What kinds of information may the implementer of the request need to communicate back to the requestor?

9.1.2 Creating a new request classification

A request classification can be created to capture this additional data. For example, based on an analysis of our most common requests for change, we may decide to model requests to change the host name of a production server. The Process Request application can already capture the details of the production server using the Target CI section, but none of the existing request classifications capture the new host name. Go to the Classifications application by selecting **Administration** → **Classifications** in the Go To menu. Click **Insert** to create a new RFC classification called HOSTREQ. Set the parent classification to PMCHG. Each process manager has a top-level request classification to help organize their request classifications. New classifications for Change requests are typically created under the PMCHG parent classification.

Figure 9-1 on page 367 shows the modification of a classification.

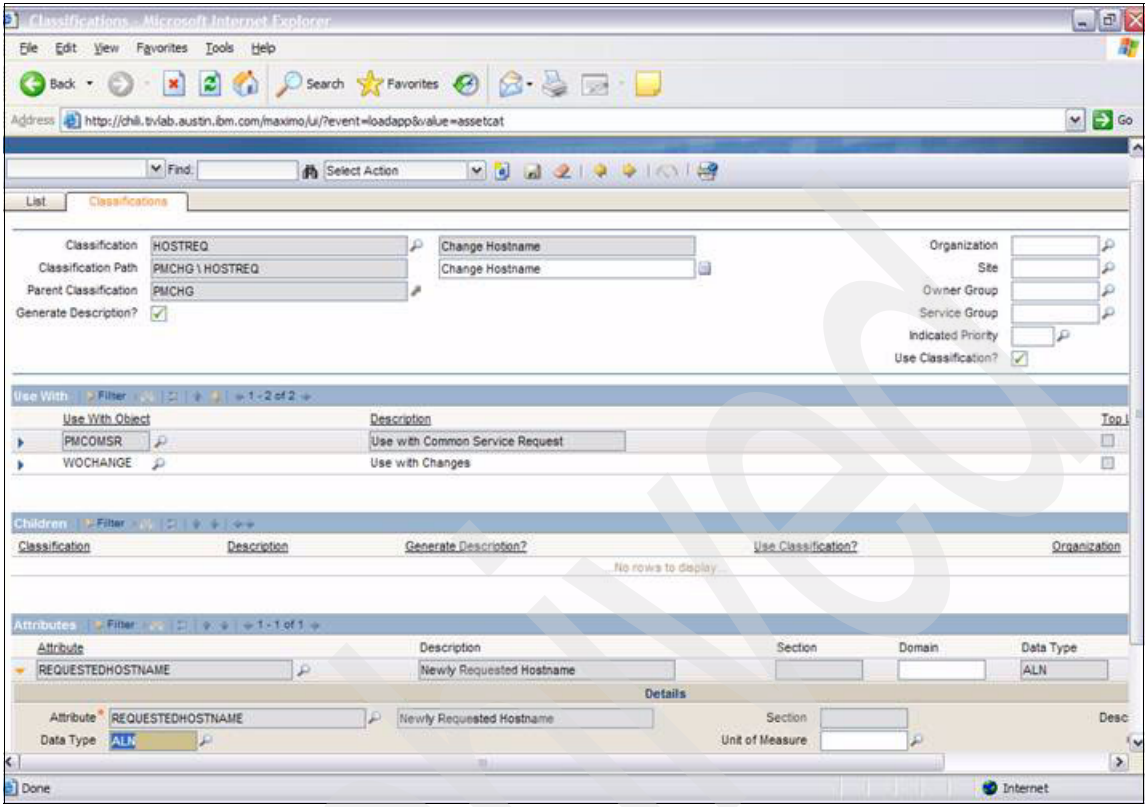


Figure 9-1 Modifying a classification

Create Use With entries for both the PMCOMSR and WOCHANGE objects. This allows the new request classification to be used with both the RFC object, and the Change workorder that is automatically created when the RFC is accepted. If you leave the Generate Description field checked, the description of the Change and RFC will be automatically generated based on the values of this classification.

Click **New Row** under the Attributes section to define a new attribute of type ALN called REQUESTEDHOSTNAME. This attribute will allow you to capture the new host name of the server. Click the **Details** button next to this new attribute, and be sure to set the attribute as **Mandatory** for both the PMCOMSR and WOCHANGE object (Figure 9-2).

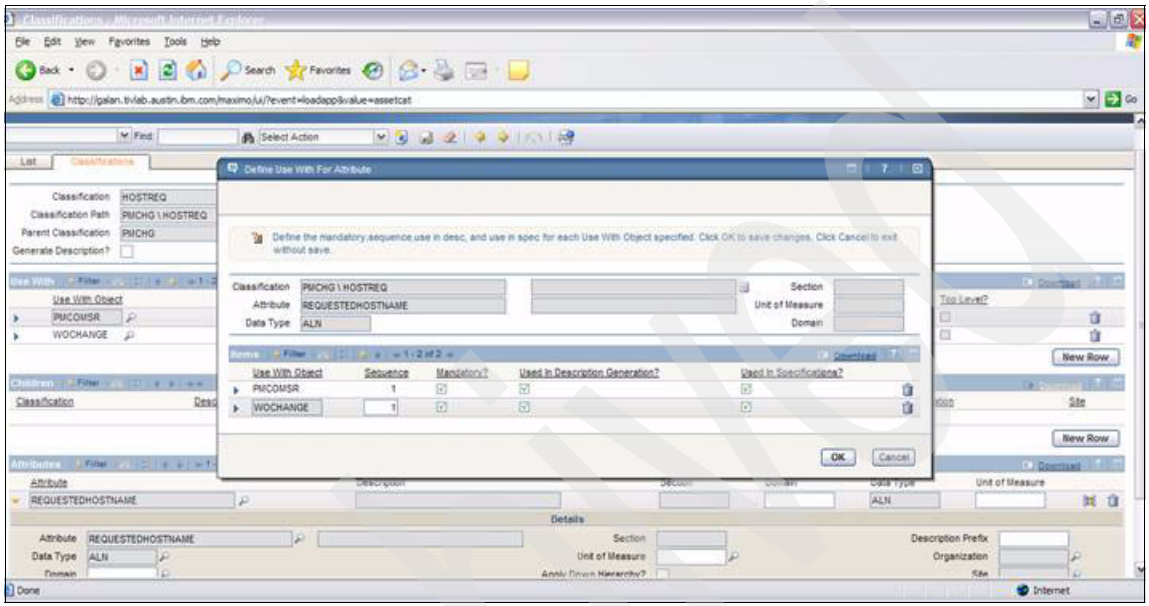


Figure 9-2 Defining new attributes

Return to the Process Request application by selecting **Service Desk** → **Process Requests** in the Go To menu. Click the **Insert** button to create a new Change request and classify the Change request as HOSTREQ. Expand the attributes section of the Change request and notice that it will now automatically prompt us for the REQUESTEDHOSTNAME attribute (Figure 9-3 on page 369).

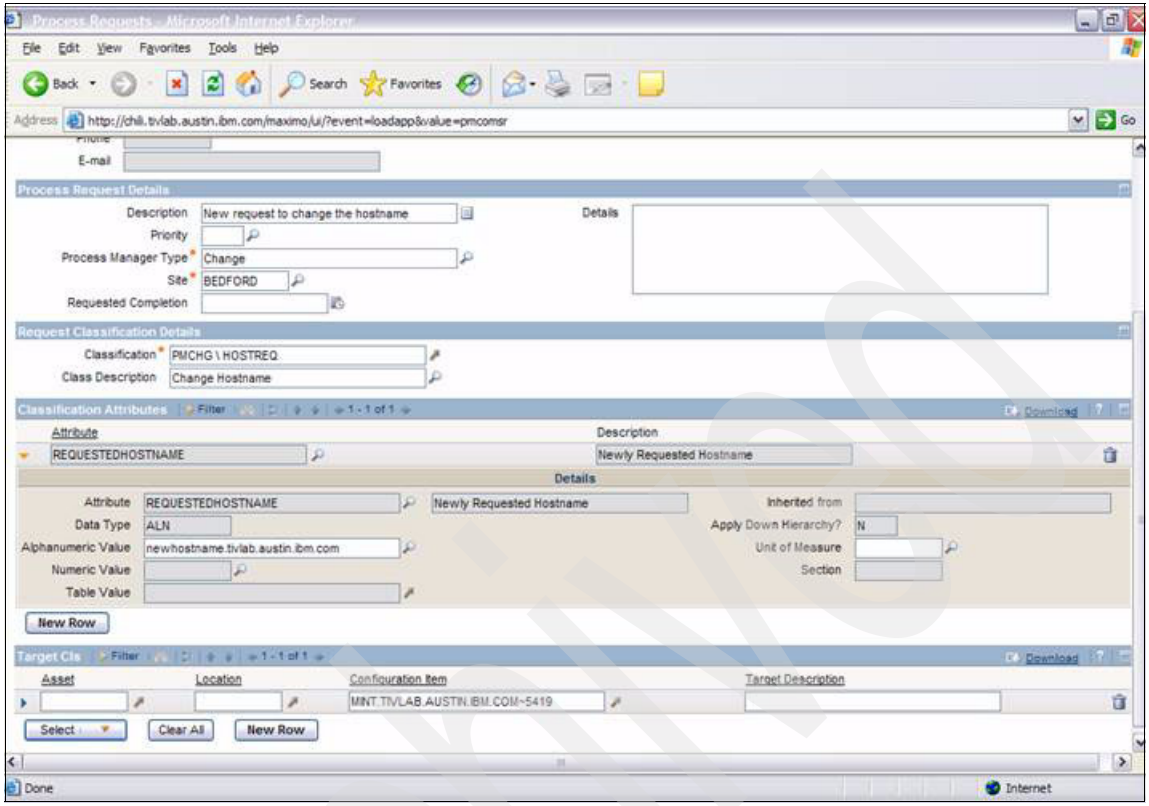


Figure 9-3 New HOSTREQ with prompt for host name

9.1.3 Modifying the choices associated with a field

Based on your Change process, you may have different choices available for the fields of a Change than are shipped by default. For example, because each customer's Change process may involve different milestones, most customers will modify the choices available for the Progress or Status fields of the Change. Choices for field values are presented to the user from a Domain. First, we will examine the types of choices you have for the domains in your Change application. For example, the choices for Assessment Type under the Impact Assessment tab of a Change are populated from a Domain. Do these values suit your assessment process? If not, please use the Domains application to customize and add additional domains. To determine which attribute is showing these domain choices, place your cursor in this field of the Changes application and press Alt-F1.

Figure 9-4 shows the details of an attribute.

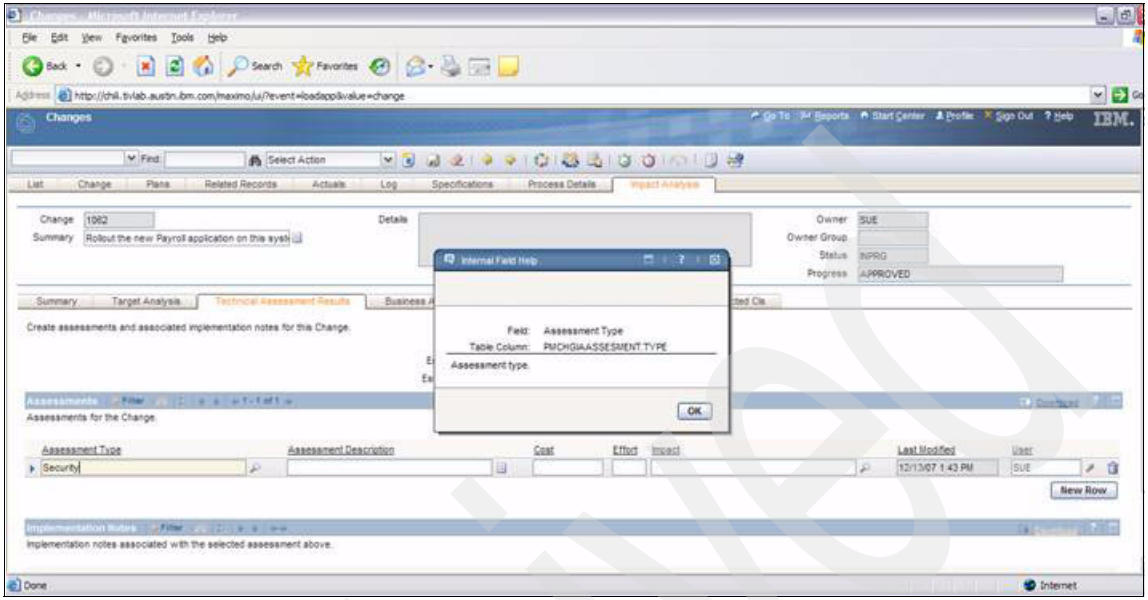


Figure 9-4 Displaying the details of an attribute

Go to the Database Configuration application and open the object that is named PMCHGIAASSESMENT. Go to the Attributes tab and locate the Type attribute (Figure 9-5 on page 371).

