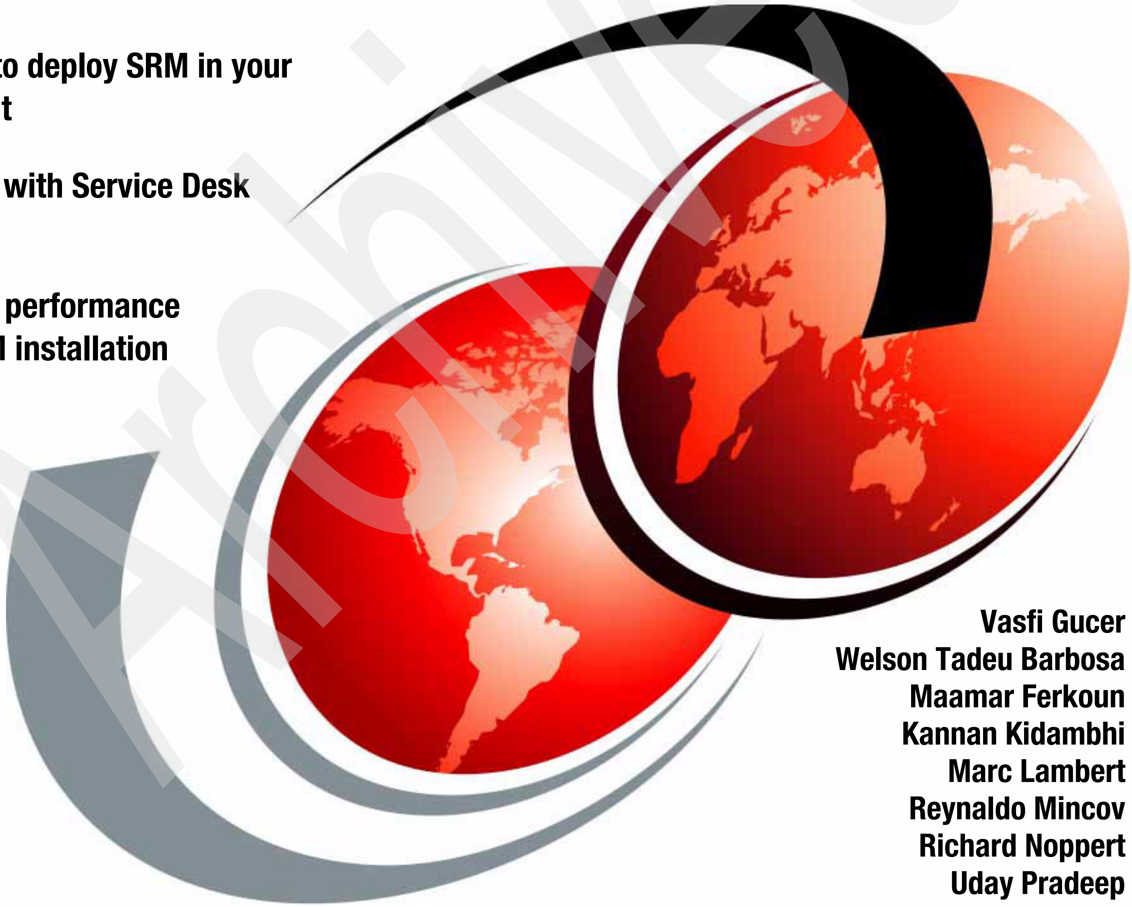
Tivoli software

Implementing IBM Tivoli Service Request Manager V7.1 Service Desk

Learn how to deploy SRM in your environment

Experiment with Service Desk functions

Improve the performance of your SRM installation



Vasfi Gucer
Welson Tadeu Barbosa
Maamar Ferkoun
Kannan Kidambhi
Marc Lambert
Reynaldo Mincov
Richard Noppert
Uday Pradeep

Redbooks

ibm.com/redbooks

International Technical Support Organization

Implementing IBM Tivoli Service Request Manager V7.1 Service Desk

August 2008

SG24-7579-00

Note: Before using this information and the product it supports, read the information in "Notices" on page xix.

First Edition (August 2008)

This edition applies to IBM Tivoli Service Request Manager Version 7.1.

Contents

Figures

Tables

Examples

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:

AIX 5L™
AIX®
DB2 Universal Database™
DB2®
Enterprise Asset Management®
IBM®
LanguageWare®
Lotus®
Maximo®
Netcool®
Rational Unified Process®
Rational®
Redbooks®
Redbooks (logo) ®
Sametime®
Tivoli Enterprise Console®
Tivoli®
WebSphere®

The following terms are trademarks of other companies:

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

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.

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

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

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

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

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

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

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

Preface

IBM® Tivoli® Service Request Manager V7.1 provides a unified and integrated approach in dealing with all aspects of service requests to enable a "one-touch" IT service experience, backed up by an optimized delivery and support process. It is a powerful solution that closely aligns IT operations and the business, improving IT service support and delivery performance.

This book provides information that can be used by clients, Business Partners, or IBM field personnel who are looking to engage in an effort to implement ITIL® based Service Desk processes in an enterprise environment utilizing the IBM Tivoli Service Request Manager V7.1 product. This book is divided into three parts:

- Concepts and components: Provides an overview of the IBM Tivoli Service Request Manager product Service Desk functions and some of the standards that drive its development. Also provides the reader with a better understanding of the various components, logical and physical, that make up the product and the functions that they provide.
- Planning and Installation: Provides information related to the actual installation of the IBM Tivoli Service Request Manager product components, including information related to hardware and software requirements.
- Demonstration scenarios and best practices: Focuses on demonstration scenarios using some of the new features of Tivoli Service Request Manager, such as reporting, survey, and search functions. We also provide best practices for fine tuning and high availability of Tivoli Service Request Manager components.

This book will be a reference for IT Specialists who will be implementing IBM Tivoli Service Request Manager Service Desk processes in client environments.

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.

Vasfi Gucer is an IBM Certified Consultant IT Specialist at the ITSO Austin Center. He started working for ITSO in January 1999, and has been writing IBM Redbooks® since. He has more than 15 years of experience in teaching and implementing systems management, networking hardware, and distributed platform software. He has worked on various Tivoli customer projects as a Systems Architect and Consultant. Vasfi is also a Certified Tivoli Consultant.

Welson Tadeu Barbosa is a Certified Sr IT Specialist and IBM IT Specialist board member, Pre-Sales Specialist in IBM Brazil. He holds a degree in Data Processing, with a post graduate degree in Business Administration. He has 12 years of experience in IT, nine years of which have been with Tivoli. He is responsible for technical pre-sales in the financial sector in Brazil, which includes banks and insurance companies. His experience includes Tivoli Performance and Availability products along with ISM family products.

Maamar Ferkoun is a Senior Product Professional with the IBM world wide Software Advanced Technology group. He is based in IBM China (Hong Kong S.A.R) and has over 20 years experience in the IT industry, over 10 years of which were with IBM. He holds a degree in Computer Science, an EXIN ITIL Manager certification, and a COBIT certification. Maamar began his career in IBM as a software field engineer engaged across the Asia Pacific region. His area of expertise covers the service management product portfolio and best practices.

Kannan Kidambhi is Computer Application Graduate from Madras University India. He is presently working in IBM Tivoli Software labs in India as a Tivoli Consultant. He has over 10 years of experience in Support, Administration, Consultation, and Implementation in the areas of IT Infrastructure Management and IT Service Management. Kannan has certifications in the following areas: ITIL Foundation, Sun™ Certified System Administrator, Cisco Certified Network Administrator, Microsoft® Certified System Engineer, and IBM Certified Tivoli Monitoring 6.1 Deployment Professional.

Marc Lambert has been employed with IBM for 13 years. For five of those years, he has been working with most of the System Management tools within the Tivoli portfolio. These five years were followed by time as a Senior IT Specialist, implementing different Service Management tools, such as HP-Peregrine, BMC Remedy, and Magic. In the last two years, he mostly has been working as an IT Architect for this particular field of business in Service Management.

Reynaldo Mincov is an IT Specialist at IBM Brazil, São Paulo. He joined IBM in 1999, where he has been working with IT Service Management tools and ITIL. He has implemented HP-Peregrine for many customers and is currently engaged in several Tivoli Service Request Manager (Maximo®) projects, including Service Provider.

Richard Noppert is a Solution Architect at MACS BV in the Netherlands. He holds a degree in Computer Science and has 14 years of experience in IT, focusing on system design, project implementation, and System Management. He is a Certified Tivoli Consultant with expertise in Service Management, ITIL processes, and Project Management. He is currently engaged in several Tivoli Service Desk implementation projects in Europe.

Uday Pradeep is a Solutions Consultant - Asset and Service Management with Birlasoft Inc in the USA. His skills include Tivoli Asset Management IT, Tivoli Service Request Management Solutions, Tivoli Maximo Enterprise Asset Management®. He is also ITIL Foundation Certified. He is an engineer in Computer Science and has expertise in visualizing solutions around the Tivoli suite of products. His current area of focus is implementing innovative integrations around Tivoli Access Manager, and Tivoli Storage Manager solutions for multiple clients by establishing best practices benchmarks for rollouts.

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

Arzu Gucer and Wade Wallace
International Technical Support Organization, Austin Center

Vijay Aggarwal, Pandian Athirajan, Russ Babbitt, Dave Calvert, John Christena, Pam Denny, Allen Gilbert, Venkatesan Govindaraju, Kyle Harding, Jennifer Lu, Tara Marshburn, Christopher Pardo, CJ Paul, Ramachandran Puthukode, Venkat Ram, Ramdas M Ramakrishnan, Sampath Sriramadhesikan, Lonnie Stone, Michael Weisskopf
IBM USA

Neil Pearson
IBM UK

The team would like to express special thanks to Praveen Hirsave from the SRM Development Team for his contributions to this document.

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Part 1

Concepts and components

In this section, we provide an overview of the IBM Tivoli Service Request Manager product Service Desk functions and some of the standards that drive its development. We also discuss the IBM Tivoli Service Request Manager architecture.

1

Introduction to Tivoli Service Request Manager

This chapter provides relevant information about how Tivoli Service Request Manager can help your organization to achieve business goals based on an IBM platform. In this chapter, we discuss:

- “IBM Service Management” on page 4
- “Establishing ITIL processes to align IT with business objectives” on page 7
- “Service desk functionality” on page 14
- “Business needs addressed by Tivoli Service Request Manager” on page 16

1.1 IBM Service Management

IBM Service Management is a new way to align your organization and all its related functions with your business. IBM Service Management encompasses the management processes, tactics, and best practices needed to deliver business services. IBM Service Management is about developing, deploying, and managing services that help reduce IT and operations costs through automated process and more effectively manage compliance. It is about increasing flexibility and getting products, solutions, and services to market more quickly, helping you to respond to changes more efficiently and effectively than ever before.

IBM Service Management lets you pull critical components, such as *people, processes, information*, and *technology* together with an array of tightly integrated solutions that can be viewed as three interconnected layers:

- IBM Process Management
- IBM Services Management platform
- IBM Operational Management

These solutions are based on IBM and industry best practices, such as the IT infrastructure Library (ITIL), Control Objectives for Information and related technology (COBIT), and enhanced Telecom Operations Map (eTOM), helping users to ensure IT and operational processes are consistently designed, automated, and executed, and are auditable for compliance adherence.

IT Service Management is the *optimal* intersection of people, processes, information, and technology, as shown in Figure 1-1 on page 5.

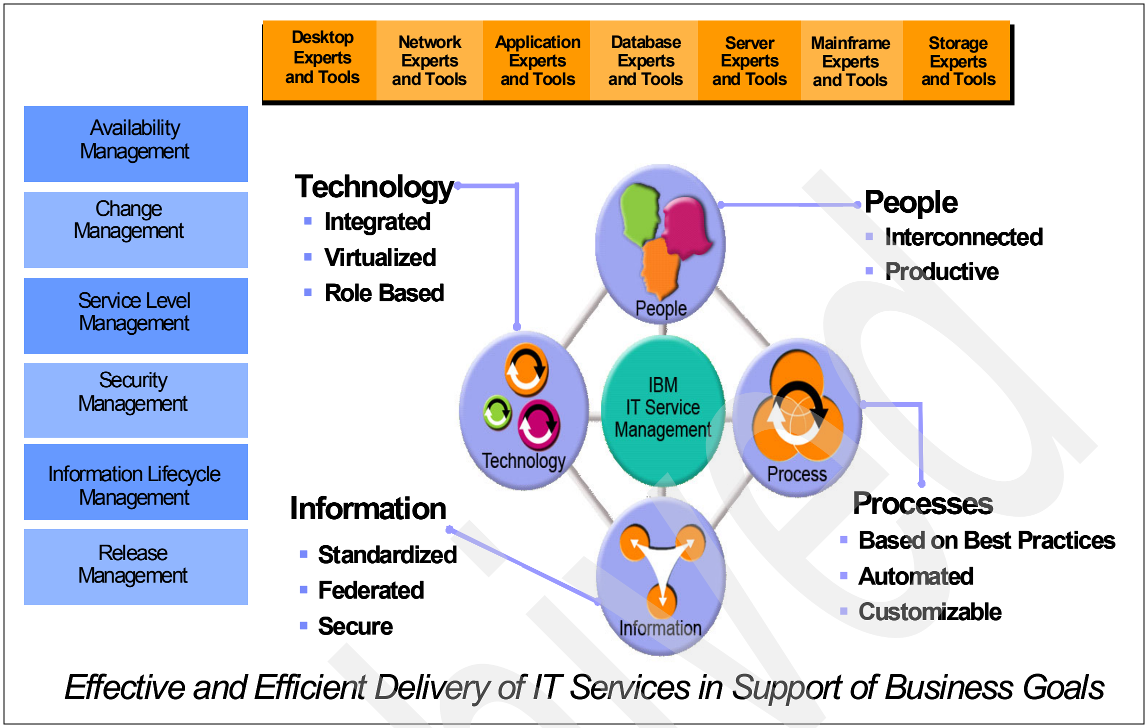


Figure 1-1 IBM Services Management interactions

1.1.1 Enabling business and technology integration

IBM Service Management provides integrated visibility, control, and automation across business and technology assets to help you overcome the roadblocks to business and technology integration and the inhibitors to innovation. This enables you to better achieve your business objectives and maximize the value from all of the assets that support your operation.

Figure 1-2 shows an overview of IBM Service Management.

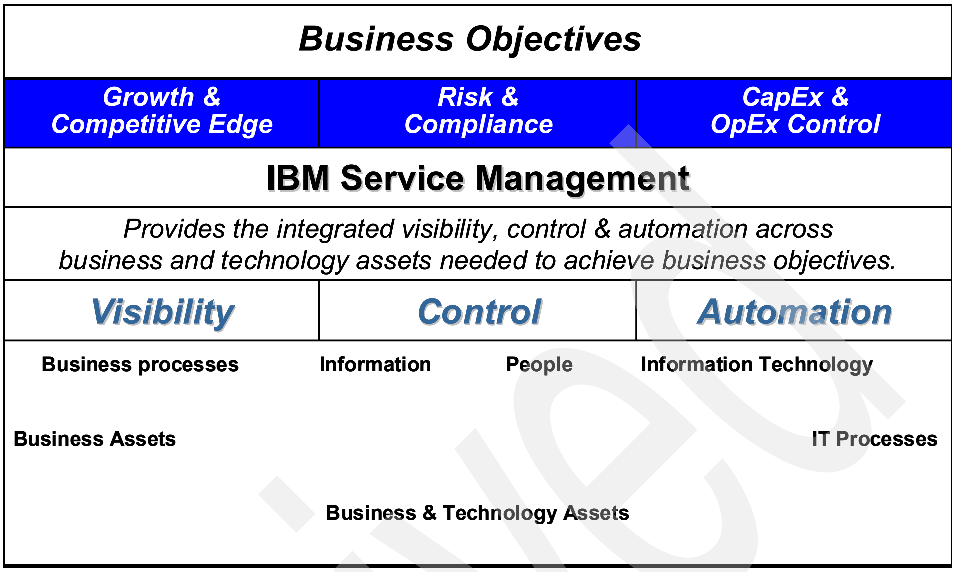


Figure 1-2 IBM Service Management overview

Innovation driven enterprises are facing an ever-growing problem to manage their IT environments in a way that becomes:

- Efficient
- Reliable
- Secure
- Consistent

At the same time, the necessary infrastructure to deliver IT-enable business services has been increasing. In most cases, IT has become so complex that is impossible to determine the dependencies between services or their business relations. This has presented a set of challenges to the organizations:

- Complexity: The root cause for IT organizations problems lies in the ever-growing complexity, sometimes caused by heterogeneity environments, interconnected applications (as known as composite applications), and so on. Architectural and organizational issues, along with worldwide spanning operations, have contributed to reducing the efficiency and effectiveness of the IT organization.

- Change: Complexity makes infrastructure vulnerable to changes. Often, a forgotten detail breaks the service. Among the top reasons for outages or service disruption are service level assurance requirements, staff turnover, and market opportunities. Statistically, almost 80 percent of service outages are caused by a lack of Change Management.
- Cost: Currently, IT labor cost represents almost 70 percent of the total IT budget. In the late 1990s, half of IT budgets were assigned to new applications development and half to operations. As IT budgets have been kept flat, or in some cases decreased, IT executives are facing a dilemma: Shift resources from new applications development or reduce the level of support for current applications. In both cases, IT efficiency and effectiveness are affected.
- Governance and compliance: New regulations, such as the Sarbanes-Oxley Act (SOX) and the Health Insurance Portability and Accountability Act (HIPAA), have put an additional burden on the IT organization to support the needs of the business to audit for compliance through the institution of better process controls and the maintenance of audit trails for IT infrastructure changes. This requires careful consideration because of the penalties of noncompliance, including criminal and civil liabilities and adverse public opinion.

1.2 Establishing ITIL processes to align IT with business objectives

Information Technology Infrastructure Library (ITIL) provides a non-proprietary framework for implementing service management best practices aligned with overall business objectives. Creating IT processes for ITIL guidelines enables organizations to more effectively manage IT changes, assets, personnel, and service levels.

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

ITIL best practices are vendor independent, and thus can be used regardless of the size of the company. ITIL books are not intended to be the solution themselves; the books are guidelines that, once followed, help an organization to optimize their IT processes and day-to-day operations.

1.2.1 ITIL Version 3

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

Service Strategy

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

Service Strategy encompasses the following processes:

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

Service Design

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

- Service Portfolio Management
- Service Catalog Management
- Service Level Management
- Capacity Management
- Availability Management
- Service Continuity Management
- Information Security Management
- Supplier and Contract Management

Service Transition

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

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

Service Operation

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

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

Continual Service Improvement

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

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

1.2.2 Successfully implementing ITIL

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

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

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

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

A successful implementation of IT Service Management should consider:

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

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

1.2.3 IBM and ITIL

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

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

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

IBM Tivoli Service Request Manager is aligned with ITIL best practices to support ITSM processes. Built with support for Incident and Problem Management, Change and Release Management, and Service Level Management, it is a part of a single platform that combines asset and services management. IBM implements ITIL through IBM Tivoli Unified Process (ITUP).

1.2.4 The IBM Tivoli Unified Process (ITUP)

IBM Tivoli Unified Process (ITUP) is a roadmap for delivering ITIL based IT Service Management using existing Tivoli and IBM solutions. ITUP links actual product names and capabilities with ITIL defined roles, responsibilities, and processes, and complements the IBM Rational® Unified Process® (a logical method of application development) to provide a mechanism and a philosophy for customers to align their IT organization and processes along business service guidelines, and develop and implement applications with additional manageability factors built in to take full advantage of these new concepts. Each ITUP process is defined by:

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

For more information, go to:

http://www.ibm.com/software/tivoli/features/it-serv-mgmt/itup/index.html

This site provides a brief overview of components, features, and how they interact with each other in the overall IBM IT Service Management strategy. ITUP is considered by IBM to be a definitive guide on how to implement ITIL in a modular yet comprehensive fashion using IBM solutions available today.

1.2.5 ITUP Composer

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

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

Table 1-1 summarizes the differences between ITUP and ITUP Composer:

Table 1-1 Differences between ITUP and ITUP Composer

Feature	ITUP	ITUP Composer
Industry best practices	Y	Y
Process-level information	Y	Y
Activity-level information	Y	Y

Feature	ITUP	ITUP Composer
Tool use guidance	Y	Y
Task-level information	Y	Y
Content customization		Y
Content creation		Y
Content publishing		Y

The ITUP Composer contains the following resources.

Process library

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

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

Tool mentors

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

Roles

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

Work products

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

Scenarios

Scenarios describe common problems and best practice solutions. With these scenarios, you are able to see how you can address real-world problems by improving and integrating processes, using the appropriate tool properly, and setting up the necessary roles and responsibilities. Underlying the ITUP processes are the layers of supporting autonomic technologies. Autonomic computing provides technologies and standards to support and enable the process environments of the IBM Tivoli Unified Process and IBM IT Service Management offerings.

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

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

1.3 Service desk functionality

According to ITIL organizations, you invest in a service desk to:

- Provide a single point of contact for users
- Deliver the high quality support critical for achieving business goal
- Help identify and lower cost of ownership for IT services as a whole
- Support changes across business, technology, and process boundaries
- Aid user retention and satisfaction
- Assist identification of business opportunities

Figure 1-3 on page 15 shows an overview of the service desk interactions.

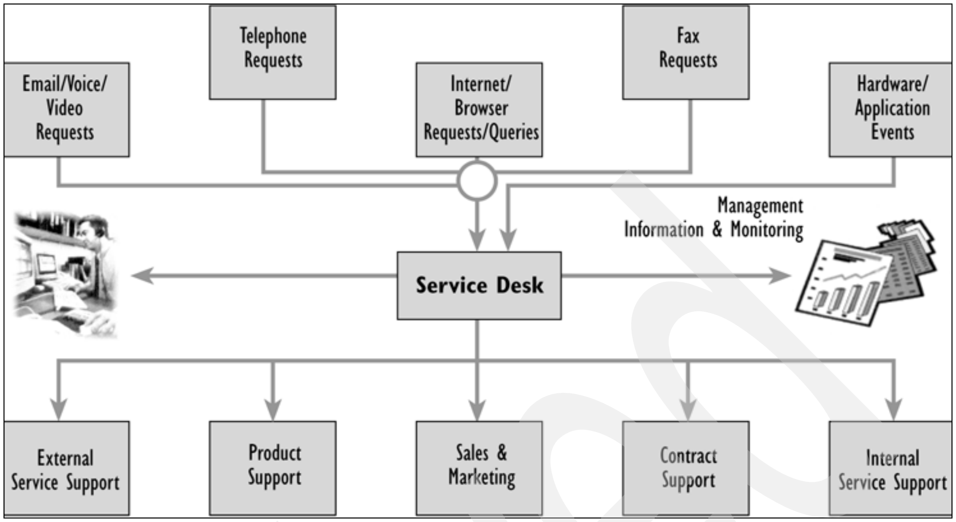


Figure 1-3 Service desk interactions[1]

Today's business environment changes as fast as the market demands. To be responsive, it is important to have a scalable infrastructure aligned with very well defined business processes, as shown in Figure 1-4 on page 16. Tivoli Service Request Manager helps you implement your Service support and Service Delivery process.

[1] Source: Reprinted, by permission of the publisher, from *ITIL Service Support*, The Stationery Office, 2002

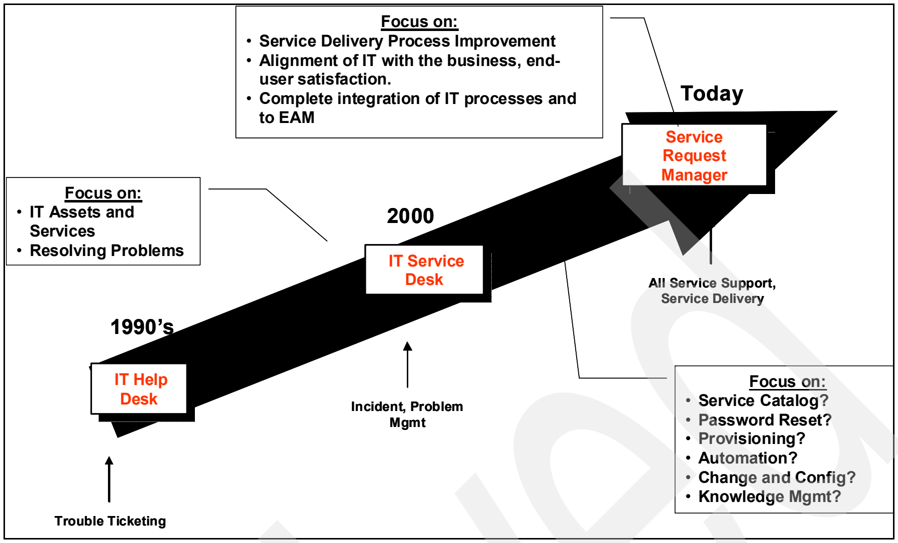


Figure 1-4 Service Desk market evolution

1.4 Business needs addressed by Tivoli Service Request Manager

Quality service support is critical for:

- Competitive advantage
- Customer retention and satisfaction
- Optimal service
- System performance
- Aligning IT with the business
- Organization efficiency
- Cost reduction, especially labor
- Service level agreements

Tivoli Service Request Manager is a solution that closely aligns IT operations and the business, improving IT service support and delivery performance. In other words, it is used to manage any type of critical enterprise asset, and lowers cost of ownership. It is an integrated platform that improves the efficiency of service delivery and offers opportunities to drive down operating costs by consolidating your service desk and systems.

Built from the ground up on the Information Technology Infrastructure Library (ITIL) framework, Tivoli Service Request Manager provides a comprehensive and modular approach to integrated Service Desk and Service Catalog management. In addition, this product enables you to establish and efficiently operate a corporate service desk for service requests around enterprise assets. Tivoli Service Request Manager offerings include tools, best practices, and service offerings for incremental value.

The two important areas for Service Management are Incident and Problem management.

- Incident management

  Incident management is the process of restoring normal service operation as quickly as possible to help minimize an incident's adverse impact on business operations. In ITIL terms, an incident is any deviation from the expected standard operation of a system or service. Best practice incident management involves immediate service restoration utilizing standard processes of investigation, diagnosis, resolution, and recovery.

  Tivoli Service Request Manager documents incidents from users, service technicians, and network systems management applications. Streamlining the process further, it leverages ticket types and classifications with powerful visual workflow escalation and e-mail notifications for quicker resolution, helping to meet customer expectations and improve service desk efficiency. Consolidation of user communication across channels, including phone, e-mail, Web, and fax, captures each incident, creating a searchable knowledge base that can vastly reduce staff response time to anomalies or outages if similar incidents reoccur. Incidents can be linked with appropriate problems or changes, and are associated with their related CIs in the CMDB.

- Problem management

  A problem is the underlying error in the infrastructure that is the cause of one or more incidents. Problem management is the process of diagnosing the root cause of the error and arranging for a correction. Furthermore, it seeks to prevent a recurrence of incidents related to these errors. Effective problem management depends on IT's ability to quickly and accurately determine the root cause and turn an unknown error into a known error, that is, problems for which the root cause is determined and attributed to a specific CI.

With Tivoli Service Request Manager, IT operations can more readily identify and classify the root cause of problems, assisting staff to quickly recognize and resolve known errors with minimal downtime. Built-in, real-time dashboards provide insight into all levels of service desk operations, so that any support staff, manager, or executive can monitor role-based key performance indicators (KPIs) in an intuitive, graphical display. Dashboards provide actionable information and can identify potential problem areas, enabling IT to take appropriate corrective actions in most cases before critical services are adversely affected. Tivoli Service Request Manager enables the creation of changes from identified problems and ties appropriate incidents to these problems.

1.4.1 What Tivoli Service Request Manager can do for you

The following are some of the benefits of using Tivoli Service Request Manager:

- It provides full service desk incident and problem management capabilities with service support and asset management processes, and service request for IT and non-IT.
- It increases service desk efficiency and effectiveness, reduces disruptions, streamlines service desk operations, improves customer satisfaction, and reduces costs.
- Incident and Problem Management supports the prioritization of fixing the user's problem through root cause analysis to increase service levels and employee productivity.
- User self service provides an easy-to-access, browser-based user interface to the service desk, enabling users to submit tickets, view updates, and search solutions.
- E-mail listener efficiently processes inbound e-mails into service requests.
- Ticket templates save time by pre-populating work order fields with information found in the service request.
- Searchable solutions provide quick access to solutions for specific service requests and help build the internal service desk knowledge base.
- Built-in Configuration Tools allow users to quickly and easily make changes to their workspace, user interface, workflows, reports, and more.
- Based on leading standards-based technology: A Web-architected platform built on J2EE™ with advanced business process management, based on SOA, Web services, and XML.
- Real-time integration with operational management products.
- Growth to true Service Catalog funcionality.

- Transparent migration to future releases.

Tivoli Service Request Manager is focused on service requests, not just problems, and those services requests might not be just IT requests; they could be for any assets or facilities that might have a service request involved.

Another key fact is that it is designed as a standards based Web application, not an aging three tier distributed architecture. That allows for great scalability and flexibility, and it is also what allows customers to easily set up self-service requests; users can check online for status for incidents problems or service requests without contacting the service desk directly.

2

IBM Tivoli Service Request Manager architecture

In Chapter 1, “Introduction to Tivoli Service Request Manager” on page 3, we discuss the business context for a Service Desk solution and what Tivoli Service Request Manager can do for you in that respect. We discuss the possible benefits organizations have using a Service Desk solution, as well as the need for the ITIL conformation and framework IBM provides. This chapter will discuss the technical aspects of the Tivoli Service Request Manager architecture and various components you can use for deployment.

This chapter will discuss the following topics:

2.1 Tivoli Service Request Manager overview

Built from the ground up on the Information Technology Infrastructure Library (ITIL) framework, Tivoli Service Request Manager provides a comprehensive and modular approach to integrated Service Desk and Service Catalog management. In addition, this product enables you to establish and efficiently operate a corporate service desk for service requests around enterprise assets.

Tivoli Service Request Manager runs on top of Tivoli's process automation engine, utilizing its services and J2EE technology with advanced business process management (based on SOA, Web services, and XML). This allows adaption, configuration, rapid deployment, robust scalability, and deep integrations with enterprise applications.

Tivoli Service Request Manager is introduced as a common package of Service Desk and Service Catalog capabilities, as shown in Figure 2-1. What Service Desk and Service Catalog have in common is the concept of satisfying user requests. In general, requests to the Service Desk are handled on a case by case basis, while Service Catalog requests are usually handled in pre-configured manner that is intended to fulfill the request in a best practice manner.

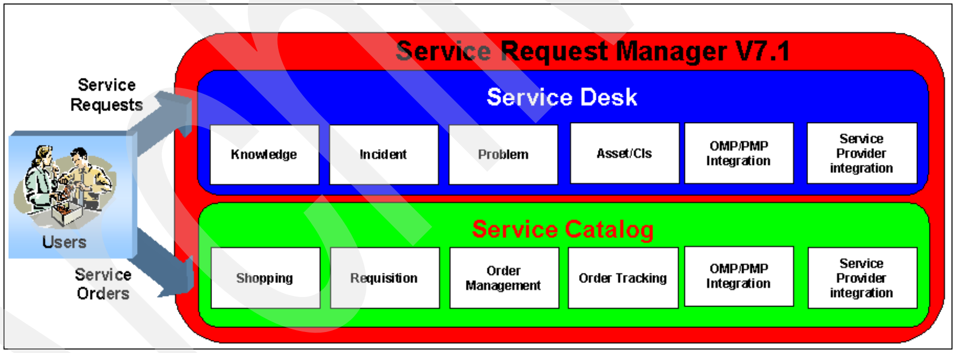


Figure 2-1 Service Request Manager

Tivoli Service Request Manager offers a unified solution for service request management, whether it is a performance problem that requires help from the service desk or a request for provisioning of new software into your workstation through Service Catalog. Users can access one application through the Web, make a phone call, or simply send an e-mail.

Tivoli Service Request Manager has built-in service desk functionality, such as incident and problem management, offers solutions to the users through knowledge management, and integrates with network/application management products to open incident tickets automatically. It also offers service delivery functionality through its catalog where users can “shop” for services they need through an online catalog. Customers need to pay for what they buy, enabling tighter control over expenses.

2.1.1 Service Request Manager V7.1 Service Desk

The Service Request Manager V7.1 Service Desk component is built upon a proven service desk platform, Maximo V6.2 Service Desk, which is already a highly functional and extremely configurable service desk product. But the Service Request Manager V7.1 Service Desk component provides many new functions over Maximo V6.2 Service Desk. At their core, these Service Desk components support generalized Service Requests and Incident and Problem Management processes. Also, the Solutions and Bulletin Board applications can provide valuable information to the user or Support Desk and operations personnel. All these processes can be significantly customized by the customer using tools such as Workflow, Activities/Ticket Templates, and the Application Designer. Further, the Maximo Enterprise Adapter tools allow you to create all sorts of integrations with external systems and tools. Finally, the Maximo framework drives significant integration between the service desk applications and all the other Maximo applications.

Figure 2-2 on page 23 provides an overview of the Maximo V6.2 Service Desk product. The Maximo V6.2 Service Desk is already a highly functional and extremely configurable service desk product.

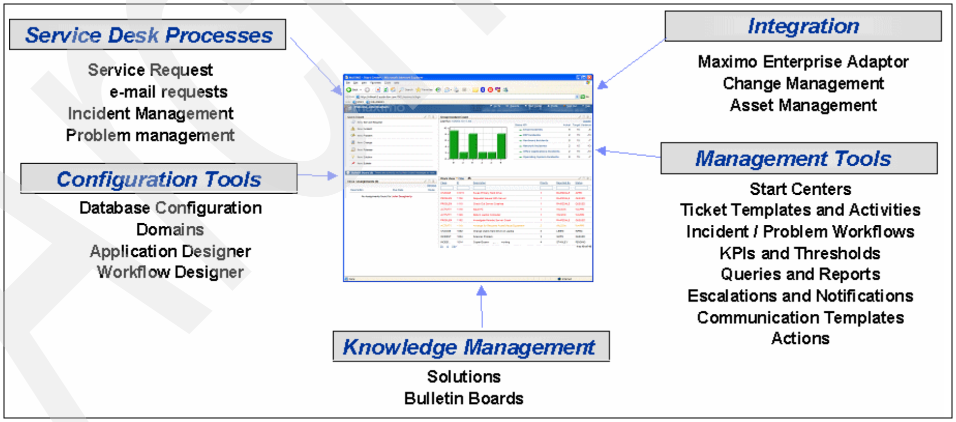


Figure 2-2 Service Desk - Basic capabilities

In Tivoli Service Request Manager V7.1, the service desk functionality (as well as the Framework) has significantly evolved, and includes a number of new or enhanced features.

Note: Note that in the V7.1 release, Maximo V6.2 Service Desk is now called Tivoli Service Request Manager V7.1 and Maximo Framework is called Tivoli's process automation engine.

Figure 2-3 on page 24 provides an overview of these new features or enhancements in Tivoli Service Request Manager V7.1 Service Desk.

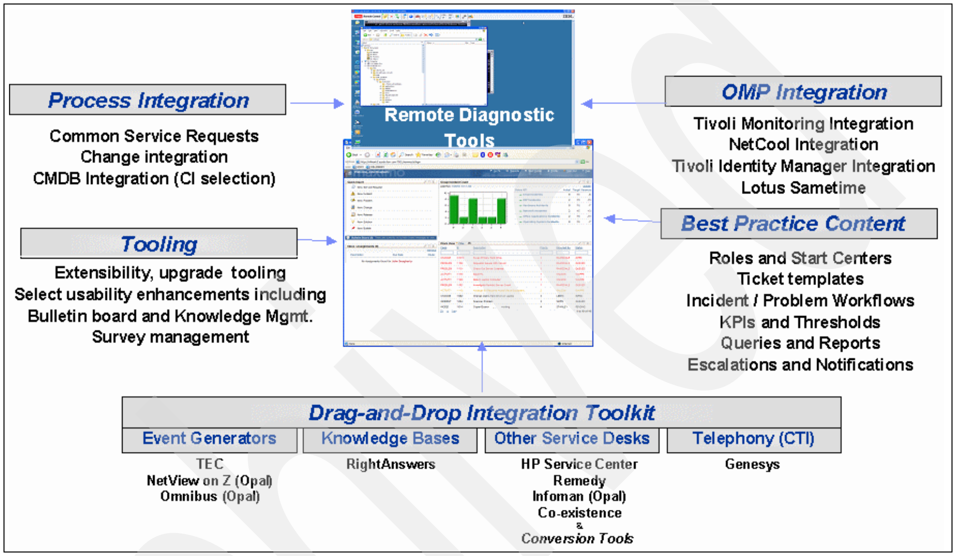


Figure 2-3 Service Desk - New features

An outline of the enhancements made to the Tivoli Service Request Manager V7.1 Service Desk are described in the following sections.

2.1.2 Process integration

The following are some of the process integration enhancements new with this release:

- CCMDB V7.1 integration
  - Actions available from the detail menu associated with CI field on a ticket:
    - Goto CI
    - View CI details
    - View related CIs
- Service Catalog integration

2.1.3 Tooling

The following are some of the enhancements in the tooling area:

- Bundled remote diagnostic capabilities
  - Provides the service desk agent with remote administration capabilities of user machines
  - Take control of the remote machine; execute commands and applications (active state), including remote reboot
  - Instant Messaging Capabilities
  - Maintain an audit record of the Remote Control session
- Password reset
  - Password reset capabilities in self-service portal
- Consumability improvements
  - Feedback from existing customers
    - Functional enhancements
    - Usability enhancements
- Survey and Global search capabilities

2.1.4 Operational Management Products (OMP) integration

The following are some of the enhancements in the OMP integration area:

- Integration with IBM Tivoli Enterprise Console/IBM Tivoli Netcool® Omnibus (refer to *Integration Guide for IBM Tivoli Service Request Manager V7.1*, SG24-7580 for more details)
  - Permits IBM Tivoli Enterprise Console/IBM Tivoli Netcool Omnibus to create incident tickets automatically based on events
  - Mutual status updates between IBM Tivoli Enterprise Console/IBM Tivoli Netcool Omnibus with Service Desk
- Integration with IBM Tivoli Identify Manager
  - To manage Tivoli Service Request Manager application users through Tivoli Identity Manager
- Integration with Lotus® Sametime® IBM
  - Allows the service desk analyst to begin a conversation session in real time with any person related to a ticket
  - Allows the person who requested the service request to talk to a service desk analyst in real time using an instant messenger application

- Integration with IBM Tivoli Monitoring
  - To view the incident related events in the Tivoli Enterprise Portal (TEP)

> **Note:** Note that current IBM Tivoli Monitoring integration is through IBM Tivoli Enterprise Console® (by viewing the TEC events in the Tivoli Enterprise Portal).

Refer to *Integration Guide for IBM Tivoli Service Request Manager V7.1*, SG24-7580 for more details about these integrations.

2.1.5 Best practice content

During installation, you can optionally install best practice content for the Service Desk. You can use this content out-of-the-box, or customize it to fit your specific business needs. The following is some of the best practice content new with this release:

- Communication templates
- KPIs and queries
- Workflows
- Security definitions/start centers for ITUP (IBM Tivoli Unified Process) defined roles
- Incident and Problem process templates
- New reports:
  - Total numbers of Requests
  - Requests Completed/closed and incomplete sorted by state, priority, impact or urgency
  - Mean elapsed time to achieve Request completion
    - Sorted by impact code
    - SLA Compliance Status (Percentage of Requests handled within agreed response time)
  - Average cost per Request

> **Note:** This best practice content will be shipped through Tivoli Open Process Automation Library (OPAL) and will save the customer time and effort in configuring their service desk. OPAL can be accessed at:
>
> http://catalog.lotus.com/wps/portal/topal

2.1.6 Drag-and-Drop Integration Toolkit

The integration toolkit is a portion of the content that is being delivered in support of the Tivoli Service Request Manager V7.1 product. It is an easy way to build data level integration with any application hosted in Tivoli's process automation engine by taking advantage of the capabilities of the Tivoli Directory Integrator (TDI). The toolkit extends the standard Maximo Enterprise Adaptor (MEA) architecture, using Maximo object structures on one end and TDI connectors on the external end.

The following are some of the enhancements related with the Drag-and-Drop Integration Toolkit:

- Computer Telephony Integration
  - CTI is a technology that allows interactions between the telephone system and the Service Desk product.
  - It pre-populates the agent's screen with the caller's information.
- Integration with IT and Enterprise asset management products
- Knowledge management extensions
  - Provides enhanced "Internet style" search.
  - Provides enhanced ranking of search results.
  - Supports importation of external content.
  - Provides a "Knowledge Provider" interface that facilitates integration with external knowledge sources and problem determination aids.
- Formalize knowledge management process
  - Develops a formal process for managing the life cycle of knowledge (solutions).
  - Provides a user ranking mechanism to identify problematic solutions.
- External service desk integration
  - Co-existence with Remedy/Peregrine.
  - Complete migration from Remedy/Peregrine.

2.1.7 Tivoli Service Request Manager V7.1 Service Catalog

Many organizations experience the same problems and are looking for answers to solve them. The main issues are often experienced from different perspectives:

- From a management perspective, Service Delivery cannot be managed effectively, because customers are not aware of which “services” are actually being provided, how frequently these services are being requested, how satisfactorily these services are being fulfilled, and unable to predict future demand for services, so planning for them is unproductive.
- From a user perspective, customers do not know what services they are entitled to and how to obtain them.
- Operations personnel do not have well defined, best practice, and fulfillment processes defined, so they fulfill the same requests in different ways. For IT services that affect critical hardware and software assets, there is a need to tightly integrate Service Delivery with established IT Management processes.