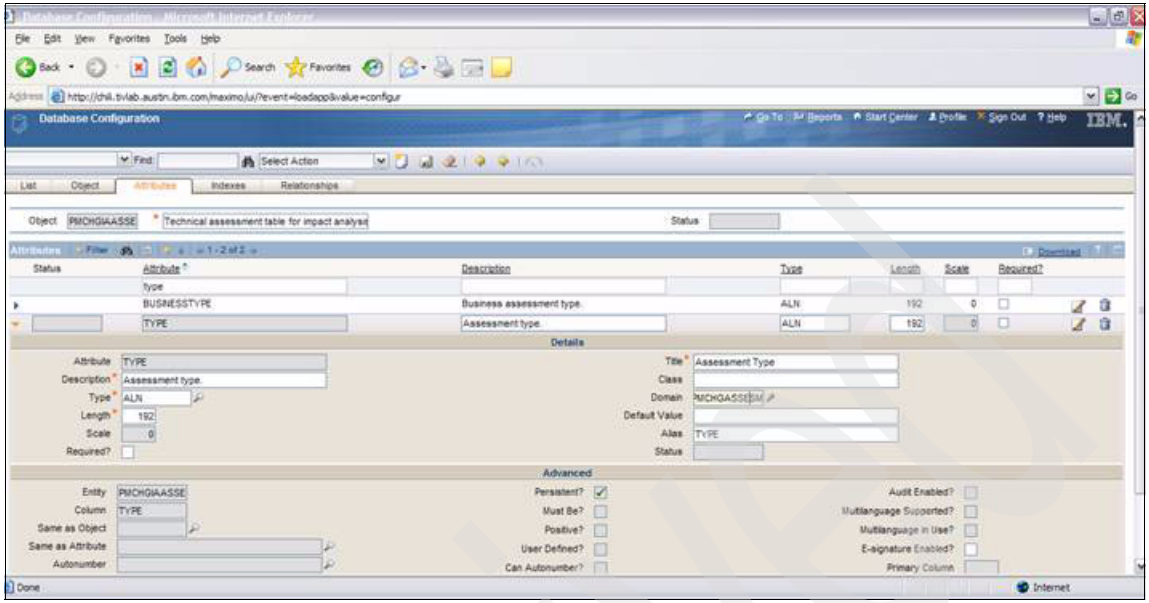


Figure 9-5 Attribute tab showing Type attribute

Notice that the Domain is set to PMCHGASSESMENTTYPE. Go to the Domains application and open this Domain (Figure 9-6).

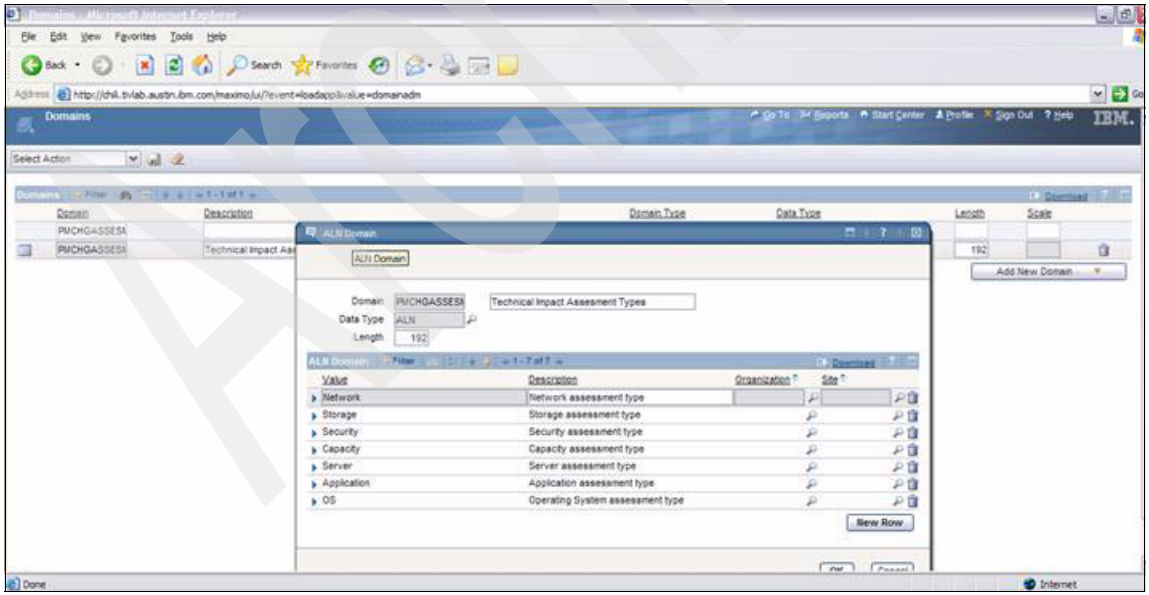


Figure 9-6 Domains application

You can add or remove new choices from this domain to more closely match your business process. If the domain is of data type SYNONYM, you can only add new values to the domain.

To modify the values shown when you attempt to change the Progress of a Change, you can add or remove values from the PMCHGPROGRESS domain in the Domains application (Figure 9-7).

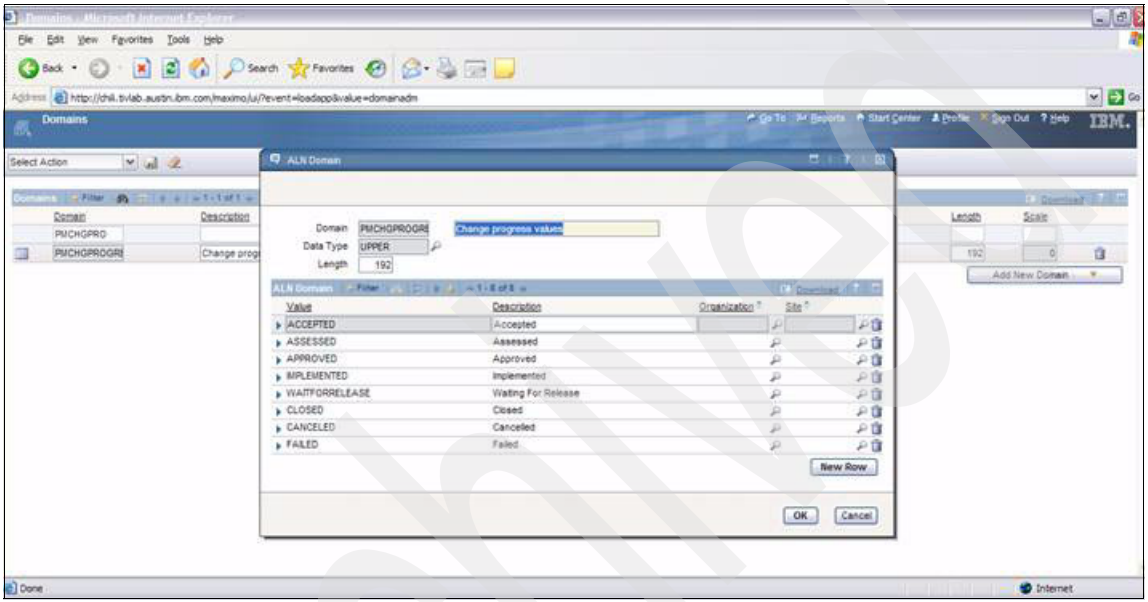


Figure 9-7 Changing values in the PMCHGPROGRESS domain

9.1.4 Customize your objects

If there is additional information that should always be captured by your RFC or Change workorder regardless of the type of Change, you can use the Database Configuration application to extend the RFC or Change objects at the database level. You can then add these additional attributes to the UI using the drag and drop capabilities of the Application Designer. None of this customization requires any coding. For example, the Process Request currently does not allow the request acceptor to specify an initial risk assessment when they are accepting the RFC. Imagine that our Change process requires this initial risk assessment be performed before the RFC is accepted, so that a different Job Plan can be applied to the Change based on this setting. We decide to add the risk field to the PMCOMSR so that the request acceptor can capture this initial Risk assessment for all types of Process Requests.

Adding an additional field to the object

First, we must create a Domain to allow our new risk field to be populated with choices. Go to the Domains application (select **System Configuration** → **Platform Configuration** → **Domains**) and click the **Add New Domain** button and add a new ALN Domain (Figure 9-8).

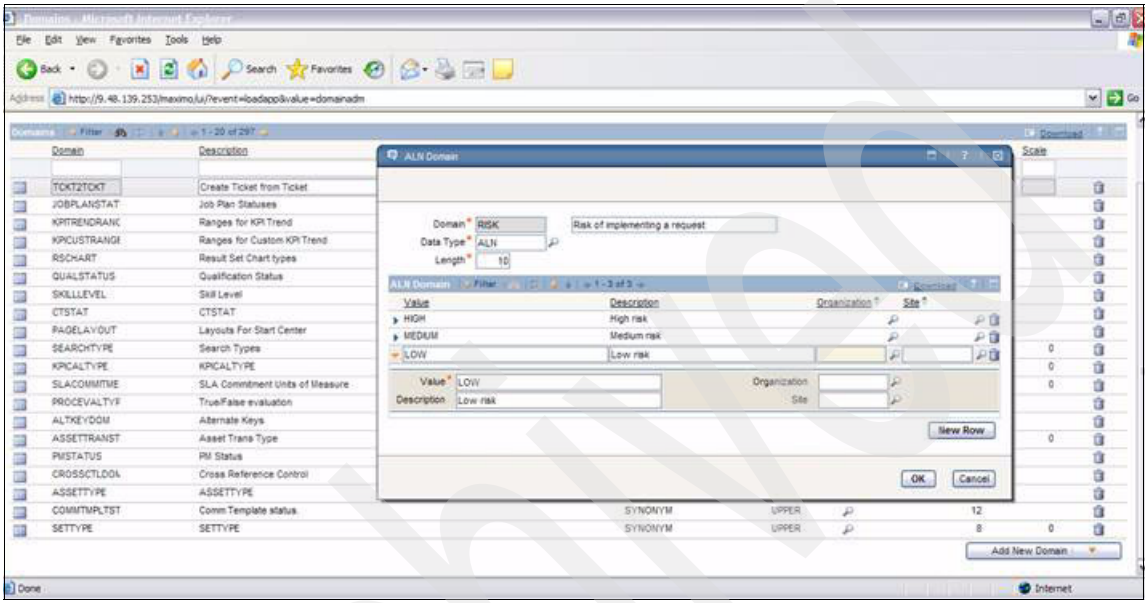


Figure 9-8 Adding a new ALN Domain

Go to the Database Configuration application and select the TICKET object (you have to modify the TICKET object because PMCOMSR extends from a ticket). Under the Attributes tab, click **New Row** and add a new attribute to capture the Risk of implementing the request. Set the Domain field to point to the RISK domain that you previously created and click **Save** at the top of the window (Figure 9-9).

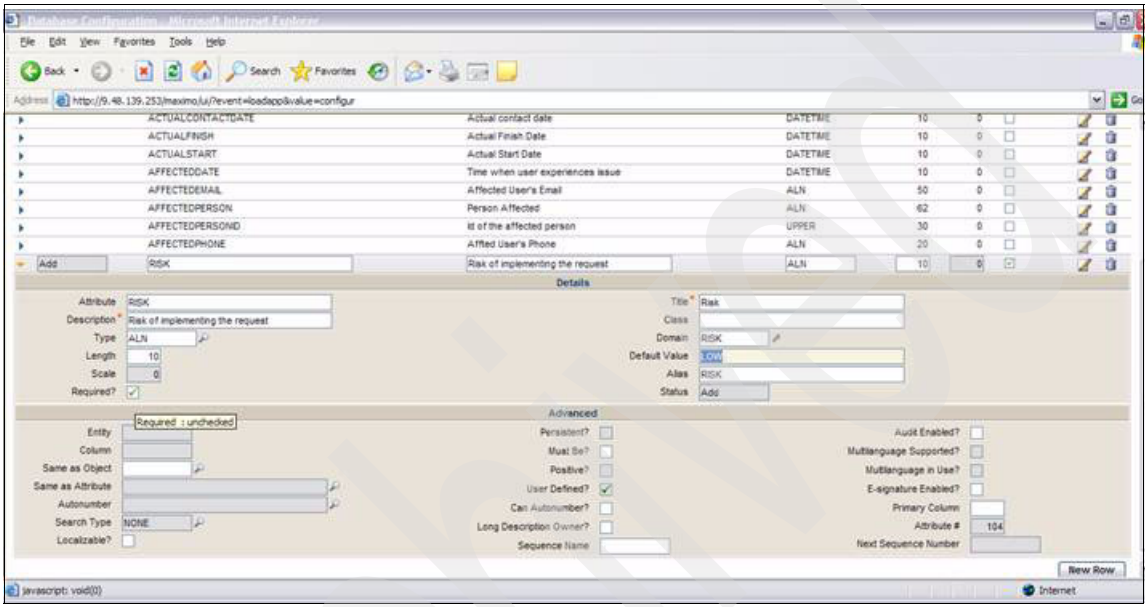


Figure 9-9 Adding a new attribute to the ticket

To actually apply this change to your database, you need to stop your WebSphere server by running:

```
C:\ibm\WebSphere\AppServer\profiles\ctgAppSrv01\bin\stopServer.bat
MXServer -user <username> -password <your password>
```

You then need to run this command to update your database:

```
C:\ibm\maximo\tools\maximo>configdb.bat
```

Now restart the MXServer:

```
C:\ibm\WebSphere\AppServer\profiles\ctgAppSrv01\bin\startServer.bat
MXServer
```

9.1.5 Adding an additional field to the UI

Now go to the Application Designer (select **System Configuration** → **Platform Configuration** → **Application Designer**) and choose the PMCOMSR application. Click the **Process Request** tab and click the **Control Palette** button to display the list of available widgets. Select the textbox from the Control Palette window and drag and drop it below the Requested Completion date field in the Process Request details (Figure 9-10).