> **Note:** You can refer to *Implementing IBM Tivoli Service Request Manager Service Catalog*, SG24-7613 for more information about Tivoli Service Request Manager V7.1 Service Catalog.

The Service Catalog supports and facilitates the processes defined by customers by addressing the issues discussed.

Service Catalog has three types of supported services:

- Descriptive services

  These services permit a company to advertise services that are delivered in *ad hoc*, manual manners. It allows a customer to put a service in the Service Catalog in a very low cost manner.
- Action

  Action services provide a cheap way to integrate existing automated services through Launch in Context to external applications or by the execution of Tivoli Service Request Manager Actions (that can do things like trigger command scripts or custom Java™ classes that call external APIs).
- Supply Chain

  The Supply Chain service fulfillment model leverages the full power of the Tivoli Service Request Manager purchasing supply chain applications.

Note: Service Catalog Supply Chain refers to the chain of Service Catalog product components that accomplishes the full set of Service Delivery tasks. The chain starts with the ability of a Service Requester to search for and requisition a service from the Service Catalog. The chain ends with the complete fulfillment of the Service Requisition.

Purchase Orders can be generated for external suppliers. Work Order Management allows one to precisely specify and track the human and non-human resources that are needed to satisfy a request for service and also to specify the manual and automated tasks that are to be performed. Work Order Management ensures that all the tasks that must be accomplished in order to accomplish a service requisition are executed in the right order.

Figure 2-4 provides an overview of the functionality and how Service Catalog interacts with Service Desk Process Management applications.

Figure 2-4 Service Catalog functionality

Figure 2-5 on page 30 shows the major new configuration tools or Service Administration tools that come with the Service Catalog product.

1. The Service Fulfillment application allows one to define the "base" portions of a service. All sorts of default parameter values can be defined for a class of service. These can be overridden at the Offering level. Among other things, a Service Fulfillment definition can define the Job Plan that indicates how to render a request for a service.

2. The Catalogs application is used to manage the Service Offerings that are visible to the user.
3. The Offerings application defines the service offerings that are available. One powerful design feature is that a single Service Fulfillment definition can be expressed in multiple Offerings. For example, a Service Fulfillment definition can be created to provide configured mobile computers. The base service supports a variety of attributes that define the hardware and software characteristics desired in the provided mobile computer. Given this base flexible sort of service, multiple offerings can be defined that provide pre-configured mobile computers at different levels of capability and price.
4. The Fulfillment Options application permits one to customize how a particular class of service will be fulfilled. Among other things, it permits alternative Service Providers for a given class of service to indicate how they would fulfill the service. When a service is requested, if there are alternative qualified Service Providers, the IT Operations Analyst can supervise the selection of a particular provider.

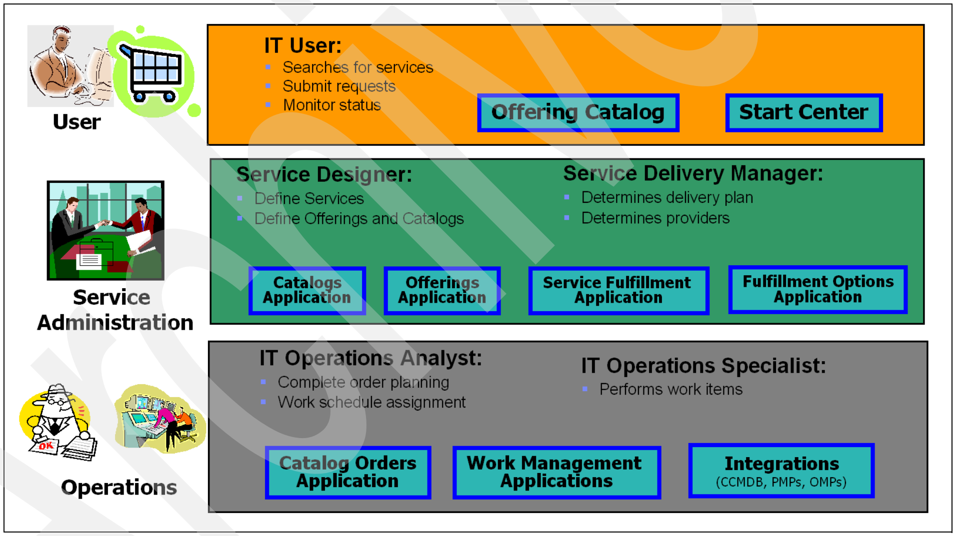


Figure 2-5 Service Catalog roles, activities, and tools

Service Catalog roles are based on the IBM Tivoli Unified Process (ITUP) roles.

- Administrative / Definitions roles
  - Service Designer (mainly uses Service Definition, Offering Definition, and Catalog Applications)
    - Manages the definition, inclusion and availability of services in the catalog.
  - Service Delivery Manager (mainly uses Capabilities application and Service Definition applications)
    - Co-author of services definition.
    - Builds and manages aggregate delivery plan, including cost and resources.
    - Maintains competence, capacity pools and profiles of the delivery teams.
  - Service Execution Manager (mainly uses Reporting and KPIs and capabilities applications)
    - Oversees the operation of the Service Catalog Supply Chain.
- Operational / Execution roles
  - User
    - User that browses the Service Catalog and submits service requests.
  - Business Line Manager
    - New role defined in Service Catalog.
    - Reviews and approves the submission, by the user, of service requests to the supply chain.
  - User Contact Analyst
    - Receives, reviews, and analyzes service request information from the user.
  - Operations Analyst (mainly uses the requisitioning management and Service Order and Work order management applications)
    - Executes all operational processes and procedures of Order Planning and Fulfillment Planning.
    - Analyzes, prepares, submits, and adjusts task planning, workload, work schedule, and work items.
  - Operation Specialist (uses ISM integration points for automated flows)
    - Perform work items

> **Tip:** More information about the IBM Tivoli Unified Process can be found at:
>
> `http://www-306.ibm.com/software/tivoli/governance/servicemanagement/itup/tool.html`

2.2 Tivoli Service Request Manager components

Similar to the Tivoli Change Configuration Management Database V7.1 (referred to as CCMDB), the two major layers of Tivoli Service Request Manager are the *Data Layer* and the *Process Layer*. Both layers are composed of multiple logical and physical components, which we will discuss in this chapter.

2.3 Logical components

Figure 2-6 on page 32 outlines a *logical component overview* for the Tivoli Service Request Manager solution and all related products.

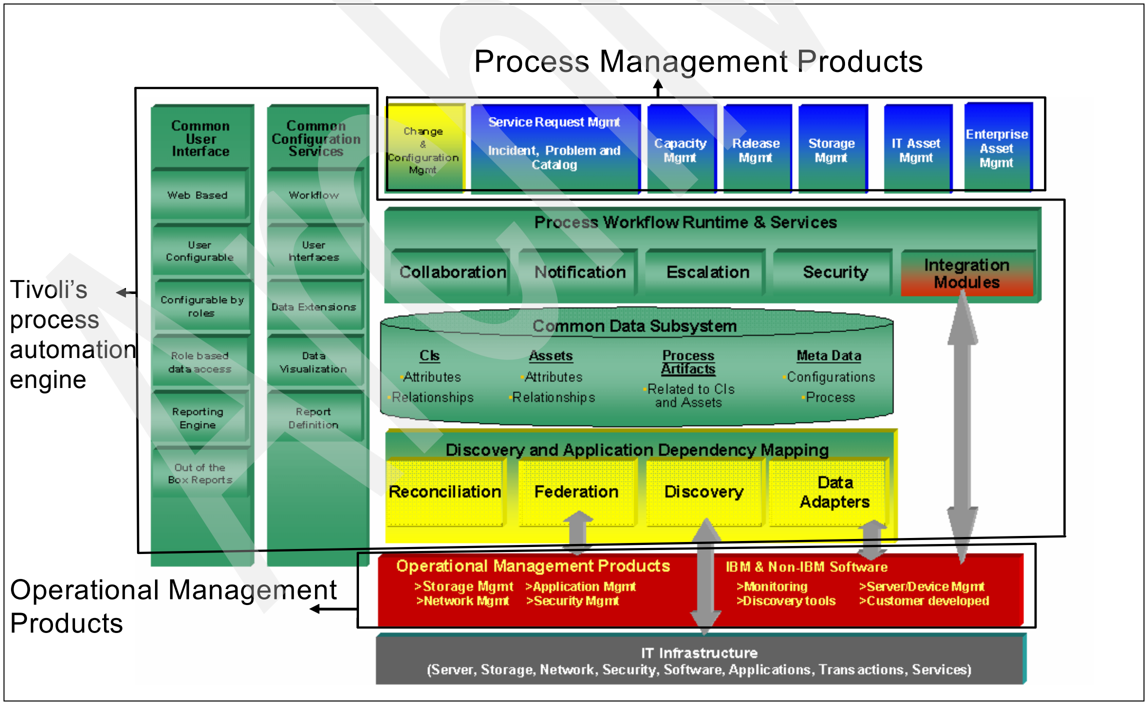


Figure 2-6 Logical component overview

Let us start with the bottom layer, Operational Management Products. Operational Management Products automate tasks to address application or business service operational management challenges. These products help optimize the performance and availability of your business-critical applications, along with the supporting IT infrastructure. They also help ensure the confidentiality and data integrity of your information assets while protecting and maximizing the utility and availability of your e-business data.

Operational Management Products can be implemented quickly to address immediate, specific IT challenges. As you implement a more comprehensive IT Service Management solution, these products can also integrate into the IT Service Management Platform and be utilized by IT Process Management Products, as shown in Figure 2-6 on page 32.

Tivoli's process automation engine provides the platform on which any of the applications, like incident management and problem management, are running. It is the foundation layer of the IBM Service Management process layer. Tivoli's process automation engine was also known by the name Base Services, which you might still find in some menus after installation. Tivoli's process automation engine provides rich tooling for configuring process flows, user interfaces, and process artifacts for the Process Managers implemented on top of it.

A Process Manager product (or application) is a system for managing the execution of a process. You can think of a process request as a ticket with a written note on it that is forwarded to various people (or entities) to perform various actions and in the end, resulting in the objective of the process.

Process Managers (also referred as Process Management Products (PMPs)) are applications that provide customizable out-of-the-box implementation of best practices processes to help customers integrate and automate IT management processes across organizational silos to improve productivity and efficiency.

The Process Managers focus on providing implementations of best practice processes as workflows, with roles, tasks, user interfaces, and integration modules stored in the process database. In addition, they provide the ability to adapt the out-of-the-box implementation to match the customer's unique process and workflow requirements, thereby allowing customers to both capture and implement their current processes as workflows, as well as provide customers with the ability to evolve to ITIL-defined best practices over time. Process Managers also provide the ability to monitor and report on process status and execution, thereby showing business owners the process bottlenecks within an organization, and providing the opportunity to improve and enhance their processes. The IBM IT Process Management products bridge organizational silos, and automate and integrate IT management.

2.3.1 Why use Process Managers

Customers implement processes out of necessity, that is, to solve particular pain points that exist at an organizational level if there was no process in place. For example, it is hard to imagine how a large organization can coordinate changes to production servers in a data center without having a well defined process. In this case, the particular pain points being addressed by using a process are the following:

1. Assessing and understanding all the impacts of a change (Pain point: Not all impacts are visible or well understood.)
2. Knowing about conflicting changes scheduled on a resource (Pain point: Direct conflicts are managed. Indirect conflicts are not known.)
3. Ensuring the right stakeholders are in the loop (for notification and approval) (Pain point: Either too many people are involved or critical people are left out.)
4. Maintaining logs for compliance / audits (Pain point: Laborious, needs extra effort, and should happen as part of the normal course of work.)
5. Understanding bottlenecks and improving organizational efficiency (Pain point: Some things just take too long, for example, several weeks to provision a server.)

Of course, this is just an example. There are many other pain points in a customer environment that can only be effectively addressed by implementing one or more processes. Process Managers help the customer implement the process and bring it to life.

2.3.2 Key terminology

Before continuing, we would like to introduce you to some of the terminology that we will be using throughout this chapter. We will elaborate on some of these terms in the subsequent sections.

- Service Organization: A department or organization that has implemented Maximo to manage their operations and processes. This might be a maintenance department, facilities department, physical plant department, IT department, call center or service desk, operations group, IT asset management group, an engineering department, or other group.
- Service Desk agent: A user who is responsible for taking calls and managing service requests, incidents, problems, and changes using the Service Desk applications.
- Requestor: A user or person who calls or e-mails the Service Desk, or reports issues or requests through the Self-Service Service Request module.

- Activities: Sequence of tasks done by a particular organizational entity (user or role). An activity is a specific class of work order used to describe general work that needs to be done on a ticket record. A ticket for a network printer problem might include three activities: check network connection, check printer, and check network. You create activities on incident and problem records to describe general, high-level work that must be performed to resolve the ticket. Activities can be assigned to different teams, different departments, or vendor or contract labor. A service desk agent can assign activities to other individuals or groups and still retain ownership of the ticket.
- Ticket: A ticket record is used to record a customer request or issue. Ticket records are created in the Service Requests, Incidents, and Problems applications. Ticket records can also be created automatically using data from e-mail messages, system monitoring tools, or external software applications.
- Work order: A work order is the details of the work performed to resolve the issue and consists of an instantiation of all work plans corresponding to an approved work request. The work orders are created in Activities, Changes, and Releases applications. Work orders are at the Site level.
- Change: A specific class of work order, used to plan the alteration of an information technology asset. This can include the addition, modification of applications, associated documentation, desktop builds, environments, hardware, networks, software, and systems.
- Change Management: The practice of ensuring that all changes to information technology assets are carried out in a planned and authorized manner. This includes ensuring that there is a reason for the change, identifying the specific assets and IT services affected by the change, planning the change, testing the change, and having a back out plan should the change result in unexpected consequences. Change records can be created in the Changes application, or through the Create Change action available from any ticket or work order application.
- Incident Management: The resolution and prevention of incidents that affect the normal operation of a company’s information technology (IT) services. Incident records can be created in the Incidents application, or through the Create Incident action available from any ticket or work order application.
- Problem Management: The resolution and prevention of problems that affect the normal operation of a company’s information technology (IT) services. This includes ensuring that all failures are corrected, preventing any recurrence of those failures, and the use of preventive maintenance to reduce the number of failures. A problem record can be created in the Problems application or from an incident record. When a problem record is created from an incident, it is for reference purposes only. The problem record is not created to resolve an incident.

- Release Management: The practice of designing and implementing procedures for the distribution and installation of hardware, software, and configuration changes to information technology systems.
- Service Request: A communication from an internal or external customer that reports an issue, requests information, or requests some sort of service. A service request record provides a means of tracking all Service Desk interactions with customers: walk up customers, phone calls, e-mail requests, and so on. You use the Service Request application to enter, view, and modify service request records. Depending on how your company chooses to implement the Service Desk module, a service request may lead to the creation of additional ticket or work order records.
- Solution: A predefined response to a problem or commonly asked question. A solution record consists of a symptom, a cause, and a resolution. Solutions can be associated with incident and problem records. You use the Solutions application to create, approve, and manage solution records.
- Ticket Template: A model for a generic or standardized service desk ticket record. You use the Ticket Templates application to create and manage generic templates that can be used to create service desk ticket records for common or high volume service desk calls.
- Tasks: A schedulable unit of work that can be assigned to a specific role.
- Role: The person or machine automation that has been assigned a specific set of tasks / activities.
- Key Performance Indicators (KPIs): Metrics to measure the effectiveness of a process.
- Job plan: A template consisting of tasks to be performed to get a particular job done (for example, changing the motor of a pump requires one to disconnect electrical connections, remove the casing, and do other tasks).
- Work plan: An instantiation of a job plan as part of a work order with the appropriate changes to tasks.
- Maximo Business Object (MBO): A Java object with business logic that encapsulates a Maximo database table (logical).
- Maximo Enterprise Adaptor (MEA): A component of Tivoli's process automation engine that facilitates sending/receiving business objects (for example, purchase orders, requests, and so on) to and from external applications. The inputs and outputs are typically MBOs.
- Action: A scheduled event that occurs when a record leaves a workflow node, that is, triggered by the routing that moves a record from one node to another.

- Escalation: Used to automatically monitor critical processes across your enterprise. The primary goal of Escalation Management is to ensure that critical tasks are completed on time, such as those defined in SLAs.
- Notification: A way to send a communication to a person or an application ID.
- Workflow: A Tivoli process automation engine feature used to automate repetitive business processes and record management processes.
- Communications template: A template to define and customize the communication.
- Work log: A log showing the status of tasks in a work order. Service Desk agents and maintenance workers create the entries in the work log, used to record telephone conversations, work performed, observations, and feedback from customers. Both the work log and the Communications Log (explained below) provide a means of communicating information about a ticket or work order.
- Communications log: A log of communications sent. Tivoli's process automation engine makes the entries in the communications log, which is used to maintain a record of e-mail communications created from the record through the Create Communication action.
- Security Groups: Used to grant access to sites, applications, and menu options. A user is assigned to one or more groups to gain access to the system.
- Common data model (CDM): A representation of CMDB entities, relationships, and their semantics.

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

1. Tivoli's process automation engine
    a. Common User Interface
    b. Common Configuration Services
    c. Common Data Subsystem
    d. Process Workflow Runtime and Services
2. Tivoli Service Request Manager solution
    a. Process Manager applications
    b. Service Desk
        i. Service Requests
        ii. Incidents
        iii. Problems

iv. Solutions

v. Global Search

vi. Customer Surveys

vii. Integration Toolkit

c. Service Catalog

2.3.3 Tivoli's process automation engine

Figure 2-7 provides an overview of the processes serviced by Tivoli's process automation engine. Tivoli's process automation engine delivers a single data subsystem to manage all of the intelligence, a common work flow that enables collaboration and cross silo activities, and a common set of configuration utilities.

Some of these functions will be described in more detail in this chapter.

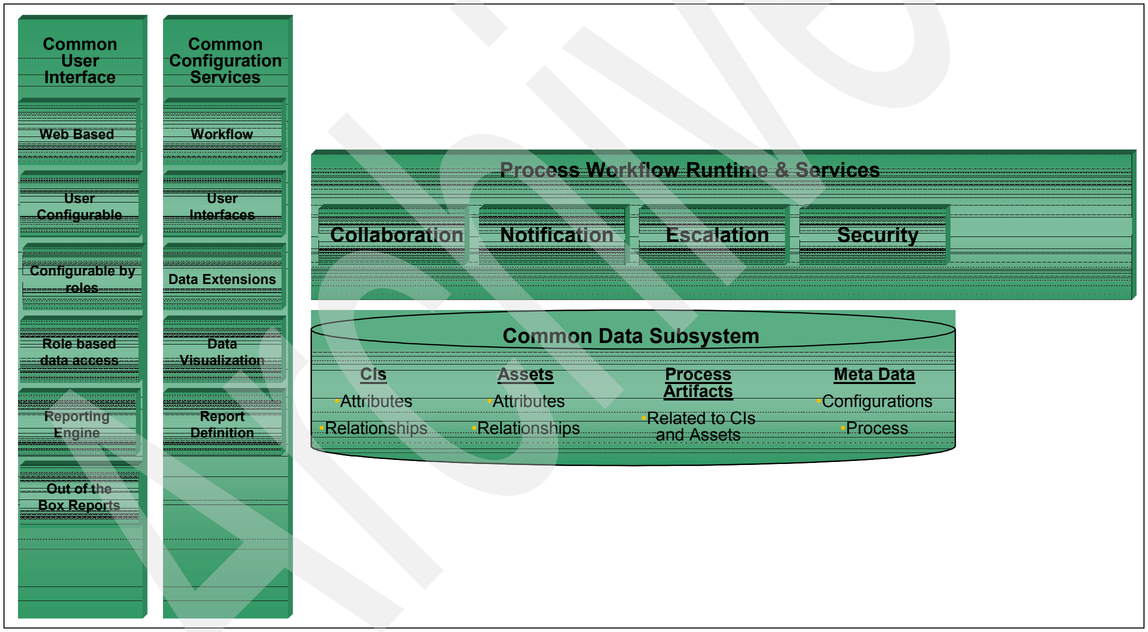


Figure 2-7 Tivoli's process automation engine functionality

Because we are describing the Tivoli Service Request Manager product, we will only focus on Tivoli's process automation engine functionality and Process Manager applications. Discovery and Application Dependency Mapping is related to CCMDB V7.1 and will therefore not be covered in this book.

How CCMDB extends Tivoli Service Request Manager's capabilities:

Although we will not cover the details of CCMDB in this IBM Redbooks publication, we do want to explain how CCMDB can extend Tivoli Service Request Manager's capabilities. The best way to do that is through an example.

CCMDB provides information to help the Service Desk team isolate the source of the problem much more quickly. Suppose a switch that is used for production goes down and affected users are calling the help desk. By looking at the applications and servers that are down (affected CIs in CCMDB terminology), a Service Desk personnel or a Specialist can understand that they are all related with a certain switch (through the CI relationship information provided by CCMDB). They can also see that this switch has caused problems before and after a closer analysis, they can understand that the software of the firmware on this switch is back level. All this information is provided by CCMDB. At this point, they can start a change request to upgrade the switch software or firmware. This change request is reviewed by the Change Manager or the change review board to analyze the impact of the change (again using the CCMDB, looking at the CIs affected) and once the change is authorized it gets implemented.

All these are ITIL processes and by integrating your help desk processes with configuration and change management processes, you greatly increase the efficiently of these processes.

Common User Interface

The Tivoli Service Request Manager Common User Interface is the user interface for administering and using the Process Manager applications. It also allows the user, given the appropriate permissions for the user, to use tools to customize the database, workflow, and user interface behavior and style.

Figure 2-8 on page 40 provides an overview of Tivoli's process automation engine Common User Interface. This section provides information about the applications and functions related to the User Interface.

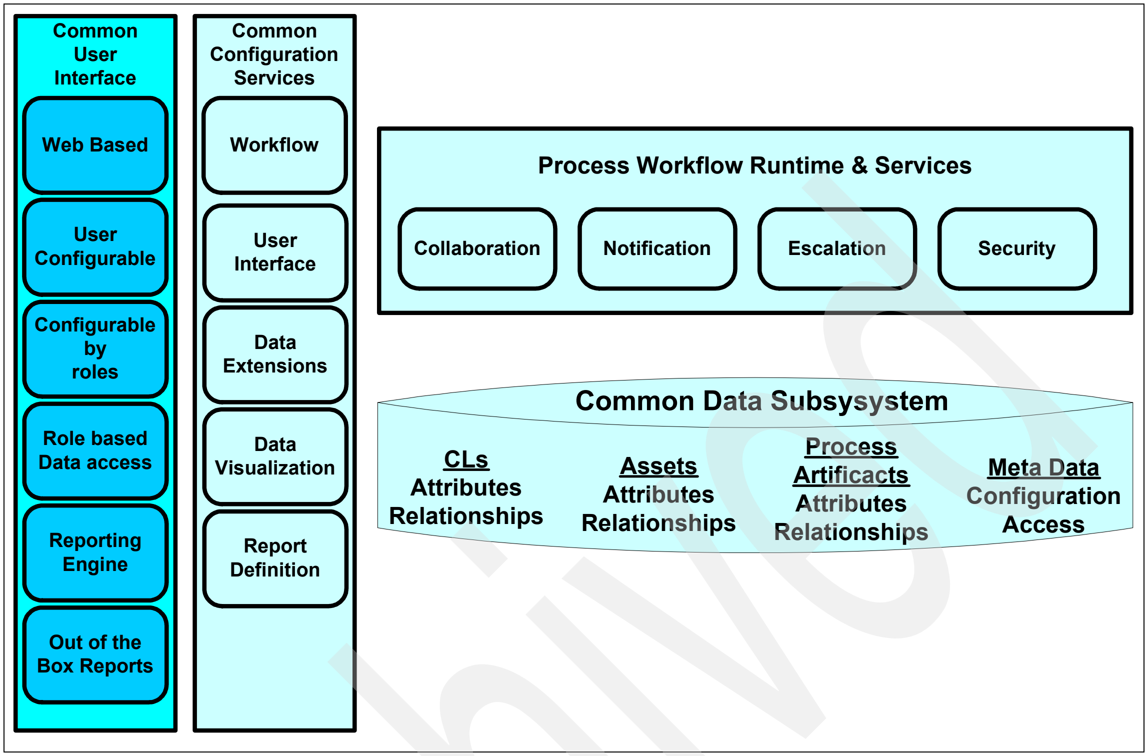


Figure 2-8 Common User Interface

Web-based

Tivoli Service Request Manager is a Web-based application suite. You access it through a Web browser by entering the Uniform Resource Locator (URL) that points to an application server that is running Tivoli Service Request Manager server. There is no client software to be installed. All users, including administrative users, access Tivoli Service Request Manager through a browser.

Web-based Tivoli Service Request Manager windows are also accessible to any supported client devices connected to the Internet. They leverage the built-in capabilities and standards of the client's Internet browser, reducing the deployment, versioning, and software conflicts that commonly occur on client devices running earlier client/server and Web-enabled applications. The rendering of Tivoli Service Request Manager windows requires no network infrastructure changes and has minimal network impact due to its Internet-optimized architecture. Anyone requiring access to Maximo data (employees, contractors, or partners) can do so from anywhere, without requiring specialized client hardware or software.

The Tivoli Service Request Manager User Interface allows launch-in-context to external systems. As shown in Figure 2-9, you can specify the target system of the launch operation as well as specific *land-into views* inside the Launch in Context application.

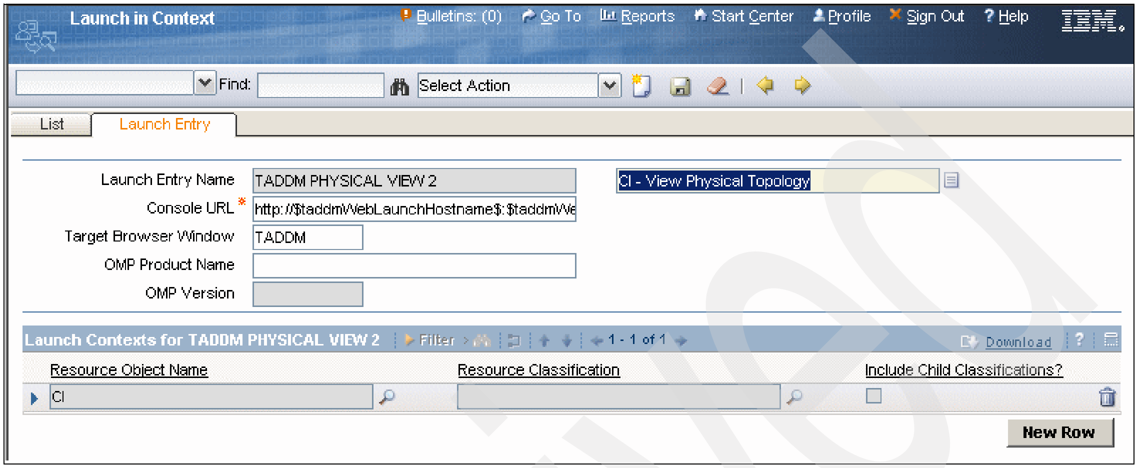


Figure 2-9 Launch in Context application

The Tivoli Service Request Manager User Interface also provides a common look and feel for all functions.

Optionally, the Tivoli Service Request Manager User Interface can be used inside the IBM Solution Console (ISC), a common console used across different IBM products, including many IBM Tivoli solutions. If you want to work with Tivoli Service Request Manager applications from the ISC console, you can do so.

User Configurable

After you sign in, Tivoli Service Request Manager displays your Start Center. A Start Center contains links to actions, applications, data, records, and reports that are relevant to your job, and it is very configurable. It allows each user to define their own start page.

> **Note:** Tivoli's process automation engine uses an XML-based screen framework layer that stores the presentation elements in XML format. This greatly improves the capabilities of application windows that utilize this framework and allows for dynamic window generation, better change tracking, and continuity across upgrades. Dynamically generated windows are also more adaptable to various screen sizes and devices compared to proprietary window technologies or even pure Hypertext Markup Language (HTML) screens.

Figure 2-10 shows an example of a Start Center, which for any Tivoli Service Request Manager user will be the first access point to use application functionality.

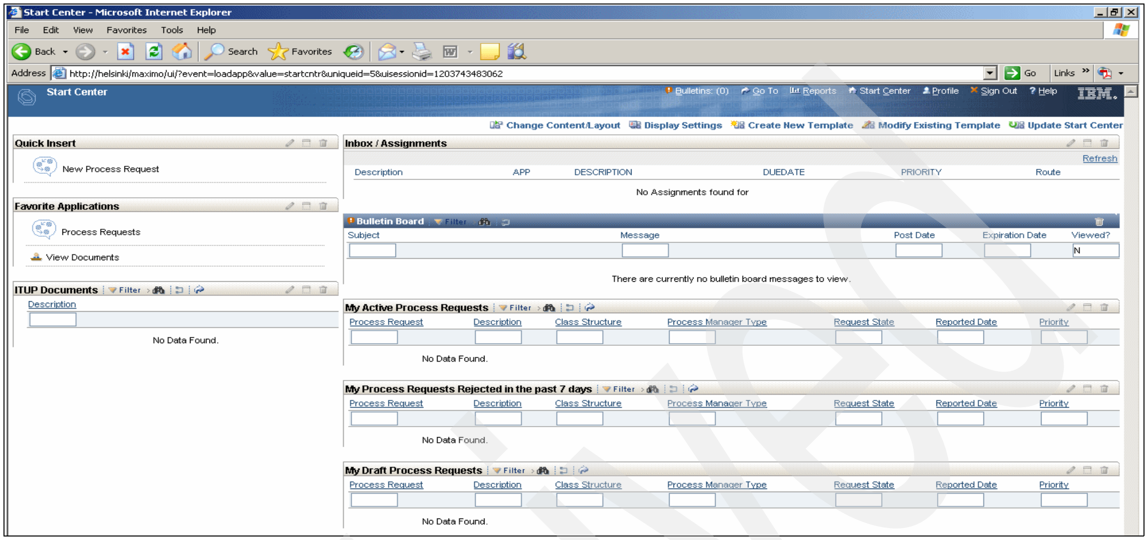


Figure 2-10 User Interface

Each portlet has an application that controls the content and configuration of the portlet. If your system administrator has granted you security permissions to a portlet's application, then you can personalize how that portlet appears in your Start Center.

Administrators and users can define content portlets that will display values based on a certain set of criteria. For example, a user may want to see all of his or her group's time cards that were entered in the past day. Another user may want to see critical asset conditions and pending work orders. Interactive KPI graphs based on user-defined criteria and set points can be placed on the Start Center for a visual representation of events and conditions. For example, the user above who wanted to see the past day's time cards may also want a graph that will show visually the number of overdue time cards or time cards awaiting approval in red. A variety of graph formats and options are provided, allowing each user to define his or her own start page.

> **Note:** Unlike the simple personalization found in many Web applications today, Start Center goes beyond static layouts to provide configurable content portlets and dynamic key performance indicators (KPIs).

Configured by roles

Tivoli Service Request Manager provides a powerful, role based security. Start Centers are assigned to security groups by a system administrator using the Security Groups application. When adding or updating users, you assign them to one or more security groups. The combination of groups determines users' profiles (set of authorizations and privileges across Sites and Organizations).

A profile is a virtual view of a user's authorizations, privileges, and settings within Tivoli Service Request Manager. Users inherit the access rights and privileges associated with assigned security groups. Tivoli Service Request Manager builds and maintains a profile for all users in the system that actively tracks a user's membership in groups.

If your user name is assigned to more than one security group, you may have more than one Start Center.

> **Note:** If you have more than one Start Center assigned to your user name, Tivoli Service Request Manager displays each Start Center as a separate tab, that is, one for each Start Center.

Role-based access

As discussed above, your role determines how you access the Tivoli Service Request Manager, meaning what applications you see and what data you can access.

Reporting Engine

Tivoli Service Request Manager contains an integrated report engine, based on BIRT technology. You use the Reports Administration application (see Figure 2-11 on page 44) to perform the following tasks:

- Open a list of reports.
- Define reports for your users.
- Change the names of reports and field titles.
- Set or view security.

As the Report Administrator, you can specify the following for users:

- Availability of reports and how they open, run, and print
- Appearance of report titles and headings
- Report security settings

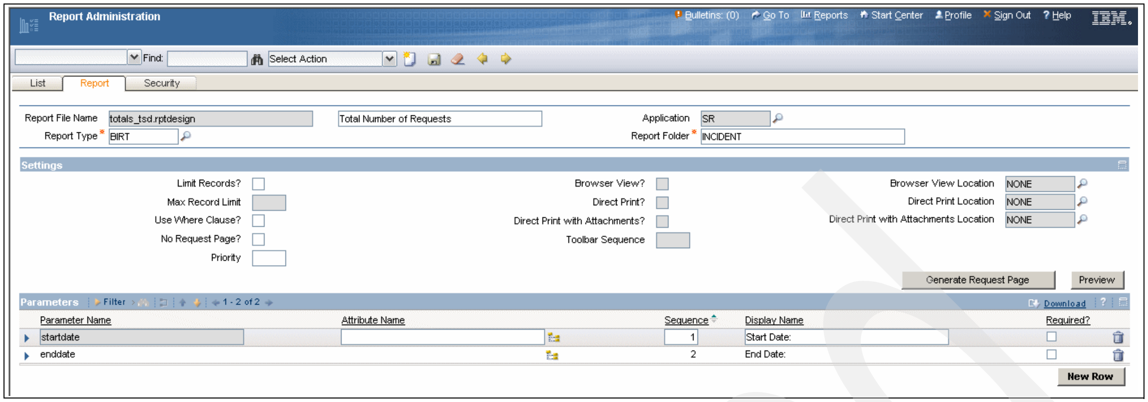
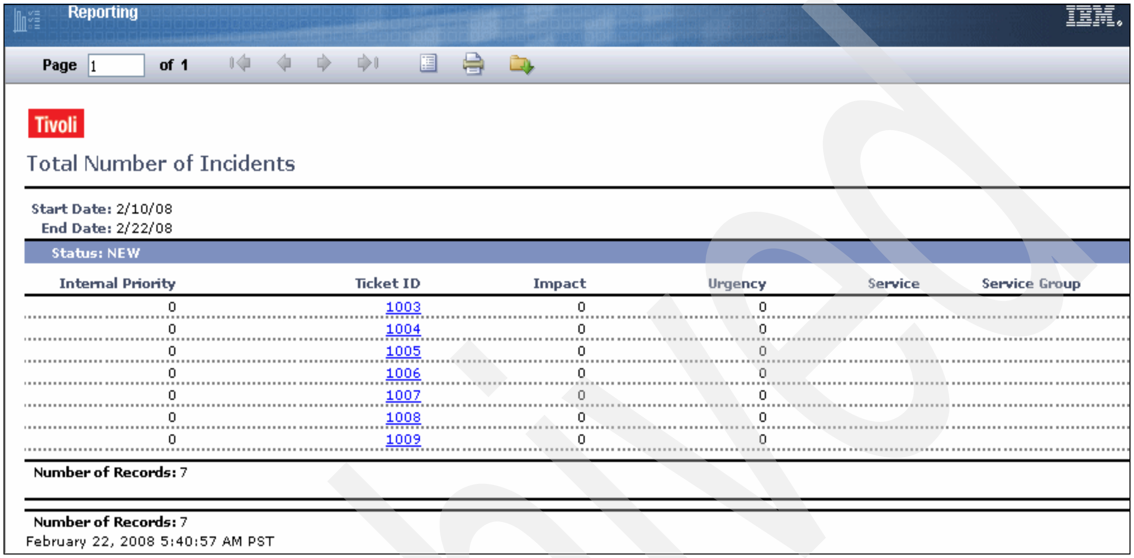


Figure 2-11 Report Administration

You can create your own reports or use the ones provided out of the box, which is discussed in "Out of the box reports".

Out of the box reports

Tivoli Service Request Manager will be shipped with out of the box reports you can immediately use or customize to your specific needs. For example, Figure 2-12 shows one of the reports that is shipped out of the box: Total Number of Incidents report.


Reporting

Page 1 of 1

Tivoli

Total Number of Incidents

Start Date: 2/10/08
End Date: 2/22/08

Status: NEW

Internal Priority	Ticket ID	Impact	Urgency	Service	Service Group
0	1003	0	0		
0	1004	0	0		
0	1005	0	0		
0	1006	0	0		
0	1007	0	0		
0	1008	0	0		
0	1009	0	0		

Number of Records: 7

Number of Records: 7
February 22, 2008 5:40:57 AM PST

Figure 2-12 Total Number of Incidents report

> **Note:** It is expected that these reports will be provided through the OPAL Web site.

Common Configuration Services

Figure 2-13 on page 46 provides an overview of Tivoli's process automation engine configuration services. This section provides more information about these services.

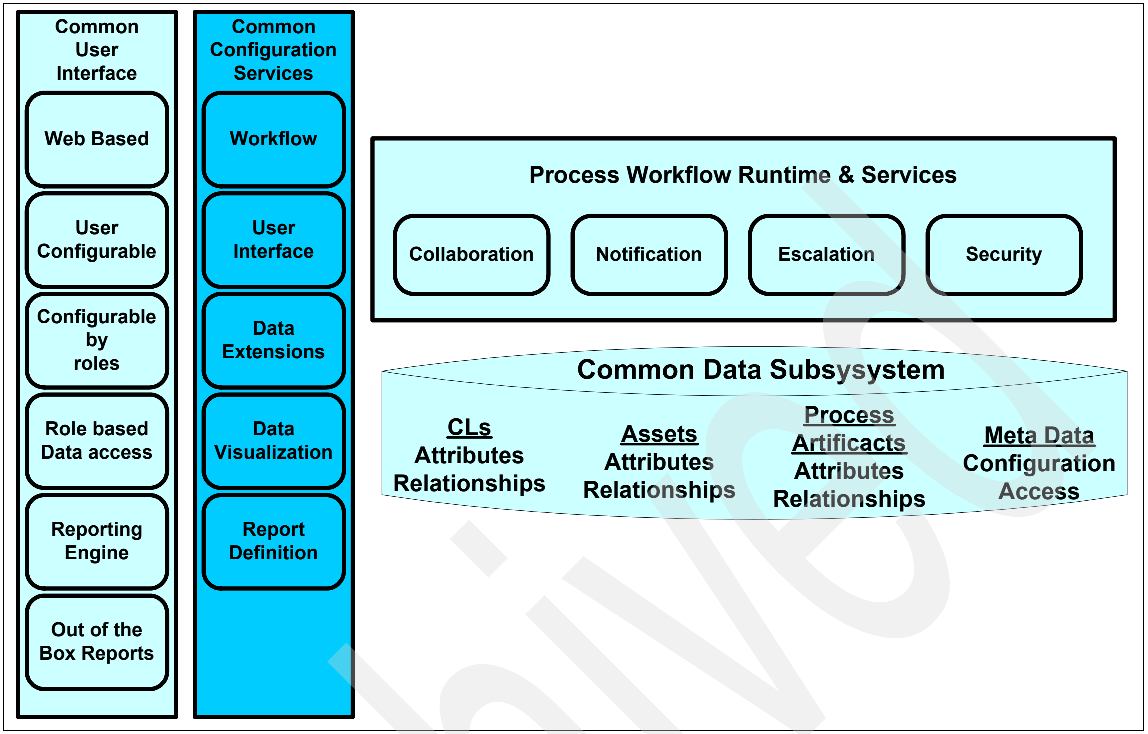


Figure 2-13 Common Configuration Services

The following applications are part of the Common Configuration Services:

- Database Configuration
- Application Designer
- Workflow Designer
- Workflow Administration

Workflow

Common Configuration Services' workflow features let you automate repetitive business processes and record management processes. Automating these processes provides a means for greater efficiency and accountability throughout your enterprise.

Tivoli's process automation engine workflow is a business process management (BPM) agent that defines and automates Tivoli's process automation engine business process flows. Besides the standard predefined workflow templates, new workflows can be defined for any business process in any Tivoli process automation engine application.

Two applications that we will discuss regarding the workflows are Workflow Designer and Workflow Administration application.

- Workflow Designer: You can use the Workflow Designer to define processes that perform many tasks, take data actions, and send notifications. These tasks, data actions, and notifications in turn depend on roles to determine the task owner(s), actions, or action groups to execute predefined tasks and communication templates to define default messages and recipients for notifications.

  You can use the Workflow Designer application to create, view, modify, and delete workflow process records that reflect your business process. An active workflow process revision defines the different paths that a record can take as it moves through the business process, and the different actions and notifications that should take place at different points in the process.

  Figure 2-14 shows the Workflow Designer application, including a simple process flow for a process request.

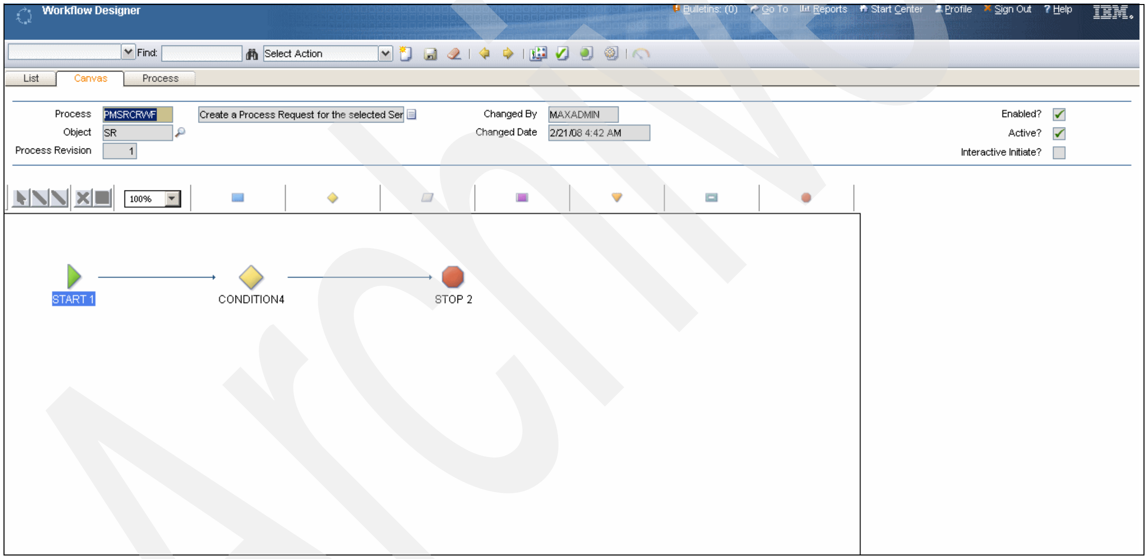


Figure 2-14 Workflow Designer

- Workflow Administration: You use the Workflow Administration application, as shown in Figure 2-15 on page 48, to view and manage active instances of Workflow processes. You can use Workflow Administration to perform the following tasks:
  - View Workflow assignments.
  - Reassign Workflow assignments.

– Delete Workflow assignments.

– Stop active processes, removing the record from the control of the processes.

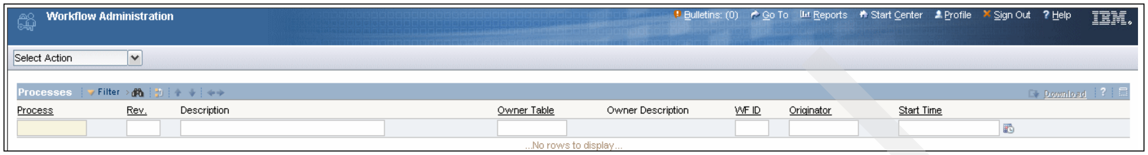


Figure 2-15 Workflow Administration

> **Note:** If you access the Workflow Administration application without any workflow active, no workflow has to be administered, and the window shown in Figure 2-15 will appear.

User interfaces

The Common Configuration Services provides integrated tooling to customize data and applications' user interfaces. Using the Application Designer, you can adjust the user interface for any application in Tivoli Service Request Manager, making sure the appropriate MBO objects and attributes are selected and used.

Application Designer lets you configure the user interface for applications to meet your specific business needs. The ease of use of the Application Designer enables you to configure applications without editing a line of code. The drag and drop feature in the Application Designer further simplifies the process of editing applications.

> **Note:** Application Designer leverages the XML layer of the screen framework to track screen elements, formatting, and upgrade transformation mappings.

Some of the common modifications you can make using the Application Designer are moving fields and sections, and creating new fields, tables, and tabs. Other advanced capabilities are creating or duplicating applications, defining signature options, and editing the tool bar or the Select Action menu (see Figure 2-16).

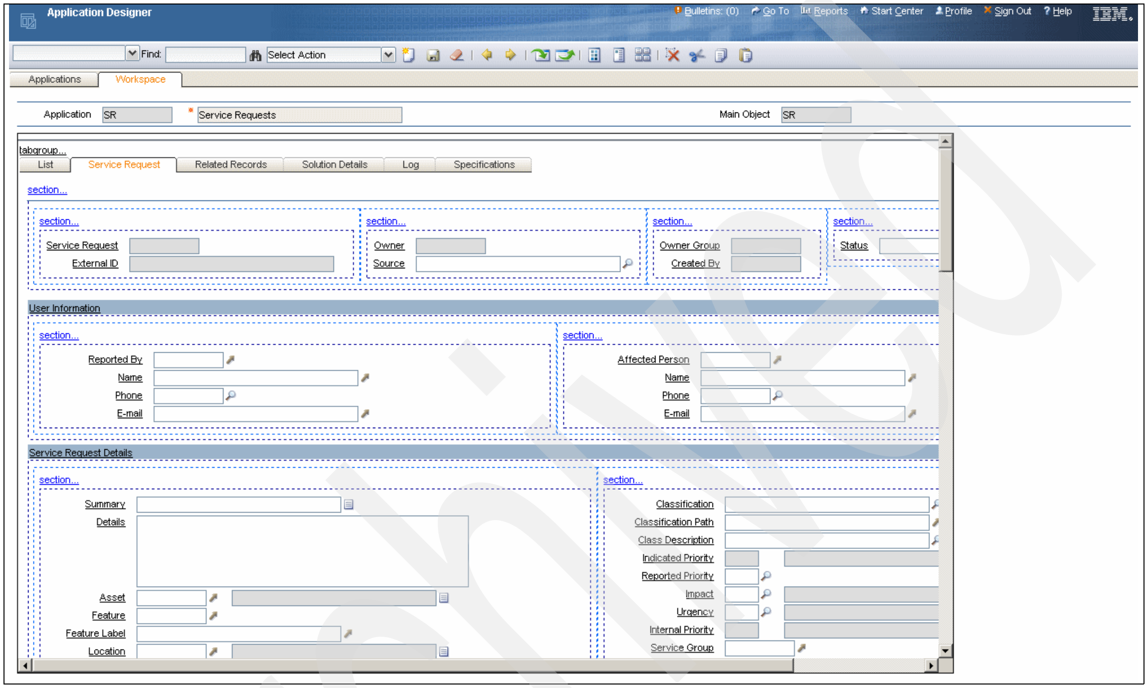


Figure 2-16 Application Designer

Data Extensions and Data Visualization

The Common Configuration Services provides tools and applications to configure the data access layer and make sure data in the database tables are accessed through appropriate business logic and encapsulated inside Java objects, also referred to as Maximo Business Objects (MBO). Once logged into Tivoli Service Request Manager with the appropriate authorization, you can use the Database Configuration application to add, change, or delete an MBO object, attributes, or relationships.

Figure 2-17 on page 50 shows the Database Configuration application.

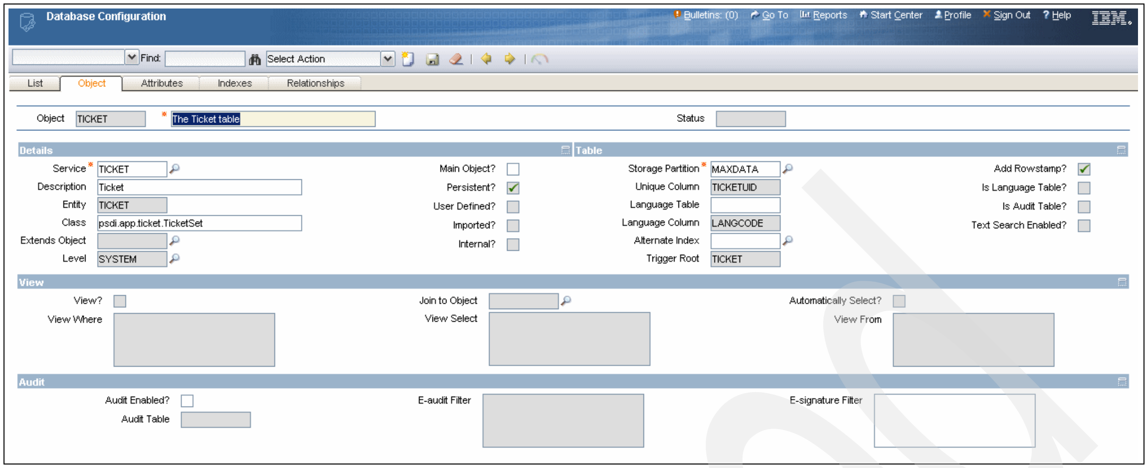


Figure 2-17 Database Configuration (object)

The MBO object provides you with information about general characteristics and values of a selected object. An object can be either a table or a view. A table object can be persistent or non-persistent. Viewed objects can be persistent or not. Only table objects can be audit-enabled.

An example of a table object is the TICKET table, which contains all the request details created. The SR view is an example of a view, which in this case will show information from the ticket table, showing only the information valid for service requests. You will see, for each view, that the *Extends Object* field contains a value, which in this case or for the SR view will be TICKET.

Figure 2-17 also contains a field named *Storage Partition*, which is the tablespace used on DB2® and Oracle®, or the file group on SQL Server®. The domain DBSTORAGEPARTITION contains the list of valid tablespaces and file groups available to the system. The database administrator must create the tablespace or file group before adding its name to the domain.

Note: On DB2 and Oracle databases, we suggest that data reside in its own table space and name this table space MAXIMO. On SQL Server, the default file group is named PRIMARY. This structure can accommodate simple databases. Larger and more complex databases that are administered by a DBA can have additional table spaces (DB2 and Oracle) and file groups (SQL Server). See your database server documentation for details.

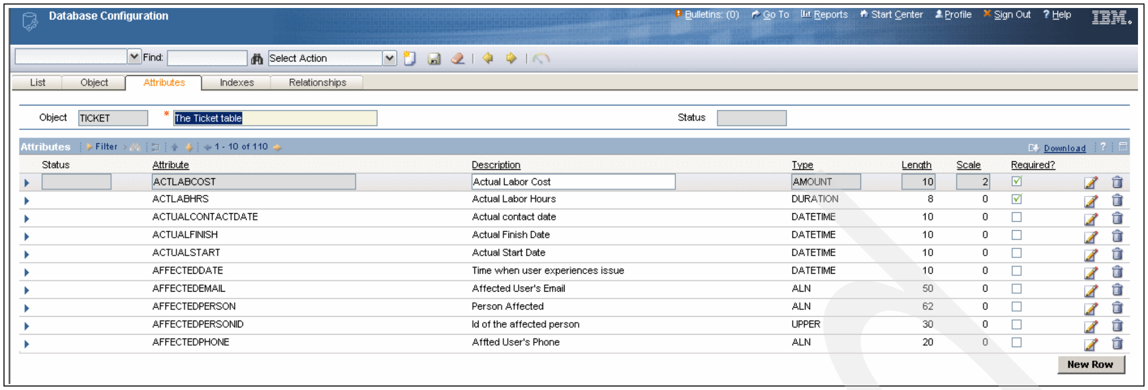


Figure 2-18 Database Configuration (attributes)

You use the Attributes tab to view existing attributes for a selected object or to add new attributes to a selected object. The attributes tab displays the following columns in the table window:

- Status
- Attributes
- Description
- Type
- Length
- Scale
- Required attributes

Click the expand arrow to the left of an attribute to display the *Details* and *Advanced* sections. The Details section contains common characteristics that apply to most situations. The Advanced section contains characteristics less commonly used, and in special cases users may need to view or modify these values.

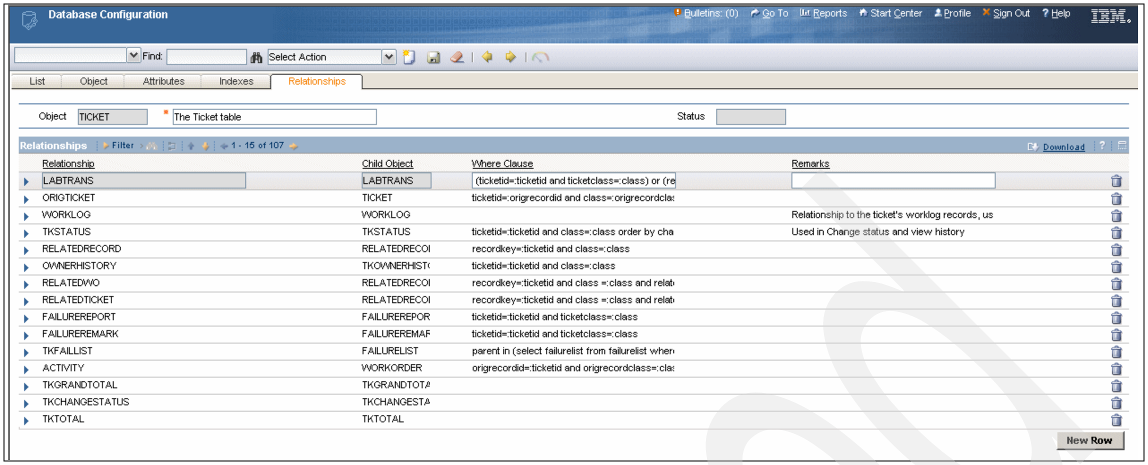


Figure 2-19 Database Configuration (relationships)

You use the Relationships tab to define SQL for JOINS, so you can create relationships between parent and child tables. A JOIN lets you link together data from multiple tables, and in the system, the Parent is the table you already have and Child is the one you are trying to get.

So, for example, Parent = MAXUSER, Child = SITE, and Name = DEFSITE means you have maxuser and want to get the site for the user's default site, as follows:

```
siteid = :defsite
```

This means site.siteid = maxuser.defsite. When the SQL gets executed, anything preceded by a colon is substituted with the value of that attribute from the parent.

Report definition

As discussed in “Reporting Engine” on page 43, Tivoli Service Request Manager contains an integrated report engine, based on BIRT technology. You can define your own reports or use the out-of-box reports shipped by Tivoli Service Request Manager.

Common Data Subsystem

The Common Data Subsystem or Data Layer contains the data space that holds Assets, CIs, process artifacts, and relationships between these objects, as shown in Figure 2-20.

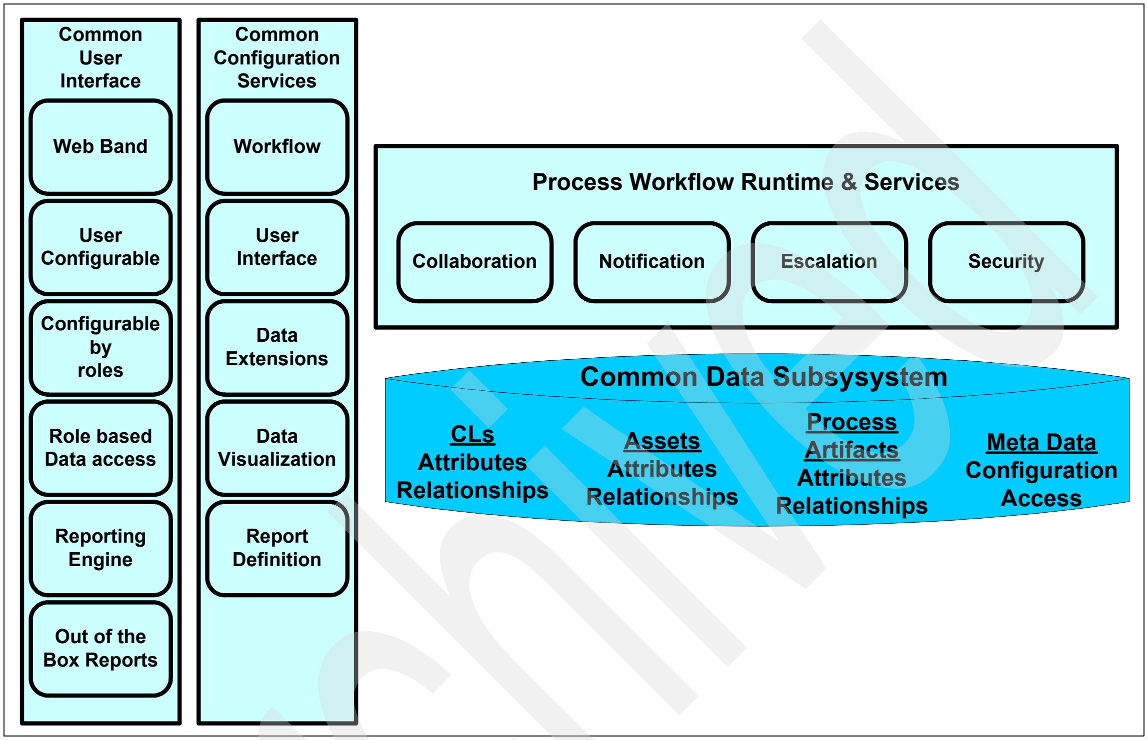


Figure 2-20 Common Data Subsystem

These objects are used to register Service Desk requests, such as service, incident, and problem requests. For this purpose, Tivoli Service Request Manager supports the Tivoli Common Data Model (CDM).

The CDM is a consistent, integrated, and logical data model that defines the general characteristics of information stored in the CMDB. The model specifies how this data is organized to correspond to real-world entities and defines the relationships between the entities. The CDM represents management information in a way that is easy for management applications to use.

The CDM is a logical information model that is used to support the sharing of consistent data definitions and the exchange of data between Tivoli management products concerning managed resources and components of a customer's business environment.

Figure 2-21 presents an abstract overview of the main sections of the Common Data Model. The important thing to understand from this diagram is what subsections exist in the CDM.

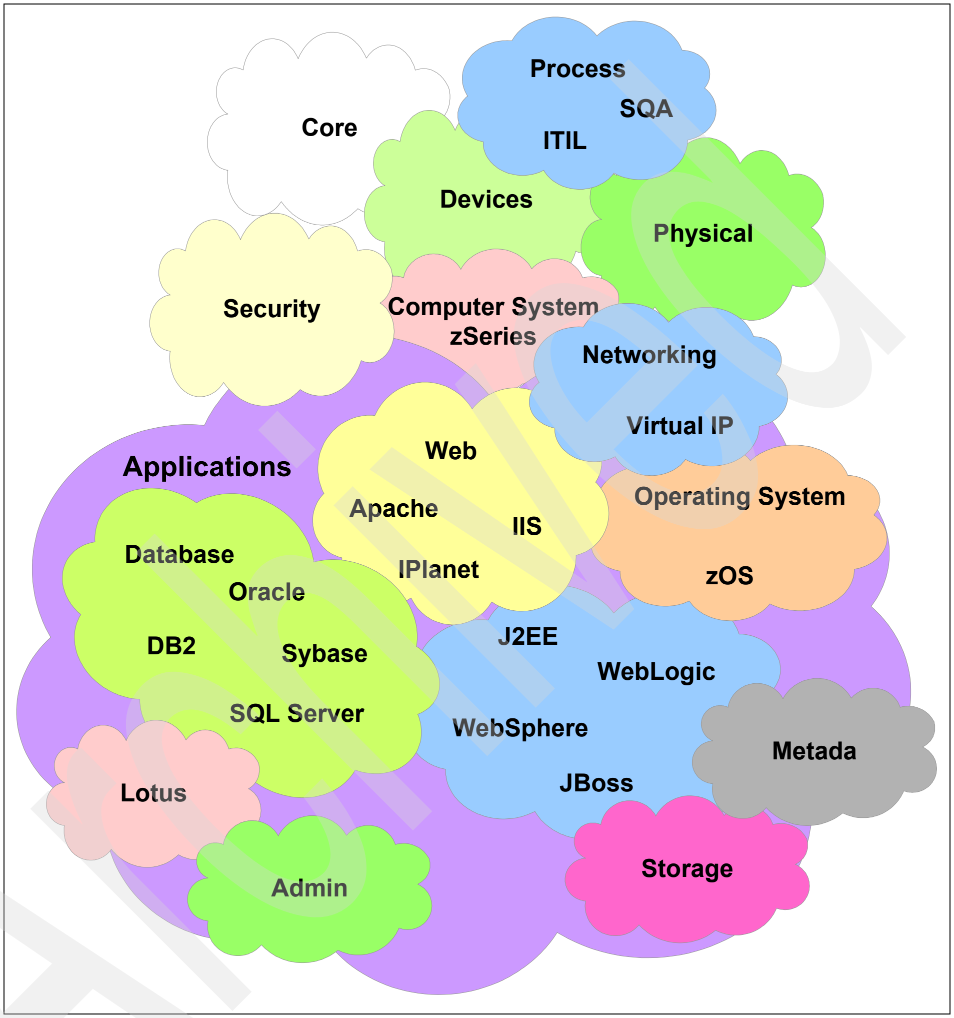


Figure 2-21 Common Data Model

An example of the subsection for databases is shown in Figure 2-22.

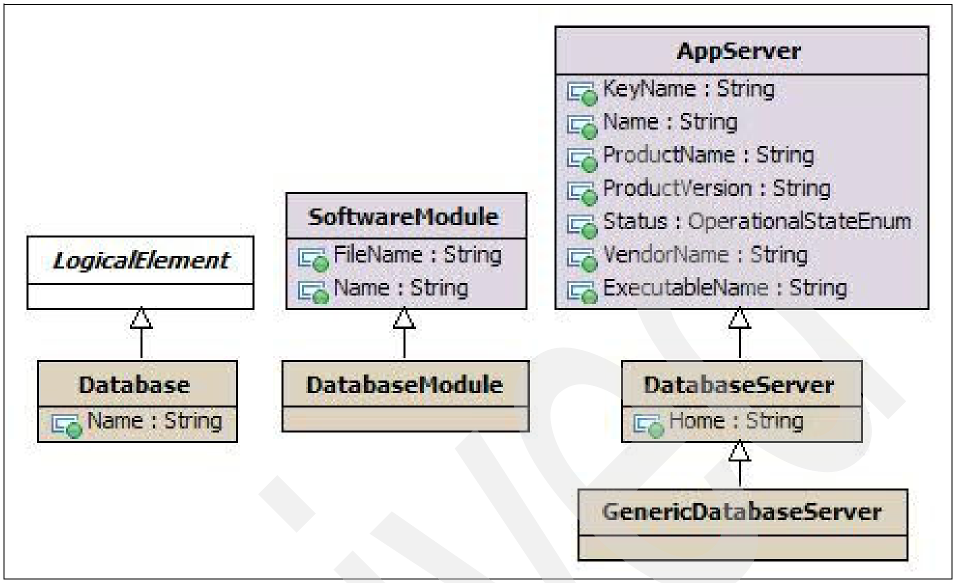


Figure 2-22 Database sub model

The database sub model provides, in a general way, information that is useful for managing databases. Each of the major database products (DB2, Oracle, SQL Server, and Sybase) is represented in a more detailed sub model derived from this model.

You can find more information about the Common Data Model in *IBM Tivoli Common Data Model: Guide to Best Practices*, REDP-4389. The latest CDM user guide can be downloaded from the Tivoli Open Process Automation Library at:

https://www14.software.ibm.com/webapp/iwm/web/reg/download.do?source=tivopal&S_PKG=1TW10CC1R&lang=en_US&cp=UTF-8#

Note: You need an IBM ID and password to be able to access the library and download the document.

Federation of external data

The CMDB allows for data sourced in an external repository to be made available to the service management processes and applications by configuring the federation capabilities of the underlying database or federation support tools and using the Tivoli Service Request Manager tooling available (Database

Configuration and Application Designer applications) to make the federated data accessible for display in applications.

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

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

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

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

Process Workflow Runtime and Services

Figure 2-23 on page 57 provides an overview of Tivoli's process automation engine Process Runtime and Services. This section provides information about the applications used to provide these services.

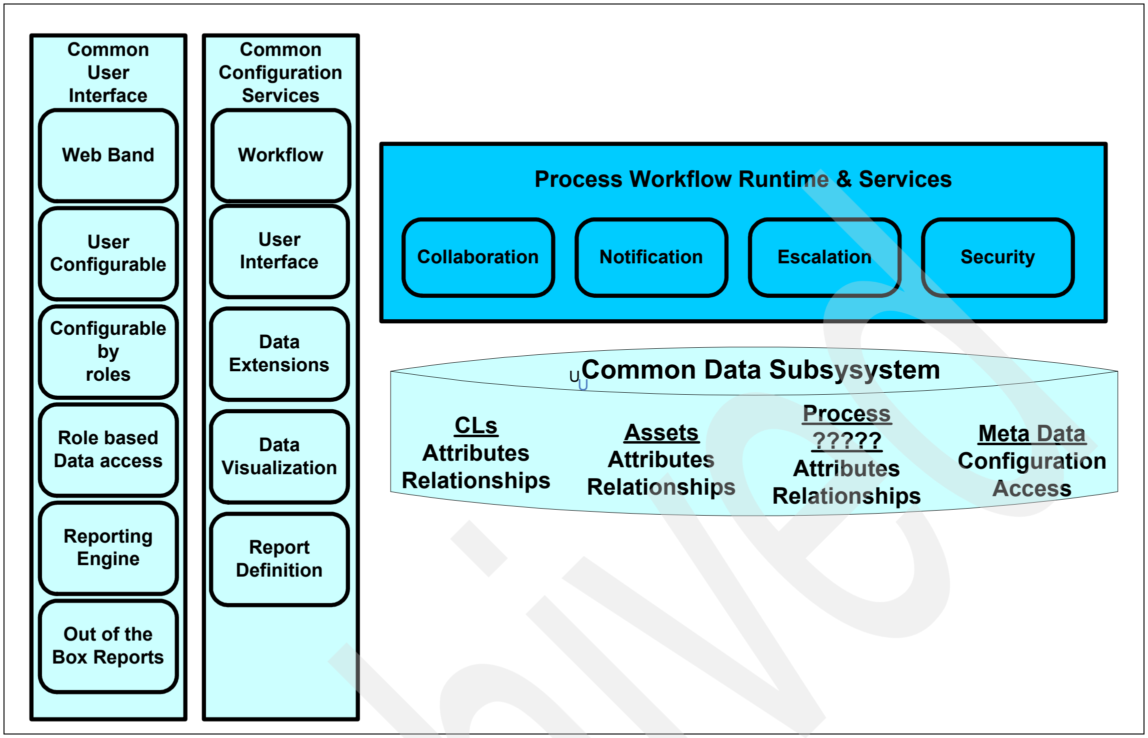


Figure 2-23 Process Workflow Runtime and Services

Collaboration

The Process Workload Runtime and Services layer provides services to automate processes, guide users through these processes, and allow collaboration among different users. There are several applications that are used in the Process Workload Runtime and Services layer, such as Job Plans, Activities, Notifications Escalations, and Security. We will talk about these applications in this section.

- Job plan: A job plan is a detailed description of the work to be performed. Job plans generally contain tasks (procedures), along with lists of estimated labor, labor hours, materials, services, and the tools required for the work. To ensure that work on an asset is performed in a safe manner, you can add work assets and associate safety plans that can be used when the work is performed.

  You can specify which organizations or organizations and sites can use the information about the plan and its tasks. If you do not specify this information, the job plan can be used in any site of any organization.

In the Job Plans application (Figure 2-24), you can break job plans into tasks or steps and enter estimates for labor, materials, services, and tools for those tasks. Using tasks can be valuable when the job plan is lengthy and complicated, and lets you track costs more accurately. For each task, you can also create a nested job plan.

Identifiers for job plan tasks must be unique, but you can enter multiple records for the same task ID in the Labor, Materials, Services, and Tools subtabs; this is useful when a task requires more than one worker, item, service, or tool.

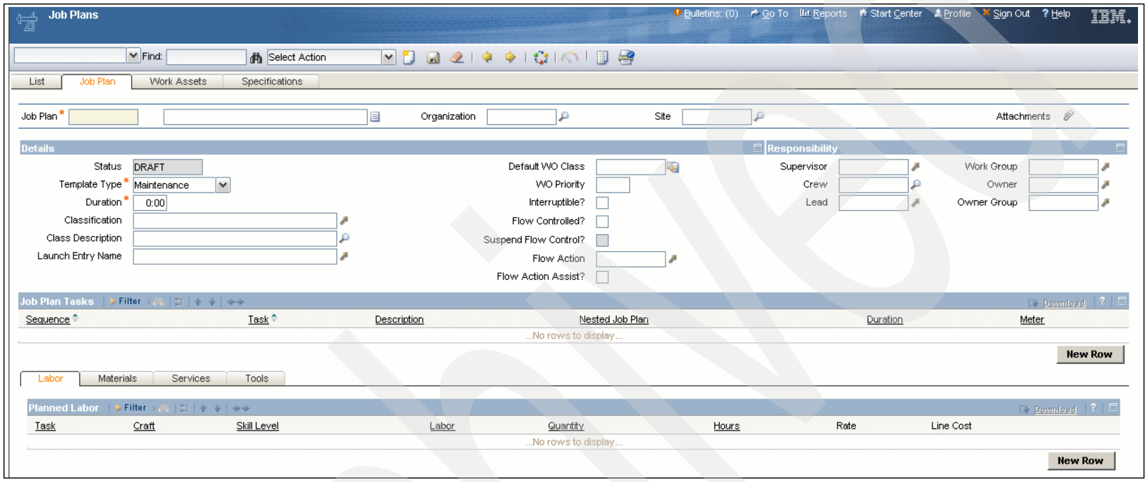


Figure 2-24 Job Plans

- Activities: An activity is used to initiate a work process and create a historical record of work being performed. Activities can be created in the Incident Management and Problem Management applications, and are planned, reviewed, and managed in the Activities application.

  An activity is a type of work order, where the work order class is automatically set to ACTIVITY. Activities are stored in the work order table, which means you can also search and modify activities in the Work Order Tracking application. The WOACTIVITY view is an extension of the WORKORDER object, which will filter work orders based on the work order class. By default, it is not possible to create activities in the Work Order Tracking application, due to restrictions of the class field.

  In the Activities and Tasks (or Work Order Tracking) application (Figure 2-25 on page 59), you can view and modify work order details, such as the work assets, work type and priority, job plan, GL account, scheduling information, ownership, and follow-up work.

You also classify an activity or a task and specify attributes to define the activity or task further. For example, an activity or task might involve the classification of a mobile computer, with attributes of memory, disk space, and speed, for which you can define values. Attributes help to categorize activities and tasks, which makes it easier to find and manage them.

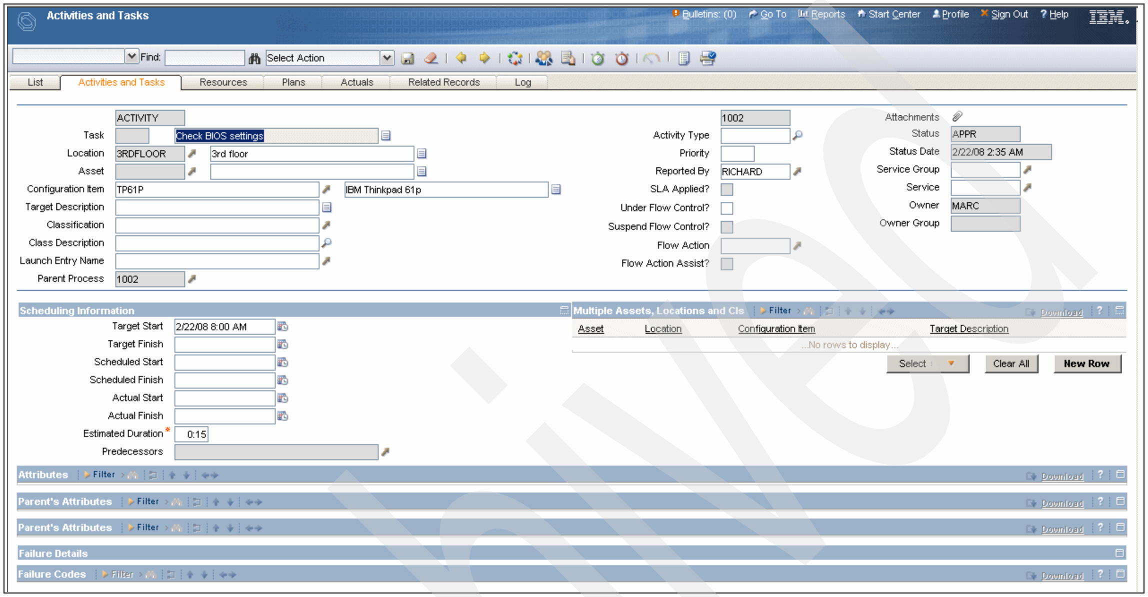


Figure 2-25 Activities and Tasks

Notifications

Tivoli Service Request Manager contains a communication platform to send notifications to persons or a group of persons. The platform is based on templates you can use to send notifications, which will be sent as an e-mail.

You use the Communication Templates application (Figure 2-26 on page 60) to create and manage generic communication templates that users can leverage to standardize frequently used communications.

For example, service desk agents can manually create and send e-mail communications from the Ticket applications (Service Requests, Incidents, and Problems) using standardized information from predefined communication templates. The recipients of these communications can respond, and agents can view the two-way dialog from the Communication Log in the Ticket applications. You can also use communication templates to create e-mail notifications for use with the automated workflow and escalation processes.

You can associate specific file attachments to a communication template, and you can also associate document folders to the template, which the system will search when a service desk user applies the template to a ticket. When a communication is actually sent, the system attaches to the communication any files that exist in the associated document folders along with those hard coded in the template itself.

The system has numerous predefined communication templates. These templates support notification functionality in the E-mail Listener Configuration, Escalations, Workflow, Service Requests, and Incidents applications.

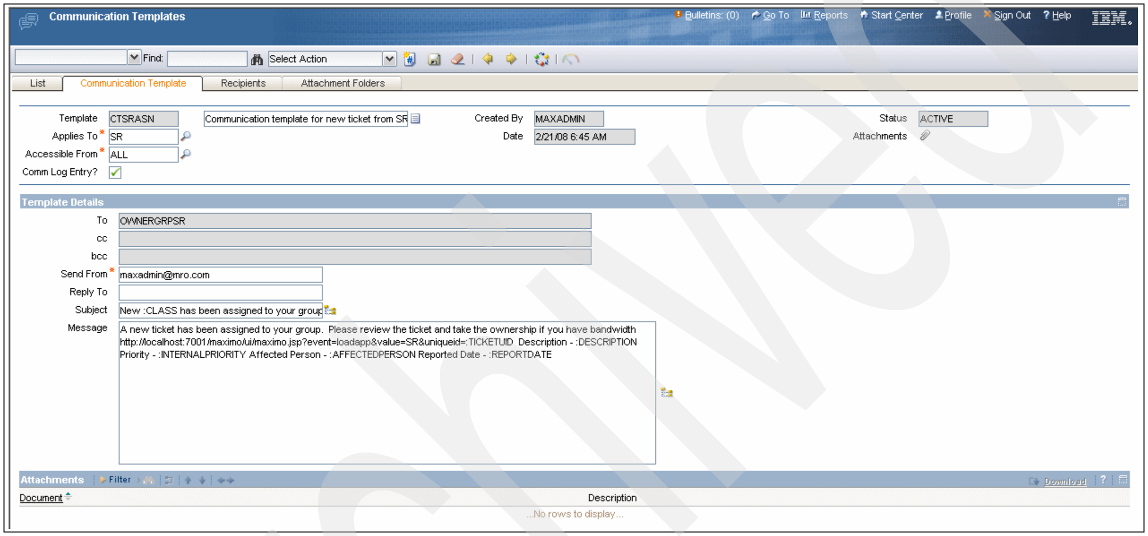


Figure 2-26 Communication Templates

Escalations

Escalation services verify predefined conditions, if they have been met and send notifications or take action(s) on exceptions. The primary goal of Escalation Management is to ensure that critical tasks are completed on time, such as those defined in Service Level Agreements. You can also use escalations for events, such as notification before contracts expire, changing the status of a record (such as for CIs), or changing the owner of a record (such as for service requests, incidents, or problems).

You can use escalations with any application. However, you are most likely to use them with the Service Desk applications, IT Asset Management applications, and workflow processes. You can create escalations at the site, organization, or system level. If you specify an organization or site, the escalation is restricted to that site or organization. If you leave both the Organization and Site fields empty, the escalation is available to all sites and organizations (system-wide).

The system provides you with numerous predefined escalations. You can tailor them to suit your business needs, or create new ones. You can also build and validate escalations incrementally while they are inactive, and activate them after validation.

Figure 2-38 on page 74 shows the Escalations menu.

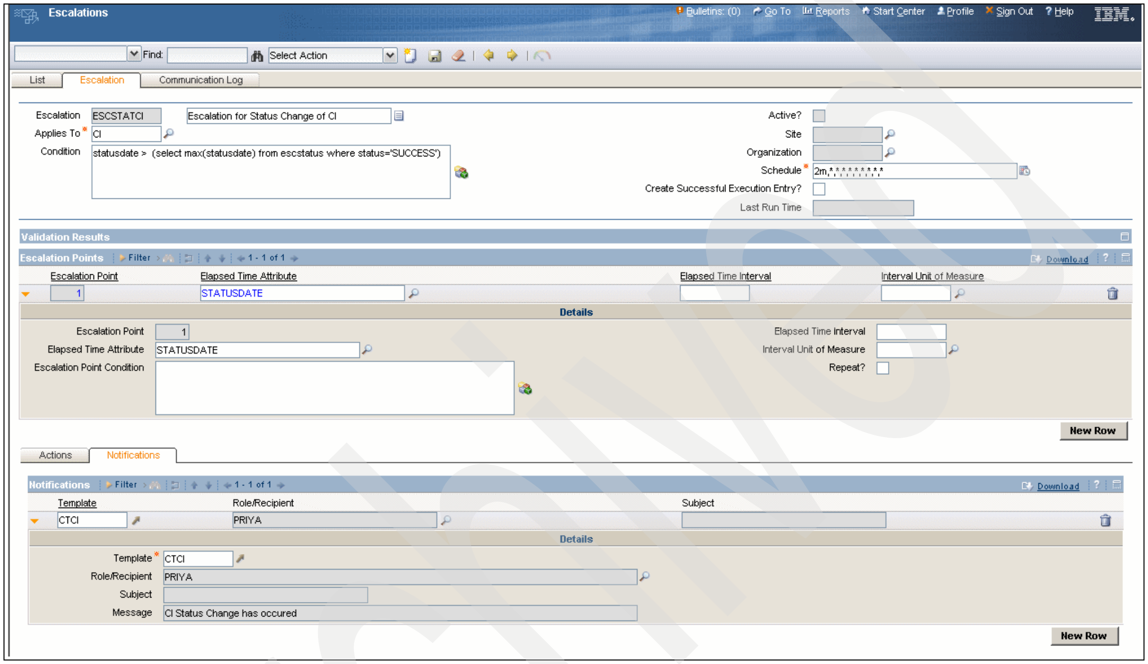


Figure 2-27 Escalations

Security

A common security model is available to support the organization and user structures. These so called security groups enable you to assign authorizations to users, allowing them to access specific applications based on roles, but also allowing them to access the data for their organizational entity.

You grant users security rights by assigning them membership in one or multiple groups. The combination of groups to which a user belongs determines an individual user's security privileges.

When you first implement the system, the Security Groups application has four groups (See Figure 2-28):

- DEFLTREG: Allows a user to change his or her password if it expires. It contains no other rights. When you insert a new Users record, the system places the user in this default group, although you can specify a different group to be the default using the Security Controls dialog box.
- MAXADMIN: Provides enough access to the system to add users and groups.
- MAXREG: A group that allows users to self register. You can use MAXREG to initiate a workflow process by which the system alerts an administrator to assign new users to the appropriate security groups.
- EVERYONE: Used for global settings that apply to all users in the system.

  You must create additional groups, with different sets of rights, to be able to assign users different sets of privileges.

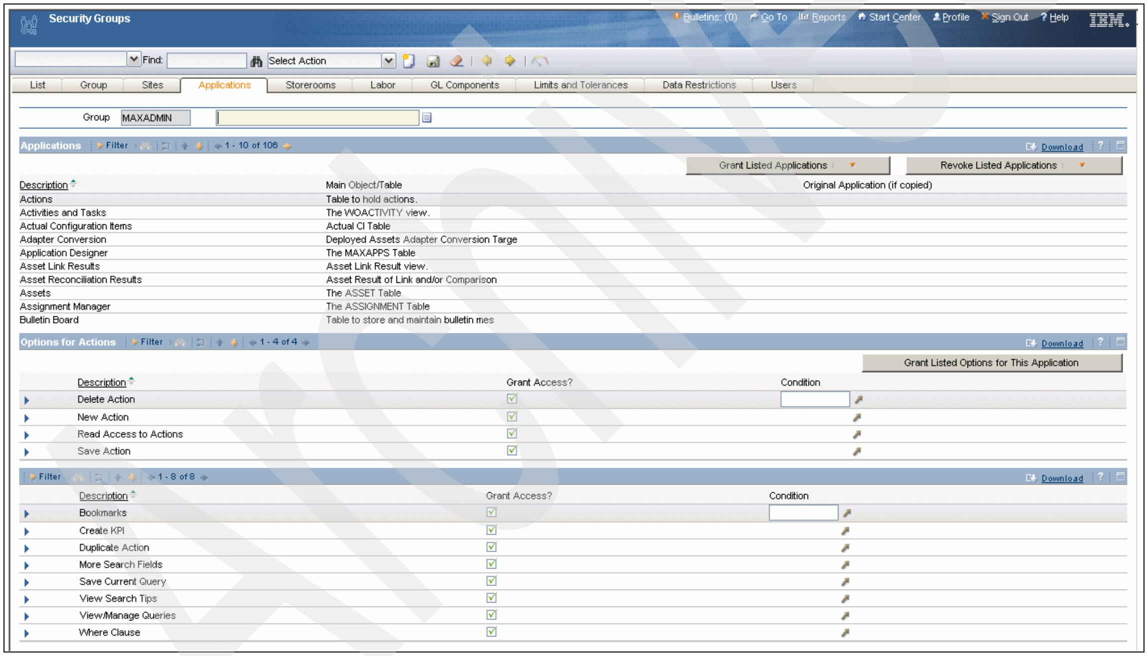


Figure 2-28 Security Groups

Note: Tivoli's process automation engine security framework fully leverages the J2EE application server security implementation for access and authentication. J2EE security features are based on well-established and proven industry-standard security features that better support the security requirements of today's complex Web-based computing environments. For example, enterprises can leverage corporate firewalls and implement digital certificates and encryptions in securing transmission security of Tivoli's process automation engine data. They can secure the data in the database with the triple DES (Data Encryption Standard) algorithm (DESede). They can simplify and streamline user access and authentication by centralizing access to all their applications through Lightweight Directory Access Protocol (LDAP) and Single Sign-On (SSO).