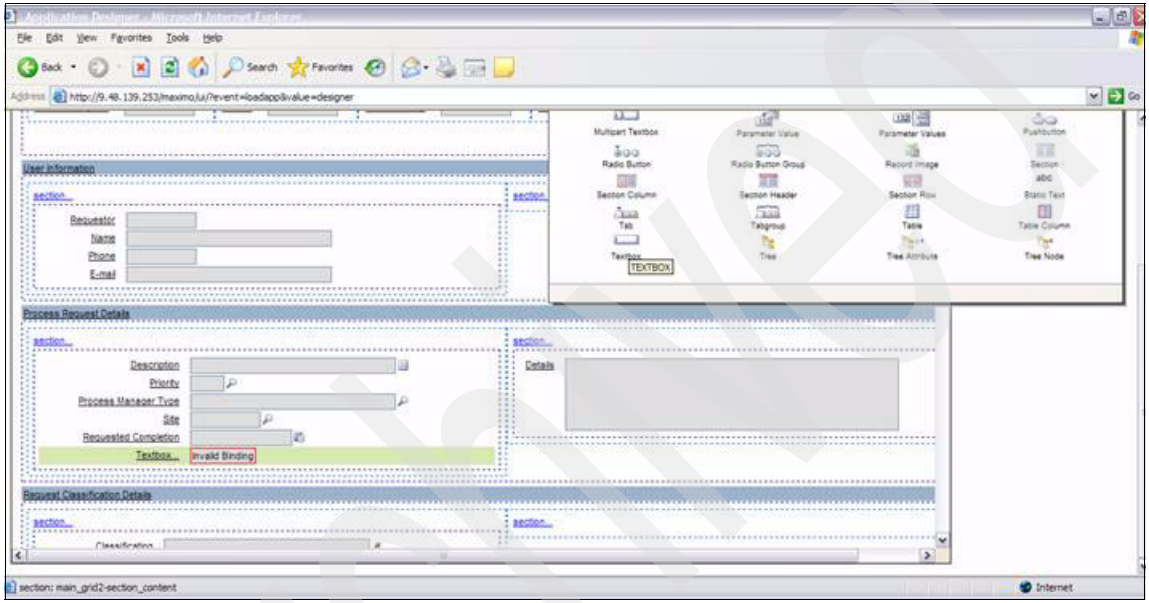


Figure 9-10 Adding a new field to the user interface

Now make sure the new textbox is selected and click the **Control Properties** button at the top of the page. This displays the Properties of this new textbox and will allow us to link the new textbox with the new field that we added to the database (Figure 9-11).

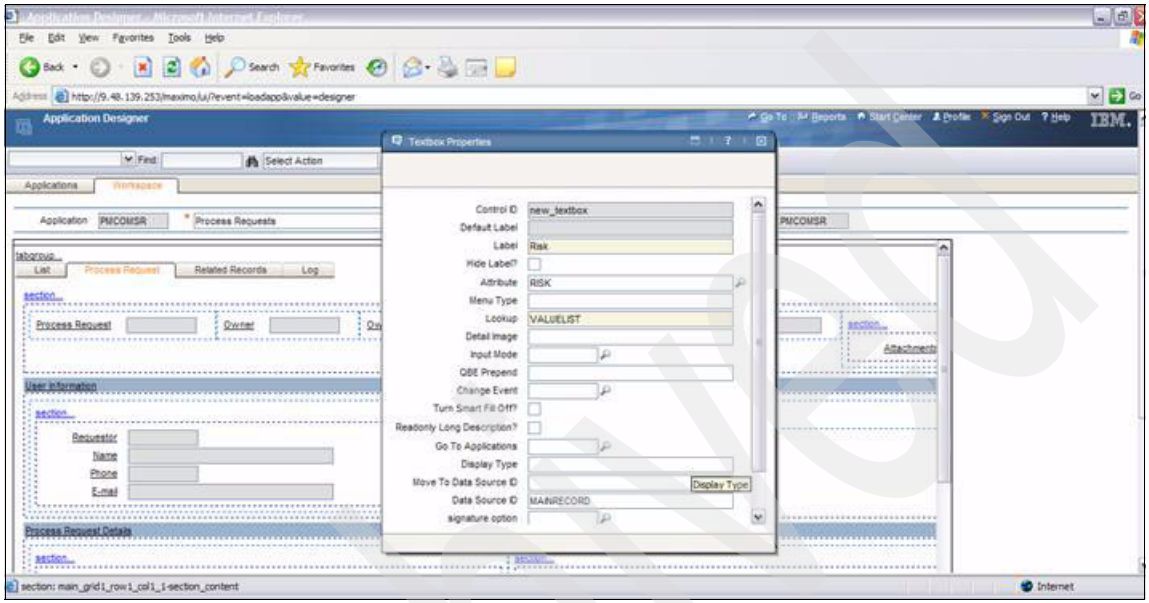


Figure 9-11 Linking the new text box to the new database field

Set the attribute field to point to the RISK attribute, and set the value of Lookup to be VALUELIST to allow the user to select their choices from the Domain we created earlier and click **Save** at the top of the window. Now we will launch the Process Requests application and see the new field we added. If you click the magnifying glass icon next to the Risk field, you will see the available choices that are populated from the Risk domain that we created (Figure 9-12 on page 377).

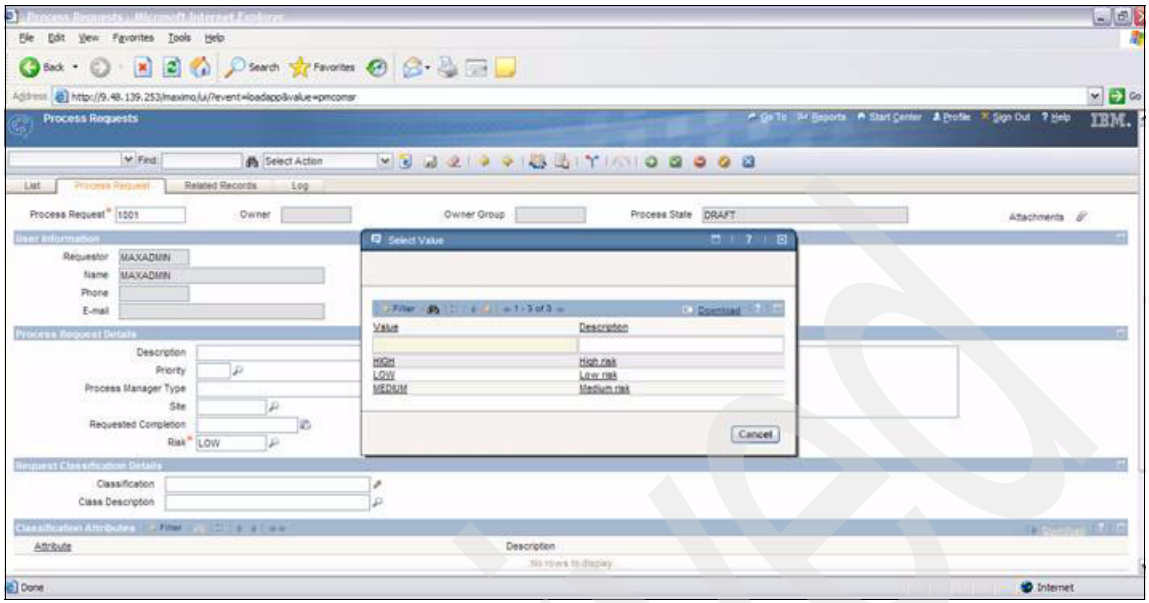


Figure 9-12 User interface prompting for new field value

For more detailed information, please read the Application Designer section in the ISM Base Services documentation.

9.2 Capturing the steps of your business process

Even though CCMDB provides some sample/standard business processes, most enterprises will want to modify the steps in these processes to match their own requirements. The following sections provide guidance on how this can be accomplished.

9.2.1 Choose a subset of your processes to map within CCMDB

Collect the IT processes across your organization. Decide the most common types of change implementation processes that occur in your environment and standardize them. It is usually a good idea to go after the types of Changes that are causing the most unplanned outages in your environment, or costing you the most money to implement. They can be collected on paper as a flowchart or mapped visually within ITUP Composer. Please see Chapter 3, “IBM Tivoli Unified Process Composer process mapping and design” on page 23 for more details on how to map a process within ITUP Composer.

Within CCMDB, each of the Change processes that you wish to capture can be collected as either workflows or Job Plans or some composite of the two. The simplest case is where the implementation of your Change is a simple series of tasks. This can be implemented by building a Job Plan.

9.2.2 Creating a new Job Plan

For example, to handle the RFC to change the host name of a production server, the Change team will need to:

- Get an approval from the CI owner.
- Schedule the Change within the approved Change Window for the system.
- Change the host name of the system.
- Send a request to the Configuration Librarian to update the host name of the Authorized CI.

First, we will create a series of tasks within the Job Plan to represent each of the steps listed above. We will assume that the first two tasks can be executed in parallel, and that the third and fourth tasks execute sequentially behind the first and second task.

Go to the Job Plans application, click **New Job Plan**, and name the Job Plan HOSTNAME (Figure 9-13).

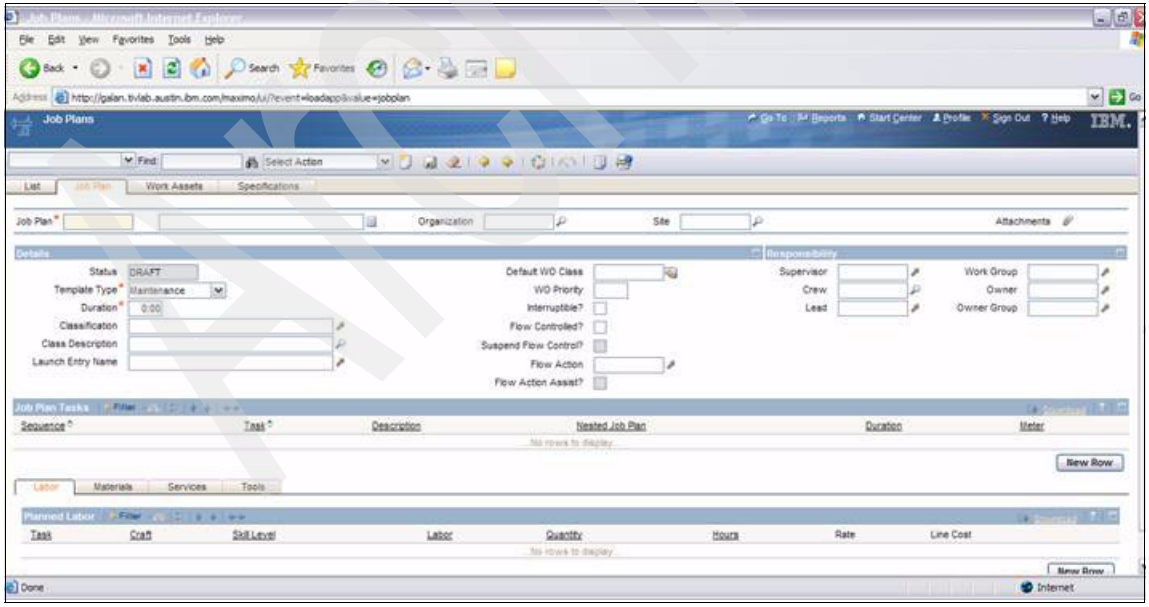


Figure 9-13 Creating a new HOSTNAME Job Plan

Click the **New Row** button to insert four tasks (Figure 9-14).

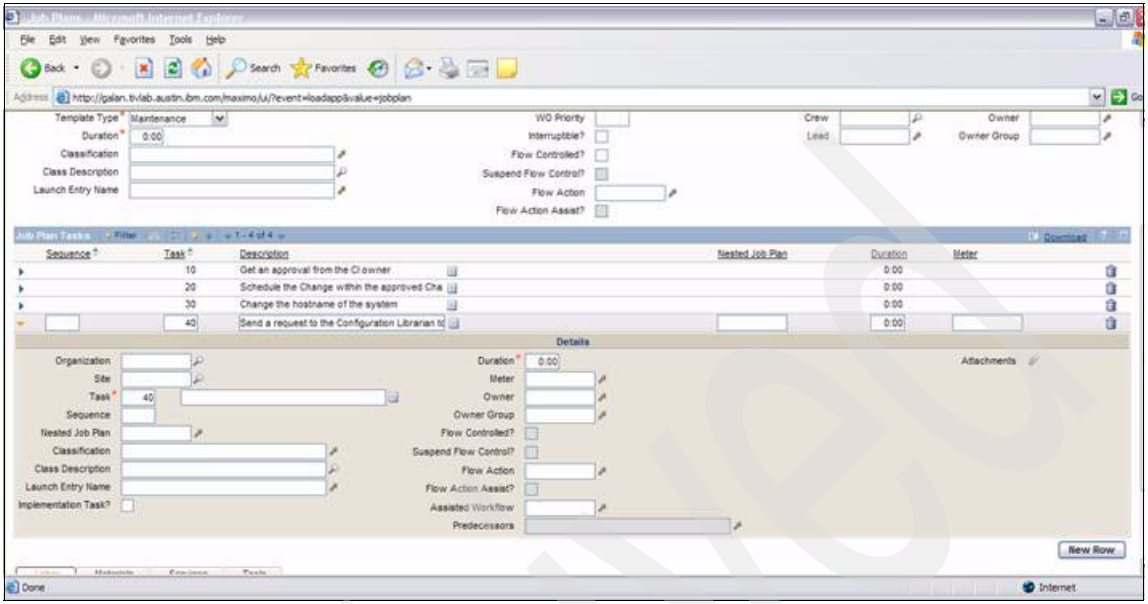


Figure 9-14 Inserting tasks to the Job Plan

For each of the tasks, expand the task details, and click the arrow next to the Predecessors field to choose the task that must complete before the current task can execute. The first two tasks that run in parallel do not have any predecessors. Set the predecessor of task 30 to be task 10 and 20. Set the predecessor of task 40 to be task 30 (Figure 9-15 on page 380).

If you know the owner for each of these tasks, you can specify it now in the owner field for each task. Otherwise, it will need to be filled in by the Change Owner after the Job Plan has been applied to a Change.

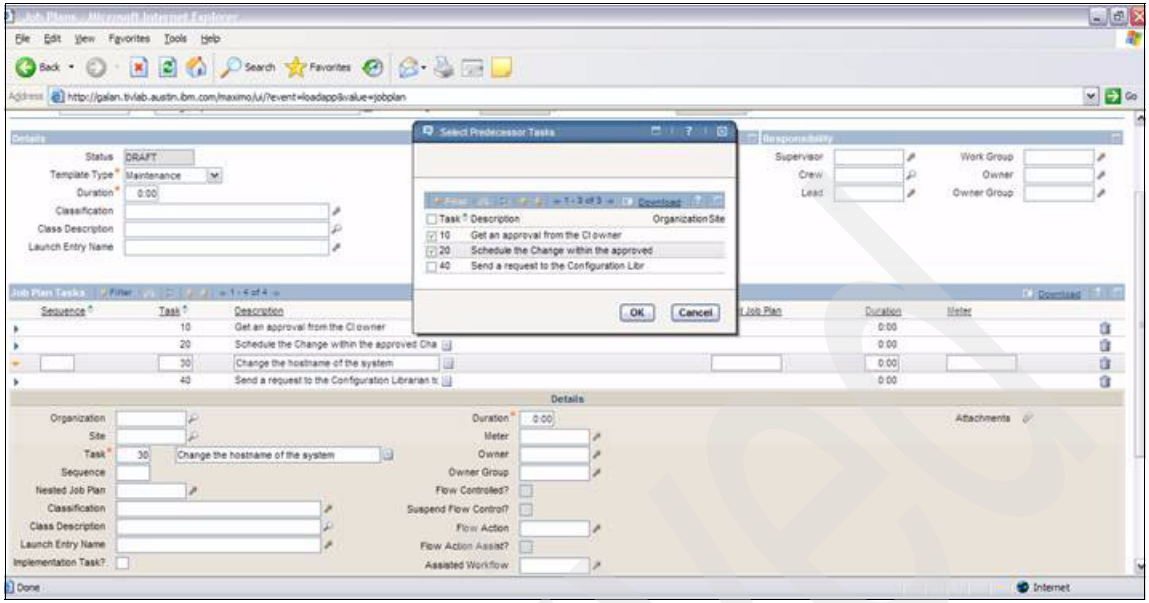


Figure 9-15 Setting predecessor tasks

9.2.3 Classifying your tasks

You can use a Classification for your tasks to organize them or to capture additional data during the task execution. For example, there is an approval task Classification to help organize all approval tasks of your Change. In the example above, set the Classification field of Task 10 to PMAPPR.

For each of the tasks in a Job Plan that will actually change a physical CI, you should mark the **Implementation Task** check box to be true. These types of tasks are very important, as they may actually cause system outages and need to be scheduled within Change Windows. Once these tasks are scheduled within the Change, they will be listed in the Change Implementation Schedule. In the example above, set the Implementation Task field of Task 30 to true.

9.2.4 Publish the Job Plan to the Change process

Check the **Flow Controlled** check box to be true. Set the Template Type field for the Job Plan to **Process** and type Change in the Default WO Class field. Now change the Status of the Job Plan to **Active**.

9.2.5 Add approvals to your Job Plan

Currently there are two approval workflows that are shipped out of the box with CCMDB7.1: APPACTWOA and APPACTWO. Both workflows send approvals to the owner of the task with which the approval workflow is associated. However, we want to send an approval record to the owner of the CI associated with the Change. This will require the development of:

- A new Role that points to the CI owners for a Change
- A new approval workflow that routes an approval to this new Role
- An associated action to call this new workflow

9.2.6 Creating a new role to point to CI owners

We will start by creating the new role. Go to the Roles application and click **New**.

Set the Name field to CIOWNERS and set the Type to “A set of data related to the record”. Set the Object field to WOACTIVITY, because the workflow will be executed on the task object. The Value field of the role can either point to an attribute of the Object, or any attribute that is accessible through any relationship that references a person. For example, to send the approval record to the task owner, you would set the Value field to :OWNER. Because we are going to send the approval record to the owner of the CI attached to the Change that is the parent of the current task, we will need to use three relationships to locate the correct approver.

To locate the CI owner from the task, the Value field needs to be set to :PMCHGTSKTOCHG.ALLCI.CI.PERSONID (Figure 9-16).

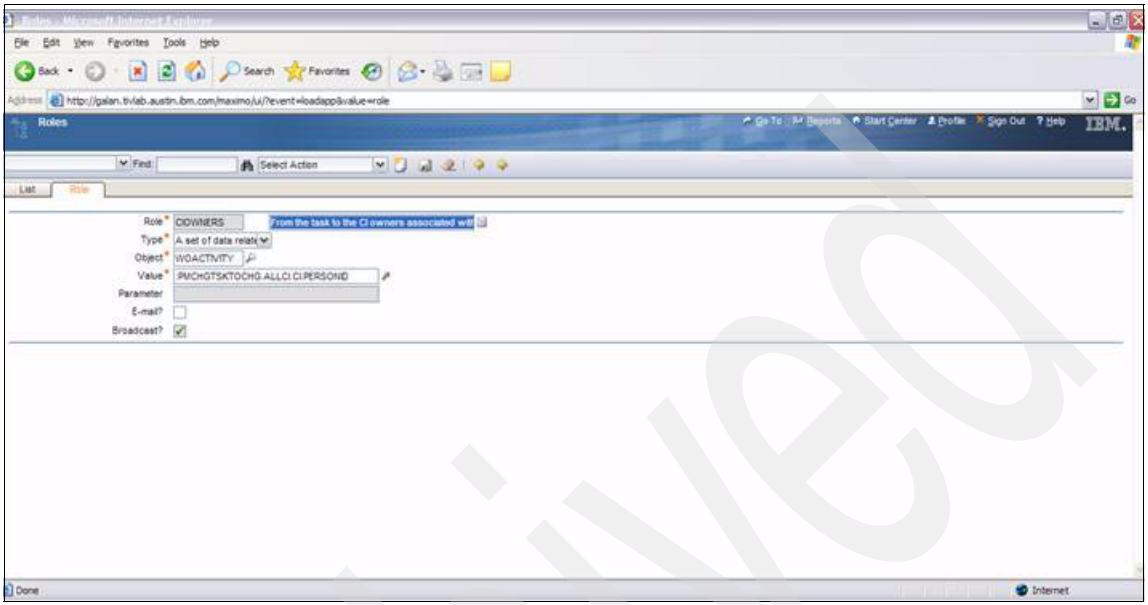


Figure 9-16 Value to show owner of task

PMCHGTSKTOCHG is a relationship between the Task and the Change, ALLCI is a relationship between the Change and the MULTIASSETLOCCI object, and CI is a relationship between the MULTIASSETLOCCI object and the CI object. PERSONID is the attribute of the CI object that contains the ID of the approver. You can find a list of the available relationships between one MBO and another MBO by querying the MAXRELATIONSHIP table in the database or by checking the Relationships tab of the Database Configuration application for each object.

9.2.7 Creating a new approval workflow

We can create the approval workflow required to send approvals to our new role by duplicating the sample workflow and modifying it. Go to the Workflow Designer and choose the APPWFWOA workflow and select **Duplicate Process**. Name the process APPCIOWN. Select the @APPROVTSK node and click the **Properties** icon. We will set the Role ID for the Approval Task node to point to the new Role that we created. Also, under Perform Accept Action, set it to **When All assignments are accepted** to handle the case where multiple CIs are attached to the Change. Save the workflow, and then choose **Enable Process** and **Activate Process** from the Action menu (Figure 9-17 on page 383).

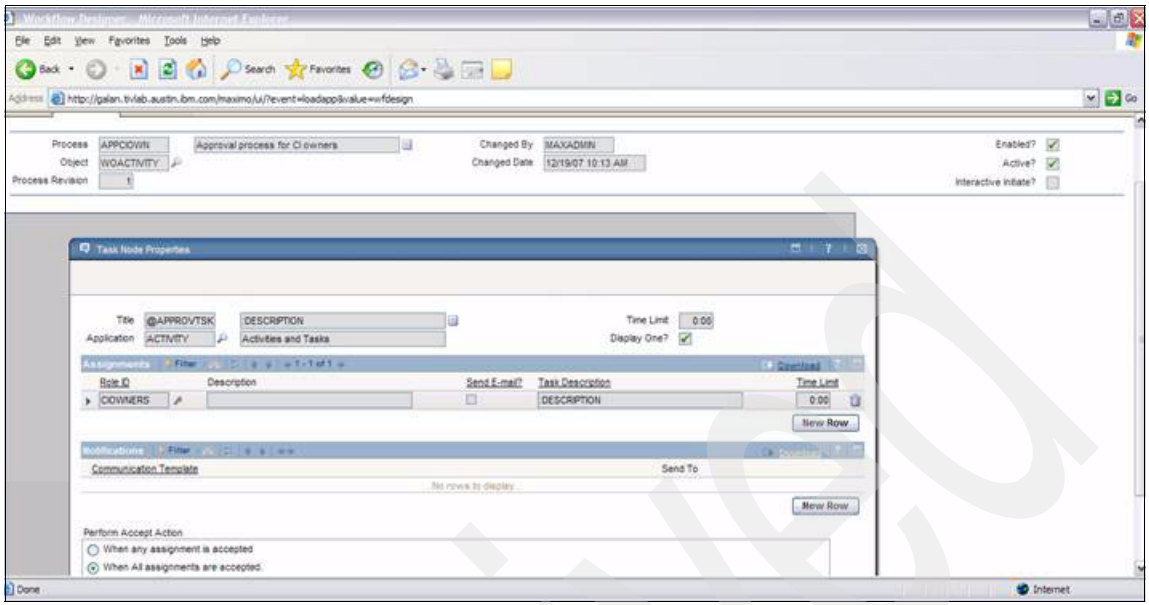


Figure 9-17 Setting Perform Accept action

9.2.8 Creating the action that invokes the approval workflow

We can create the action that calls the workflow by duplicating the existing APPACTWOA action and modifying it to call the new approval workflow. Go to the Actions menu and choose APPACTWOA and click **Duplicate Action**. Set the name of the action to be APPCIOWN. Set the Parameter field to APPCIOWN to have the action initiate the new workflow we created (Figure 9-18).

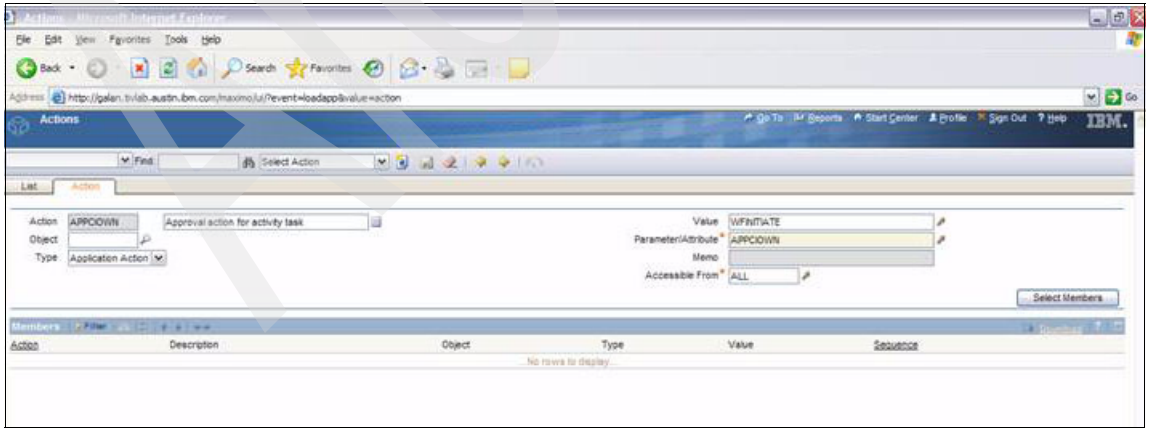


Figure 9-18 Setting the action to call a new workflow

Finally we need to attach this action to the correct task in the Job Plan, so that the workflow to collect these approval records is automatically kicked off at the appropriate time.

Open the Job Plans application for the HOSTNAME Job Plan, and expand task 10. Set the Flow Action field to point to the APPCIOWN action you created (Figure 9-19).