2.3.4 Tivoli Service Request Manager product

As previously explained, the Tivoli Service Request Manager package contains Service Desk functionality, as well as the Service Catalog, which you can install as separate solutions on the same process automation engine instance.

Process Manager applications

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

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

Service Desk

The Service Desk contains several Process Management applications and supporting applications to manage execution of three Service Management processes, such as:

- Service Request
- Incident
- Problem
- Service Request
- Solutions
- Search
- Customer Survey
- Integration Toolkit

Service Request

Service Request Management provides an application to support the Service Request process. A Service Request can be created to resolve an issue, obtain new service, obtain information, or change a current service. Either a service desk agent or a customer can create a service request.

An agent creates a service request record to track all contacts from a requestor, capture information from the requestor and determine what, if any, further action is needed. A requestor can either contact the service desk agent or create a service request through e-mail or other form of communication. A requestor also can create a self-service request through the Create Service Requests application. The agent views these requests in the Service Requests application and either resolves them or delegates them to another party for resolution.

Service request records are a type of ticket. Other ticket types are incidents and problems. The ticket applications are closely related and share many features, including the ability to define relationships between tickets, link them together for information purposes, and view the linkages and details in the appropriate applications.

Figure 2-29 shows the Service Requests window.

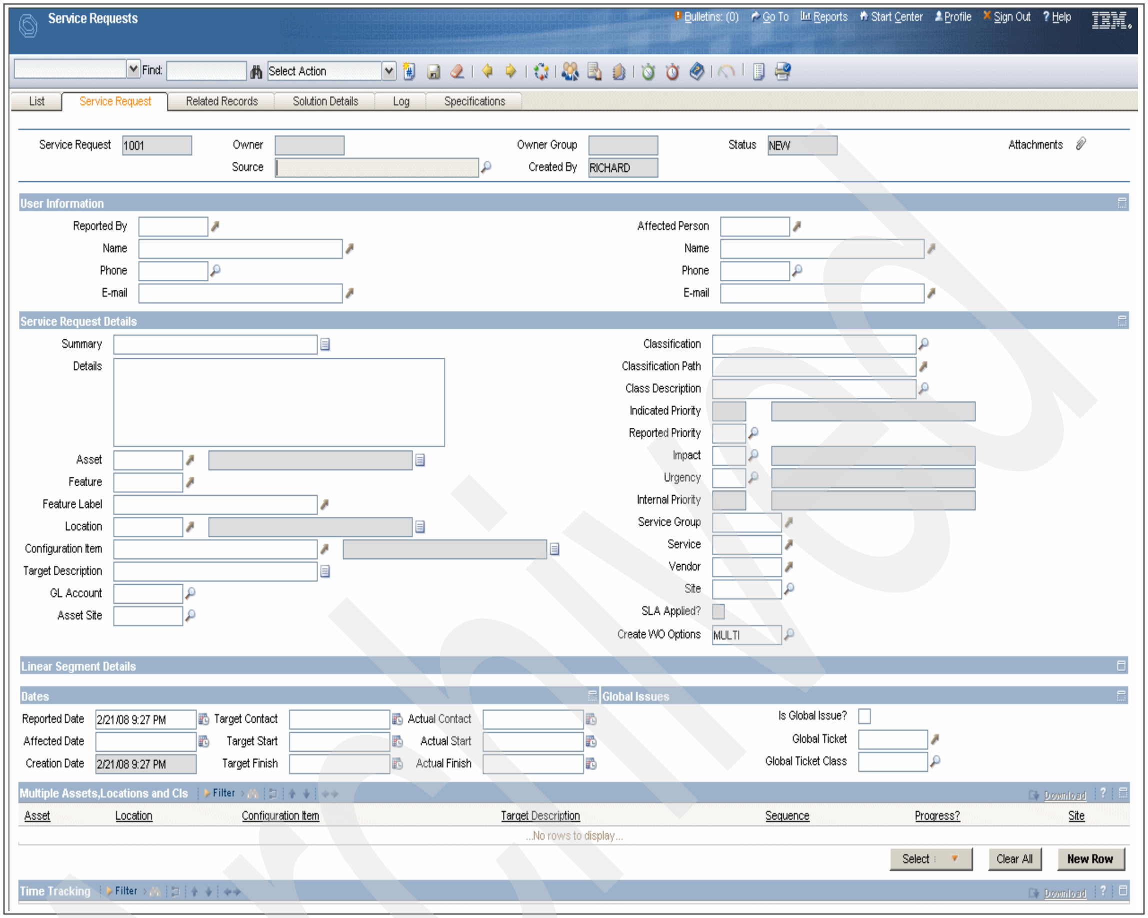


Figure 2-29 Service Requests

Incident

Incident Management provides you the Incidents application to support the Incident Management Process of your organization, based on ITIL best practices. You can create incident records to capture information about an event that deviates from standard service or an event that might disrupt the quality of that service. Users might or might not be aware of the event. The Incidents application enables you to create and modify incident records.

Figure 2-30 shows the Incidents window.

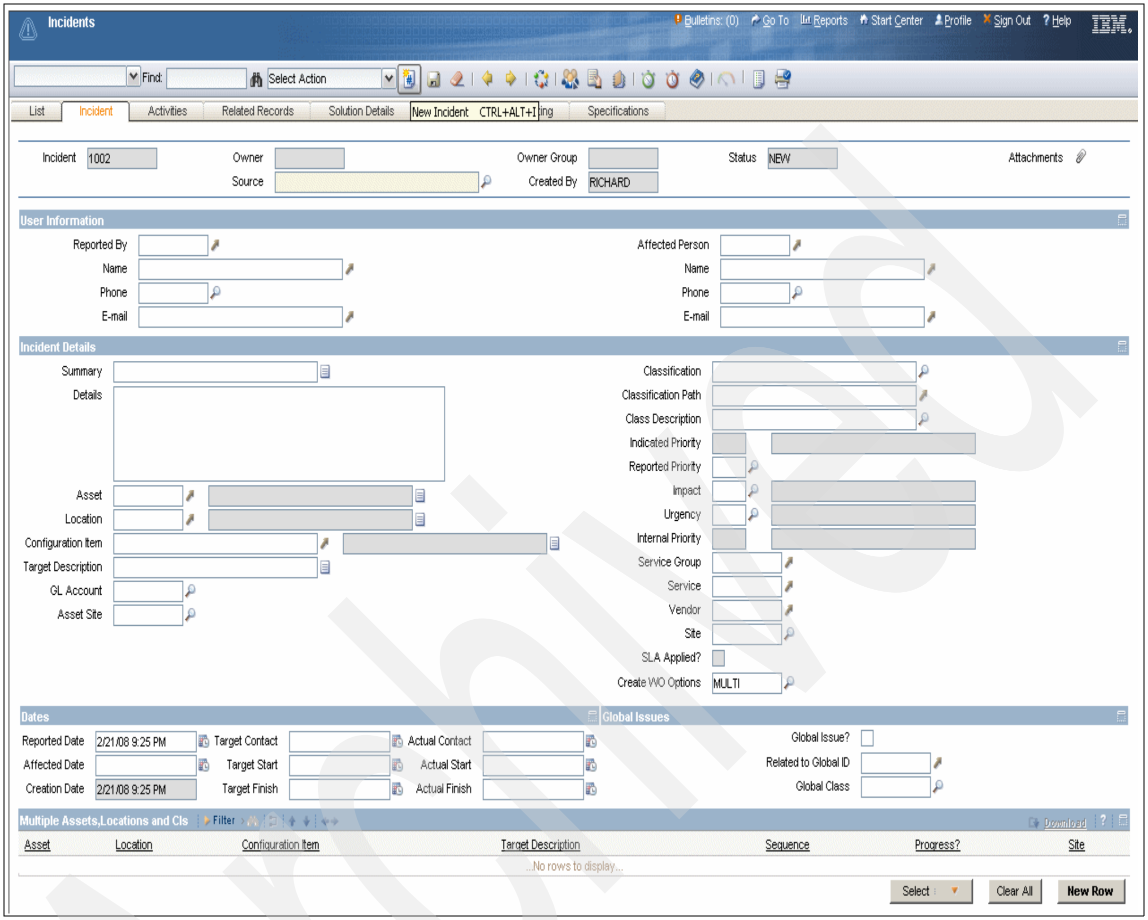


Figure 2-30 Incidents

To support the fast and easy creation of high-volume incident records, you can use predefined incident templates (Figure 2-31). An applied template can specify information such as internal priority, classification, owner, service, vendor, and activities.

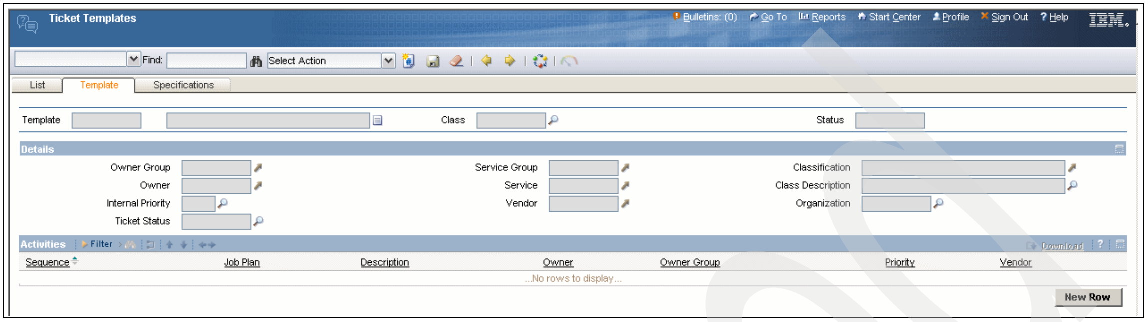


Figure 2-31 Ticket Templates

Applying templates can significantly reduce the amount of time needed to create incident records, since you can insert information simply by applying the correct template. Application of templates is a flexible process, since you can modify applied information until the incident is in a resolved or closed status.

The Incidents application also provides predefined communication templates that you can use when sending a communication to other users. Communications that use predefined templates are automatically sent to the Reported By and Affected Person users shown on the incident record when certain conditions occur, such as a change in the status of the incident.

An important part of managing the volume of service desk tickets is to define a priority for each ticket. Ticket priorities, as determined by service desk analysts, drive the order in which the tickets are handled and the allocation of resources for resolving tickets. In the ticket applications (Service Requests, Incidents, and Problems), this priority is referred to as the *internal priority*, a numerical value that is entered in the Internal Priority field on the ticket record.

The Priority Matrix application enables you to create a priority matrix that defines, in advance, internal priorities for Service Desk tickets that specify given combinations of impact and urgency. After a Service Desk agent fills in the Impact and Urgency fields on a ticket record, the Internal Priority field is automatically filled based on the values in the priority matrix. The internal priority values on ticket records help to determine the order in which tickets are handled.

A priority matrix is based on the ITIL concept that impact and urgency are the primary determinants of the relative priority in which a series of items, such as Service Desk tickets, must be addressed.

Problem

Problem Management is represented by the Problems application, which you can find in the Service Desk module (Figure 2-32). The Problem Management application will support the Problem Management process, based on ITIL best practices.

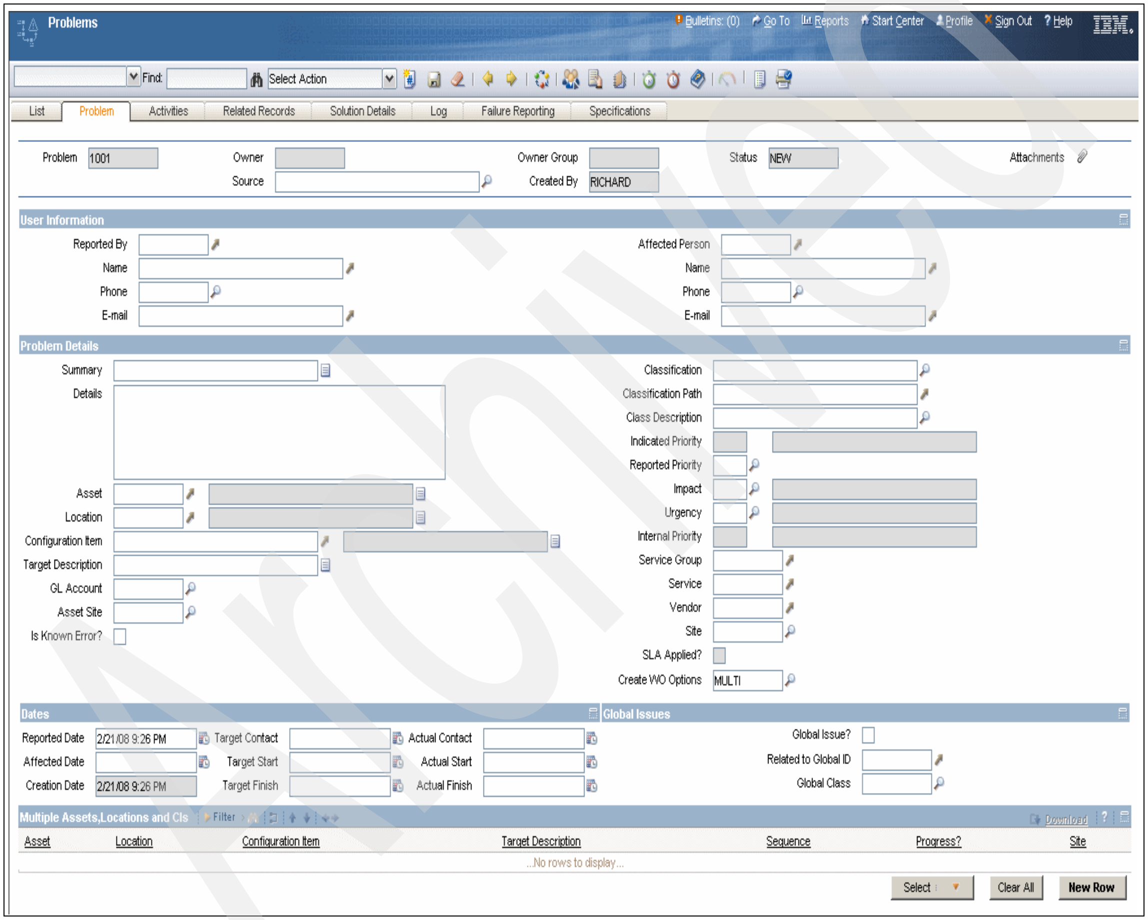


Figure 2-32 Problems

You can use the Problems application to create and modify problem records. You create a problem record to capture an unknown, underlying cause of one or more incidents. You resolve a problem when you identify its root cause so that similar incidents in the future are prevented or have a lesser business impact.

Solutions

A solution is a predefined response to a commonly asked question or problem. A solution record defines a symptom, a cause, and a resolution. You can use the Solutions application to create and manage solution records within the Service Desk environment.

The Solutions application is the administrative application, separate from the Search Solutions application that customers can use to find solutions and resolve problems on their own. When you create a solution, you can specify whether the solution record is available to customers from the Search Solutions application by selecting the **Self-Service Access?** option.

Solution records can also be associated with a service request, incident, or problem ticket. You must set the status of a solution to ACTIVE to make it accessible from other applications. See Figure 2-33.

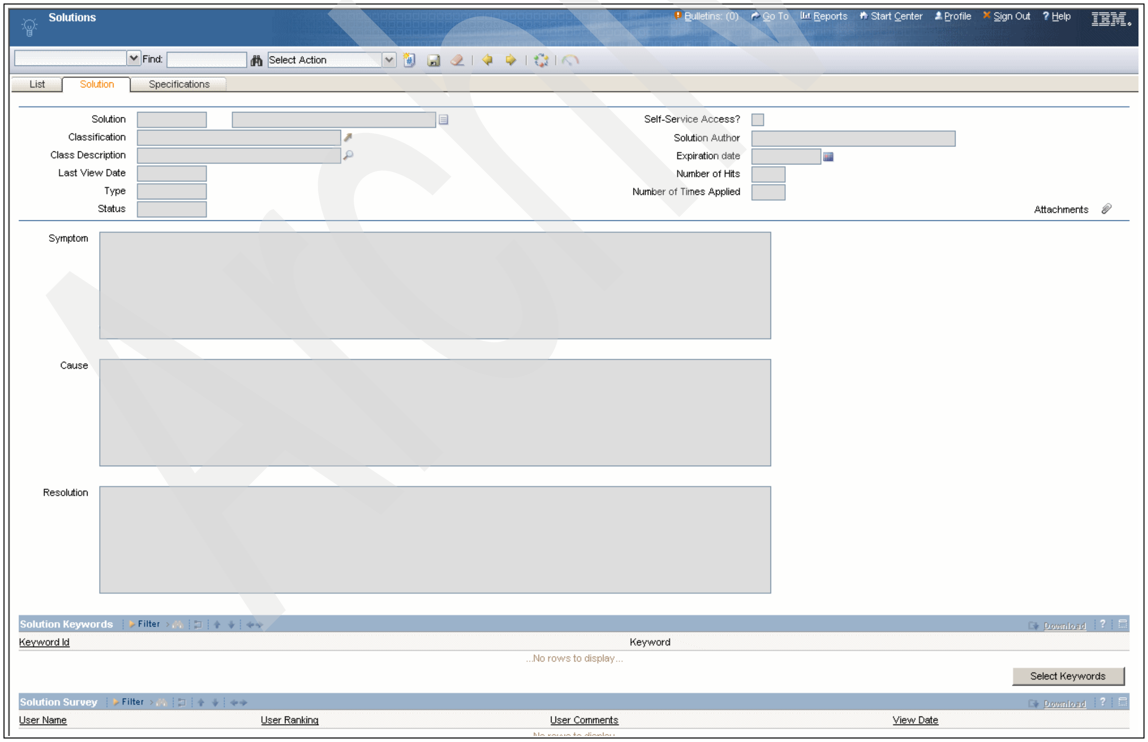


Figure 2-33 Solutions

Search

Search functionality is not new to Tivoli Service Request Manager. Many applications in previous versions of Tivoli Service Desk have a list tab, enabling you to search for records in an application. What previous versions of Tivoli Service Desk did not have was search functionality to search the entire database for relevant information, which is now possible with the new Global Search application.

Global Search is a very helpful and powerful new feature in Tivoli Service Request Manager V7.1. It provides intelligent search functionality to a Service Desk Analyst or reduced search functionality to a self service user, based on the open source Lucene information retrieval library.

You can find the Global Search application in the Service Desk module of the GoTo menu, as shown in Figure 2-34.

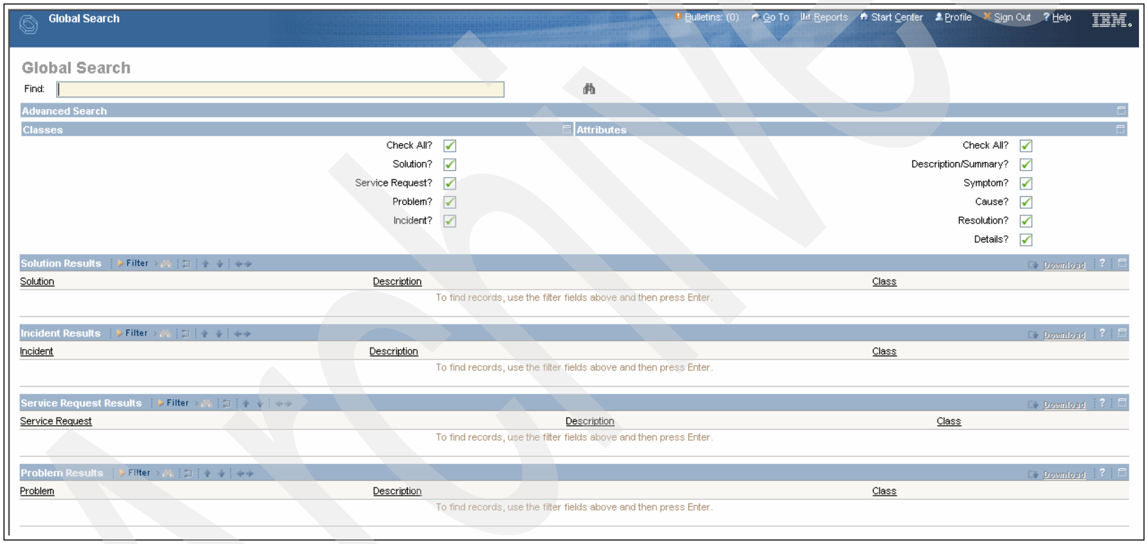


Figure 2-34 Global Search

You will also find the Service Request Manager Search in the Self Service module, which provides customers a search engine with reduced functionality and access to search for relevant records they have access to, as shown in Figure 2-35.

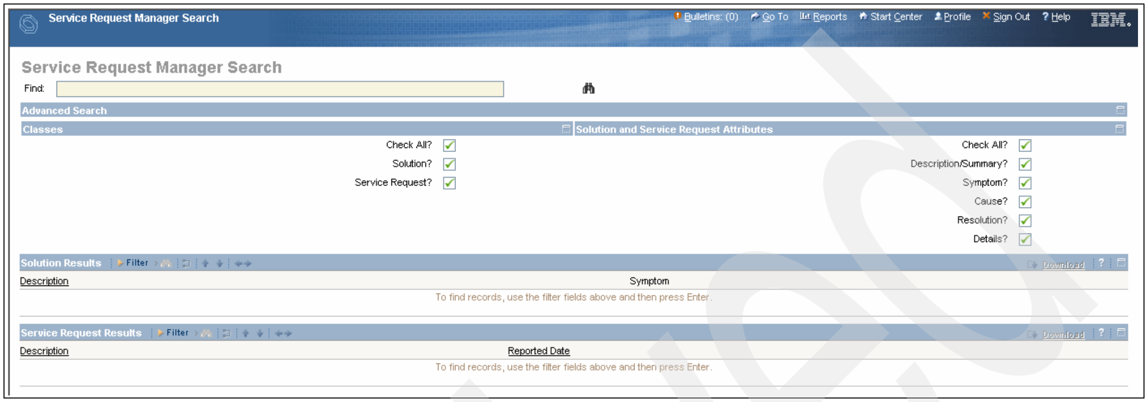


Figure 2-35 Service Request Manager Search

In 5.2, "Survey" on page 311, we will describe in more detail the Global Search function in Tivoli Service Request Manager V7.1.

Customer Survey

Survey functionality addresses the Service Desk organizations need to gather ratings for the quality of service provided by the Service Desk and have a better understanding of how customers perceive the Service Desk's customer service. Customer Satisfaction Surveys are used to gather these qualitative evaluations which, in conjunction with a Service Desk's performance metrics, enhance management's ability to have an overall idea of Service Desk performance.

Figure 2-36 on page 72 shows the Survey List window. Refer to 5.2, "Survey" on page 311 for a detailed discussion about Customer Survey.

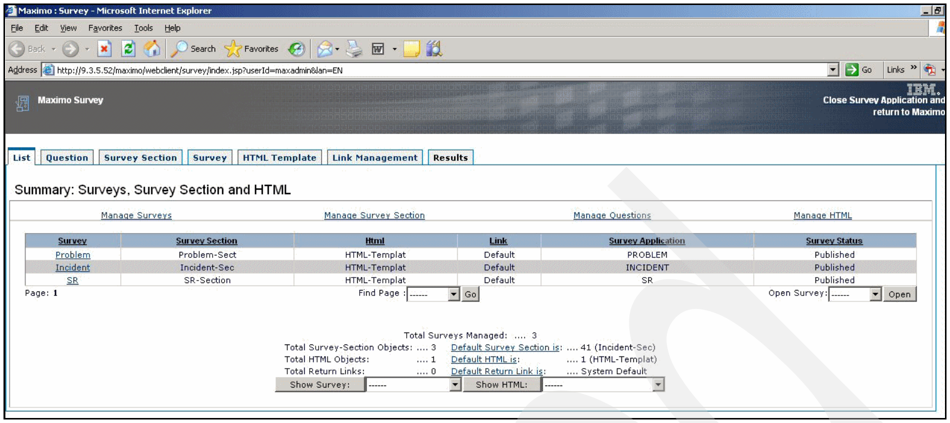


Figure 2-36 Survey List window

Integration Toolkit

There are further applications that are provided within Tivoli Service Request Manager that we do not explain in detail, for example, the foundation to interconnect a Process Management application to an Operational Management Product (OMP) such as Tivoli Provisioning Manager or Tivoli Configuration Manager in order to initiate an action from within a step in a process into the OMP. An application can initiate a software distribution action inside the Tivoli Provisioning Manager as the OMP. Tivoli's process automation engine provides the technology for the integration between the Process Manager application and OMP layer. It is not providing concrete implementations of integrations, but only provides the foundation.

For example, the Integration Modules, the Logical Management Operation, or the Launch in Context applications inside the Integration Module Menu are provided by Tivoli's process automation engine.

Figure 2-26 on page 60 shows the integration module menus.

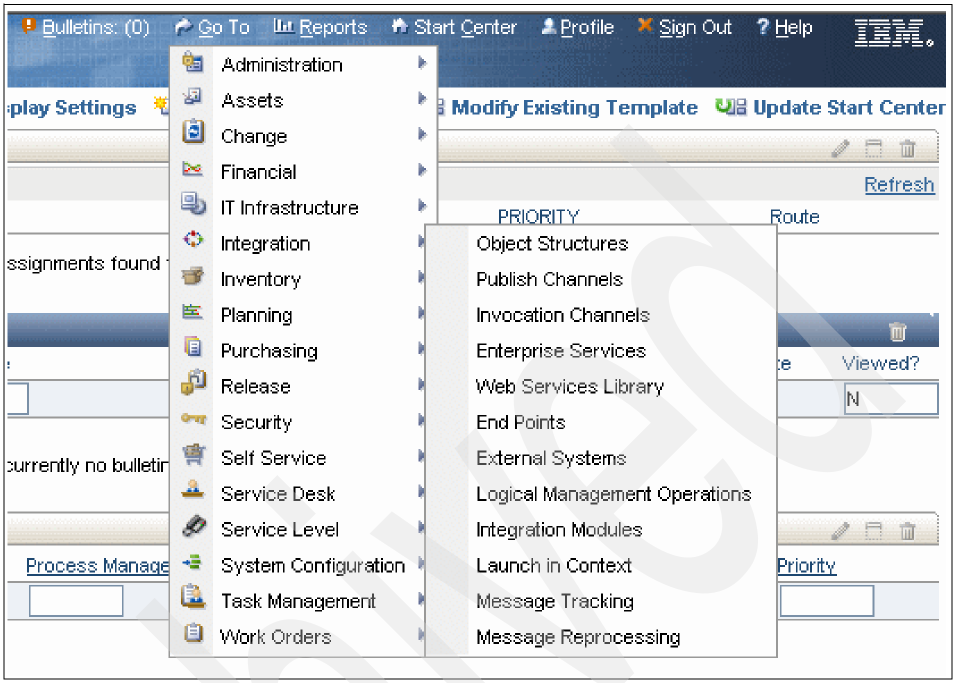


Figure 2-37 Integration module menus

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

Service Catalog

Service Catalog is a new and separate application just like Service Desk is. You can install both solutions separately, although Service Catalog is dependant on Service Desk. Because of the new functionality and complexity, the Service Catalog will be discussed in a separate IBM Redbooks publication, *Implementing IBM Tivoli Service Request Manager Service Catalog*, SG24-7613.

2.4 Physical components

In this chapter, we so far have described the components of the Tivoli Service Request Manager V7.1 solution from a functional perspective. Our focus has been on the logical components, such as Tivoli's process automation engine, the Process Manager Products, Data Access Services, or the Data Layer in general rather then the physical runtime environment.

In this section, we now describe the runtime environment itself is used to host the logical components that we previously described.

First, we describe the major building blocks of the physical runtime environment, their role in the overall solution, and their relationships to each other.

Second, we describe the environment from the perspective of an operational model. We explain the protocols used for component interaction and different options for deploying the solution.

Figure 2-38 highlights the components of the Tivoli Service Request Manager V7.1 runtime environment.

Note: Although the components are drawn in separate boxes, they do not necessarily have to be deployed on different physical systems.

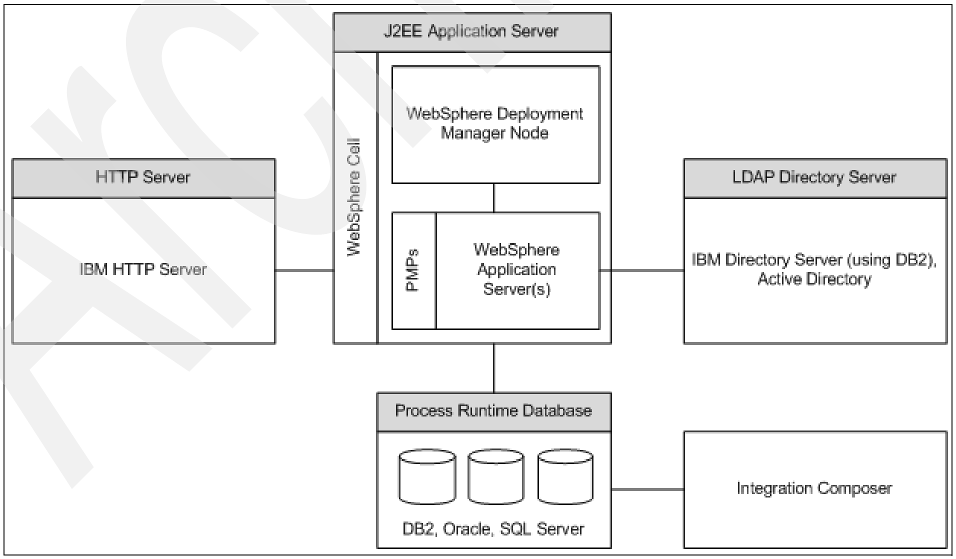


Figure 2-38 Physical component and relationship overview

The HTTP Server, the WebSphere Application Server, the Tivoli Service Request Manager Process Runtime Database, and the LDAP Directory Server comprises the environment to host the Process Manager Products. In other words, this is the environment that is needed to run service management process implementations.

To be able to use the full capabilities of a Service Management implementation with Tivoli Service Request Manager, you can integrate Tivoli Service Request Manager V7.1 with CCMDB V7.1, which will give you a robust CMDB with change and configuration management capabilities, including service desk processes to support ticket creation on each of the configuration items. This chapter will only discuss the components used for the most common Tivoli Service Request Manager deployments, and CCMDB will not be covered. For more information about IBM CCMDB V7.1, refer to the CCMDB documentation, or to the IBM Redbooks publication *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565.

2.4.1 J2EE server configurations

The most difficult part of any Maximo installation has always been the J2EE server configuration. With Tivoli Service Request Manager V7.1, only IBM WebSphere is currently supported, but BEA Weblogic support is expected 90 days after general availability. You will typically need the IBM WebSphere Application Server Network Deployment and IBM HTTP Server for WebSphere Application Server products for Tivoli Service Request Manager to work.

The Tivoli Service Request Manager installation provides you an automated install and configuration of all middleware components, which is the recommended option for installation. Manual configuration of each component is still possible, but not preferred, due to the large number of parameters that you need to configure. In the following chapters, we will discuss best practices for deployment that can help you plan, install, and configure a solution that will meet your requirements.

We discuss the security architecture, of which the J2EE server is a key component, in more detail in 2.5, “Tivoli Service Request Manager security architecture” on page 76.

2.4.2 Database server configurations

Tivoli Service Request Manager can be installed using one of three databases: Oracle, IBM DB2 Universal Database™, or Microsoft SQL server. The middleware installation provides a DB2 Enterprise Server Edition installation, where Oracle or SQL server have to be installed separately. The Tivoli Service

Request Manager installation procedure does provide the necessary tools to configure any of the databases mentioned automatically, as long as the appropriate credentials and required information is provided.

2.4.3 Directory Server configurations

Lightweight Directory Access Protocol (LDAP) is an application protocol for querying and modifying directory services running over TCP/IP. Due to its basic design, LDAP is often used by other services for authentication, but can be used for other purposes where a set of objects with similar attributes has to be organized in a logical and hierarchical manner. Tivoli Directory Server is just one way to implement a Tivoli Service Request Manager environment, but Microsoft Active Directory® is also supported. Using LDAP in a Tivoli Service Request Manager deployment is optional, but can provide benefits to any organization.

We discuss the security architecture, of which the Directory Server is a key component, in more detail in 2.5, "Tivoli Service Request Manager security architecture" on page 76. We will discuss in more detail the installation and configuration steps for the Tivoli Directory Server and alternatives.

2.5 Tivoli Service Request Manager security architecture

In this chapter, we explain the security architecture of the Tivoli Service Request Manager V7.1 solution in some detail. As already explained, the Tivoli Service Request Manager solution is a concept that consists of multiple physical and logical components. In order to support a sophisticated user experience, a common security model has been implemented.

In this section, we primarily focus on components of the solution that are involved in the common security model for authentication, authorization, and Single Sign-On (SSO). Our focus is not on other aspects of security, such as transport layer security or encryption standards, being used in the Tivoli Service Request Manager solution.

We provide insight in key areas of the system that are either installed and customized by default or need to be adjusted to your organizational needs. We provide example windows in order to ease your understanding of what you need to customize.

The major Tivoli Service Request Manager component with respect to security is the process environment hosted inside the J2EE WebSphere Application Server runtime environment. We consider the process environment the leading component of the solution, since most of the users of Tivoli Service Request Manager will be users of the service management processes.

2.5.1 Authentication models

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

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

Virtual Member Manager

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

Tivoli Service Request Manager V7.1 uses the Virtual Member Manager technology to shield the process runtime environment from the LDAP provider being used. All interactions between the applications and the directory server flow through the Virtual Member Manager. Its common interface masks the differences of the LDAP provider implementation, being Tivoli Directory Server, Active Directory Server or different implementations in the future.

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

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

There can be multiple directory server implementations at runtime that get virtualized by the VMM technology. In our environment, we use one Tivoli Directory Server instance to hold user and group data. We used the default settings at installation time. In this case, you can see in the WebSphere Application Server Admin console's security definition that a realm called ISMRealm has been defined. In addition, you can see that the definition of the base entry has been defined as ou=SWG,o=IBM,c=US. This is the root entry that is the starting point in the repository for our user and group data.

Figure 2-39 shows the ISMRealm definition.

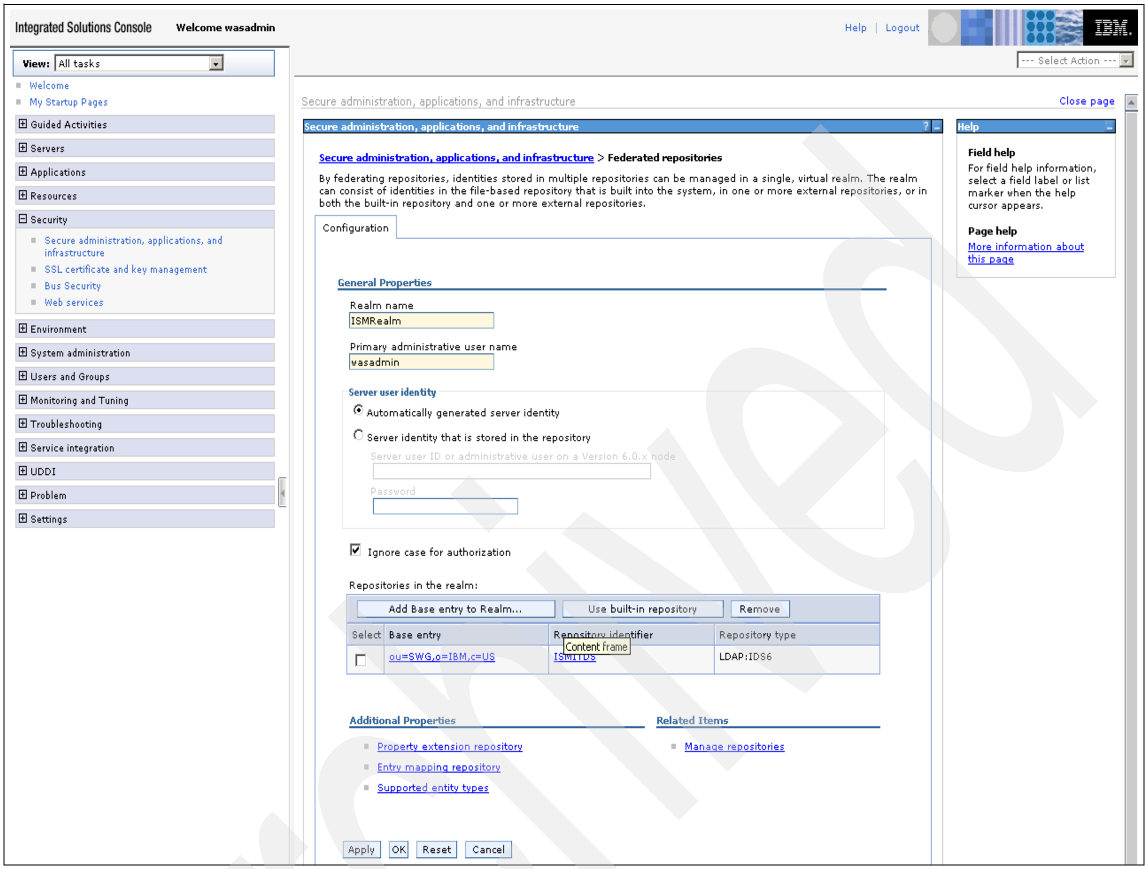


Figure 2-39 ISMRealm definition

The identifier is set to ISMITDS by default.

Once user and group data has been entered into the Directory Server, they are synchronized into the Tivoli Service Request Manager process runtime database. The synchronization between LDAP and the process environment is one-way. It is controlled by a cron task that is defined at Tivoli Service Request Manager installation time. The name of the cron task is *VMMSYNC*.

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

The default definition for the VMMSYNC cron task definition is shown in Figure 2-40 on page 80.

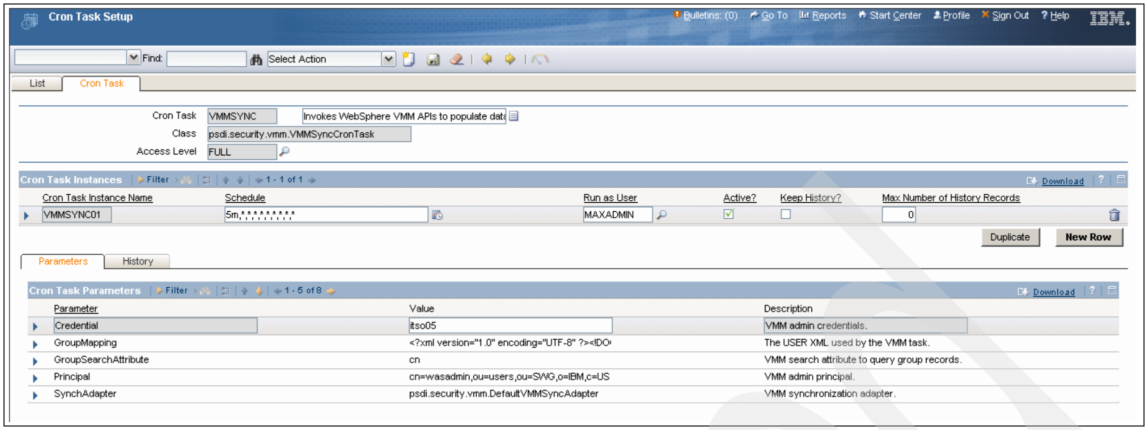


Figure 2-40 VMMSYNC Cron Task definition

VMM is the only supported option in Tivoli Service Request Manager V7.1 for setting up a connection to the Directory Server. Although technically possible, it is not supported in Version 7.1 to connect directly to LDAP without going through the VMM interface.

Secure Token Services

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

Tivoli Service Request Manager V7.1 uses the Lightweight Third Party Authentication (LTPA) mechanism of the WebSphere Application Server security. LTPA usually is used in an interoperability context when multiple applications need to share a common security token that identifies a user so that the user does not have to log in into multiple applications but rather is provided a single sign-on experience. The instance of the WebSphere Application Server security model that generates and validates LTPA tokens is called the Secure Token Service, sometimes also referred to as the *Embedded Security Service (ESS)*.

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

The Secure Token Service is the second major security facility that Tivoli Service Request Manager leverages from the WebSphere Application Server security model in order to pass a security token from the process environment into any other environment when a user tries to launch-in-context from a Process Manager Product.

2.5.2 Authorization model

In 2.5.1, “Authentication models” on page 77, we explained the authentication model of the Tivoli Service Request Manager solution, which relies on WebSphere security components and LDAP directory server implementations. The directory server manages the users and groups, as well as their relationships.

Authorizations to Tivoli Service Request Manager applications (for example Incident and Problem Management applications) are managed in the Security Groups application. In the Security Groups application, you can define what a user, as a member of a group, can access and execute in the system. You can provide access and set restrictions to organizations, sites, applications, and specific data restrictions, as well as defining limits and tolerances for financial transactions.

Users and groups are the key entities involved in determining the authorization or security profile of a user. As mentioned, users and groups, including their relationships, are by default created and maintained in an LDAP Directory Server. They are synchronized into the Tivoli Service Request Manager process runtime database using the VMM cron task.

The User application is used to maintain Tivoli Service Request Manager specific attributes of user records that are not maintained in LDAP. That means you can add user specification attributes in the database with additional data that has not been synchronized through the VMM cron task from the LDAP Directory server.

A user is a member of one or more Security Groups. When a user tries to access an application, security checks are performed in order to see what the maximum access is based on the combining of the group memberships the user has. Figure 2-41 on page 82 shows a calculated security profile, based on the combination of Security Groups the user belongs to.