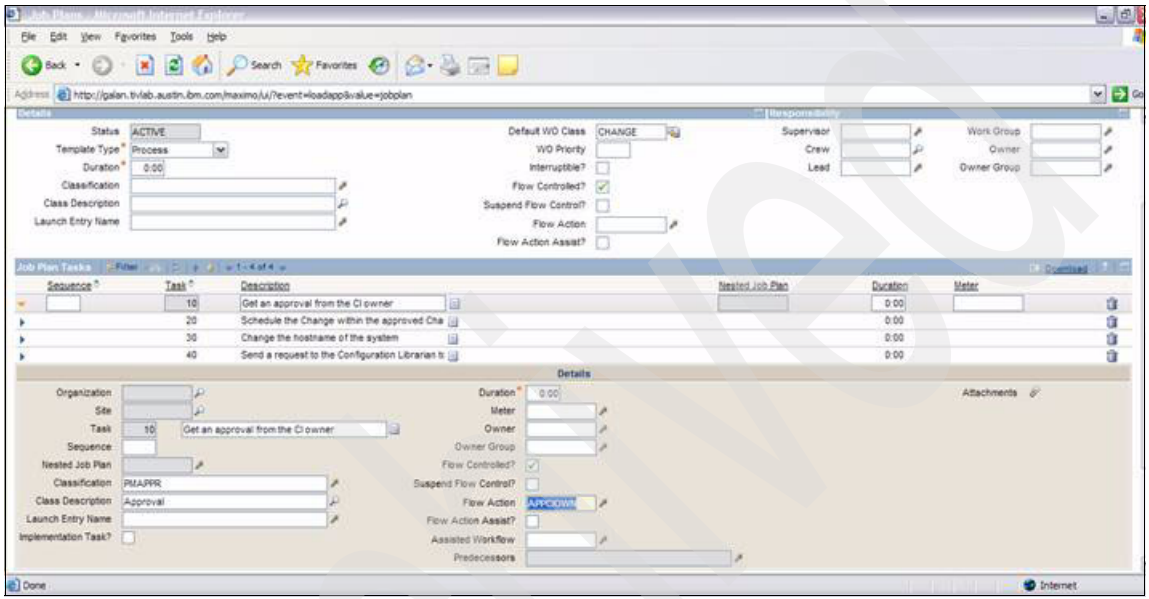


Figure 9-19 Setting Flow Action field to new action

9.2.9 Automate the steps of your Job Plans

You can create actions to automate any of the steps of your Job Plan using custom Java code. Here are the steps necessary to automate a Job Plan task:

- Create, compile, and deploy the custom Java code.
- Create an Action that will call the custom Java code from CCMDB.
- Create an Action Group that will call the new Action and then complete the current Job Plan task.
- Associate this Action Group with the correct Task within the Job Plan.

We will create an automated Java action to simulate changing the system host name.

9.2.10 Writing and deploying custom Java code

First, we will create a new file on your CCMDB server called ChangeHostnameAction.java under the directory C:\IBM\Maximo\applications\businessobjects\src. Our custom Java code will look up the value of the new host name from the parent Change and name of the production server associated with the parent Change and will print this information to the WebSphere system output logs.

Here is our custom Java code:

```
import java.rmi.RemoteException;

import psdi.common.action.ActionCustomClass;
import psdi.mbo.MboRemote;
import psdi.mbo.MboSetRemote;
import psdi.util.MXException;

public class ChangeHostnameAction implements ActionCustomClass {

   public void applyCustomAction(MboRemote mbo, Object[] params)
         throws MXException, RemoteException {

      //Use the MBO relationship to lookup the Change MBO Set from the
Task
      MboSetRemote changeSet = mbo.getMboSet("PMCHGTSKTOCHG");
      //Lookup the Change MBO from the Set
      MboRemote change = changeSet.getMbo(0);
      //Lookup the CI to change the hostname of from the Change
      MboSetRemote targetSystems = change.getMboSet("ALLCI");
      //Get the specific CI to change the hostname of
      MboRemote targetSystem = targetSystems.getMbo(0);
      String cINum = targetSystem.getString("CINUM");
      System.out.println("Change the hostname of: "+cINum);

      //Lookup the classification values for the Change, to find the
desired hostname
      MboSetRemote classificationValues =
change.getMboSet("WORKORDERSPEC");
      //Filter for the specific classification attribute we're looking
for
      classificationValues.setQbe("ASSETATTRID", "REQUESTEDHOSTNAME");
      //Lookup the specific classification attribute we're looking for
      MboRemote hostnameAttribute = classificationValues.getMbo(0);
      String newHostname = hostnameAttribute.getString("ALNVALUE");
```

```
      System.out.println("Change the hostname to: "+newHostname);

      //Insert your callout to the external automation system to change
the hostname here
      System.out.println("Callout to external system to change hostname
of "+cINum+" to new hostname: "+newHostname);
   }

}
```

You can compile the Java code on your CCMDB server with this sample ant build.xml file.

```
<?xml version="1.0"?>

    <project name="build" default="compile" basedir=".">
        <property name="MAXIMO_BUILD_DIRECTORY" value="c:/IBM/Maximo"/>
    <!-- Classpath to be used for all targets -->
    <path id="maximo.classpath">
        <dirset
dir="${MAXIMO_BUILD_DIRECTORY}/applications/maximo/businessobjects/clas
ses">
            <include name="**/" />
        </dirset>
        <dirset
dir="${MAXIMO_BUILD_DIRECTORY}/applications/maximo/commonweb/classes">
            <include name="**/" />
        </dirset>
        <dirset
dir="${MAXIMO_BUILD_DIRECTORY}/applications/maximo/maximouiweb/webmodul
e/WEB-INF/birt/script/classes">
            <include name="**/" />
        </dirset>
        <dirset
dir="${MAXIMO_BUILD_DIRECTORY}/applications/maximo/maximouiweb/webmodul
e/WEB-INF/classes">
            <include name="**/" />
        </dirset>
        <dirset
dir="${MAXIMO_BUILD_DIRECTORY}/applications/maximo/mboejbclient/classes
">
            <include name="**/" />
        </dirset>
```

```
        <dirset
dir="${MAXIMO_BUILD_DIRECTORY}/applications/maximo/mboweb/webmodule/WEB
-INF/classes">
            <include name="**/" />
        </dirset>
        <dirset
dir="${MAXIMO_BUILD_DIRECTORY}/applications/maximo/meaweb/webmodule/WEB
-INF/classes">
            <include name="**/" />
        </dirset>
        <dirset
dir="${MAXIMO_BUILD_DIRECTORY}/reports/birt/scriptlibrary/classes">
            <include name="**/" />
        </dirset>
        <dirset
dir="${MAXIMO_BUILD_DIRECTORY}/reports/birt/tools/classes">
            <include name="**/" />
        </dirset>
        <dirset dir="${MAXIMO_BUILD_DIRECTORY}/tools/maximo/classes">
            <include name="**/" />
        </dirset>
        <fileset
dir="${MAXIMO_BUILD_DIRECTORY}/applications/maximo/lib">
            <include name="**/*.jar" />
        </fileset>
    </path>

    <target name="compile"
        description="Compile custom business objects for CCMDB">
       <echo>Compiling custom business objects for CCMDB</echo>
        <javac
srcdir="${MAXIMO_BUILD_DIRECTORY}/applications/maximo/businessobjects/s
rc"

destdir="${MAXIMO_BUILD_DIRECTORY}/applications/maximo/businessobjects/
classes"
            debug="on"
            deprecation="off"
            target="1.5">
            <classpath>
                <path refid="maximo.classpath"/>
            </classpath>
        </javac>
</target>
```

```
</project>
```

Update the MAXIMO_BUILD_DIRECTORY to point to your own location and run this command to compile the code:

```
ant -file build.xml compile
```

The compiled class should now exist in the c:\IBM\Maximo\applications\maximo\businessobjects\classes directory.

To add the new class to the CCMDB EAR, run the command **`c:\IBM\Maximo\deployment\buildmaximoear`**.

Once the EAR has been rebuilt, it should reside in c:\IBM\Maximo\default\deployment\Maximo.ear.

To update Maximo.ear on your CCMDB server (we are assuming a default WebSphere environment), log in to the administration console at `https://localhost:9043/ibm/console`, expand the **Applications** tab, and choose **Enterprise Applications**. Select the MAXIMO application and click **Update** (Figure 9-20).

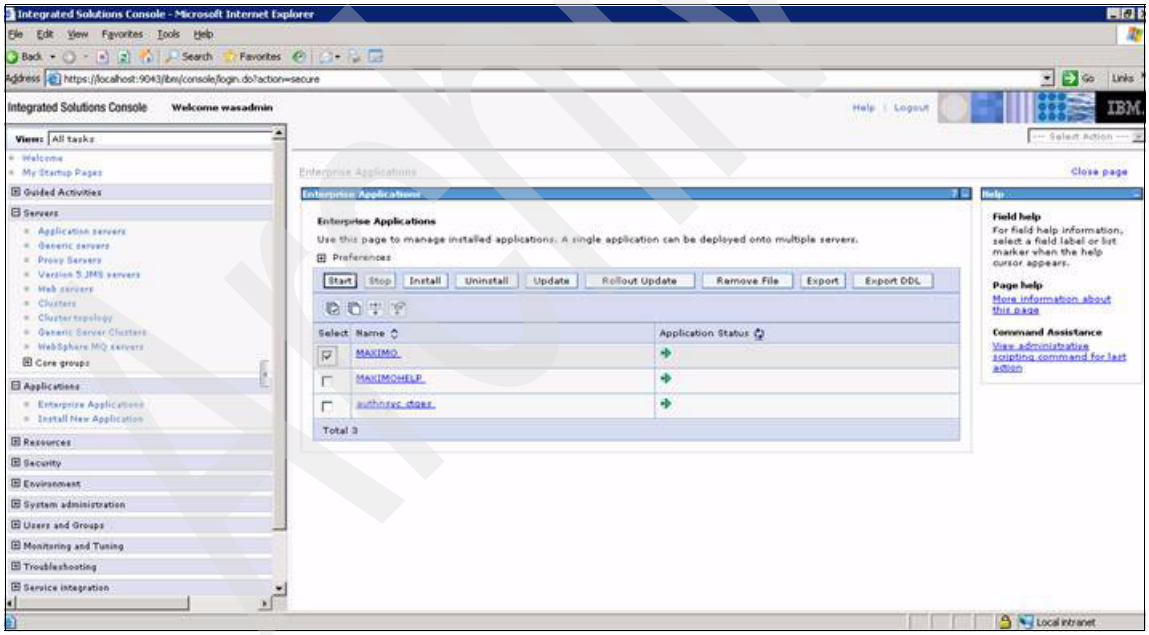


Figure 9-20 Updating the Maximo application

Set the full path of the replacement ear file to C:\ibm\maximo\deployment\default\maximo.ear and click **Next** three times, and then click **Finish**. Wait for the redeployment to finish and then click **Save** (Figure 9-21).

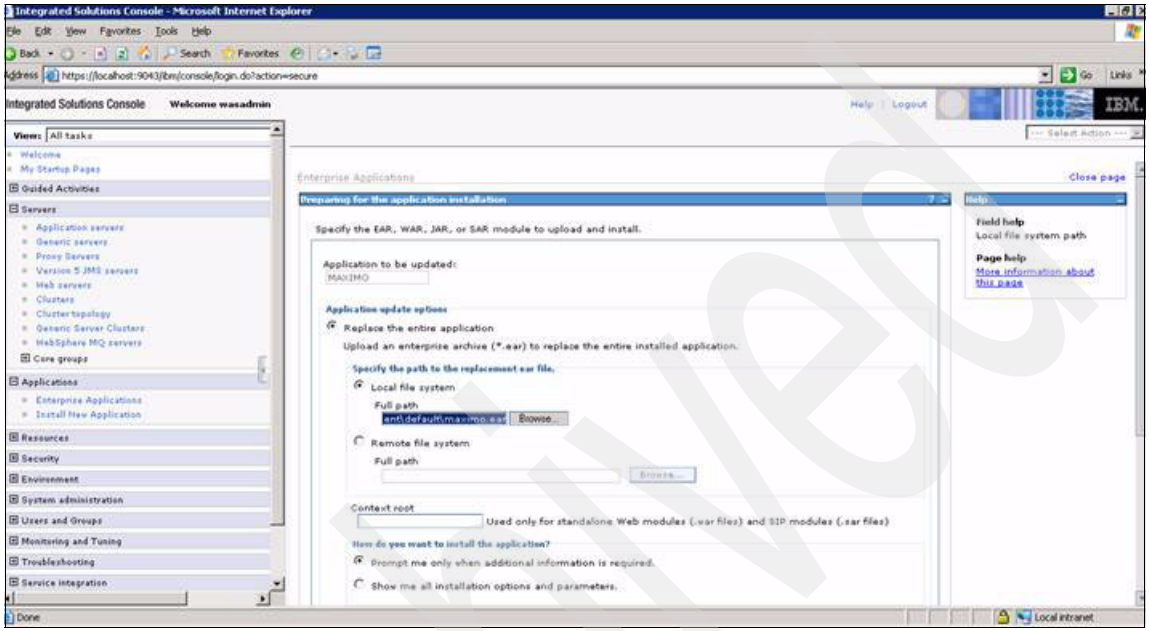


Figure 9-21 Setting the full path of the EAR file

9.2.11 Defining the custom action

Now we will create the action that will call this custom Java class. Log in to the CCMDB UI and go to the Actions application and create a new custom Java action that points to your ChangeHostnameAction class.

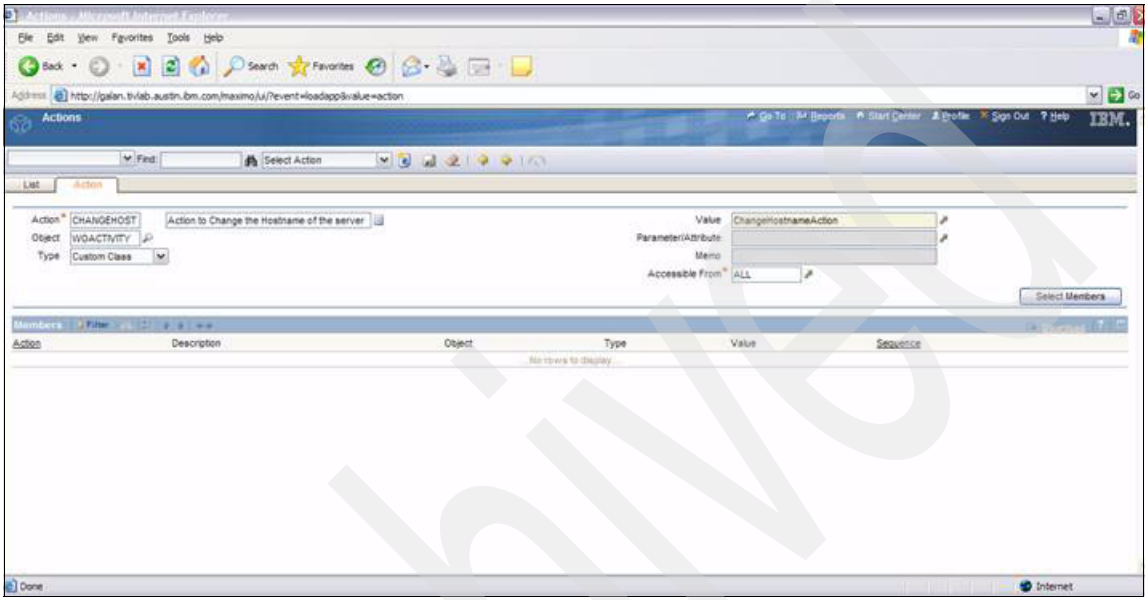


Figure 9-22 Creating new custom Java action

We need to create an action group that will call this action to change the host name and then set the status of the current task to the complete state so that the Change tasks automatically continue their execution after the host name has been changed. Go to the Actions application and create a new action group called CHANGEHOSTGRP of type Action Group. Click the **Select Members** button to choose the CHANGEHOST and PMCHGCOMPTASK actions as the members of this task (Figure 9-23).

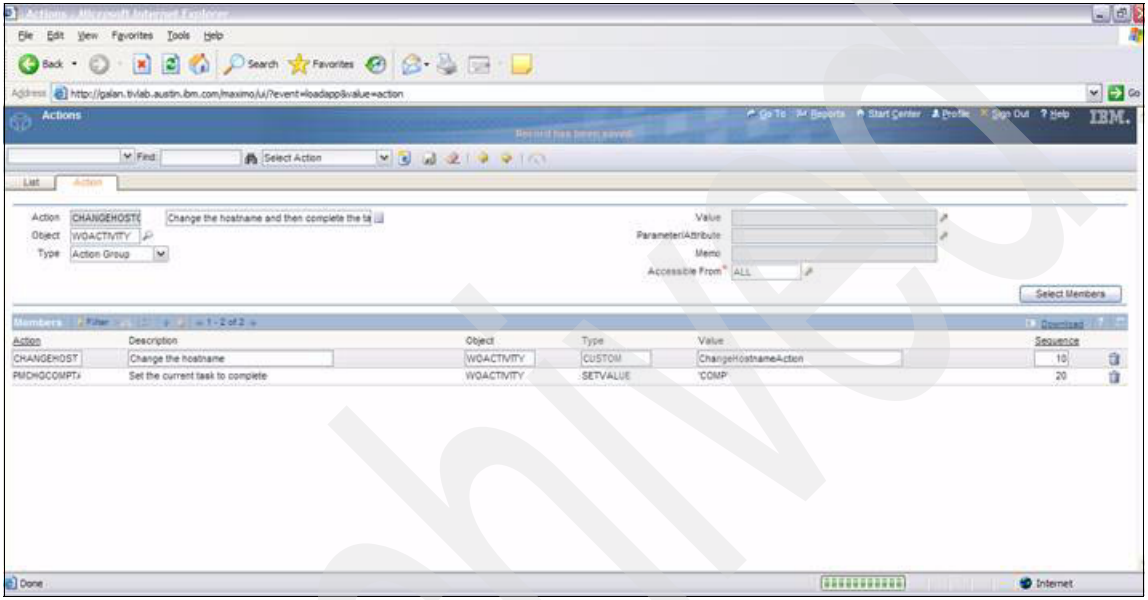


Figure 9-23 Selecting members of the task

Finally, you need to attach this action group to the correct task within your Job Plan. Go to the Job Plans application and choose the HOSTNAME Job Plan. Expand task 30 and set the Flow Action field to CHANGEHOSTGRP. Once this task is started, it will automatically call the Java code to change the host name of the CI and then complete the task (Figure 9-24).

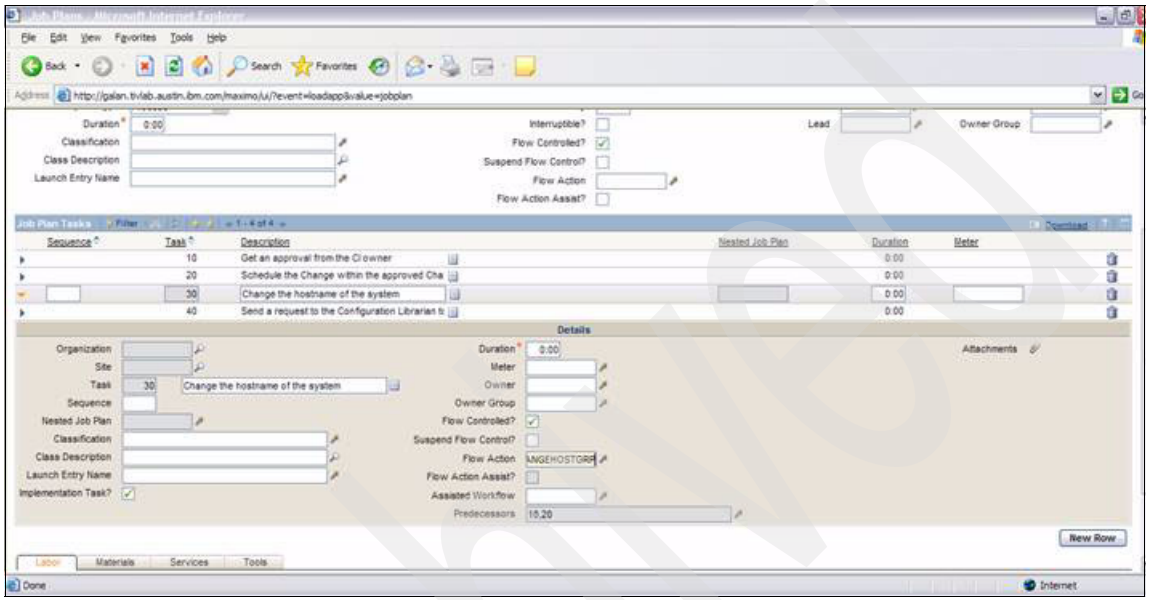


Figure 9-24 Change the Flow Action to CHANGEHOSTGRP

9.2.12 Automatically setting the fields in your Change when your RFC is accepted

While all of the classification attribute values that are set in the RFC are automatically copied to your Change when the RFC is accepted, it is very common to automatically set additional fields in your Change record based on the type of RFC. For example, you might want to modify your request acceptance workflow to automatically assign a Job Plan to your Change based on the classification of your RFC.

First, go to the ISMACCEPT workflow in the Workflow Designer and from the Action menu choose **Deactivate Process** and **Disable Process**. This is the top-level workflow that is triggered when you accept an RFC from the Process Request application.

Now open the PMCHGACC workflow in the Workflow Designer; this is the subprocess that the ISMACCEPT workflow calls to accept RFCs (Figure 9-25).

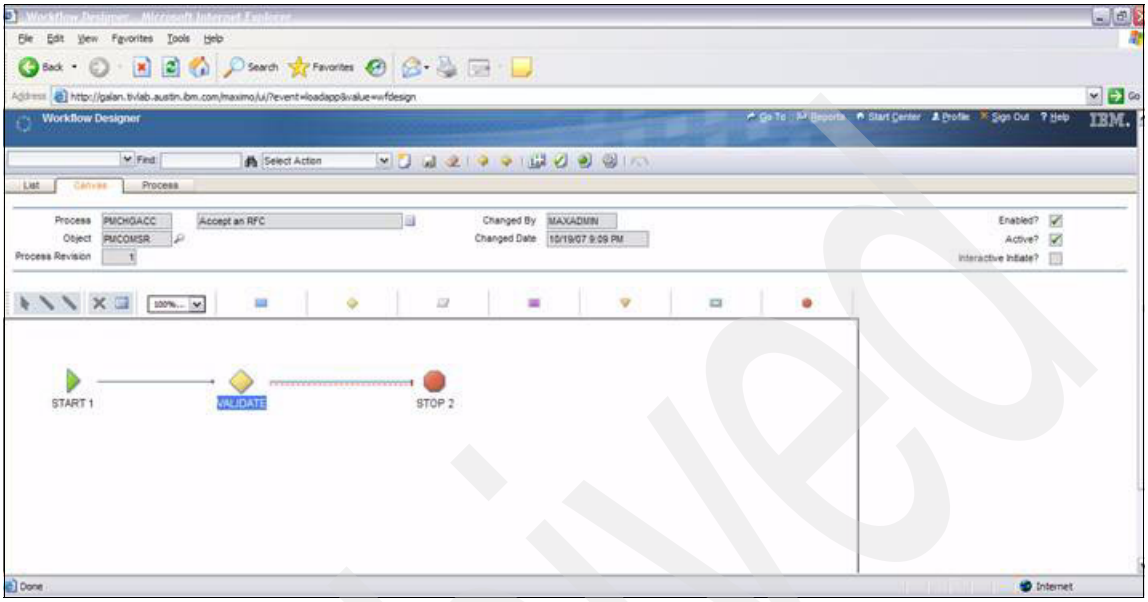


Figure 9-25 Default PMCHGACC workflow in Workflow Designer

From the action menu, choose **Deactivate Process** and **Disable Process**. Create a new workflow revision so you can edit it, and add a condition called IS STD CHG that examines the classification ID of the created change. Create a new positive output from the VALIDATE condition node to your new condition node and remove the old positive output from the VALIDATE to the STOP 2 node. Add a new positive and negative output from your IS STD CHG workflow to the STOP node.

The updated workflow should look like Figure 9-26.

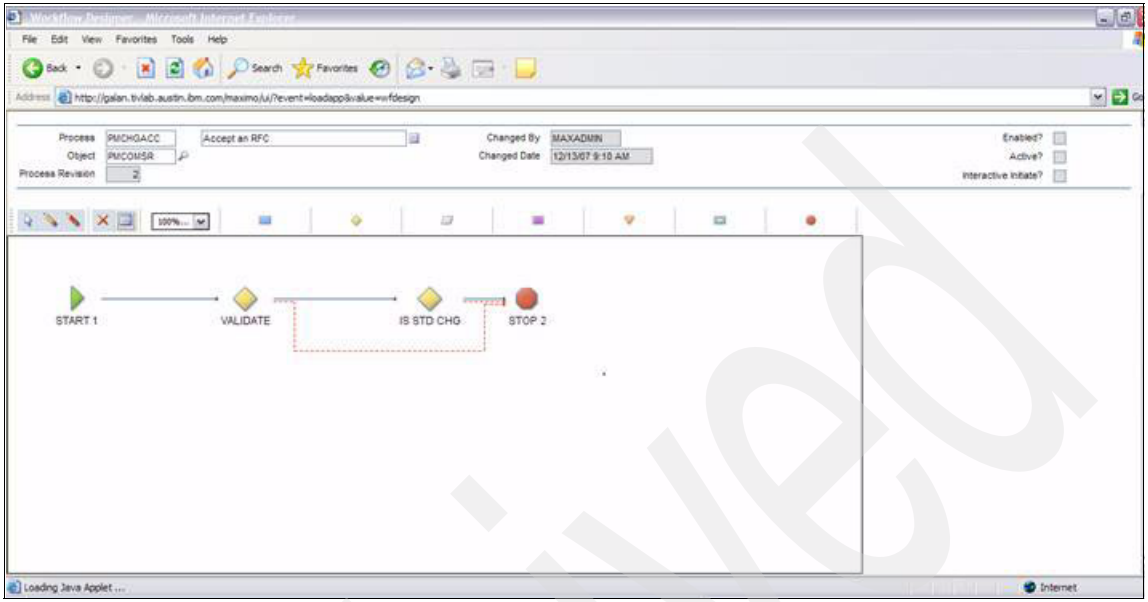


Figure 9-26 Updated workflow

Modify the properties of the new positive output from the VALIDATE to the IS STD CHG condition node to look like Figure 9-27 on page 395.