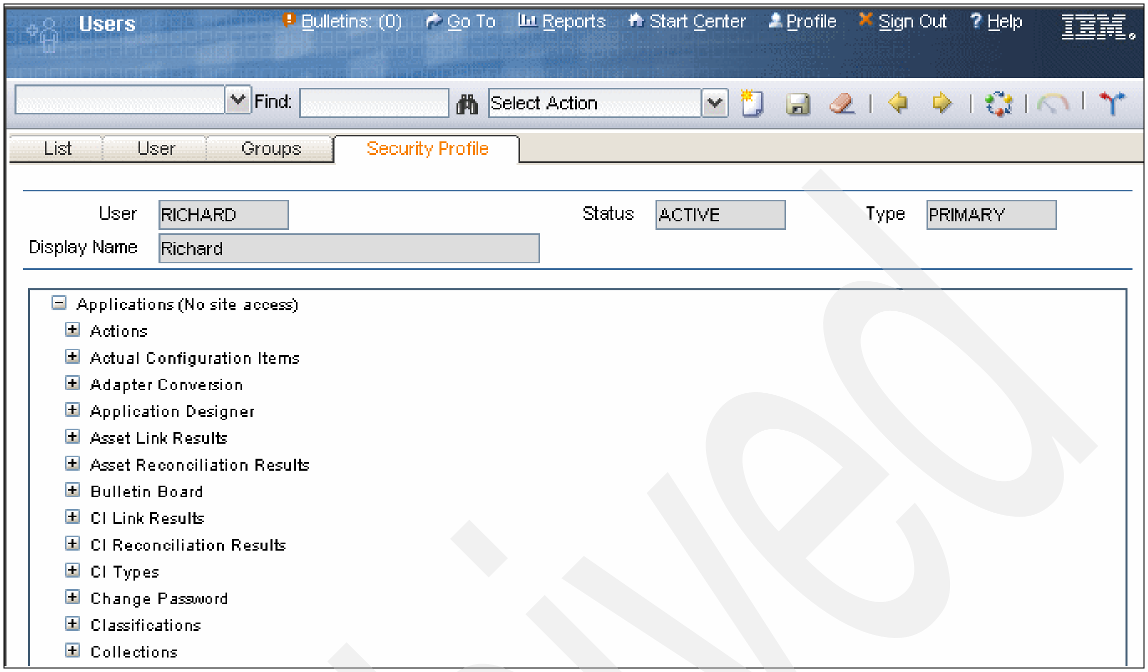


Figure 2-41 Users

You define permissions in the Security Groups application. Permissions are never defined for a user directly, but always to groups that a user needs to belong to in order to get access to specific objects and data. A user only inherits the permissions from the groups he belongs to, depending on the configuration of these Security Groups.

In Figure 2-42, you can see several tabs that allow you to define different types of permissions. We do not explain all the possibilities in detail, but just highlight the most important ones.

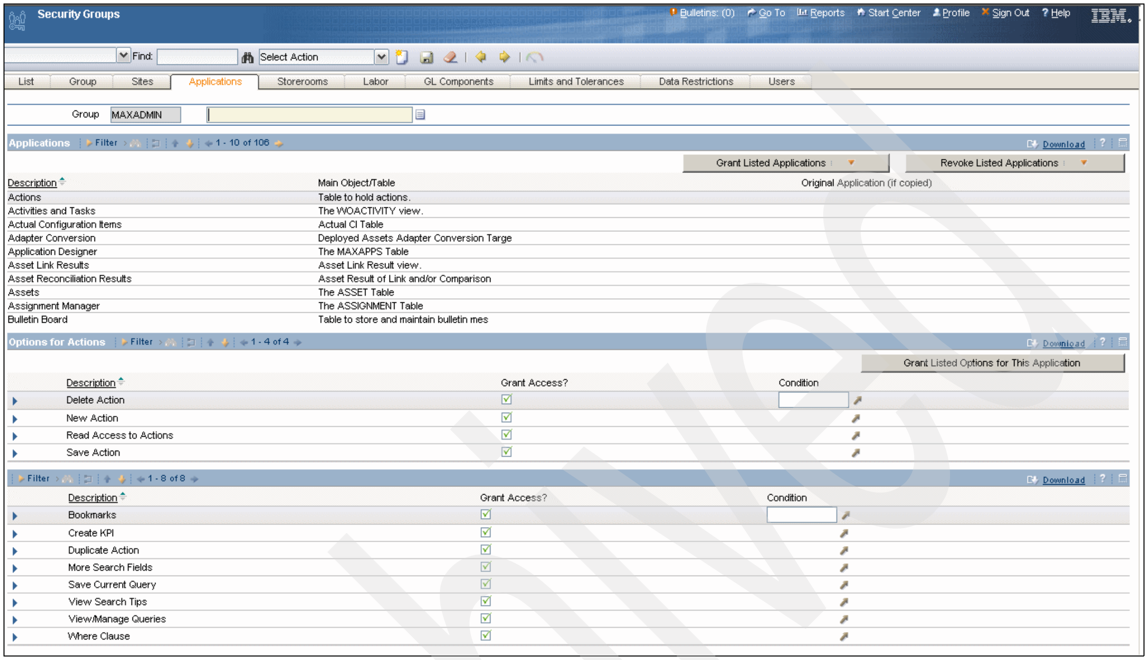


Figure 2-42 Security Groups

The *Applications* tab you see allows you to specify which applications in the process environment a user should be able to access. You see different applications listed in Figure 2-42, which are all part of the PMPs installed.

In addition to the specification of which application a user can get access to, you specify which type of access a user is allowed for the specific application. Furthermore, you can select options that are usually accessible in the menu bar or Action Menu of the appropriate application.

The *Sites* tab allows you to specify if a user has access to data from all Sites or a specified Site. A Site is an entity in the database to partition the data for organizational purposes. An organization could be the IT department of your company while a site could be the Austin location in Texas. A Site always belongs to one organization, while organizations consist of one or more sites.

You can also specify which Start Center a user will see for each specific Security Group when he logs into the system. In Figure 2-43 we defined Start Center template number 3, which is a template for an Incident Manager role.

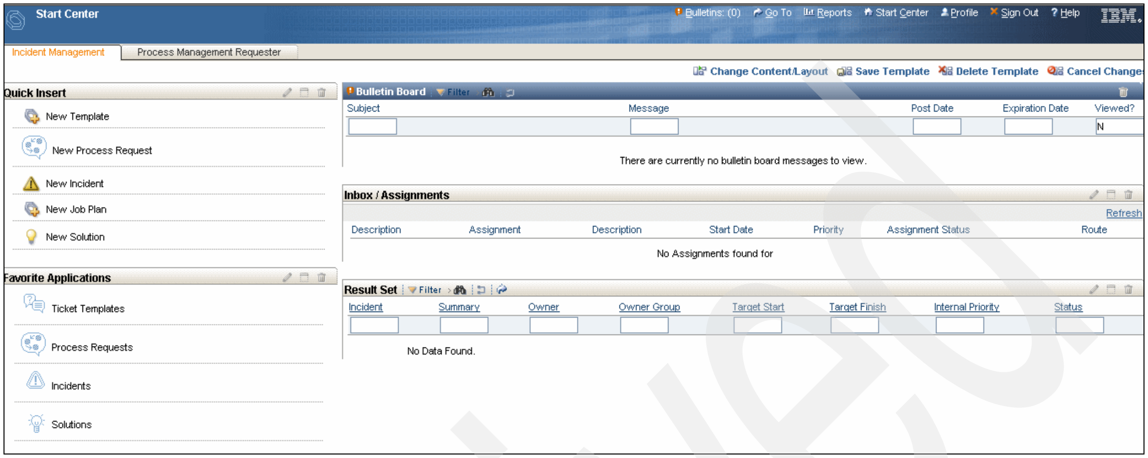


Figure 2-43 Start Center

If a user is assigned to more then one Security Group and these groups all have their own Start Center template defined, the user will see a tab for each Security Group containing a template.

2.6 Integration technologies

IBM Tivoli Service Request Manager V7.1 provides various interface technologies that can be used to exchange data between solutions for different purposes. These technologies are bundled in what is called the integration toolkit. The integration toolkit is build on a number of applications, covered in the Integration Module.

By default, a number of integrations are supported and provided as pre-configured installation packages. You can use these integrations to connect Tivoli Service Request Manager to third-party tooling very easy, enabling data exchange between the solutions. Introducing Tivoli Directory Integrator as an integration hub enables you to connect Tivoli Service Request Manager with many products, using a single point of configuration and intuitive User Interface.

Further information about Tivoli Service Request Manager integrations with CCMDB will be covered in the IBM Redbooks publication *Integration Guide for IBM Tivoli Service Request Manager V7.1*, SG24-7580.

Part 2

Planning and installation

In this part, we provide information related to the actual installation of the IBM Tivoli Service Request Manager product components, including information related to hardware and software requirements and other planning considerations.

3

Planning for installation

This chapter provides a guide for the planning an installation of the Tivoli Service Request Manager along with its components. You should consider scenarios, topologies, hardware, and software that must be analyzed before starting the installation.

Pre-installation planning is critical for a successful installation of Tivoli Service Request Manager.

In this chapter, the following topics are discussed and described:

3.1 Installation overview

The installation of Tivoli Service Request Manager V7.1 consists of installing the following items:

- Middleware
- Base Services
- Service Desk
- Service Catalog (optional)
- Survey application (optional)
- Integration Software (optional)

> **Note:** You can find the prerequisite middleware, middleware fix packs, base services, and the Service Request Manager on a set of DVDs that comes with the product.

3.1.1 DVD layout

IBM Tivoli Service Request Manager ships on a set of DVDs that contain the prerequisite middleware, middleware fix packs, base services, and the Service Request Manager product code.

The following DVDs contain the files for the Service Request Manager V7.1 features:

- Service Request Manager V7.1

  Contains the middleware installer, the base services installer, and the Service Request Manager installer and launchpad, as well as other functions including IBM Agent Controller installer, ITUPC installer, Log and Trace Analyzer-WebSphere Process Server context tool, and the Base Services Language Pack installation program for Windows® 2003.

- Middleware Images

  A middleware DVD is provided for each supported operating system:

  - Middleware for Windows 2003

    Contains the prerequisite middleware software for Windows 2003.

  - Middleware for Red Hat Enterprise Linux® V4

    Contains the prerequisite middleware software for Red Hat Enterprise Linux Version 4.

– Middleware for AIX® 5L™ V5.3

Contains prerequisite middleware software for AIX 5L Version 5.3.

3.2 Prerequisites for installation

This section provides information about the components required to install Service Request Manager V7.1. We divided this section as follows:

- Components
- Hardware and software prerequisites

3.2.1 Components

The components required to run Service Request Manager consists of the following:

- Middleware
- Tivoli's process automation engine (also known as Base Services)

Bellow, we detail each of the components required to run a Service Request Manager application. Once the installation of those components is complete, you will be able to start the Service Request Manager installation.

The Service Request Manager installation, middleware installation, and base services installation programs install and configure IBM products that make up the Service Request Manager architecture. For some IBM products already deployed in the enterprise, provided that they meet minimum release level requirements, the Service Request Manager installation programs can configure them to be part of the Service Request Manager deployment. In some cases, the Service Request Manager installation programs can even configure previously deployed products from other vendors to be used with Service Request Manager. For more information about middleware configuration, refer to "Base services" on page 92.

Middleware

Tivoli Service Request Manager requires several middleware products to be installed in your environment prior to deployment.

> **Note:** The installation needs to create a workspace folder for deployment; the process will check automatically the available space on disc.

The workspace contains the following items:

- Deployment plan

  The deployment plan is a collection of installation steps, configuration parameters for those steps, and target machine information. It is generated through the Tivoli middleware installer and it resides in the workspace directory.

  When deployment steps are changed, the existing deployment plan is deleted and replaced with the new deployment plan.

  After you have successfully configured the plan, the middleware associated with your deployment choices will be installed.

- Topology file

  The topology file is a properties file that describes the configuration parameters of the Tivoli Service Request Manager middleware deployment. This file is created and then updated after every deployment or undeployment. If you have not defined a workspace that is centrally located and accessible to all the systems that will be receiving Tivoli Service Request Manager middleware, this file will have to be copied to the workspace of each machine where Tivoli Service Request Manager middleware is being deployed. The contents of this file can be used by the Tivoli Service Request Manager installation program to populate its panels with meaningful default values.

- Logs

  Log files that contain information about the deployment can be found in the workspace directory. In addition, log files native to the Tivoli Service Request Manager middleware itself are also contained in this directory.

Once you have the middleware installed, you have two options of configuring the servers for use with Service Request Manager. You will be presented with the option of either allowing the base services installation program to automatically configure middleware, or, you can elect to manually configure each middleware component, which must be done prior to running the base services installation program.

Auto-configure

The base services installation program will automatically configure the middleware to work together with base services. This option is recommended if you are installing new instances of middleware components using the Tivoli middleware installation program, or if you have existing middleware instances that are not governed by policies that restrict programmatic configuration.

Manual

You can manually configure middleware that either already exists in your environment, or has been installed by the Tivoli middleware installer. This configuration must be completed prior to running the base services installation program. You might choose this deployment path if you have policies in place that dictate certain procedures and guidelines when configuring systems in your environment.

The middleware applications can be installed on Windows, AIX, and Linux operating systems. You may consider installing new middleware instances or reusing existing ones. The reuse of existing middleware is discussed in 3.5, "Reusing middleware" on page 104.

The middleware consists of the following products:

- IBM Rational Agent Controller Version V7.0.3
- Database server: DB2 Enterprise Server Edition Version V9.1.2
- J2EE server: WebSphere Application Server ND Version V6.1.0.9
- Directory server: IBM Tivoli Directory Server Version V6.1
- HPPT server: IBM HTTP Server Version V6.1

IBM Rational Agent Controller Version 7.0.3

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

Database server: DB2 Enterprise Server Edition Version 9.1.2

A database server is required to store information about configuration items and how they relate to each other. When using the launchpad interface, DB2 is the default database used by Tivoli Service Request Manager. You can optionally install and use SQL Server or Oracle databases.

J2EE Server - WebSphere Application Server Network Deployment Version 6.1.0.9

Tivoli Service Request Manager is built using Java 2 Enterprise Edition technology, which requires a commercial application server to host and run business applications such as Tivoli Service Request Manager. WebSphere Application Server Network Deployment is the default J2EE application server used by a Tivoli Service Request Manager installation. You can optionally install and use BEA Weblogic Application Server.

Directory server (optional)

A directory server is used to secure a Tivoli Service Request Manager application. IBM Directory Server is the default directory server used by Tivoli Service Request Manager installation. Optionally, you can choose to install MS Active Directory.

> **Note:** A directory server is required when you plan to install Service Request Manager integrated with Tivoli Change Configuration Management Database (CCMDB).
>
> A directory server is not required if you decide not to integrate with CCMDB. In this case, you can use Service Request Manager security.

Prior to deciding whether or not installing base services, refer to 3.3, “Planning to deploy Service Request Management” on page 96.

Base services

The base services provides basic infrastructure to deploy a Tivoli Service Request Manager application. The base services software is included with the Service Request Manager media and can be installed only on Windows operating systems designated as the *administrative workstation*.

The administrative workstation is used to complete tasks such as gathering information about your base services deployment and configuration, copying files to your local system, and performing configuration tasks using the values you have specified. It is also used to install program patches, product upgrades, or just running configdb[1] and similar utilities in which case you must stop the application server from running. The Enterprise Archive (EAR) files are also built using base services software and subsequently deployed to the application server.

Administrative workstation is not used for daily operations, although is an important Service Request Manager support component for administrative activities.

3.2.2 Hardware and software prerequisites

This section describes hardware and software requirements to install Tivoli Service Request Manager. Each product version listed reflects the minimum requirement for use with Service Request Manager. We discuss the hardware requirement for an IBM portfolio only. Refer to vendor specifications if you are planning not to use an IBM portfolio.

[1] Configure database

Hardware

We list hardware requirements for Service Request Manager in Table 3-1. For hardware requirements for software not listed below, refer to the product documentation provided with that product.

Table 3-1 Hardware requirements for an IBM portfolio

Software	Hardware
IBM DB2 UDB	Minimum 20 GB disk space
IBM WebSphere Network Deployment	2 GHz processor 20 GB disk space 2 GB RAM
IBM Tivoli Directory Server	For AIX and Linux, 1 GB of space available in the /opt directory

Note that if you are installing all middleware onto a single machine, the RAM requirement increases to 3 GB.

Software

Table 3-2 through Table 3-9 on page 95 lists supported software for Service Request Manager.

Table 3-2 Software requirements: browser

Software	Operating systems
Browser: Service Request Manager supports Internet Explorer® Version 6 and later.	Any operating system that supports IE V6 and later.

Table 3-3 Software requirements: database

Software	Operating systems
Database: The following products can serve as the database component of a Service Request Manager deployment: ► DB2 UDB V9.1 FP 2 (installed by the Tivoli middleware installer) or V8.2 FP 14 ► Oracle 10g Rel2 ► Microsoft SQL Server 2005, Standard or Enterprise version.	► Windows Server® 2003 SP2 ► Red Hat Enterprise Linux V4 (Intel®) ► Red Hat Enterprise Linux Advanced Server V4 (Intel) ► IBM AIX 5L V5.3 ML level 5300-06Microsoft SQL Server 2005, only Windows is supported. For DB2 on AIX and Linux systems, ensure you have a minimum of 8 GB (binary) free of space in the DB2 database instance home directory (/home/ctginst1) in order to meet the default table space disk space requirements of the DB2 install. For DB2 on Windows, ensure you have a minimum of 8 GB of free space in the DB2 installation directory.

Table 3-4 Software requirements: application server

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

Table 3-5 Software requirements: integration options

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

Table 3-6 Software requirements: HTTP server

Software	Operating system
HTTP server: The following product can serve as the HTTP server component of a Service Request Manager deployment: IBM HTTP Server V6.1 FP9	► Windows Server 2003 SP2 ► Red Hat Enterprise Linux V4 (Intel) ► Red Hat Enterprise Linux Advanced Server V4 (Intel) ► IBM AIX 5L V5.3 ML level 5300-06

Table 3-7 Software requirements: directory server

Software	Operating system
Directory server: The following products can serve as the directory server component of a Service Request Manager deployment: ► IBM Tivoli Directory Server V6.1 ► Microsoft Windows Server 2003 SP2 Active Directory Microsoft Active Directory Application Mode (ADAM) is not supported.	► Windows Server 2003 SP2 ► Red Hat Enterprise Linux V4 (Intel) ► Red Hat Enterprise Linux Advanced Server V4 (Intel) ► IBM AIX 5L V5.3 ML level 5300-06

Table 3-8 Software requirements: administrative system

Software	Operating system
Administrative system: The following product can serve as the administrative system component of a Service Request Manager deployment: Service Request Manager administrative system	Windows Server 2003 SP2 (32-bit)

Table 3-9 Software requirements: data migration

Software	Operating system
Data migration: The following product can serve as the data migration component of a Service Request Manager deployment: IBM Tivoli Integration Composer V7.1	► Microsoft Windows 2003 Server with Service Pack 2 ► IBM AIX 5L V5.3, AIX Technology Level 5300-06 ► Red Hat Enterprise Linux V3 and V4 (Intel) ► Red Hat Enterprise Linux Advanced Server V4 (Intel)

3.3 Planning to deploy Service Request Management

Before installing Tivoli Service Request Manager, you need to understand the different possible scenarios that are supported by Service Request Manager and also the different deployment topologies for your environment and business needs. We divided this section as follows:

- Installation scenarios
- Deployment topologies

3.3.1 Installation scenarios

Here are some of the scenarios that are supported by Service Request Manager. You can install Service Request Management standalone or integrated with IBM Tivoli Change and Configuration Management Database (CCMDB). Your installation procedures may vary depending on what software you have already installed in your environment.

Service Request Management standalone

When installing Service Request Manager without other specific software that may have already installed software needed by Service Request Manager, follow these steps:

- Middleware
- Base Services
- Service Desk
- Service Catalog (optional)
- Survey application (optional)
- Integration Software (optional)

Service Request Management integrated with CCMDB

Use this scenario if you are planning to integrate Service Request Manager with Tivoli CCMDB.

Installing Service Request Manager before CCMDB is installed

Here are the steps when installing Service Request Manager before CCMDB is installed:

- Middleware
- Base Services
- Service Desk

- Service Catalog (optional)
- Survey application (optional)
- Integration Software (optional)

> **Note:** Note that some of the CCMDB installation steps might not be needed because you have already installed some software as a part of the Service Request Manager installation.

A directory server is required if you decide to use this scenario. Refer to 3.2, “Prerequisites for installation” on page 89 for more information.

Installing Service Request Manager after CCMDB is installed

Use this scenario when installing Service Request Manager where IBM Tivoli Change and Configuration Management Database (CCMDB) are already installed. Note that the middleware and base services are already installed.

- Service Desk
- Service Catalog (optional)
- Survey application (optional)
- Integration Software (optional)

A directory server is required if you decide to use this scenario. Refer to 3.2, “Prerequisites for installation” on page 89 for more information.

3.3.2 Deployment topologies

To effectively plan an installation, you need to select a deployment topology. In this section, we discuss the two primary deployment strategies:

- Simple
- Custom

Simple (Single-server)

The single-server topology is a configuration where all components reside on the same machine, as shown in Figure 3-1 on page 98. In this configuration, there is one Windows machine serving as the Service Request Manager administrative system and a second machine hosting all other software used with Service Request Manager. This would be done typically for proof-of-concept purposes, as a demonstration, or as a learning environment. For managing enterprise assets and processes, you would typically implement a custom topology.

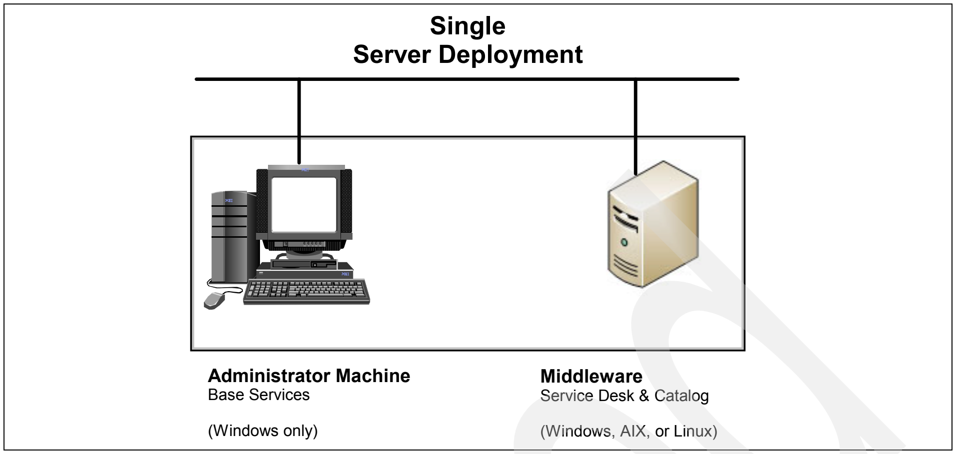


Figure 3-1 Single-server topology

Custom (Multiple-server)

This section details deploying Tivoli Service Request Manager across multiple machines. Service Request Manager should be deployed on multiple machines in order to provide load balancing, availability, reuse, and redundancy. This is the recommended deployment topology for a production environment.

In Figure 3-2, Service Request Manager has been deployed on a collection of systems.

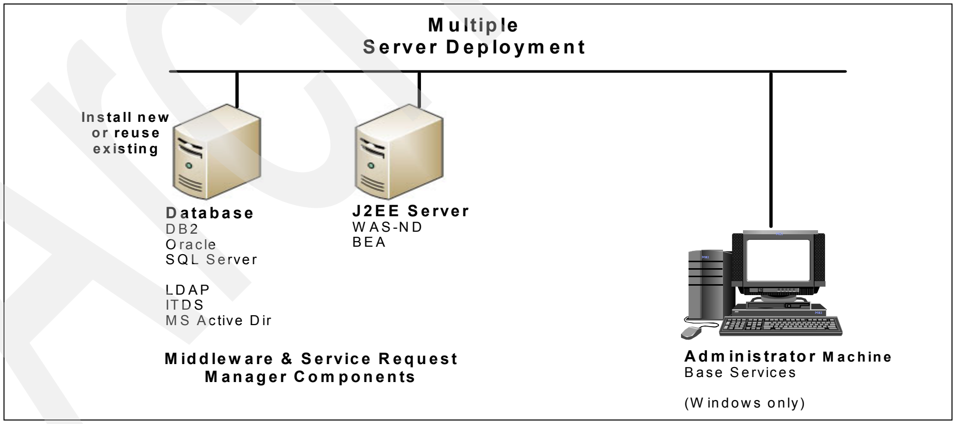


Figure 3-2 Multiple-server topology

3.4 Deployment considerations

This section provides information to help you ensure good system performance. We discuss basic and enterprise configuration deployments of Tivoli Service Request Manager.

The larger and more complex the deployment of Tivoli Service Request Manager, the more challenging it is for you to keep Tivoli Service Request Manager performing well for users. Because some of the greatest challenges are faced by those who deploy Tivoli Service Request Manager across large, global enterprises, this document has a special focus on improving performance in advanced enterprise configurations.

Considerations in system performance

Tivoli Service Request Manager system performance depends on more than the Service Request Manager applications and the database. The network architecture affects performance. Application server configuration can hurt or improve performance. The way that you deploy Tivoli Service Request Manager across servers affects the way that it performs. Many other factors come into play in providing the user with a good experience of system performance.

The following factors can influence system performance:

- System architecture setup
- Application server configuration
- Scheduled tasks (cron tasks)
- Reporting
- Integration with other systems with the Maximo Enterprise Adapter
- Network issues
- Bandwidth
- Load balancing
- Database tuning
- SQL tuning
- Client workstation configuration

3.4.1 Basic Service Request Manager configuration

A basic Service Request Manager configuration consists of Tivoli Service Request Manager running on a single application server. The application server connects to a single instance of the Tivoli Service Request Manager database that is available on a database server. The application server also connects to a report server.

If the Maximo Enterprise Adapter is also configured for deployment, then additional messaging queues are set up. Tivoli Service Request Manager uses the additional queues to send data to external systems and to receive data from external systems.

Suitable for up to 50 users

Your testing and QA configuration environment (to test the development work) should closely simulate your eventual production environment. The basic configuration is appropriate for development configuration and for QA configuration if your production system is expected to have a user load of 50 or fewer users. It is suitable for a production system with an actual user load of 50 or fewer users.

Figure 3-3 illustrates the high-level components of this configuration.

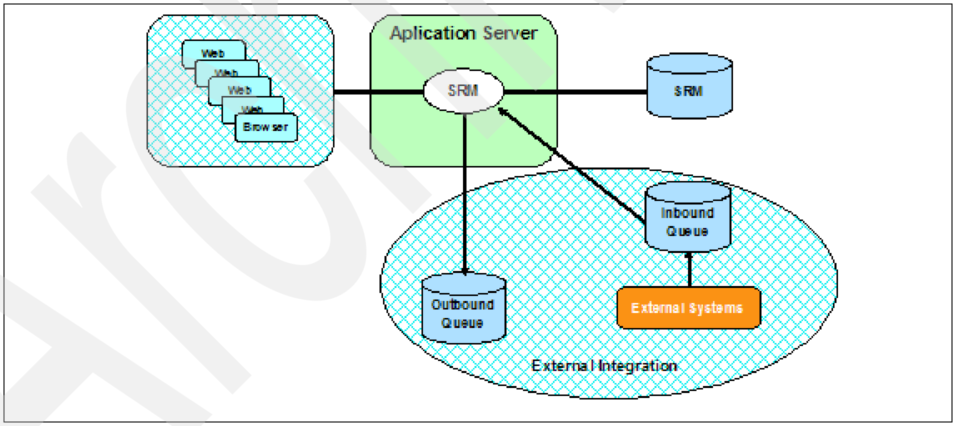


Figure 3-3 Basic Service Request Manager configuration

3.4.2 Advanced Tivoli Service Request Manager configuration

If you have a small to medium-size Tivoli Service Request Manager implementation, follow the standard Tivoli Service Request Manager installation and setup instructions and application server configuration instructions. If your production system must support hundreds of users, the basic configuration will likely prove insufficient due to memory and processing limitations. A single instance of the application server to support the Tivoli Service Request Manager application cannot handle the load that is required in enterprise deployments.

Because Tivoli Service Request Manager is an interactive application, the user who uses Tivoli Service Request Manager from a browser expects a response from the server to be immediate, or nearly so.

Other processes in Tivoli Service Request Manager, such as cron tasks and inbound messages from external systems, do not require user interaction. Response time for these processes does not need to be immediate. These processes can be configured to run in separate clusters or on separate hardware.

Deploying Tivoli Service Request Manager in a multiple cluster

Use load distribution in a heavily used system that experiences unacceptable delays in Tivoli Service Request Manager applications and overall throughput. Load distribution separates the user-interface traffic from all other processing in Tivoli Service Request Manager.

Deploying Tivoli Service Request Manager on more than one server (or clustering) is an effective way to distribute user load and improve the user experience of system performance. When you use clustering, implement the cluster configuration so that the system can scale well. You can set up multiple clusters, each consisting of multiple Java virtual machines (JVMs). The number of JVMs depends in part on the overall hardware and software limitations of the environment. You can tune the setup based on traffic and performance numbers. A system of separate clusters with multiple JVMs as cluster members provides advantages in system administration. In such a setup, Tivoli Service Request Manager users are not affected by any problem in the queue cluster or the Tivoli Service Request Manager cron task cluster that requires a cluster restart.

Figure 3-4 shows Tivoli Service Request Manager in a multiple cluster setup.

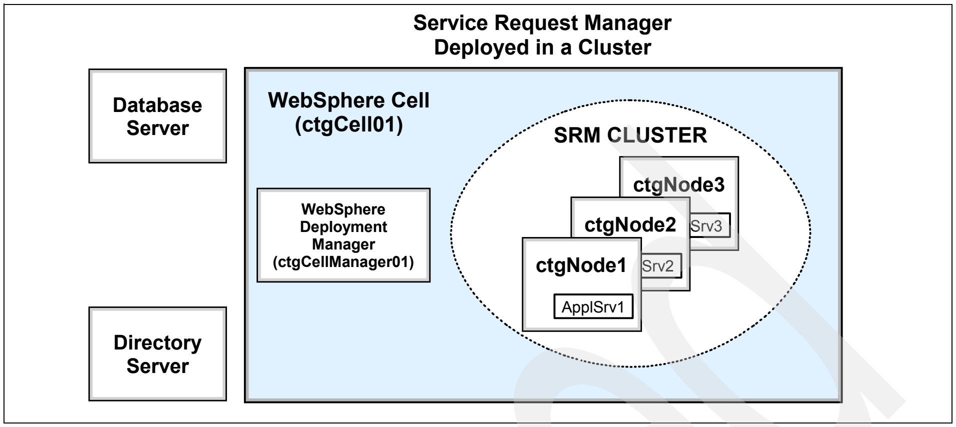


Figure 3-4 Tivoli Service Request Manager in a multiple cluster setup

Isolating Tivoli Service Request Manager user applications

For your system to perform well, isolate the Tivoli Service Request Manager user-interactive applications to one cluster, and the asynchronous processes to a separate cluster. Such a configuration also helps the system to scale well. You can further enhance each cluster simply by adding servers, depending on your needs. Table 3-4 on page 94 illustrates the high-level components of an advanced configuration.

Isolating reporting

The use of Tivoli Service Request Manager application reports can also put a large burden on the Tivoli Service Request Manager user-interactive applications. To remove the impact of reports on the user interface performance, run reports that require significant system resources as scheduled reports on separate clusters. To help the overall system performance, run only simple reports from the user interactive application.

Figure 3-5 on page 103 shows the overall Advanced Service Request Manager configuration.

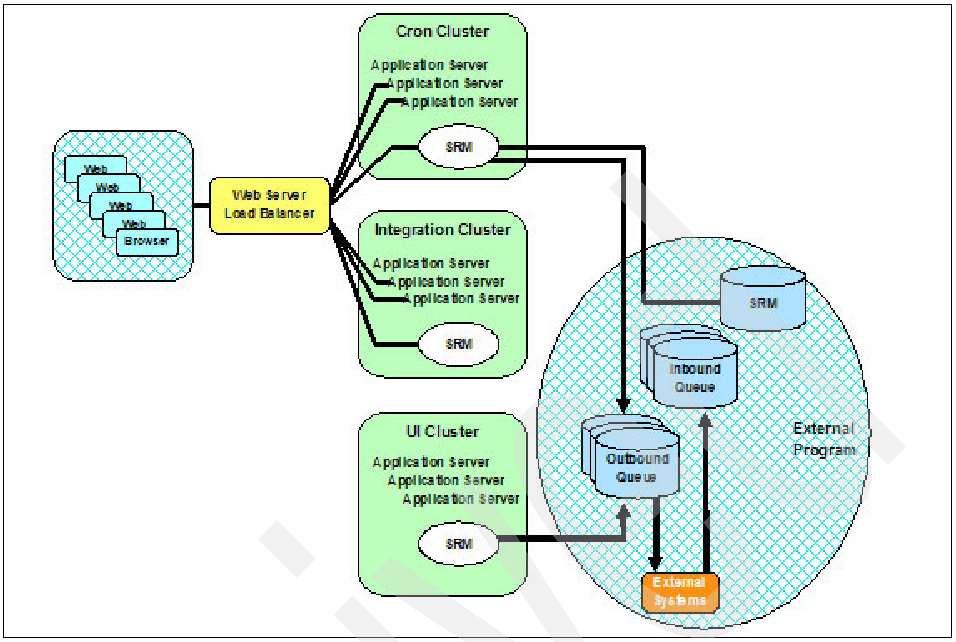


Figure 3-5 Advanced Service Request Manager configuration

Application server hardware configuration

Providing the following minimum hardware for each 50 concurrent users on the application server can help ensure good system performance:

- One 800 Mhz or faster processor
- 2 GB or more of memory

On a four-processor computer, dedicate one processor and 1 GB of memory to the operating system and application server. The typical four-processor server can support approximately 150 concurrent users. For example, to support approximately 150 concurrent users at the minimum recommended configuration, you should have three Java virtual machines (JVMs). Each JVM™ should have a processor and 2 GB of memory dedicated to it.

3.5 Reusing middleware

You can reuse some existing middleware installations as Tivoli Service Request Manager components. For example, you might have an instance of DB2 or Oracle in an existing database server farm that already has established access policies, redundancy measures, and backup plans in place. Reuse of an existing WebSphere server is not supported. If you plan to reuse existing middleware, ensure that they are at the level supported by Tivoli Service Request Manager. The middleware and Tivoli Service Request Manager installation programs do not provide a mechanism for patching back-level servers or provide remote prerequisite checks to ensure they are at the right level.

After through middleware has been installed through the Tivoli middleware installer, you have two options of configuring the servers for use with Service Request Manager. You will be presented with the option of either allowing the base services installation program to automatically configure middleware, or you can elect to manually configure each middleware component, which must be done prior to running the base services installation program.

4

Tivoli Service Request Manager installation

This chapter provides information about the steps needed to perform an installation of the IBM Tivoli Service Request Manager V7.1.

This chapter has the following sections:

4.1 Before you begin

This section describes the steps that you must take before you run either the Tivoli middleware installer or the Tivoli Service Request Manager installation program.

Ensure that you are logged in as a user with administrator privileges.

4.1.1 Disabling the firewall

This procedure disables the firewall should one be present. For Windows, open the Windows firewall and select **Off** on the General tab.

4.1.2 Hardware and software used for this IBM Redbooks publication installation

We used the systems listed in Table 4-1 for our Windows and Linux Tivoli Service Request Manager installation scenarios.

Table 4-1 Systems used in our lab

Platform	Operating system	Host name	Memory	CPU
Windows	Microsoft Windows 2003 Server with Service Pack 2	swat	3 GB	3 GHz
Linux	Red Hat Enterprise Linux Advanced Server V4	tsd34	3 GB	3 GHz

The following software has been installed on both systems:

- IBM Rational Agent Controller Version 7.0.3.1
- DB2 Enterprise Server Edition Version 9.1.4
- IBM Tivoli Directory Server Version 6.1.0.5
- WebSphere Application Server Network Deployment Version 6.1.0.13
- IBM HTTP Server Version 6.1.0.13
- Embedded Security Services Version 6.1
- Tivoli Service Request Manager Version 7.1

Note: For details about the installation scenarios, refer to Chapter 3, "Planning for installation" on page 87.

4.2 Tivoli Service Request Manager installation overview

Installation and deployment of the IBM Tivoli Service Request Manager consists of the following tasks:

1. Installation and configuration of the required Tivoli Service Request Manager middleware software products
2. Installation and configuration of base services
3. Installation of Language Pack for base services
4. Installation and configuration of Tivoli Service Request Manager

   This module is subdivided into three different sections:

   a. Best Practice Content Service Desk

   b. Sample Data Classification based on the United Nations Standard Product Service codes for Service Desk

   c. Sample data for Service Desk
5. Installation of Service Catalog

Note: You can choose to either select the options you deem required for your installation or select all the options. Some of the options that you may select will automatically entails the selection of additional options within the list.

4.2.1 Tivoli Service Request Manager middleware workspace

This middleware installer is designed to record all the information that was gathered at the time of the installation in the workspace directory. The information gathered from the different components is then used to deploy a single application from the different components. There is no possibility, however, for the middleware to be reconfigured for additional options, which would require the middleware to be undeployed and then redeployed to include the additional, options.

4.2.2 Tivoli Service Request Manager launchpad

The Service Request Manager launchpad serves as a centralized interface for launching a collection of installation programs.

This application indicates the different components to install and the order in which they are to be installed.

4.2.3 Starting the Tivoli Service Request Manager launchpad

This section contains the information to start the Service Request Manager launchpad.

To begin the Service Request Manager installation, run the launchpad by completing the following:

1. Log on as an administrator.
2. Insert the Service Request Manager CD into the CD drive.
3. Start the launchpad:
   - Windows

     The launchpad starts automatically. If it does not launch automatically, run the launchpad program by navigating to the root directory or the downloaded installation image and run the following command at a command prompt (or by double-clicking it in Windows Explorer):

     `launchpad.exe`
   - AIX and Linux

     Navigate to the root directory or the downloaded installation image and run the command **`launchpad.sh`**. The Welcome to IBM Tivoli Service Request Manager V7.1 window displays.
4. Select the launchpad language by clicking the **Language selection** list.
5. Click **Install the product** in the left pane.

The IBM Tivoli Service Request Manager window displays.You can now install the Service Request Manager component.

4.2.4 Tivoli Service Request Manager middleware installation

Several Tivoli Service Request Manager middleware products have to be deployed before the Tivoli Service Request Manager can be installed. The Tivoli Service Request Manager launchpad provides a convenient interface for installing and deploying the Tivoli Service Request Manager middleware. This middleware will extract the information gathered from the middleware installation when the Tivoli Service Request Manager is installed and will subsequently deploy the Tivoli Service Request Manager based on that information.

The middleware installer installs and deploys the following software:

- Database server

  Tivoli Service Request Manager uses the Maximo database to store details about DB2 UDB V9.1, or using a preexisting instance of DB2 UDB V8.23 or DB2 UDB V9.1. If you chose to use MS SQL Server 2005, or Oracle 10 for your Tivoli Service Request Manager deployment, you will have to install and configure them separately. See 4.5, “Using Oracle Database” on page 252 for details of using Oracle with the Tivoli Service Request Manager on Oracle Database.

- Directory server

  The directory server is used to secure the J2EE application. You do have the choice of using an existing IBM Tivoli Directory Server or installing a new version of IBM Tivoli Directory Server. In the case of a new installation of Directory Server, you do have to install a new DB2 UDB instance or reuse an existing one.

- J2EE server

  J2EE is the application server that serves and manages the Tivoli Service Request Manager base services application. This a new instance of WebSphere Network Deployment. Besides this J2EE deployment, you need to ensure that a Directory Server is securing the J2EE application.

  The J2EE application is composed of the following elements:

  – IBM HTTP Server

  – IBM HTTP Server Plug-in

The Tivoli Service Request Manager can deploy on a single machine or more machines. However, you need to run the installer from every machine. Please refer to Chapter 3, “Planning for installation” on page 87 for more details.

The Tivoli middleware installer deploys software on a single machine. To deploy Service Request Manager middleware on multiple machines, the Tivoli middleware installer must be invoked on each machine in the topology configuration you have chosen. Ensure that you have a strategy for deploying the middleware for each system you plan to use in your Service Request Manager deployment. If you deploy a Service Request Manager component using the Tivoli middleware installer on a system, for example, DB2, and then later decide you would also like to add IBM Tivoli Directory Server to that same system, you will have to undeploy DB2 before redeploying it in the same Tivoli middleware installer deployment plan that included IBM Tivoli Directory Server. When installing the middleware on a system, you must install all of the middleware intended for that system at one time.

A process ID is generated each time the Tivoli middleware installer is used to install or uninstall a set of middleware products. The process ID will appear on the files system in various places related to logs and generated files, such as file names, directory names, and log messages. The process ID is used to group logs and other generated files that are related to the same invocation of the Tivoli middleware installer. It also separates logs and other generated files that are related to different invocations of the Tivoli middleware installer. The process ID is a string of the format [operation_MMdd_HH.mm], where operation is a string indicating the operation being performed, such as __INSTALL__ or __UNINSTALL__, MM is a two-digit number (1-12) indicating the current month, dd is a two-digit number (1-31) indicating the current day in the month, HH is a two-digit number (0-23) indicating the current hour, and mm is a two-digit number (0-59) indicating the current minute. Here are some examples of process ID values:

```
[INSTALL_0924_15.45] An installation started on September 24 at 3:45pm
[UNINSTALL_1216_09.59] An uninstallation started on December 16 at
9:59am
```

In addition to installing and configuring Service Request Manager middleware, the Tivoli middleware installer performs additional tasks. If you choose to not run the middleware installer because you intend to perform the necessary configuration on existing middleware resources manually, you will need to perform the following tasks.