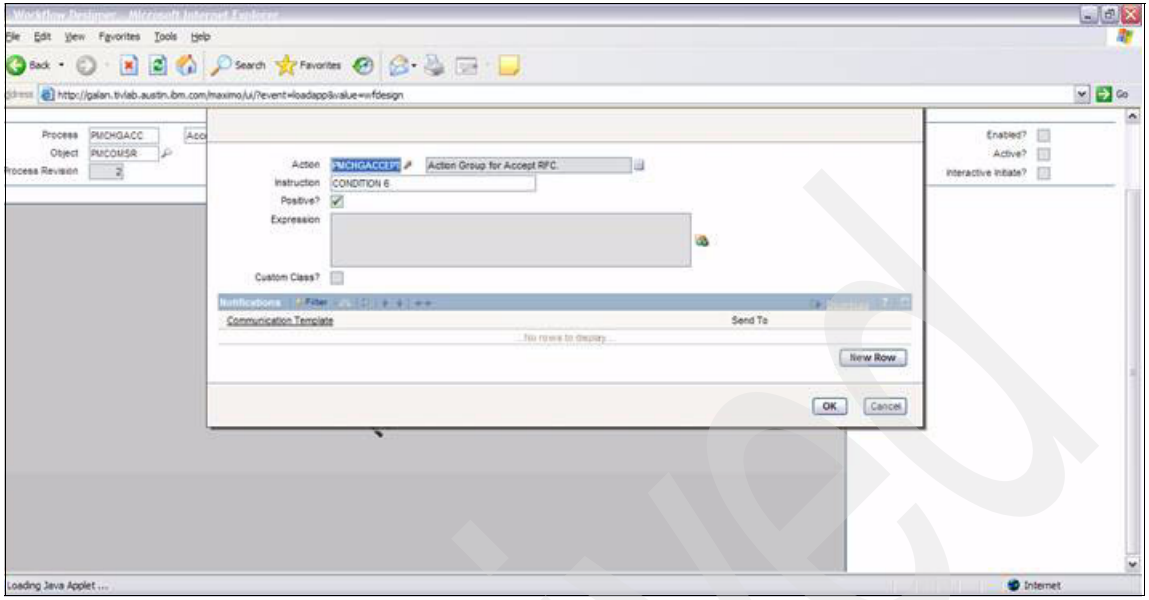


Figure 9-27 Update properties of the IS STD CHG condition node

Now modify the condition block of the new IS STD CHG condition to check the classification ID of the WOCHANGE.

Figure 9-28 is an example condition that identifies standard changes.

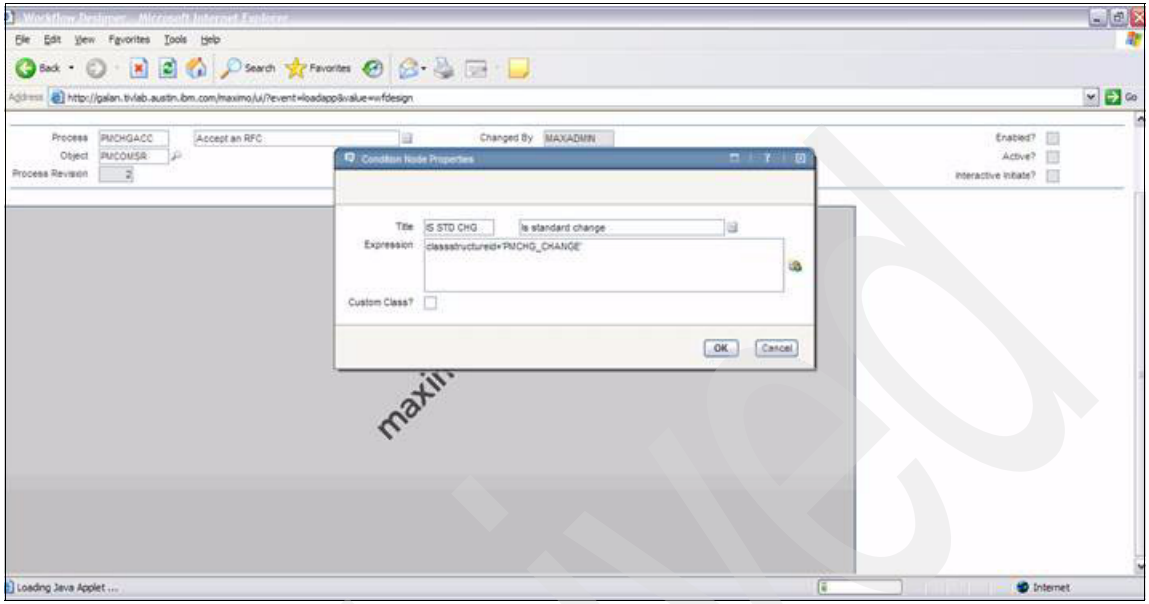


Figure 9-28 Sample condition for identifying standard changes

Select the positive output from the IS STD CHG to the STOP node, and click **Properties** to modify what Action is performed when the Change is a standard one. From the display next to the Action name, select **Go To Actions** to create a new action to set the Job Plan to the standard Change Job Plan (Figure 9-29).

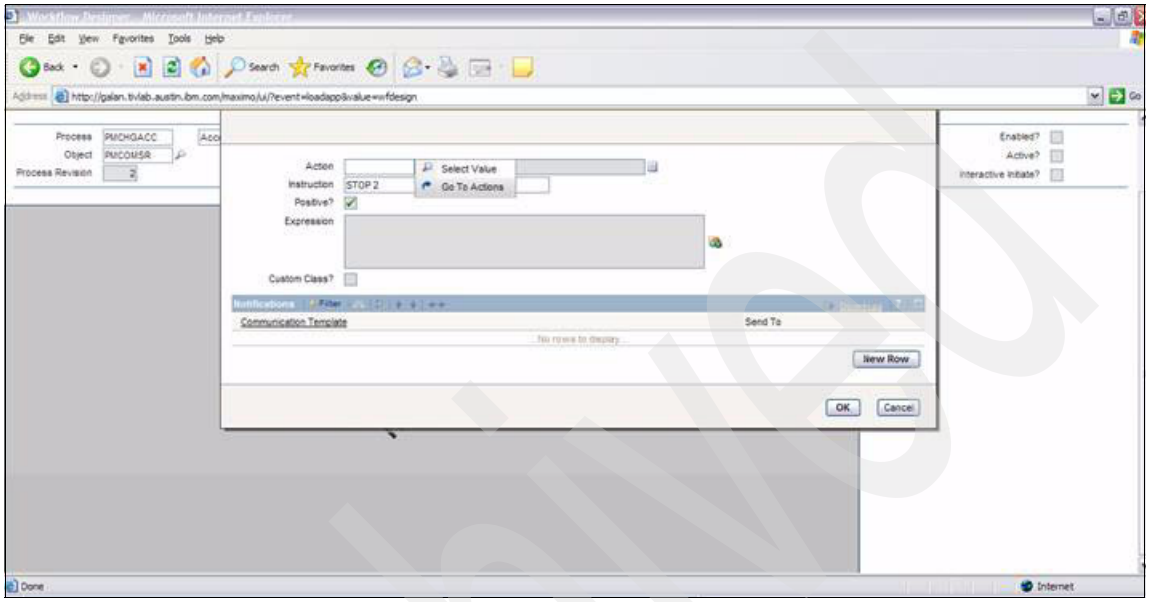


Figure 9-29 Setting the Job Plan to the standard Change Job Plan

Launch the Actions application, and create a new action that sets the Job Plan of the workorder as related to PMCOMSR. The action should look like Figure 9-30, where Value is the JPNUM of the Job Plan that you wish to automatically set when the Change is of the standard Classification.

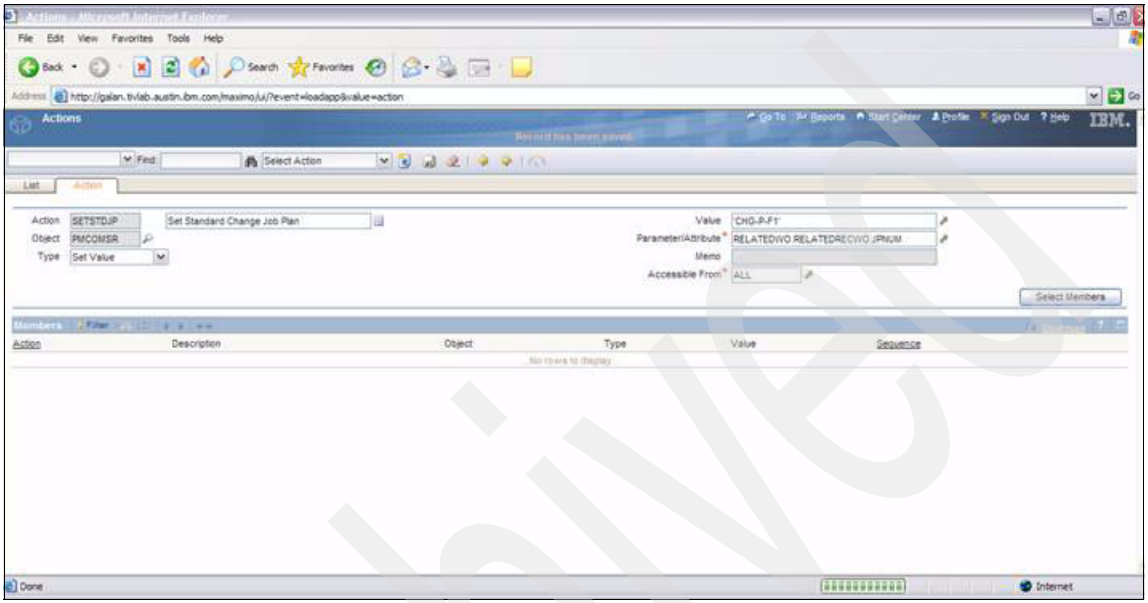


Figure 9-30 Values for new action

Click **OK**, and then **Save** the modified Workflow. From the Actions menu, choose **Enable Process** and then from the Actions menu choose **Activate Process**. Both the Enabled and Active check boxes should now be checked.

Go back to the ISMACCEPT workflow and enable and activate the process.

Now create a new PMCOMSR of the desired classification and accept it to test that the Change is created with the correct Job Plan.

If your modified workflow is not working, enable logging of the workflow engine in the Logging application by checking the **Active** box next to the workflow logger and setting it to the DEBUG log level. Then select **Apply Settings** from the Actions menu. The workflow engine logs will be written to the SystemOut.log of your WebSphere server.

Figure 9-31 on page 399 shows the process of enabling logging.

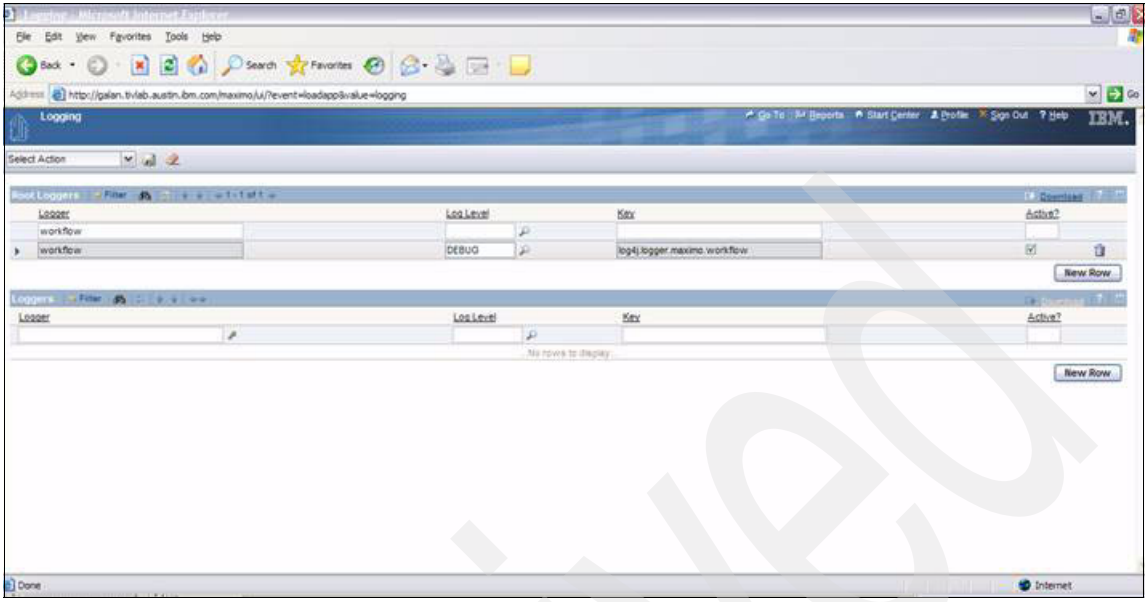


Figure 9-31 Enabling logging

Depending on the size of the Job Plan, the application of the Job Plan to a Change can be slow, so for large Job Plans you may consider applying the Job Plan to the Change using an escalation rather than through the accept workflow.

9.2.13 Customize your security rules using the dynamic UI

You may choose to limit the access your users have to each object based on their role or the current state of the object. For example, you may determine that a Change that has already been approved can no longer allow additional targets to be assigned to it.

Go to the Conditional Expression® Manager (select **Administration** → **Conditional Expression Manager**) and define a new condition called ISNOTAPPR (Figure 9-32).