4.2.5 The Tivoli middleware installer workspace

The Tivoli middleware installer is designed to record the options you select during install in a directory referred to as the *workspace*, and then configure the components selected as a single deployed application. Once a plan has been deployed, the Tivoli middleware installer cannot subsequently deploy additional features and products onto the machine at a later time. The existing plan must first be completely undeployed through the Tivoli middleware installer before a different set of features and products can be deployed.

The composition and details of the deployment, as well as any logs generated by the Tivoli middleware installer process, are located in the workspace.

By default, the Tivoli middleware installer workspace is defined as follows:

- Windows: C:\ibm\tivoli\mwi\workspace
- AIX and Linux: /ibm/tivoli/mwi/workspace

The workspace can be defined on a shared resource that is made available to all the systems that will run the Tivoli middleware installer. Locating the workspace on a shared resource avoids the need to copy files, such as the topology file, manually from one machine to another.

The workspace contains the following items.

Deployment plan

The deployment plan is a collection of installation steps, configuration parameters for those steps, and target machine information. It is generated through the Tivoli middleware installer and it resides in the workspace directory.

When deployment steps are changed, the existing deployment plan is deleted and replaced with the new deployment plan.

The deployment plan configuration files contain information about the deployment plan itself. Whenever a deployment plan is modified, which includes reconfiguring existing deployment choices, the deployment plan configuration files will be deleted and regenerated when the deployment plan is redeployed.

Topology file

The topology file is a properties file that describes the configuration parameters of the middleware deployment. This file is created and then updated after every deployment or undeployment. If you have not defined a workspace that is centrally located and accessible to all the systems that will be receiving middleware, this file will have to be copied to the workspace of each machine where middleware is being deployed. The contents of this file can be used by the installation programs to populate their panels with meaningful default values.

This file is saved in <workspace>/topology.xml.

Logs

Log files that contain information about the deployment can be found in the workspace directory. In addition, log files native to the middleware itself are also contained in this directory.

4.2.6 Installing and configuring the middleware

This procedure explains how to use the Tivoli middleware installer to create a deployment plan that is responsible for installing and configuring prerequisite middleware products.

The instructions provided in this section are for a typical installation using default values. In addition, these instructions assume you are using the Tivoli middleware installer to install a complete set of middleware for use with Service Request Manager on a single machine. If you intend to deploy middleware products across an array of machines, you will have to run the Tivoli middleware installer on each machine, choosing which piece of middleware to install on that each particular machine. In this case, you will encounter a subset of the panels included in these instructions that are relevant to the middleware you have chosen to install on a machine.

The Tivoli middleware installer can also configure existing middleware products. If you intend to reuse existing middleware products for your Service Request Manager deployment, refer to 3.5, "Reusing middleware" on page 104. Refer to Chapter 3, "Planning for installation" on page 87 to learn about using custom values during a custom installation.

In some cases, fields and labels displayed within the Tivoli middleware installer are not correctly displayed on the window when installing through remote sessions. We recommend that you use the Tivoli middleware installer locally on the system that will host the middleware. If you do experience this phenomenon, first minimize and then maximize the install wizard to force it to re-display the panel.

Avoid using localhost for host name values in the installation program. Specify the actual fully-qualified host name of the system on which you are installing. For Linux systems, ensure that the command `hostname -f` returns a fully-qualified host name. If it does not, consult the appropriate documentation for your operating system to ensure that the `hostname` command returns a fully-qualified host name

Important: When entering LDAP values for Service Request Manager installation window fields, entries in LDIF files, or values you enter directly into an directory instance using the directory server's own tools, be aware of the product-specific syntax rules for using special characters in an LDAP string. In most cases, special characters must be preceded by an escape character in order to make it readable by the directory server. Failing to escape special characters contained in an LDAP string used with Service Request Manager will result in errors.

Many directory server products consider a blank space as a special character that is part of the LDAP string. Therefore, if you mistakenly enter an LDAP string that contains a blank, at the end of a field value, for example, and you do not precede the blank character with an escape character, you will encounter errors that will be difficult to troubleshoot. Refer to the product documentation for your directory server for more information about special characters in LDAP strings.

4.2.7 The Tivoli middleware log files

The Tivoli middleware log files are located in the workspace directory. This directory was defined during the installation.

To install the prerequisite middleware products for Service Request Manager, complete the following steps.

If you have not started the Service Request Manager launchpad, then:

1. Log on as an administrator.
2. Insert the Service Request Manager CD into the CD drive.
3. Start the launchpad.

Windows

The launchpad starts automatically. If it does not launch automatically, run the launchpad program by navigating to the root directory, or the downloaded installation image, and run the following command at a command prompt (or by double-clicking it in Windows Explorer):

```
launchpad.exe
```

AIX and Linux

Navigate to the root directory, or the downloaded installation image, and run the command **`launchpad.sh`**. See Figure 4-1.

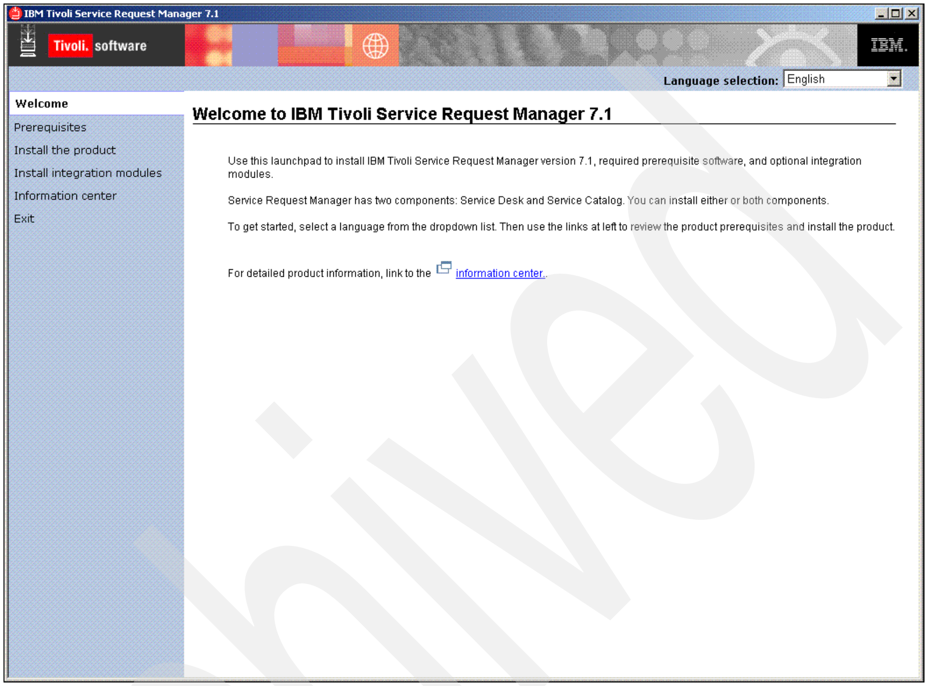


Figure 4-1 Launchpad

4.3 Uninstalling the middleware

Uninstalling the middleware for Service Request Manager consists of running the Tivoli middleware installer and using it to undeploy the previously deployed deployment plan.

> **Note:** You must use the Tivoli middleware installer to uninstall any Service Request Manager middleware installed by the Tivoli middleware installer. The Tivoli middleware installer creates a registry when installing Tivoli Service Request Manager middleware. If you use the native middleware uninstall programs, this registry will be out of sync with what is deployed. This will cause errors if you then try to reinstall Service Request Manager middleware using the Tivoli middleware installer.

Note: At points during the uninstall process, the Tivoli middleware installer uninstall progress bar might appear to pause. This is normal behavior. In most cases, the Tivoli middleware installer uninstall progress bar will resume shortly after pausing. If you suspect your uninstall process has experienced an error, refer to the Tivoli middleware installer log files.

To uninstall the J2EE server, ensure the directory server (IBM Tivoli Directory Server or Active Directory) is active. Do not uninstall the directory server until the J2EE server has been uninstalled.

4.3.1 Middleware installation on Windows

Note: For the middleware installation on Linux, please refer to 4.3.2, “Middleware installation on Linux” on page 158.

1. Log in as administrator on Windows.
2. Navigate to the root directory, or the downloaded installation image, and run the `launchpad.exe` command or double-click it in Windows Explorer.

3. To launch the Tivoli middleware installer from the launchpad, start the launchpad, as shown in Figure 4-2.

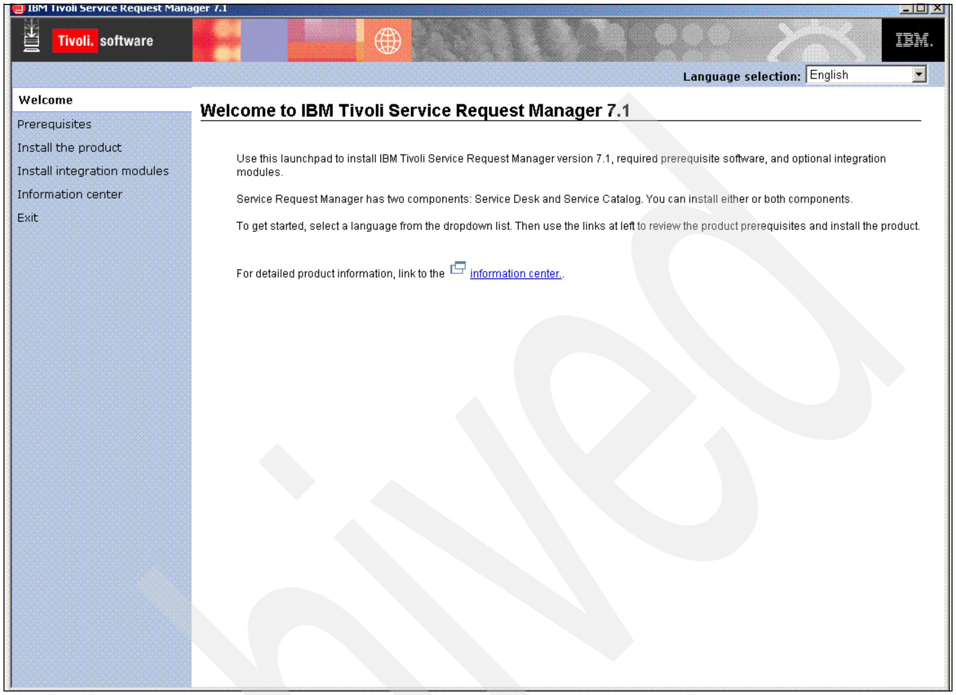


Figure 4-2 Launchpad

4. Select **Install the Product** (Figure 4-3 on page 117).

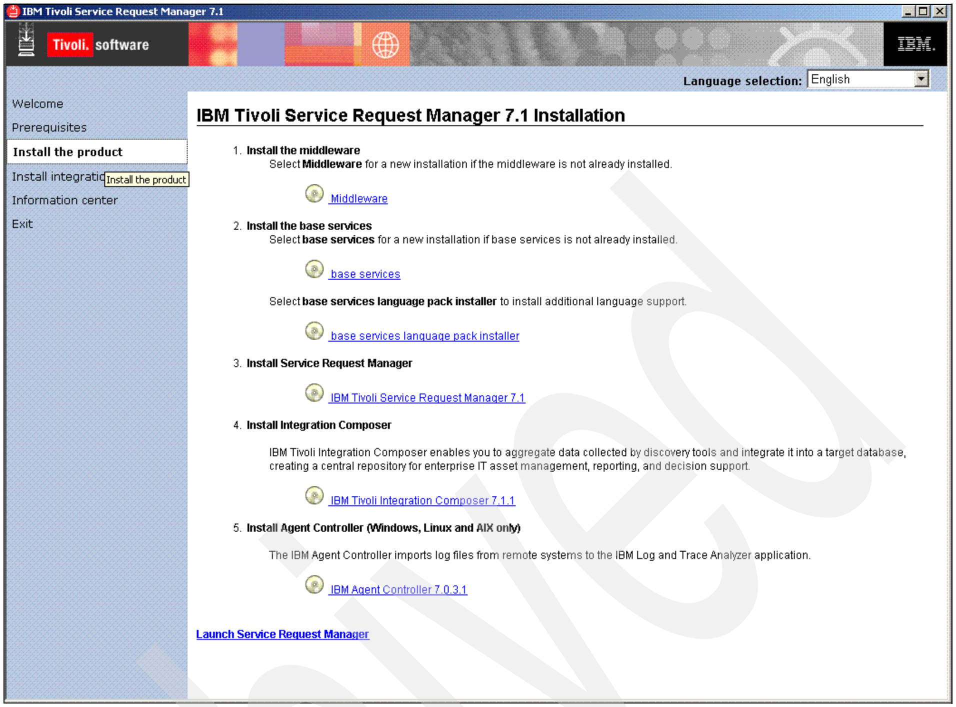


Figure 4-3 Launchpad

5. Select a language for the installation and click **OK** (Figure 4-4).

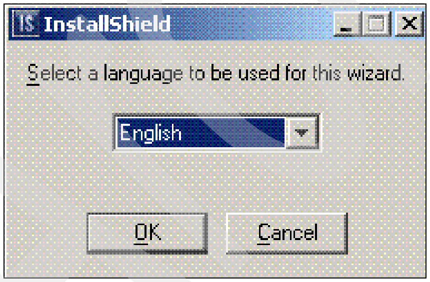


Figure 4-4 Language selection

6. From the Welcome window, click **Next** (Figure 4-5).

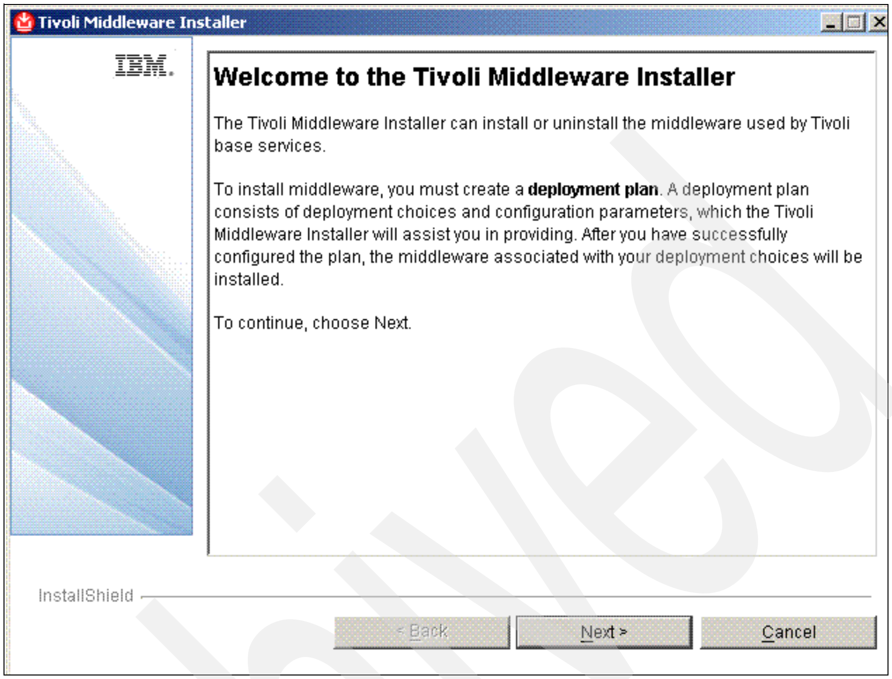


Figure 4-5 Tivoli Middleware Installer

7. The Tivoli middleware installer license agreement window is displayed. Read the license information and select **I accept both the IBM and the non-IBM terms** if you agree with the terms. Click **Next** (Figure 4-6).

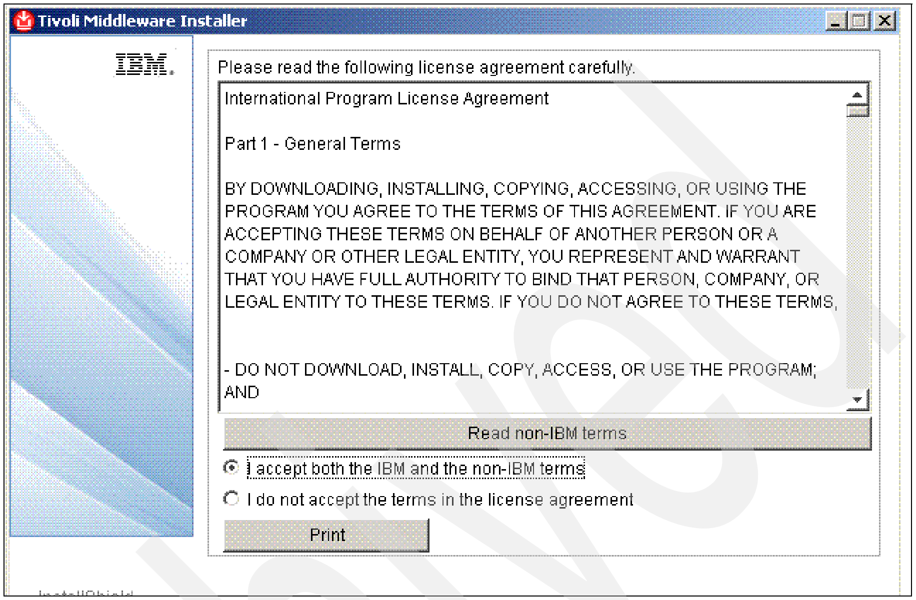


Figure 4-6 Licence agreement

8. From the Choose Workspace window, specify the workspace directory containing the currently deployed plan, and then click **Next** (Figure 4-7). The default location for the workspace will be the last workspace location specified. The default location for the workspace is c:\ibm\tivoli\mwi\workspace.

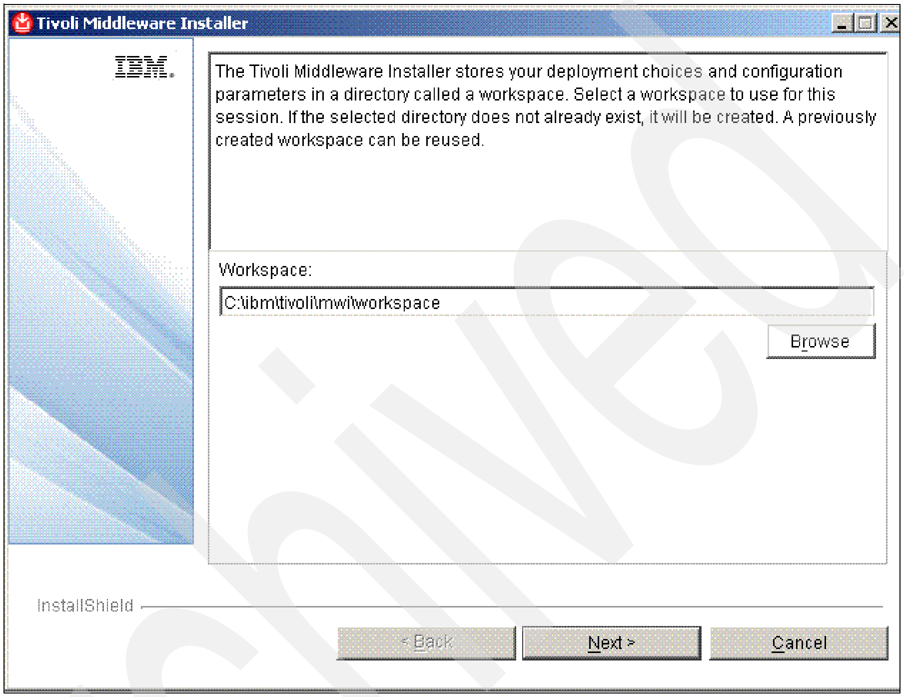


Figure 4-7 Workspace

9. Click **Next** (Figure 4-8) for the IBM Autonomic Engine to proceed with the installation.

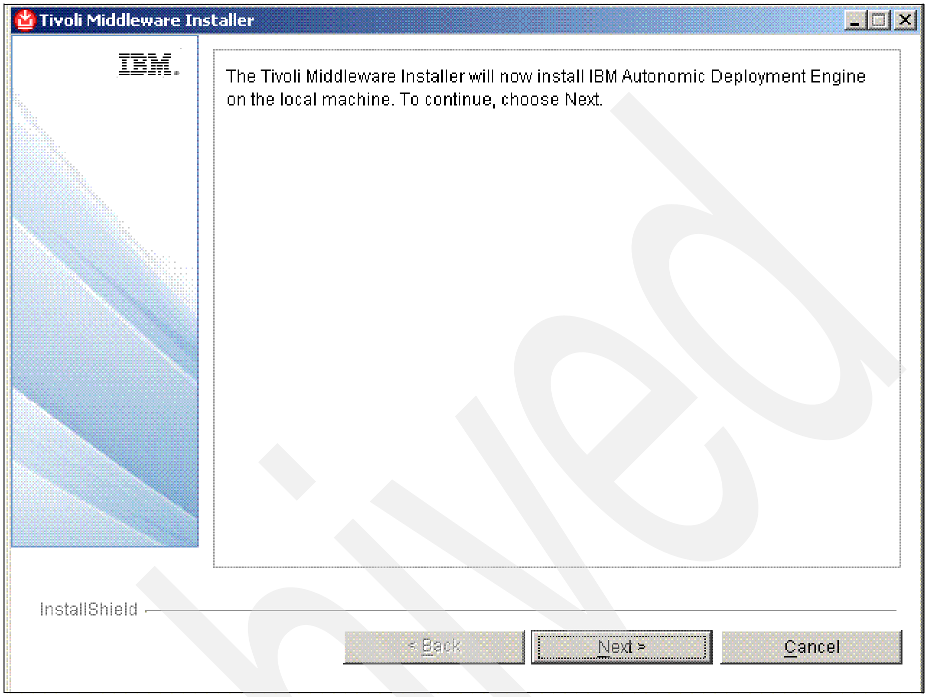


Figure 4-8 Tivoli Middleware Installer

10. The installation proceeds and you are notified of the different stages of the installation. Click **Next** (Figure 4-9)

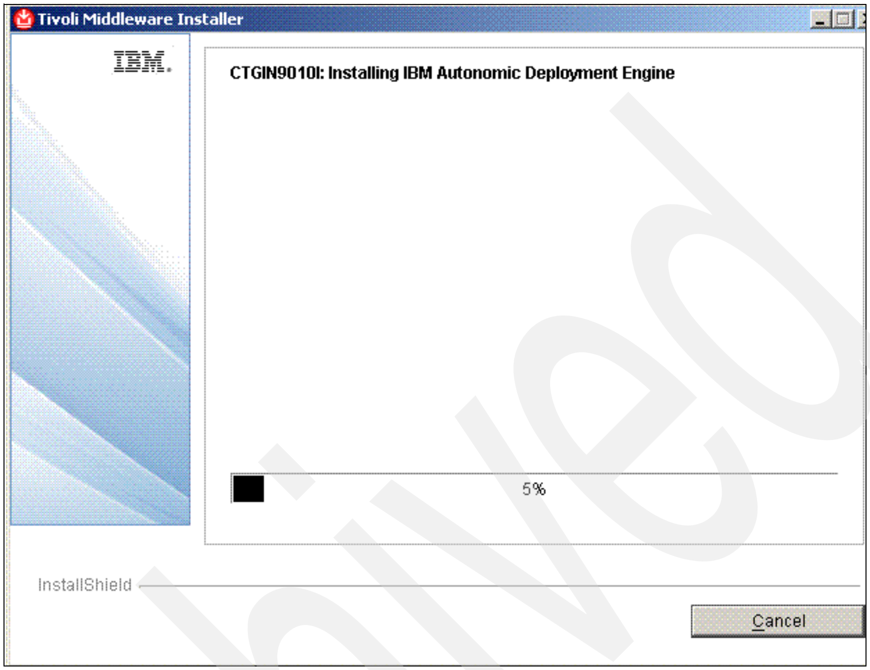


Figure 4-9 Tivoli Middleware Installer

11. The installation checks that all prerequisites have been met. Click **Next** (Figure 4-10).

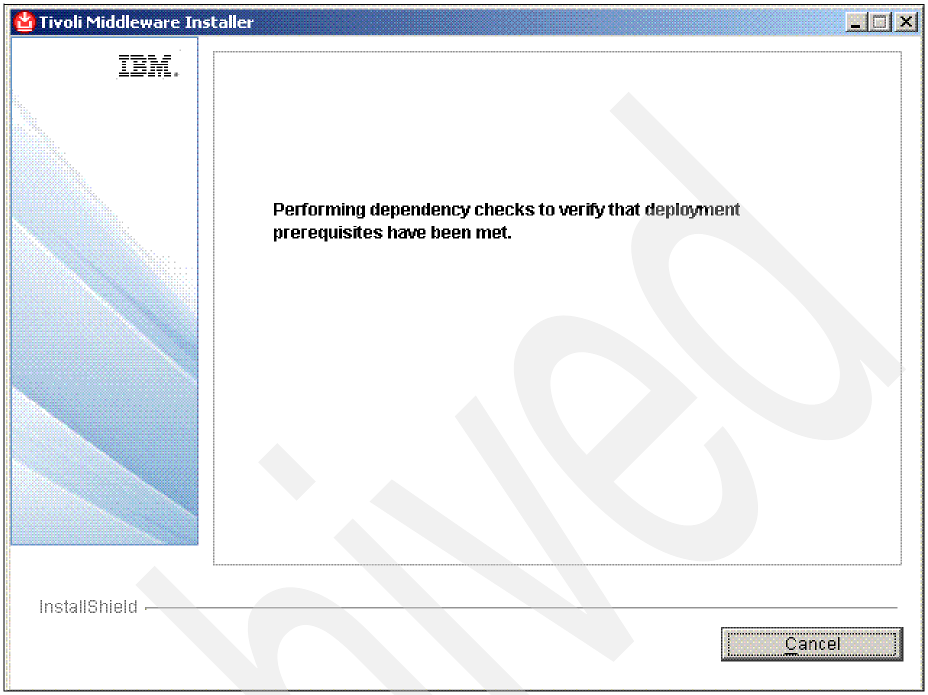


Figure 4-10 Tivoli Middleware Installer

12. The installation then looks for any software components that are already installed onto the machine and if these can eventually be reused for the current installation. Click **Next (**Figure 4-11).

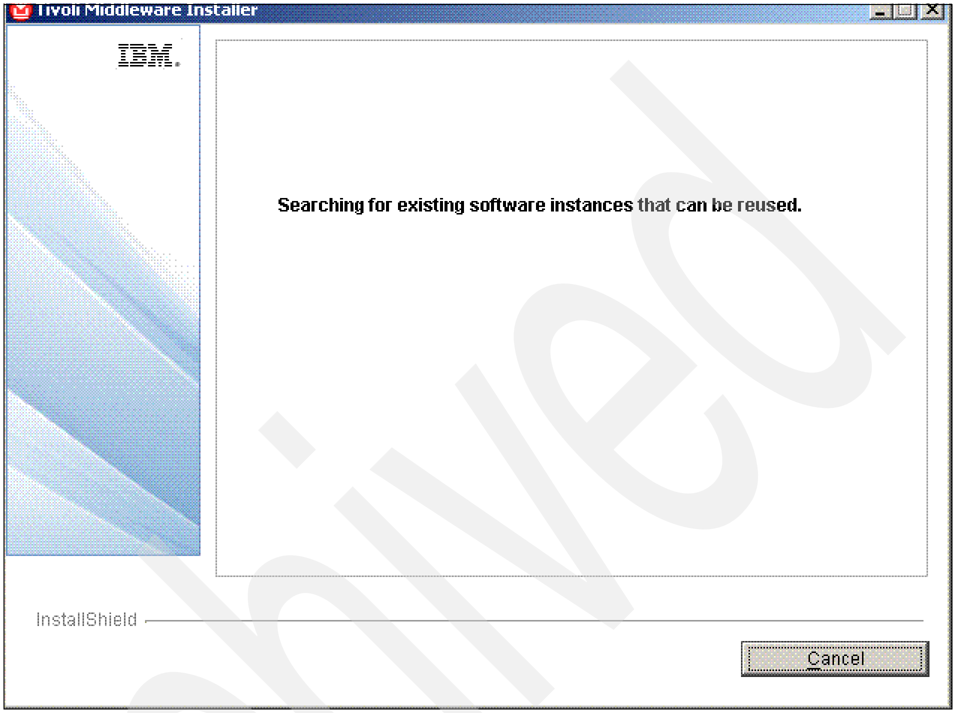


Figure 4-11 Tivoli Middleware Installer

13. Once this process is complete, you will be presented with the options shown in Figure 4-12. Select all the options and press **Next**.

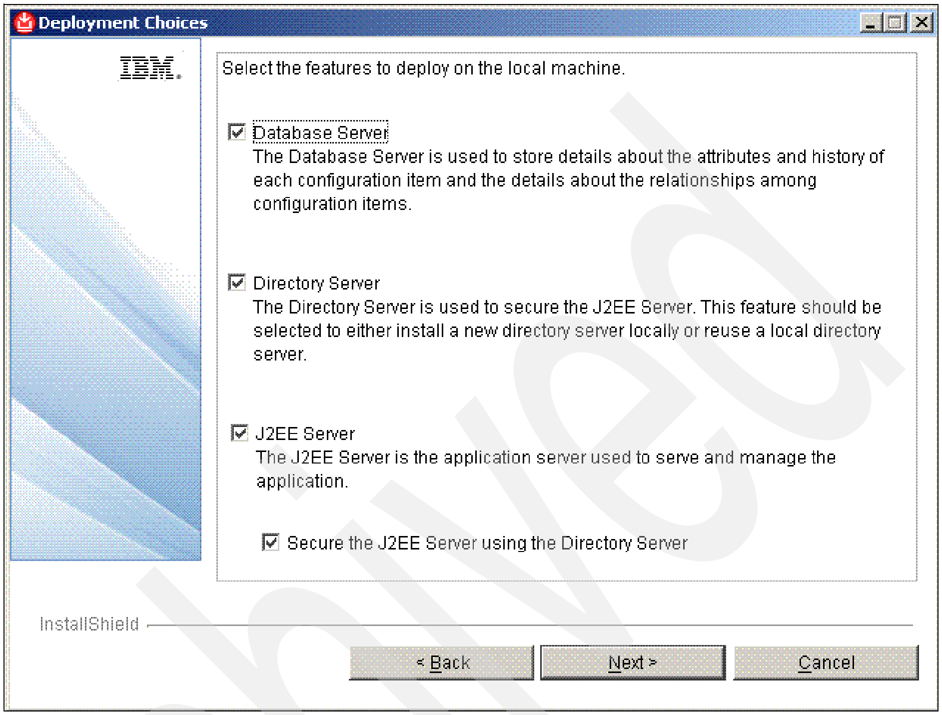


Figure 4-12 Deployment Choices

14. The installation proceeds with the generation of the *deployment plan*. Click **Next** (Figure 4-13).

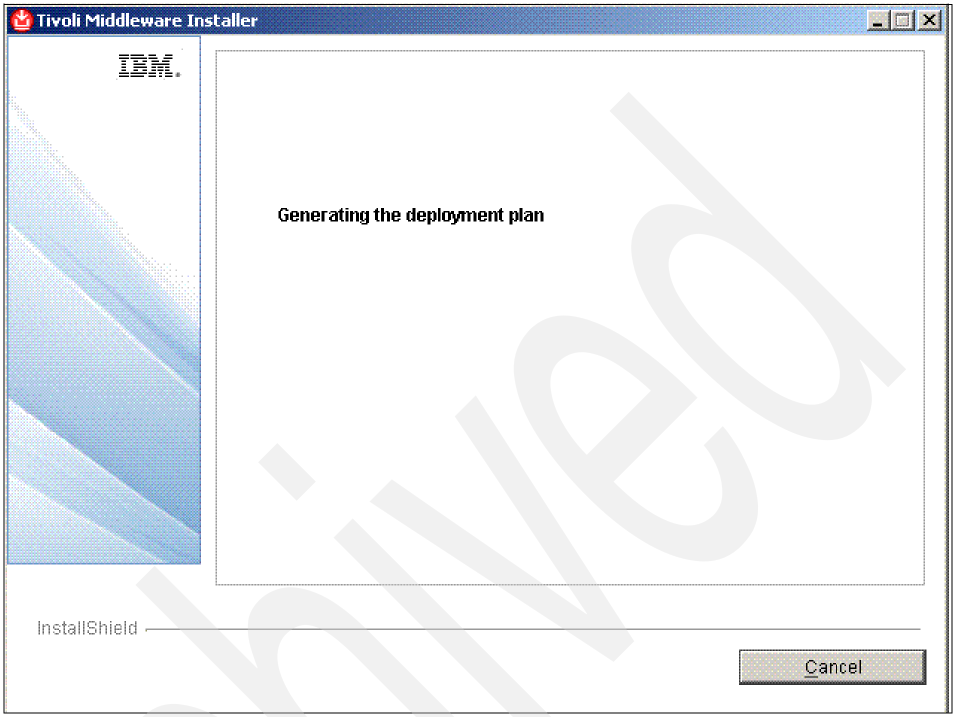


Figure 4-13 Generating the deployment plan

15. Once the deployment plan has been generated, you are then presented with the list of applications that will be installed. Click **Next** (Figure 4-14).

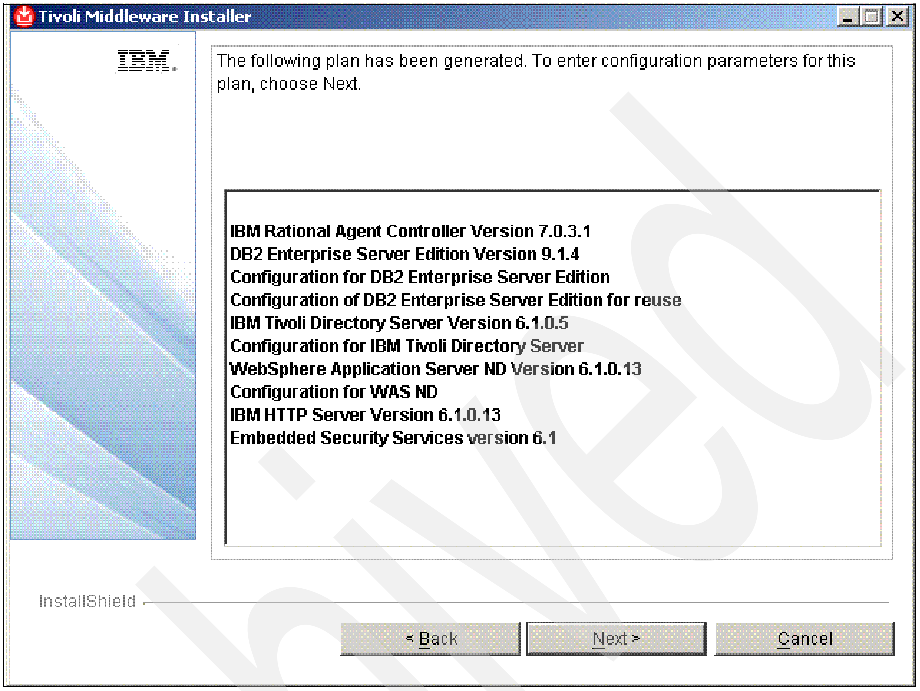


Figure 4-14 Generating the deployment plan

16.Click **Next** to continue with the installation. You are now presented with the option of using the same password for the subsequent password options. Should you need to use different passwords, do not select **Use this password as the value for all subsequent passwords**. Click **Next** (Figure 4-15).

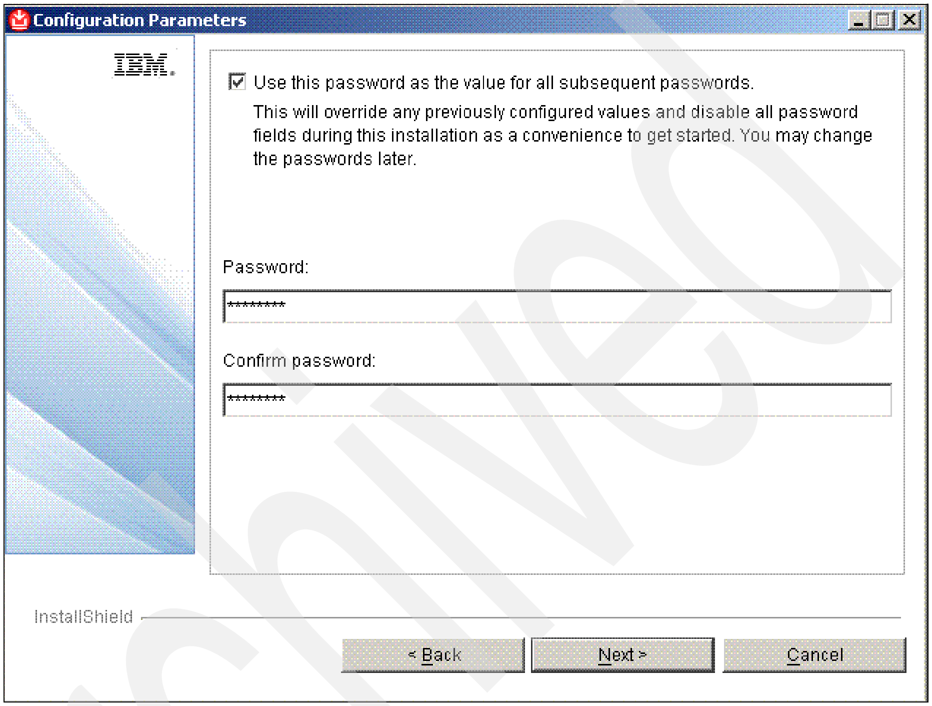


Figure 4-15 Configuration Parameters

17. Enter the requested information (for example, the password if you have selected not to use the same password for all subsequent options in the previous step). The values are entered by default. Click **Next** (Figure 4-16).

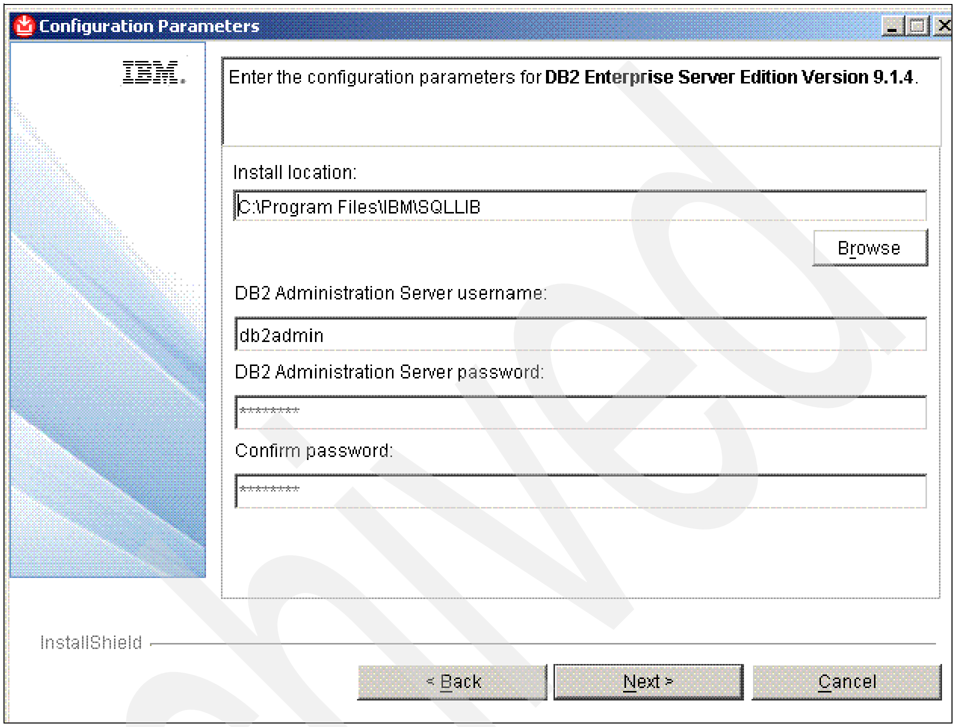


Figure 4-16 Configuration Parameters

18. Modify the parameters as you see fit and click **Next** (Figure 4-17).

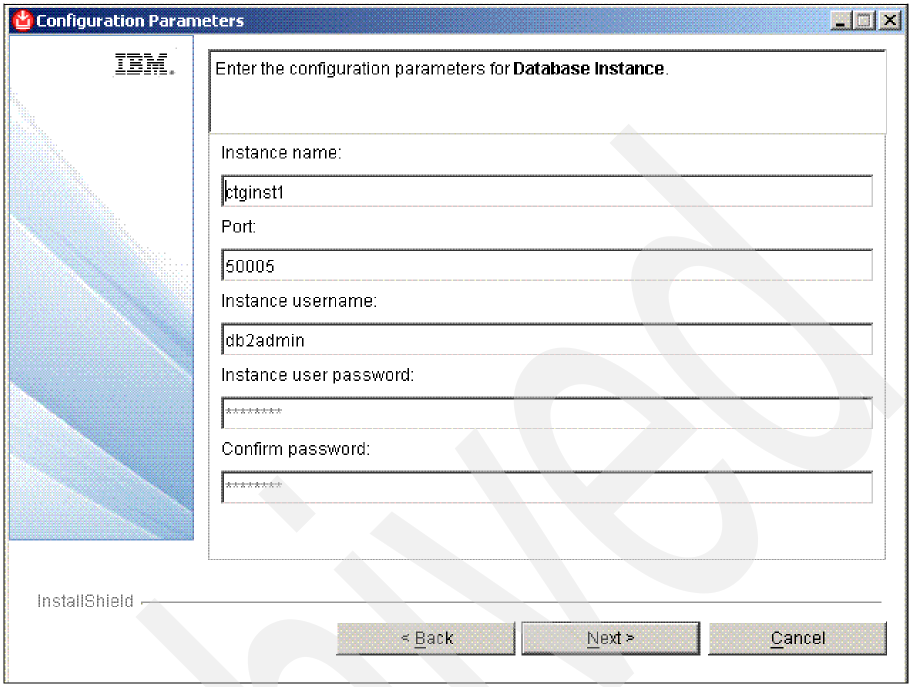


Figure 4-17 Configuration Parameters

19. The options are again filled in by default. Click **Next** (Figure 4-18).

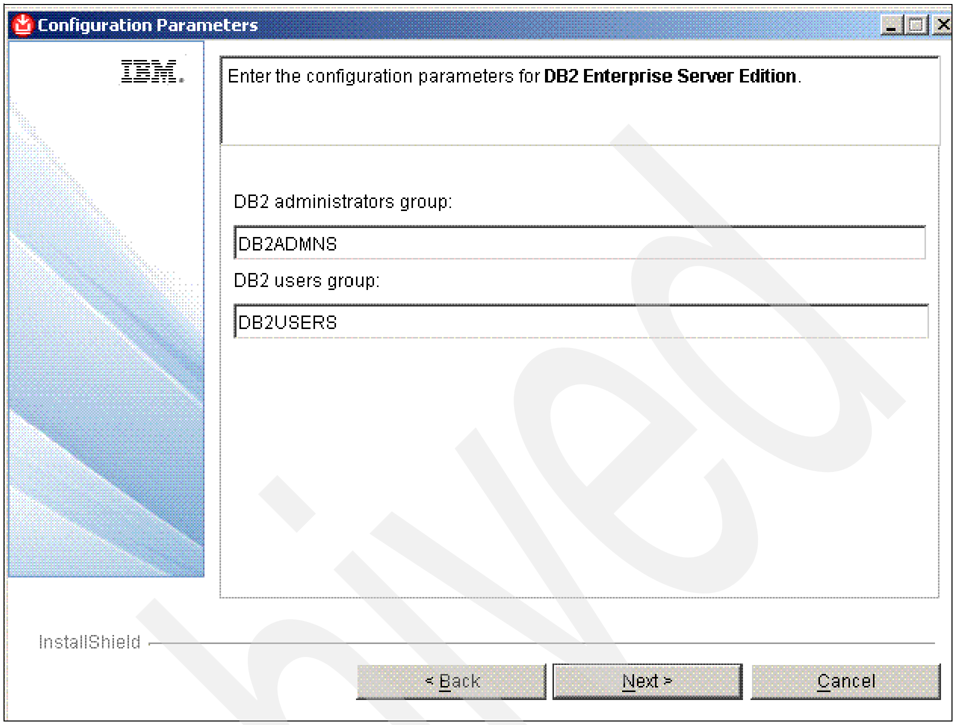


Figure 4-18 Configuration Parameters

20. The options are again filled in by default. Click **Next** (Figure 4-19).

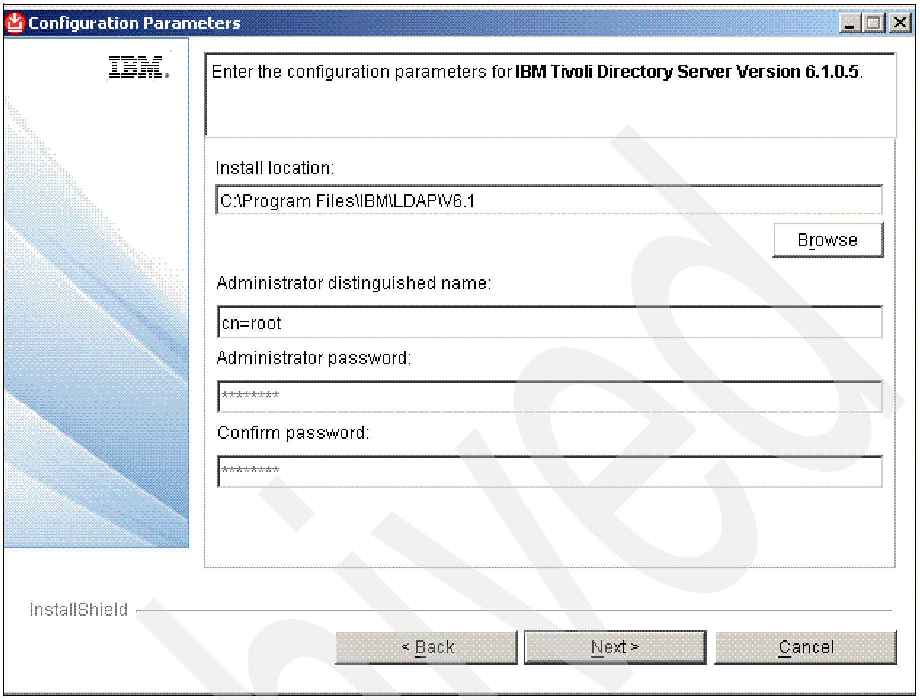


Figure 4-19 Configuration Parameters

21.Accept the default values and click **Next** (Figure 4-20).

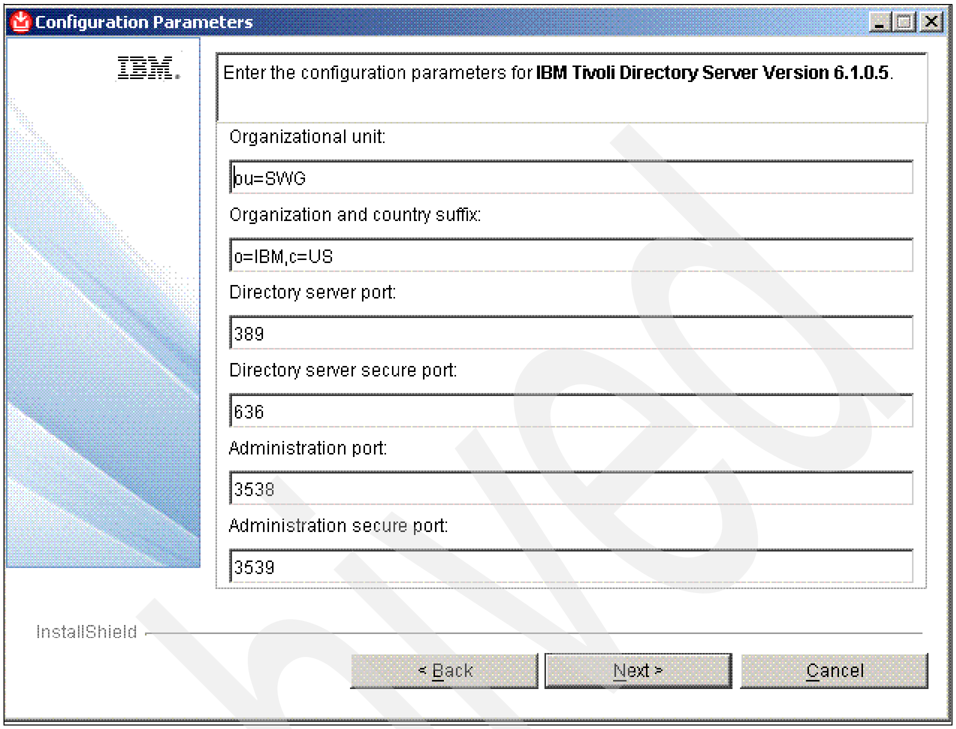


Figure 4-20 Configuration Parameters

22. Accept the default values and click **Next** (Figure 4-21).

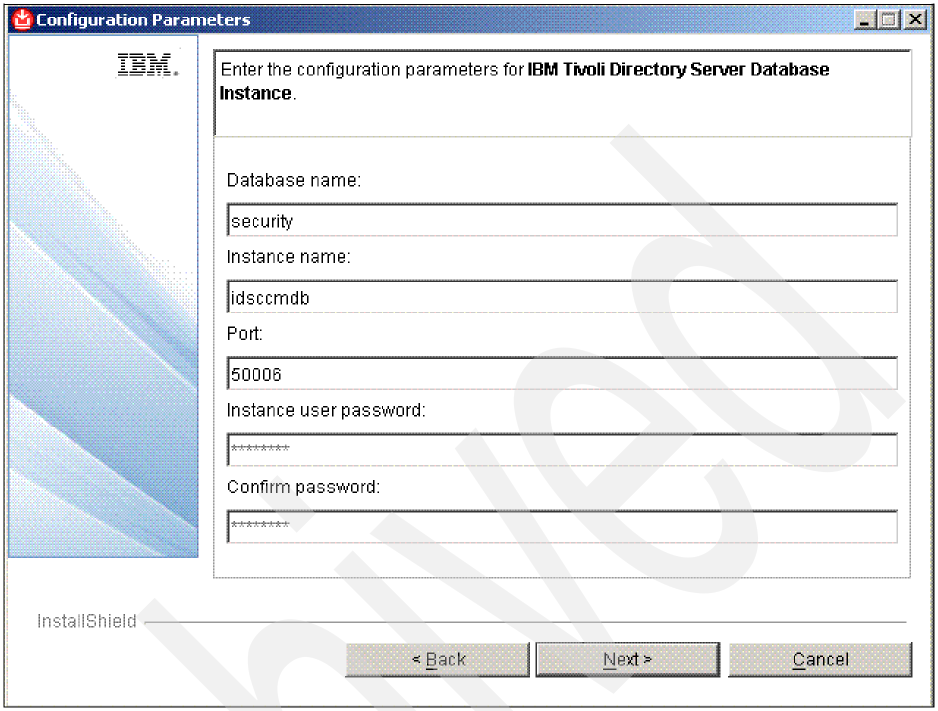


Figure 4-21 Configuration Parameters

23. Accept the default values and click **Next** (Figure 4-22).

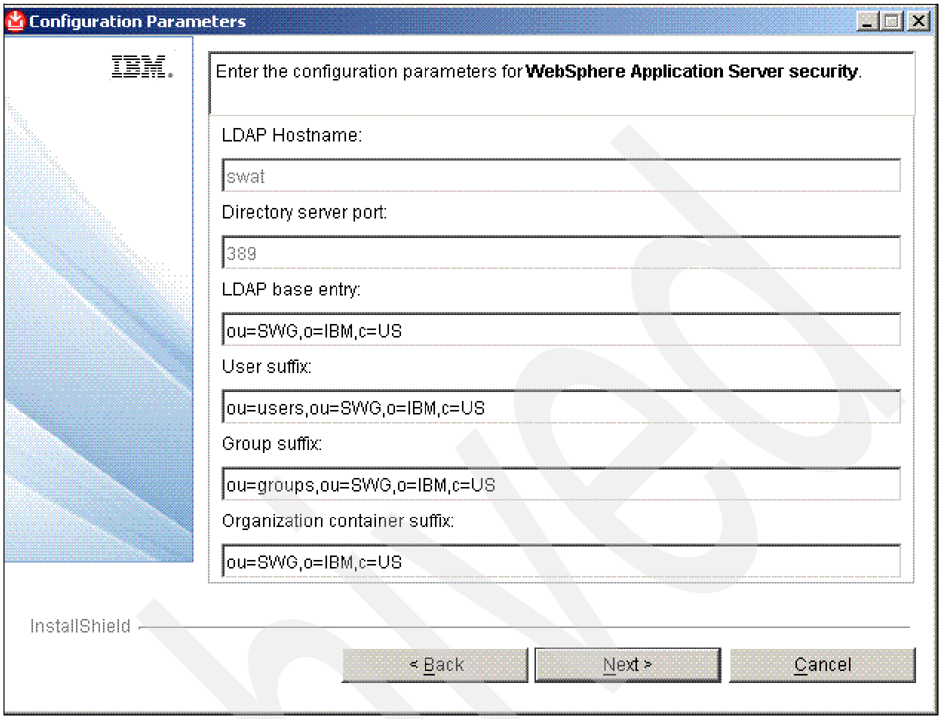


Figure 4-22 Configuration Parameters

24. Accept the default values and click **Next** (Figure 4-23).

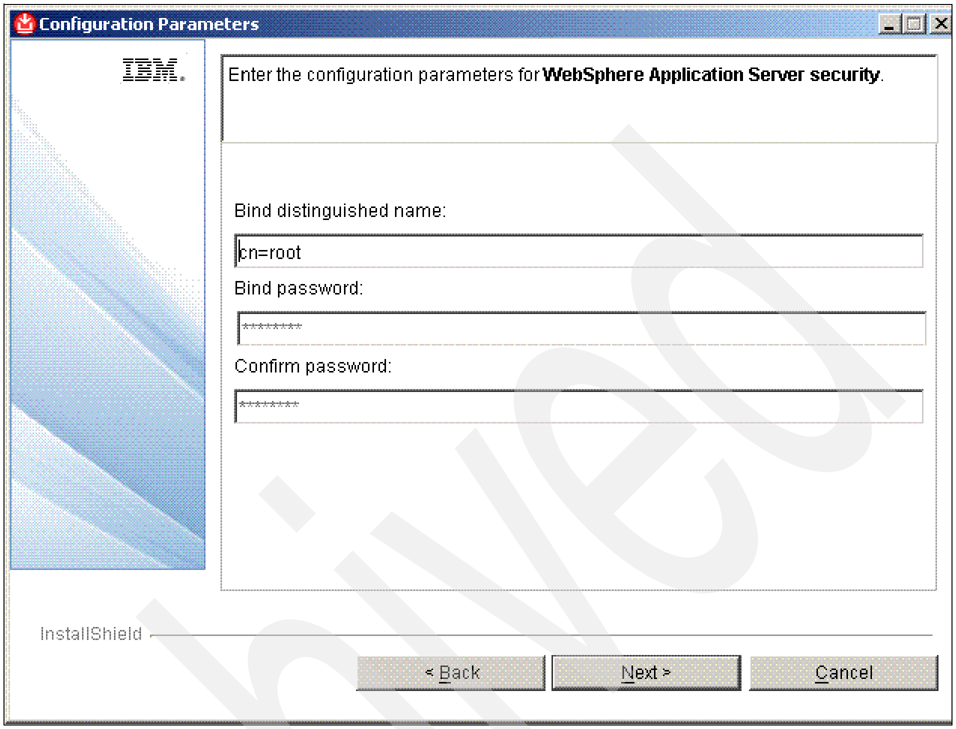


Figure 4-23 Configuration Parameters

25. Accept the default values and click **Next** (Figure 4-24).

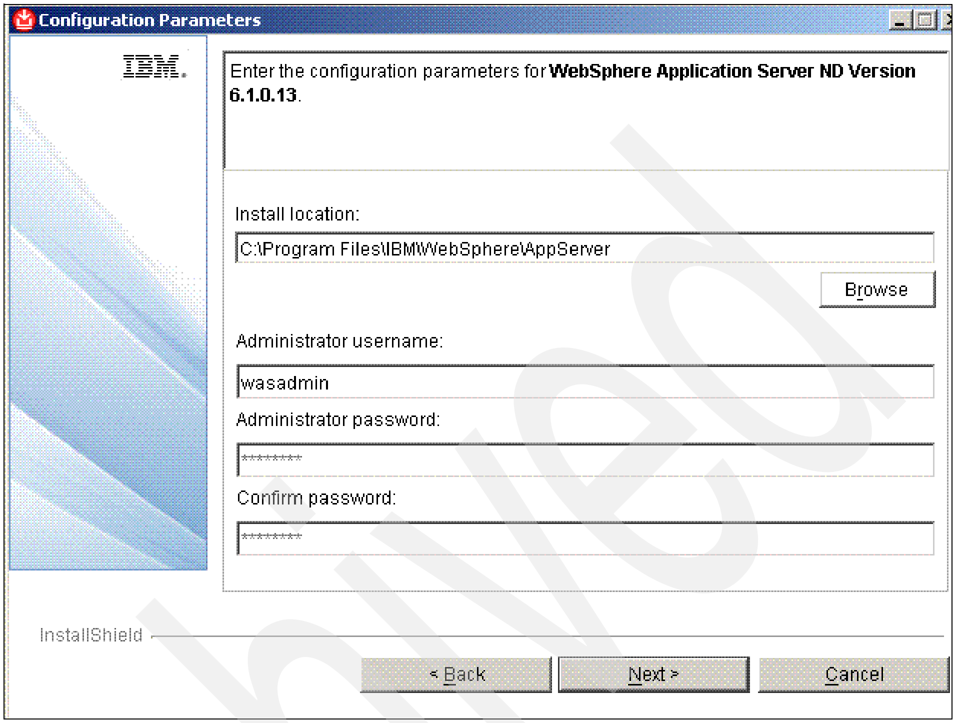


Figure 4-24 Configuration Parameters

26. Accept the default values and click **Next** (Figure 4-25).

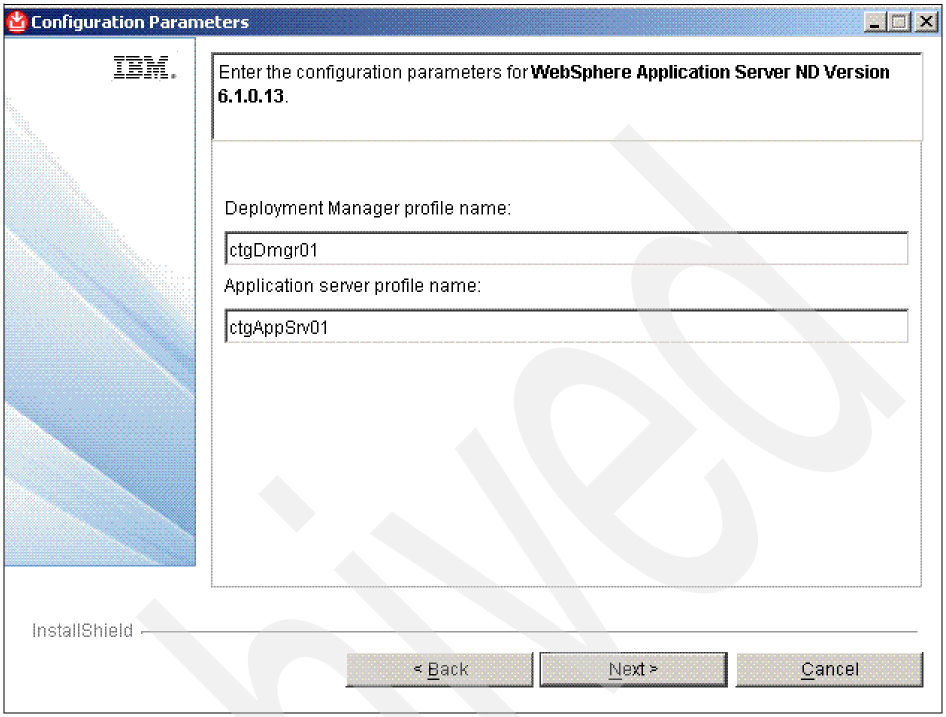


Figure 4-25 Configuration Parameters

27. Accept the default values and click **Next** (Figure 4-26).

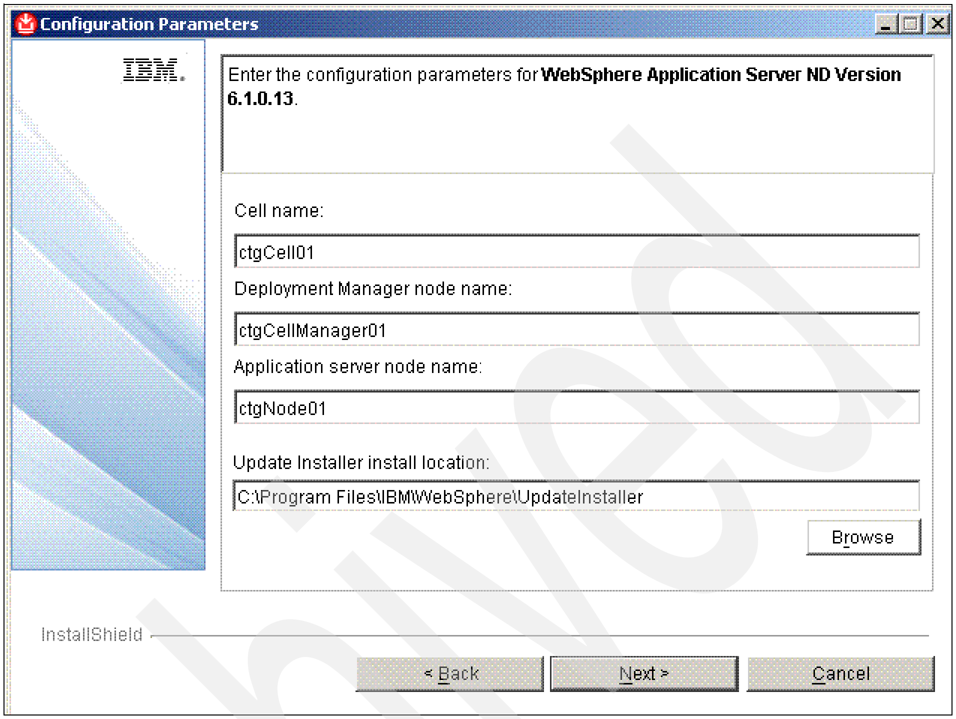


Figure 4-26 Configuration Parameters

28. Accept the default values and click **Next** (Figure 4-27).

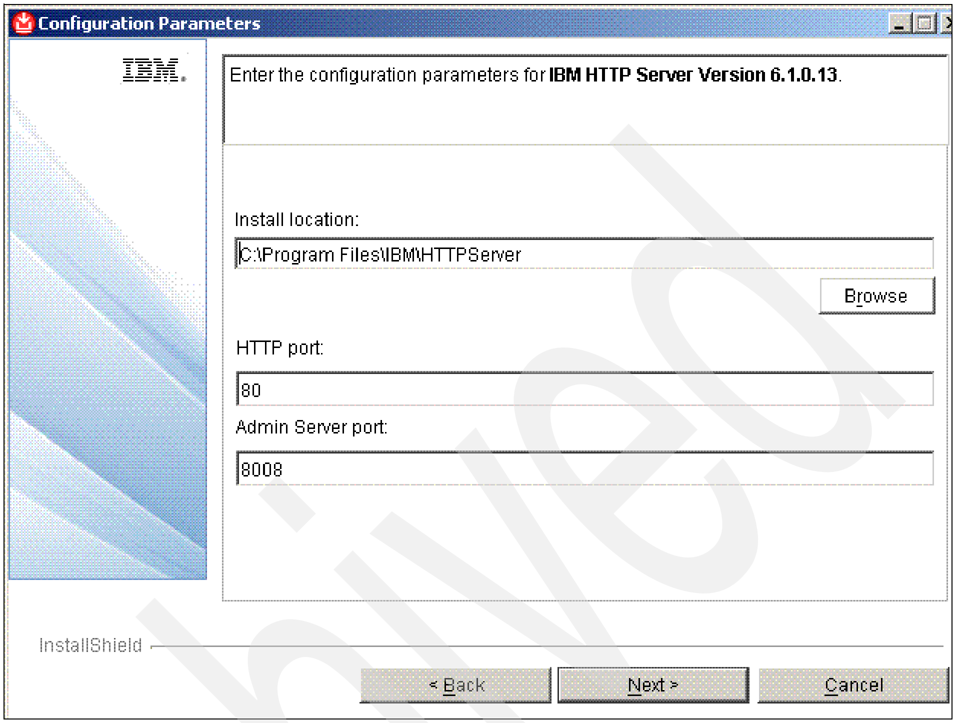


Figure 4-27 Configuration Parameters

29.Click **Next** (Figure 4-28).

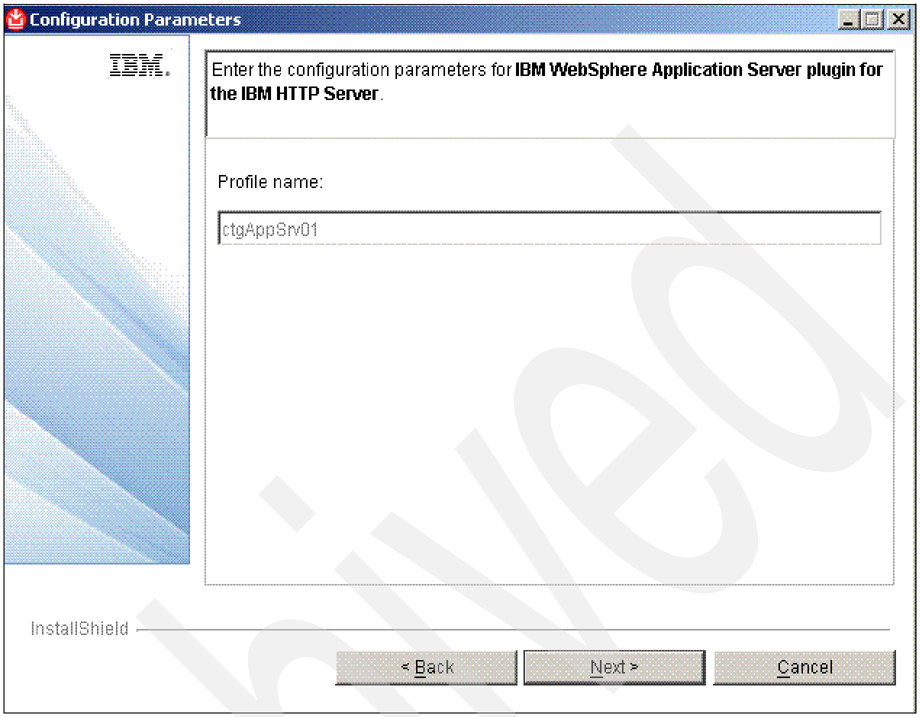


Figure 4-28 Configuration Parameters

30.Click **Next** (Figure 4-29).

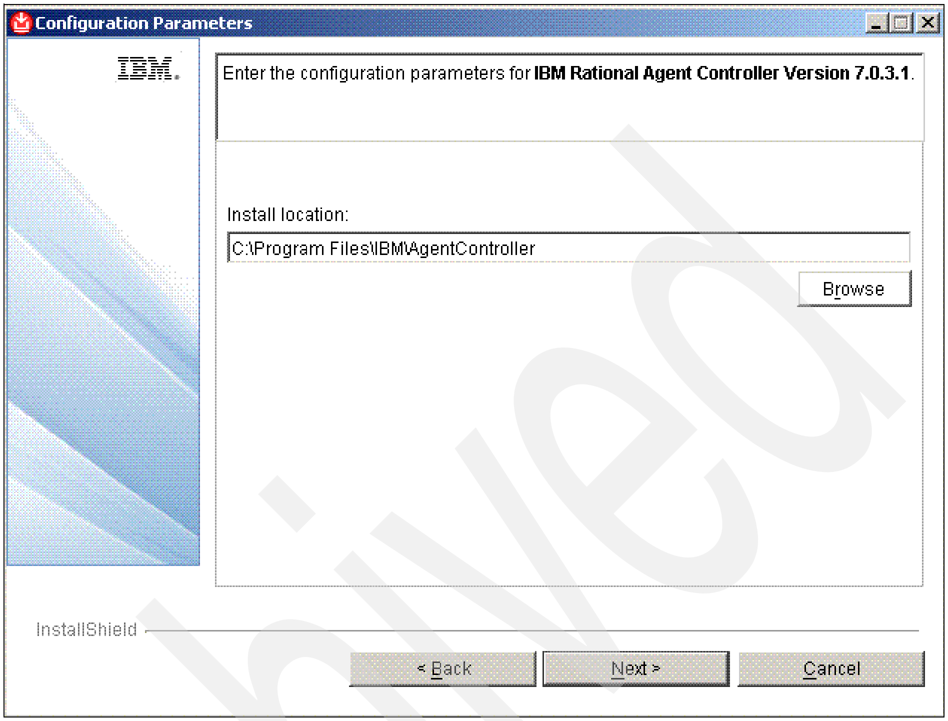


Figure 4-29 Configuration Parameters

31. You must locate the directory containing all the software images to proceed with the installation. You can either copy these files from a source, such as a CD, or just point to an existing directory containing the images. In our case, we specify an existing directory containing all the images. Click **Next** (Figure 4-30) to continue.

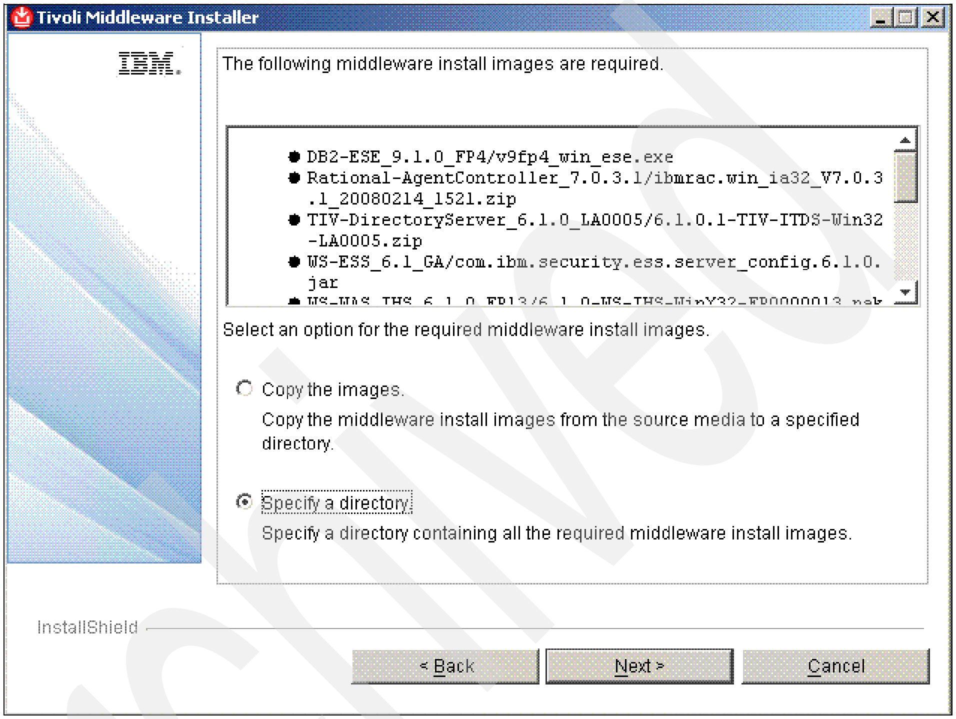


Figure 4-30 Location of the images

32. Point to the directory where all these images are located and click **Next** (Figure 4-31).

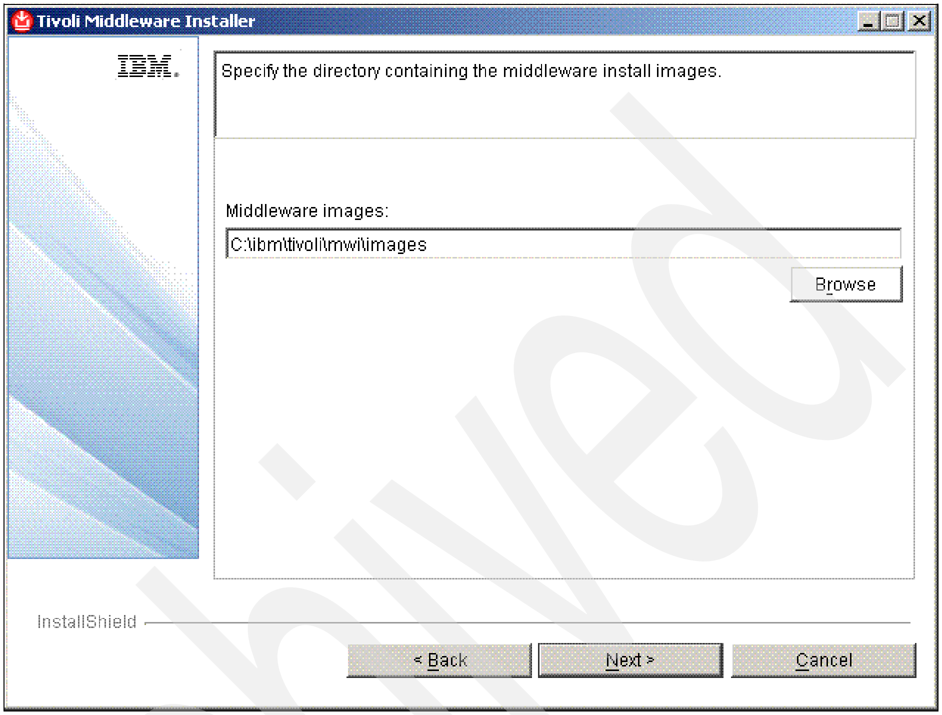


Figure 4-31 Location of the images

33. In our installation the directory was C:\source\middleware. Verify that the images are correct and then click **Next** (Figure 4-32 on page 145) to continue and proceed with the installation.

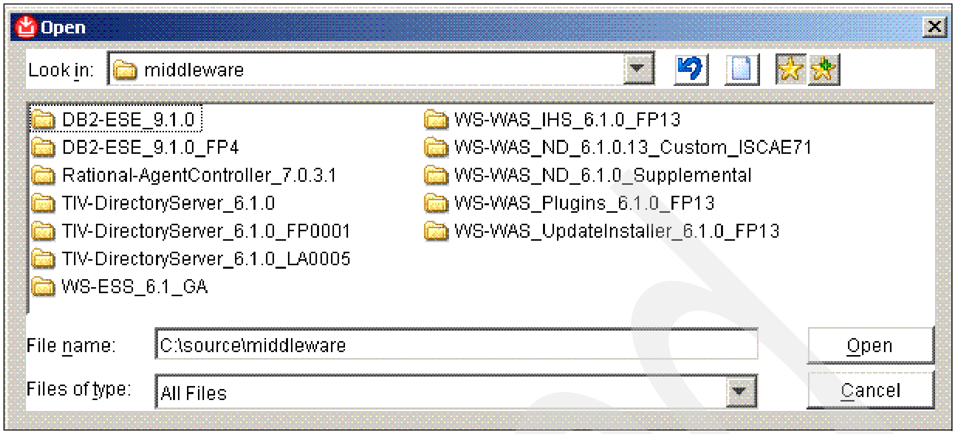


Figure 4-32 Location of the images

34. The installation program will use a temporary location for the installation. Chose the default location or locate one of your choice. Click **Next** (Figure 4-33).

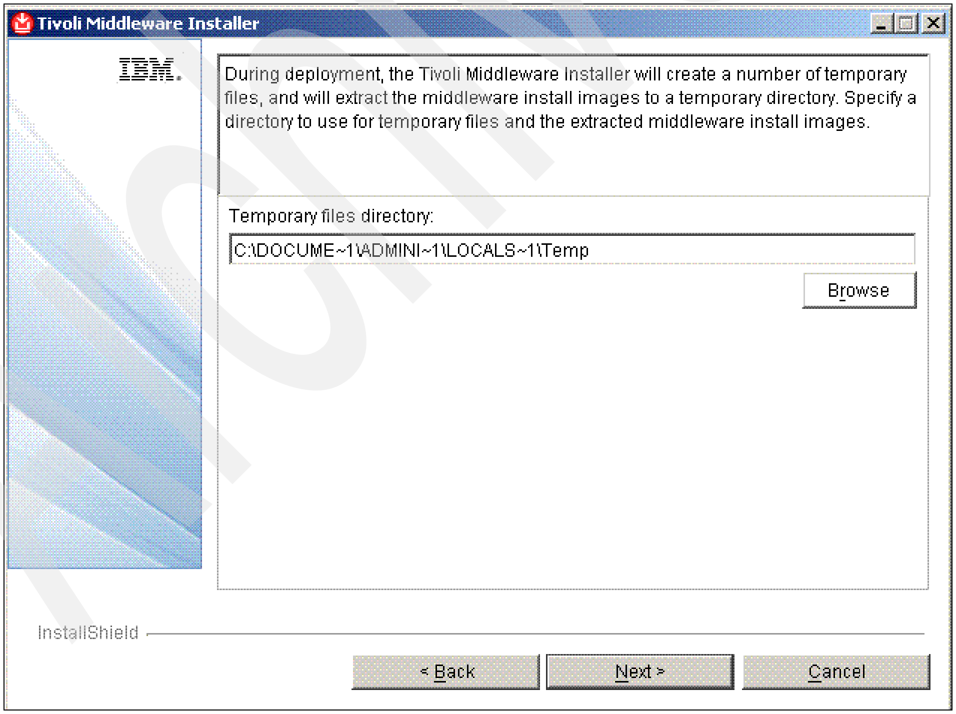


Figure 4-33 Temporary files directory

35.Click **Next** (Figure 4-34) to continue with deploying the plan.

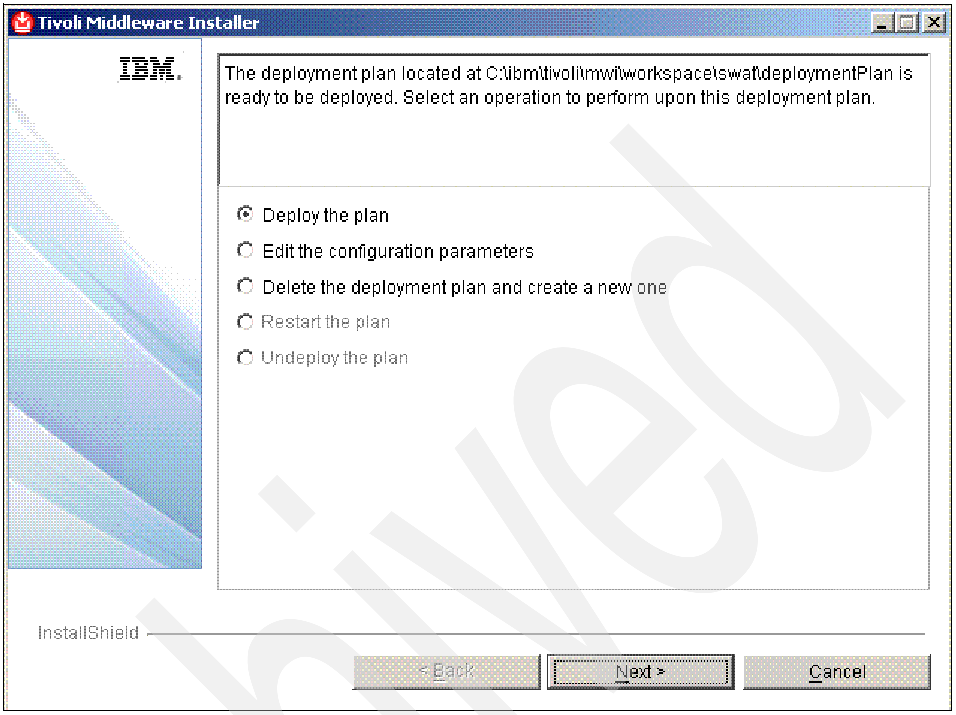


Figure 4-34 Deploy the plan

Note: The first time you run the installation, the default option **Deploy the plan** will be selected, while during subsequent installations, you will be allowed to select the other options, such as for deleting the plan or editing the plan.

36. The installation will now perform disk space checks. Click **Next** (Figure 4-35).

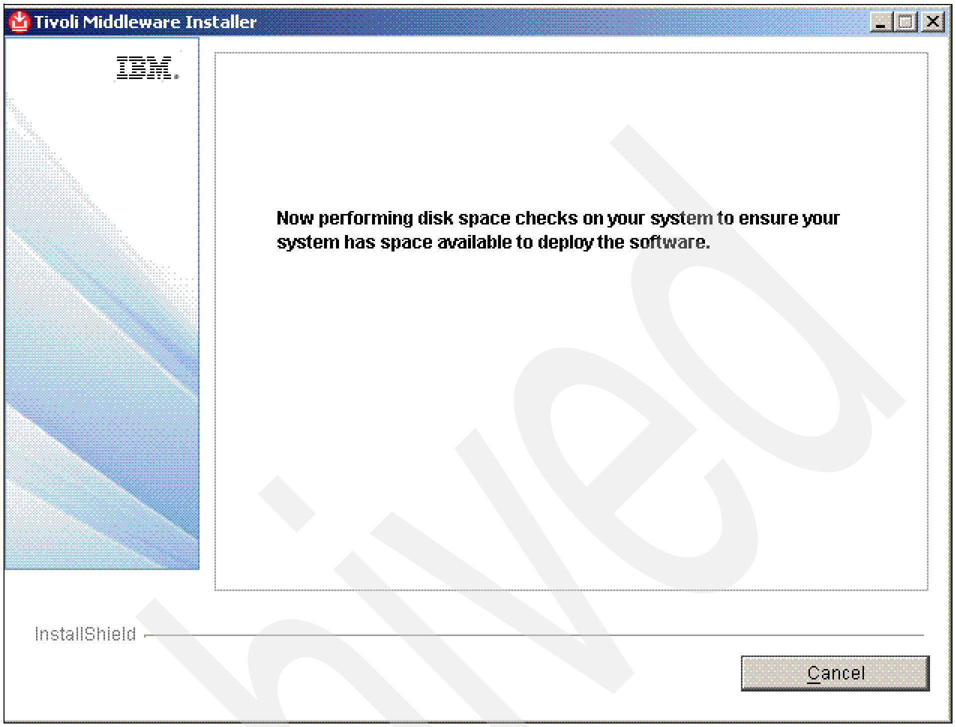


Figure 4-35 Disk space checks

37. Once the installation program has verified that enough disk space is available, you are presented with the results of the disk space checks. Click **Next** (Figure 4-36).