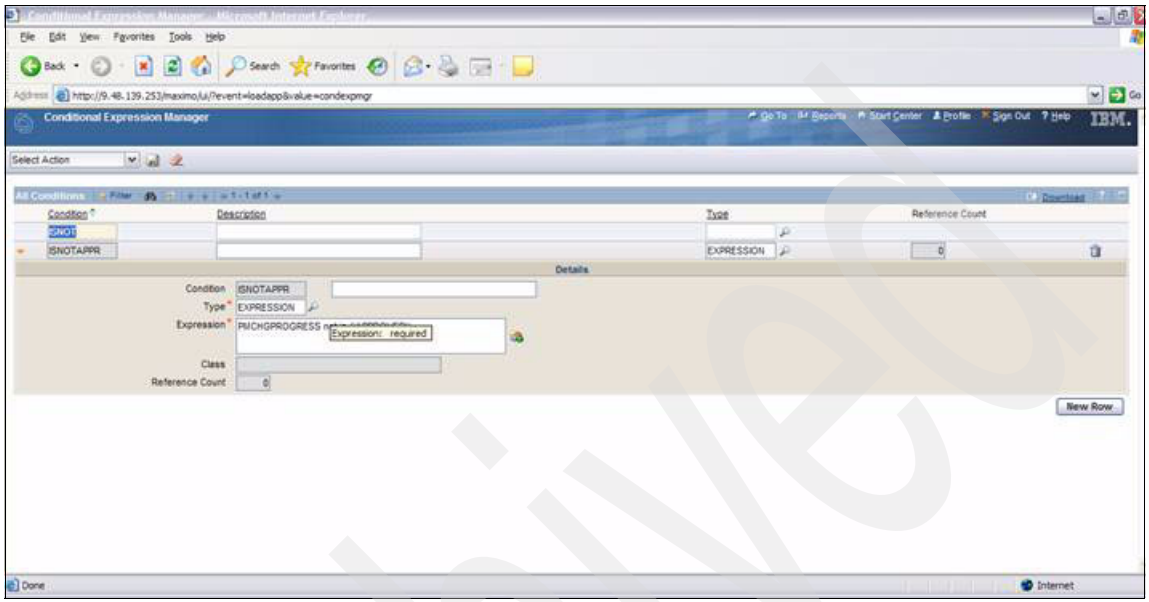


Figure 9-32 Defining a new condition ISNOTAPPR

Now to use this conditional security rule, go to the Application Designer for the Changes application. You will need to create a new Signature Option to control the Select Targets action. From the Actions menu, choose **Add/Modify Signature Options**, click **New Row**, create a new Option that looks like the one shown in Figure 9-33, and click **OK**.

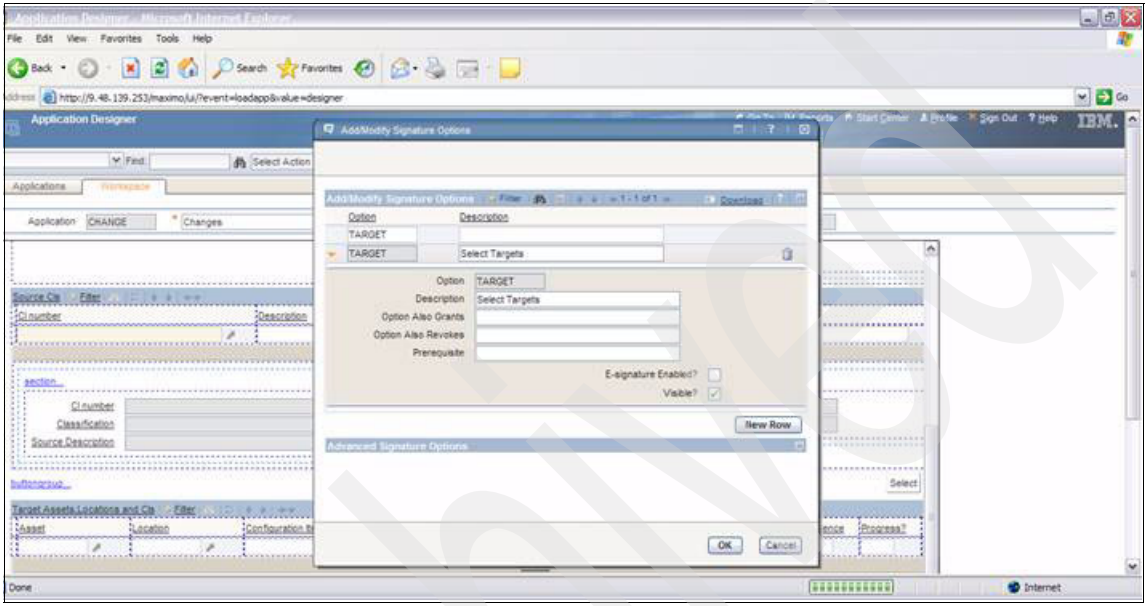


Figure 9-33 Adding new Signature option

Now choose the **Select** button next to the Targets section, which will display the Properties dialog for the Select button. Set the Signature Option to the new TARGET signature option that you created, set the Sig Option Data Source ID to MAINRECORD, and click the **Save** button.

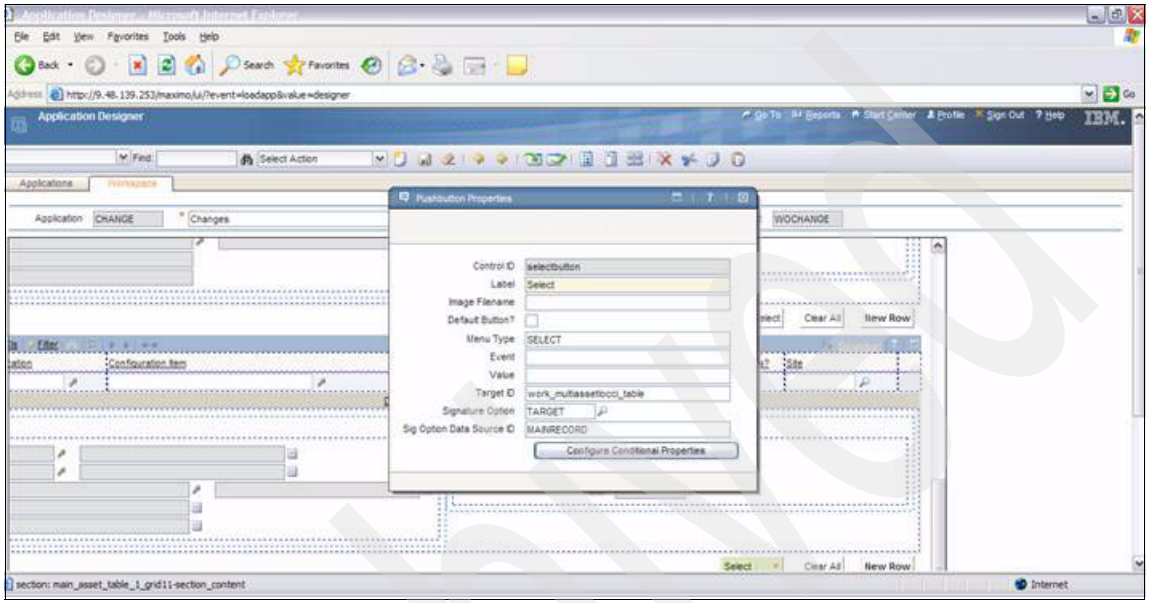


Figure 9-34 Setting options

Now go to the Security Groups application (select **Security** → **Security Groups**) and search for the maxadmin group. Under the Applications tab, find the Changes application and under the Options section, find the Select Targets option. Check the **Grant Access** check box to allow access to this button, type ISNOTAPPR in the condition field, and click the **Save** button (Figure 9-35 on page 403).

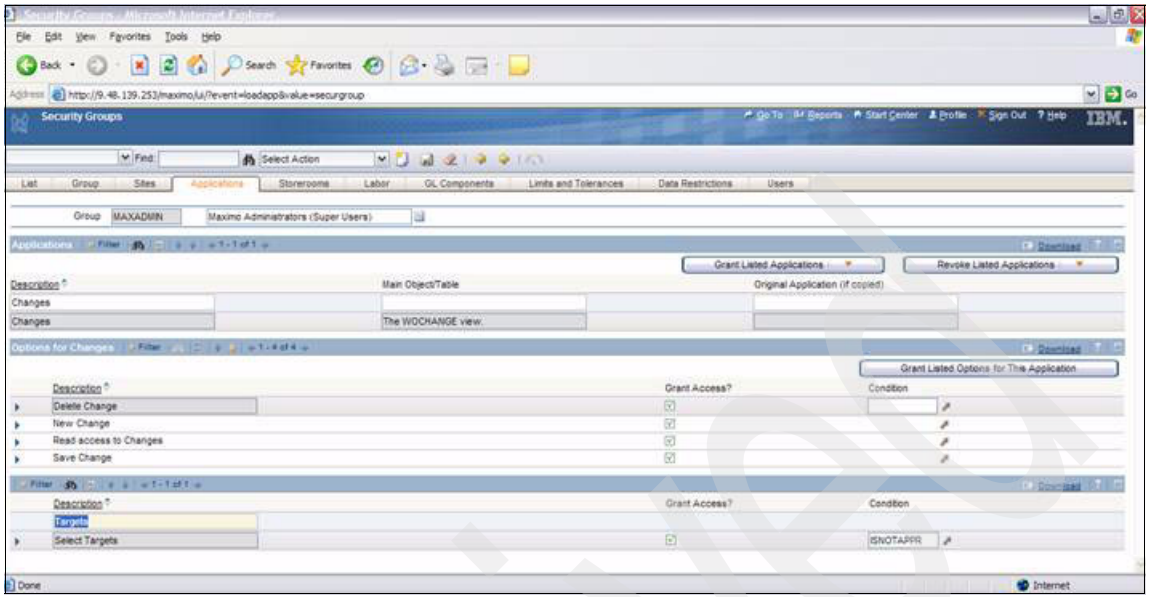


Figure 9-35 Granting access to maxadmin group

This will set the Changes application so that the Select button used to assign new Targets will only be visible if the Change has not already been approved for the maxadmin group.

Log in as maxadmin, create a Change, and select **Change Progress** from the action menu to set the Change to APPROVED. Notice that the buttons to allow selection of targets have gone away (Figure 9-36).

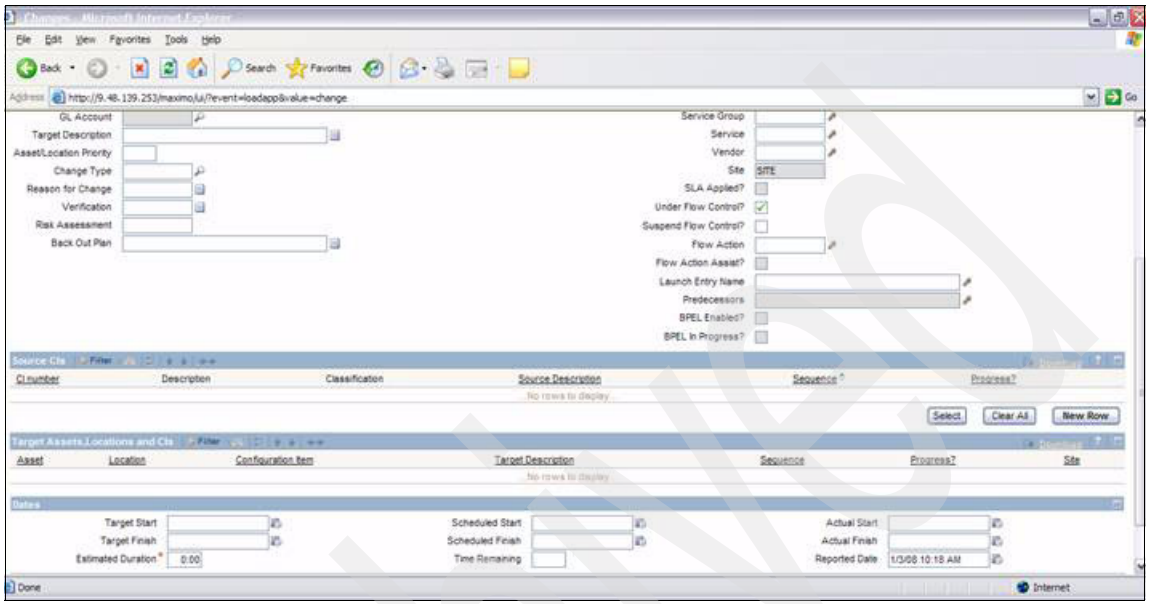


Figure 9-36 Modified UI

9.3 Summary

This chapter has shown simple examples of how the Change Management process Manager can be easily customized to meet your own data and process requirements.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 405. Note that some of the documents referenced here may be available in softcopy only.

- *IBM Tivoli Application Dependency Discovery Manager Capabilities and Best Practices*, SG24-7519
- *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565

Online resources

These Web sites are also relevant as further information sources:

- Tivoli Product Documentation Information Center

  http://publib.boulder.ibm.com/tividd/td/tdprodlist.html

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

D

E

F

N

O

P

R

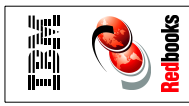
S

T

U

V

W



IBM Tivoli CCMDB Implementation Recommendations

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

IBM Tivoli CCMDB Implementation Recommendations

Gather requirements and document processes

Extend the CCMDB data model

Customize processes for your needs

The IBM Tivoli Change and Configuration Management Database (CCMDB) is one of the key components of the IBM Service Management (ISM) strategy. It is the foundation for automating and supporting change and Configuration Management processes as described by the Information Technology Infrastructure Library (ITIL).

This IBM Redbooks publication provides information valuable to those who want to plan for, customize, and use the IBM Tivoli CCMDB product to automate and manage change and configuration processes in their environments. It includes three parts:

- Understanding and documenting requirements
- Using and customizing the CCMDB data model
- CCMDB Process Engine and PMPs

A companion book, *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565, provides a more general overview of the CCMDB product and information related to planning and installation of the 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

SG24-7567-00 ISBN 0738485276