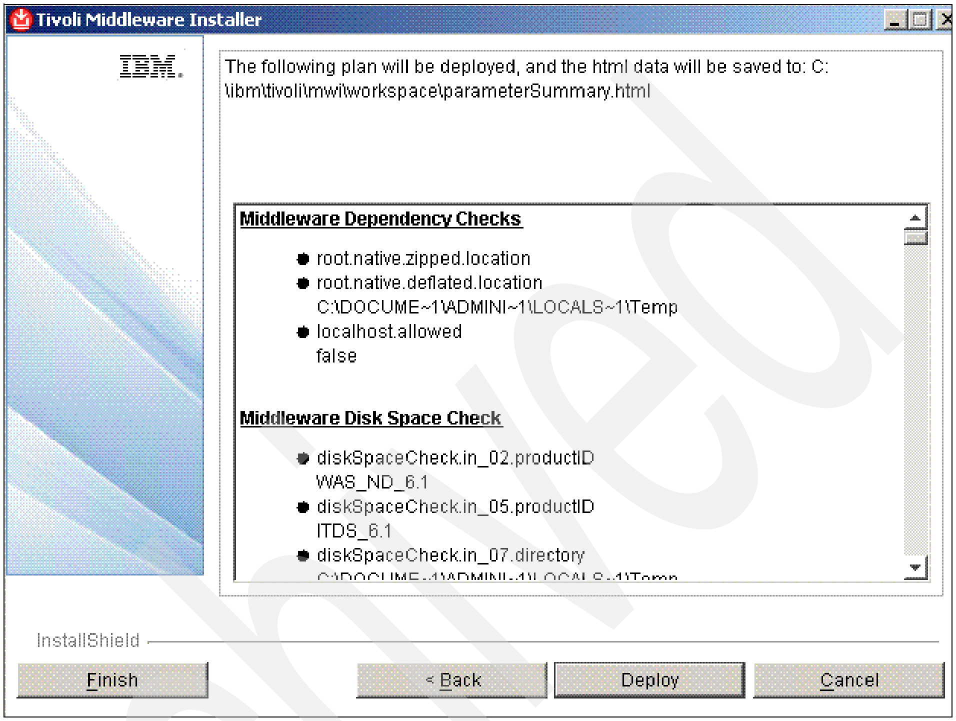


Figure 4-36 Disk space checks

38. You are now taken through the middleware installation process and notified about the different stages of the progress. Click **Next** (Figure 4-37).

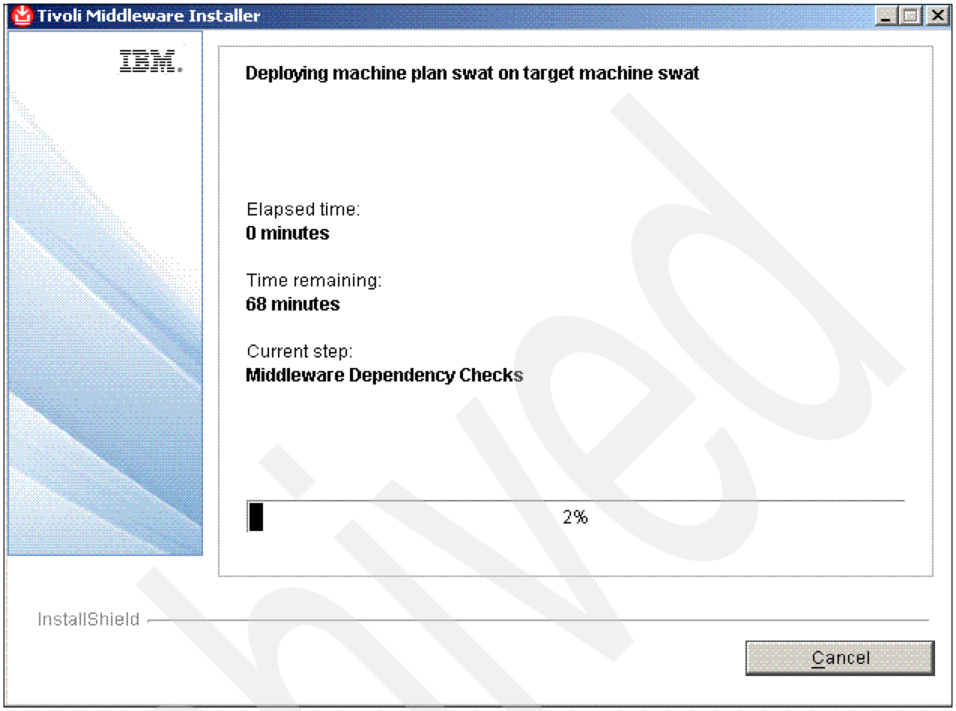


Figure 4-37 Progress bar

39.The installation proceeds with the installation of the IBM Rational Agent Controller. Click **Next** (Figure 4-38).

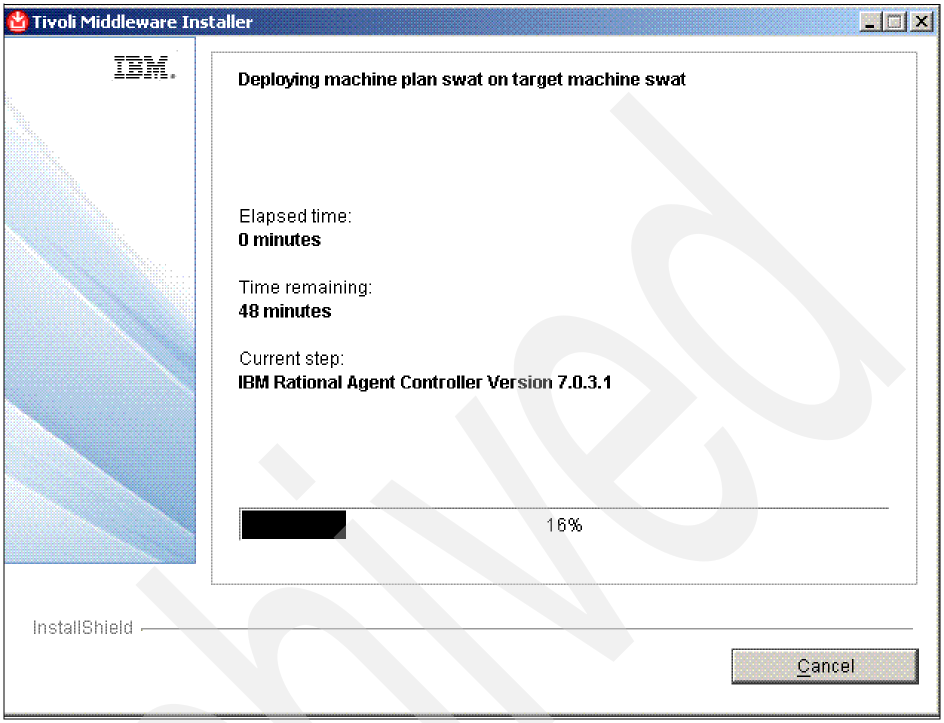


Figure 4-38 Installation of the IBM Rational Agent Controller

40. The installation proceeds with the installation of the IBM DB2 Enterprise Server Edition. Click **Next** (Figure 4-39).

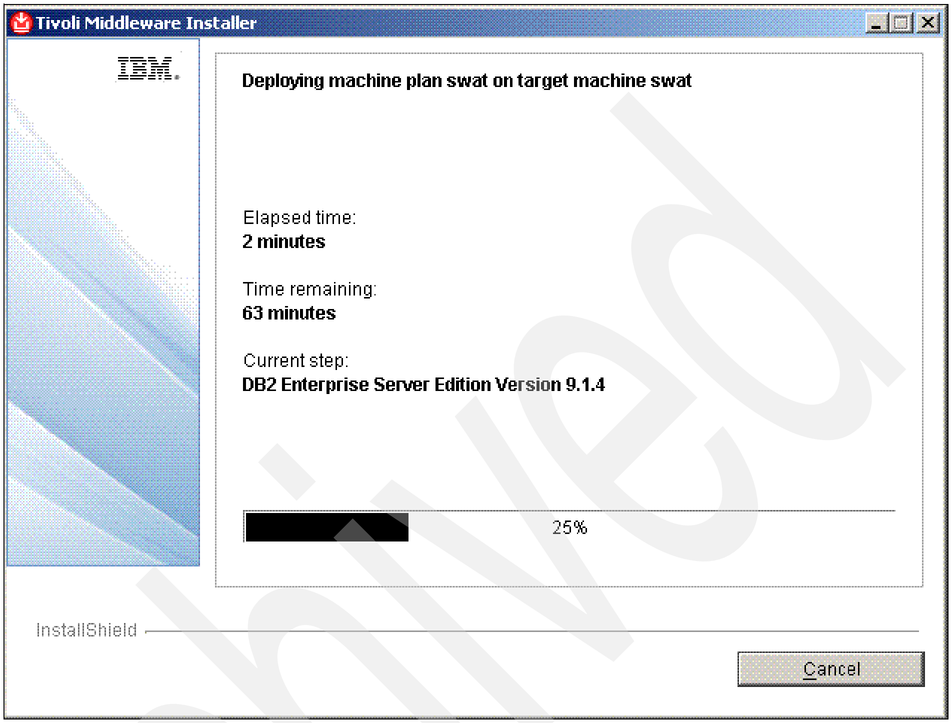


Figure 4-39 Installation of the IBM DB2 Enterprise Server Edition

41. The installation proceeds with the installation of the IBM DB2 Enterprise Server Edition configuration. Click **Next** (Figure 4-40).

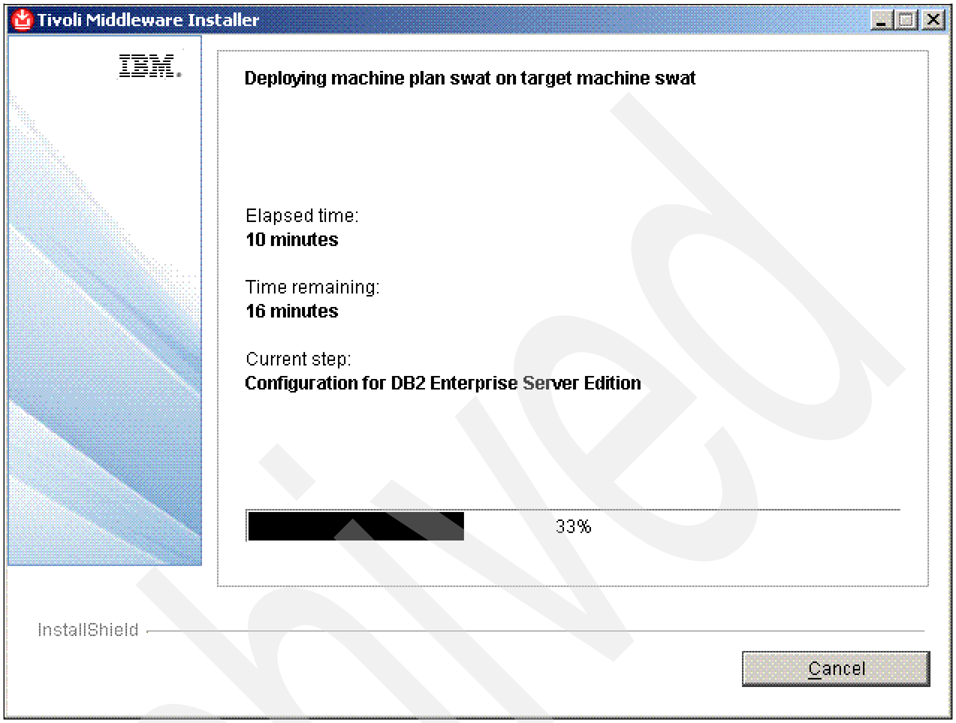


Figure 4-40 IBM DB2 Enterprise Server Edition configuration

42. The installation proceeds with the installation of the IBM Directory Server installation. Click **Next** (Figure 4-41).

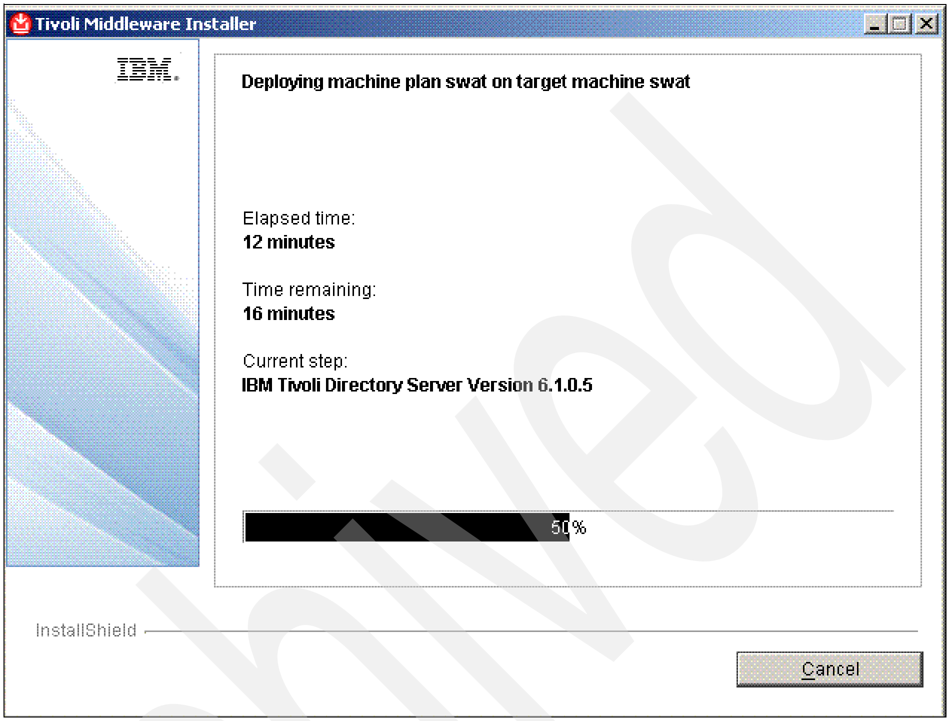


Figure 4-41 IBM Directory Server installation

43. The installation proceeds with the installation of the IBM Directory Server configuration. Click **Next** (Figure 4-42).

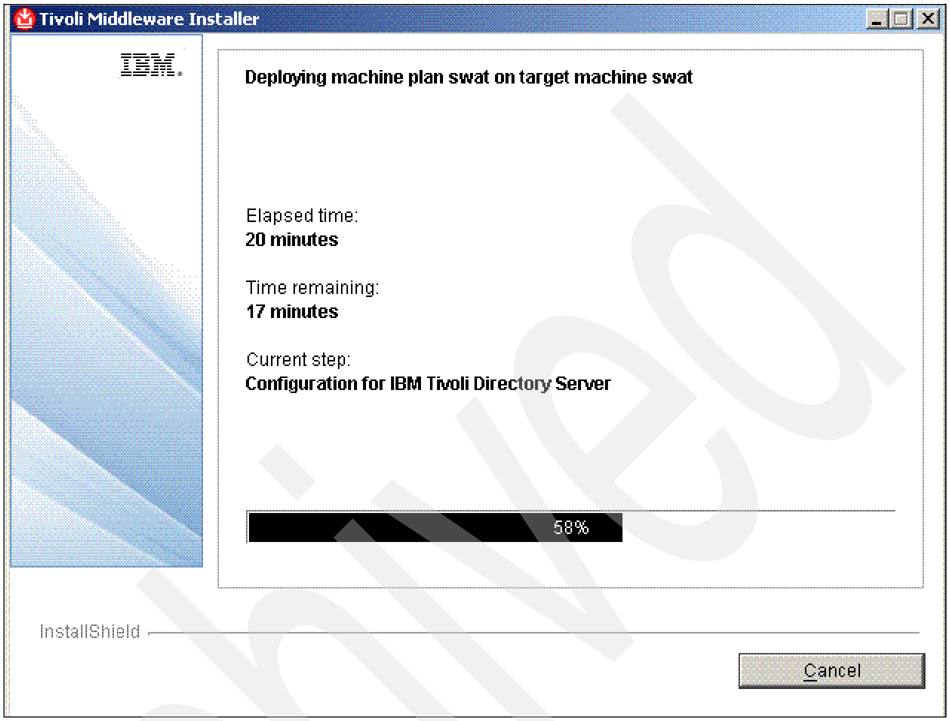


Figure 4-42 IBM Directory Server configuration

44. The installation proceeds with the installation of the IBM WebSphere Application Server installation. Click **Next** (Figure 4-43).

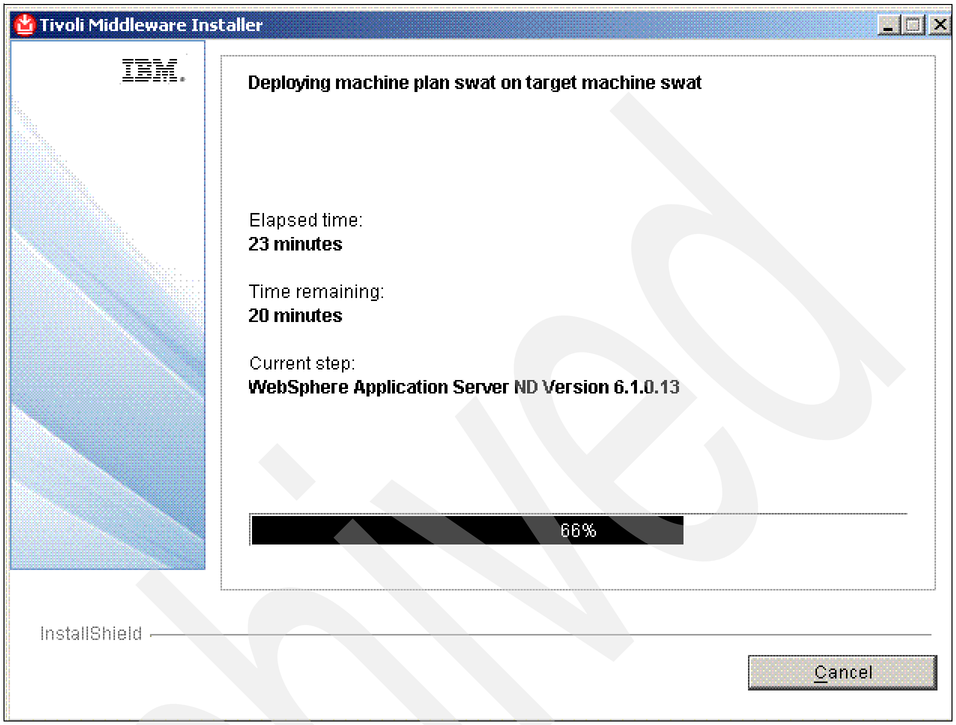


Figure 4-43 IBM WebSphere Application Server installation

45. The installation proceeds with the installation of the IBM WebSphere Application Server configuration. Click **Next** (Figure 4-44).

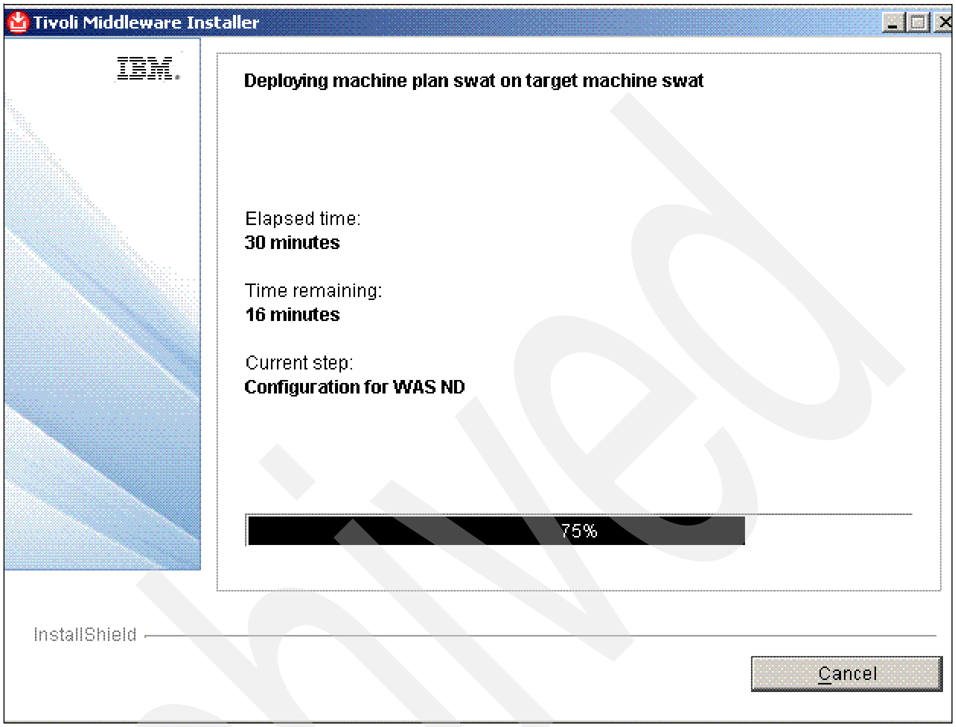


Figure 4-44 IBM WebSphere Application Server configuration

46. The installation proceeds with the installation of the IBM HTTP server. Click **Next** (Figure 4-45).

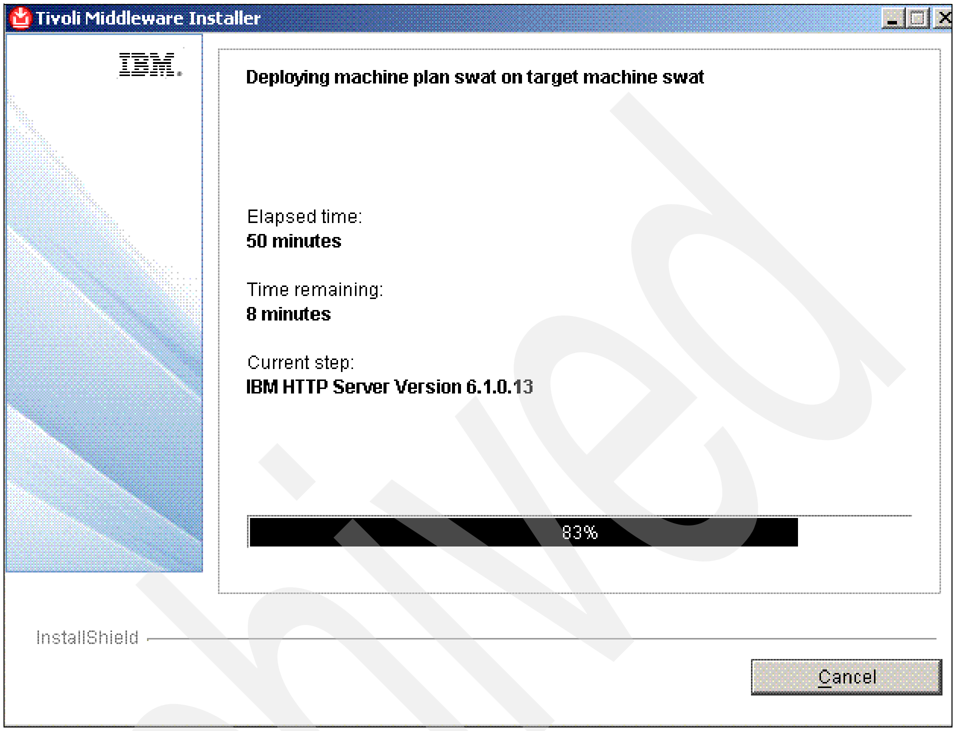


Figure 4-45 Installation of the IBM HTTP server

47. Once all the middleware applications have been installed, you are presented with a window summarizing the installation. Click **Next** (Figure 4-46).

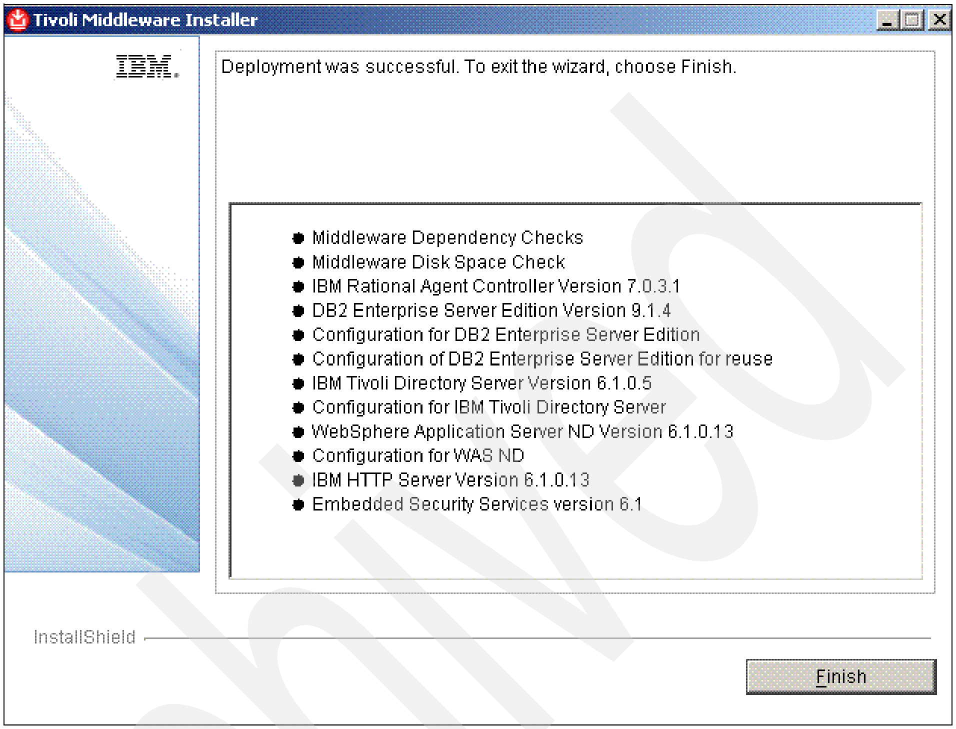


Figure 4-46 Summary of installations

48. Once all the middleware applications have been installed you are presented with a window summarizing the installation. Click **Finish** to close the window.

4.3.2 Middleware installation on Linux

In this section, the middleware installation will be performed on a Linux platform.

1. To start the middleware installation on the Linux machine, navigate to the directory where the launchpad is located and type **`./launchpad.sh`**, as shown in Figure 4-47.

Figure 4-47 Starting the launchpad

The IBM Tivoli Service Request Manager V7.1 window will be displayed, as shown in Figure 4-48.

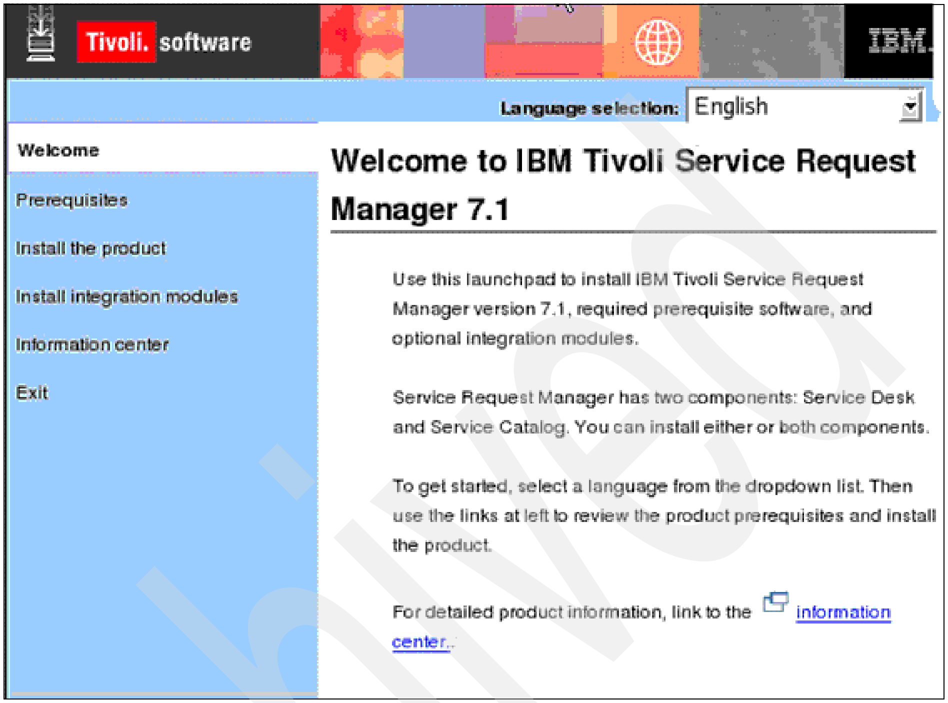


Figure 4-48 IBM Tivoli Service Request Manager V7.1 window

2. Select the **Install the product** option on the left and press Enter (Figure 4-49).

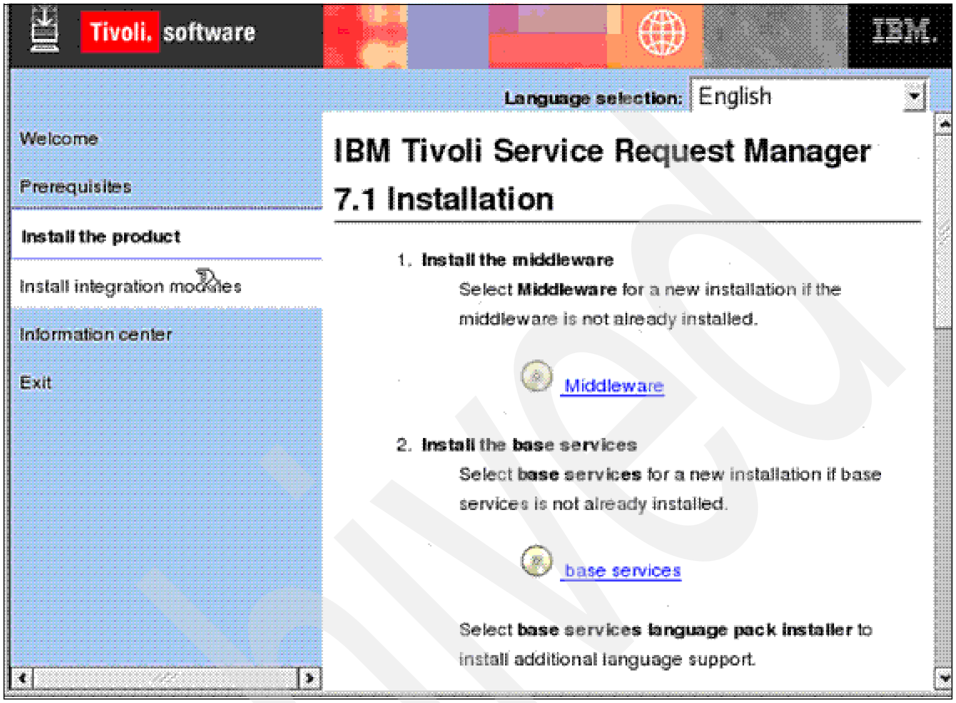


Figure 4-49 Install the product option

3. Select the **Install the product** option on the left and press Enter. Click the **Middleware** option on the right. The middleware installation is initiated.
4. Select the language for the installation and then click **OK** to continue (Figure 4-50 on page 161).

Figure 4-50 Language for the installation

5. At the Welcome window, click **Next** (Figure 4-51) to continue.

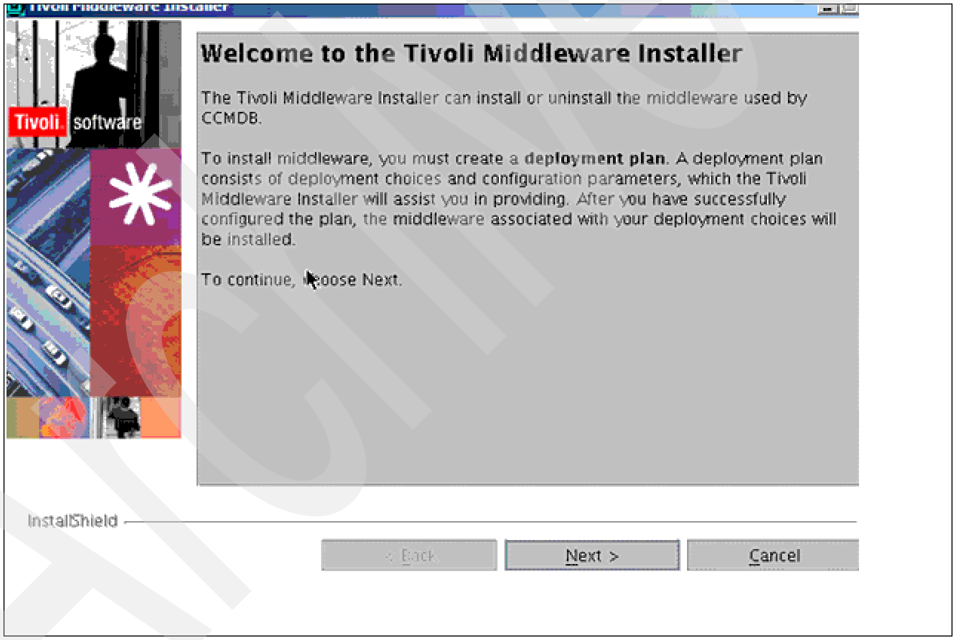


Figure 4-51 Welcome window

6. The Tivoli middleware installer license agreement window is displayed. Read the license information and select **I accept both the IBM and the non-IBM terms** if you agree with the terms. Click **Next** (Figure 4-52).

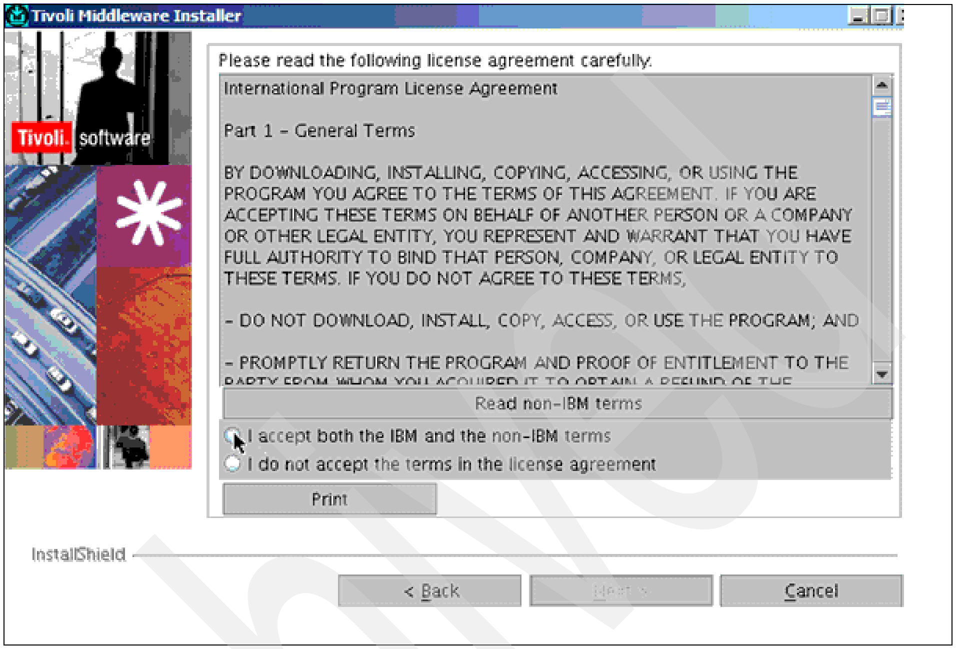


Figure 4-52 License agreement

7. You must specify the location of the middleware installation images. Click **Next** (Figure 4-53).

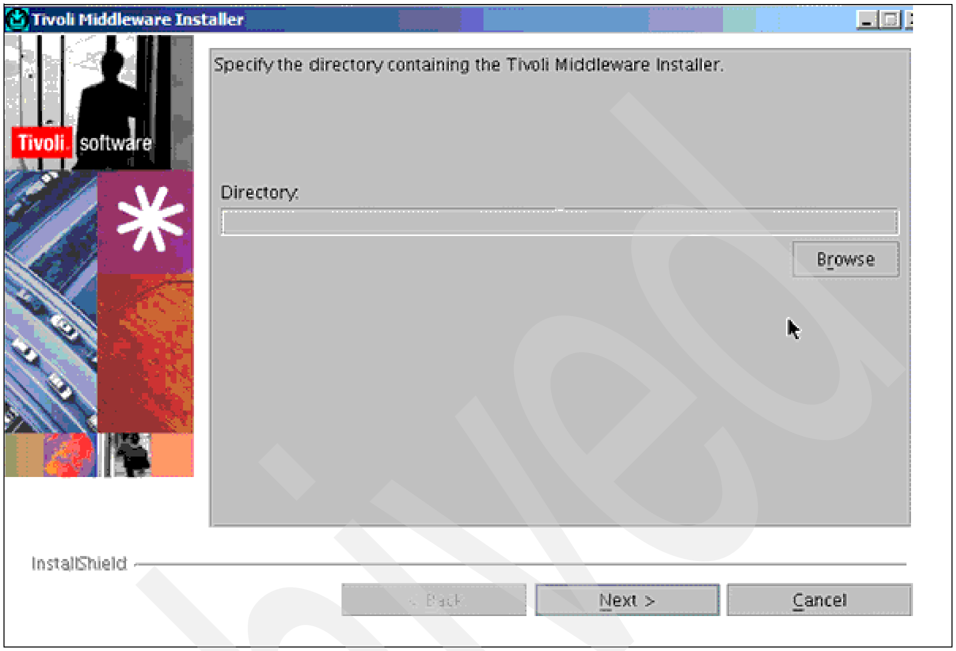


Figure 4-53 Location of middleware installer

8. Browse to the directory where the middleware installer is located and click **Open** (Figure 4-54).

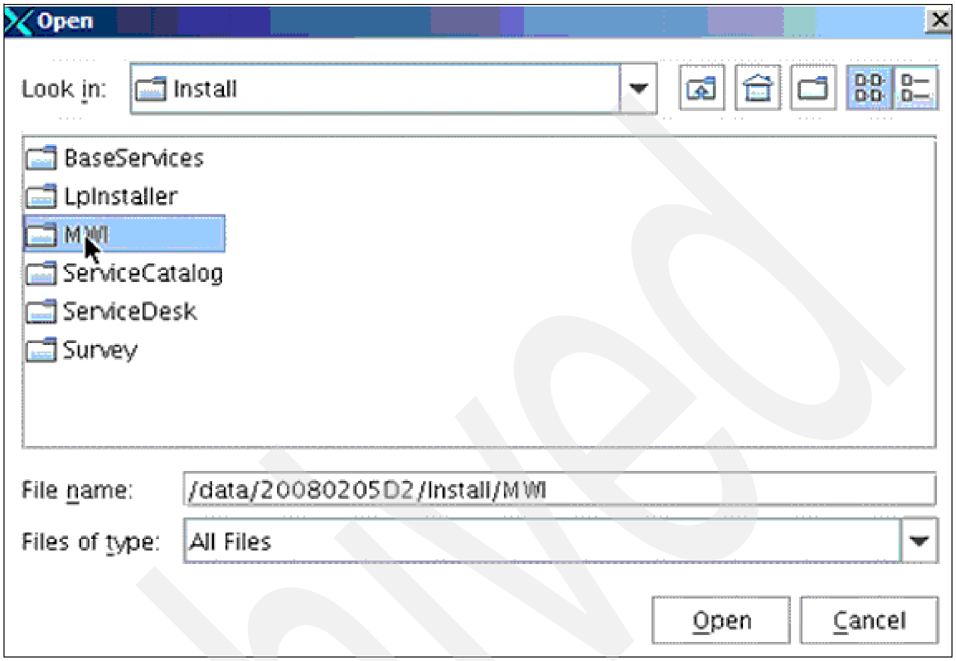


Figure 4-54 Tivoli Middleware Installer

Note: The location of the middleware installer might differ from the path described in this book.

9. You have now selected the directory for the middleware installer. Click **Next** (Figure 4-55) to continue.

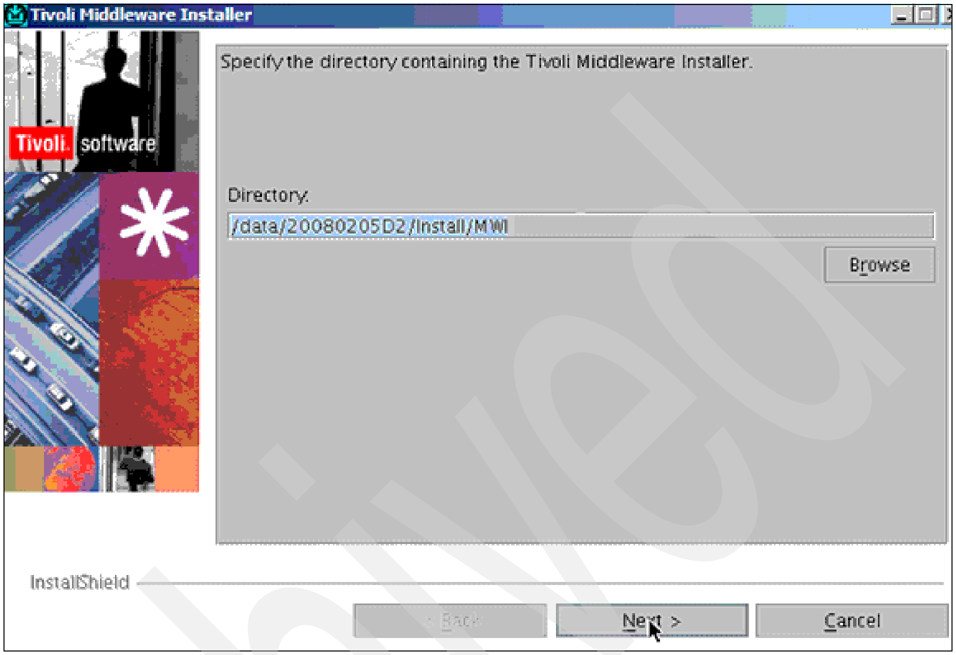


Figure 4-55 Tivoli Middleware Installer

10. The deployment choices and the configuration parameters are now stored in the workspace directory. You can use a different directory to store these choices and parameters. Click **Next** (Figure 4-56) to continue.

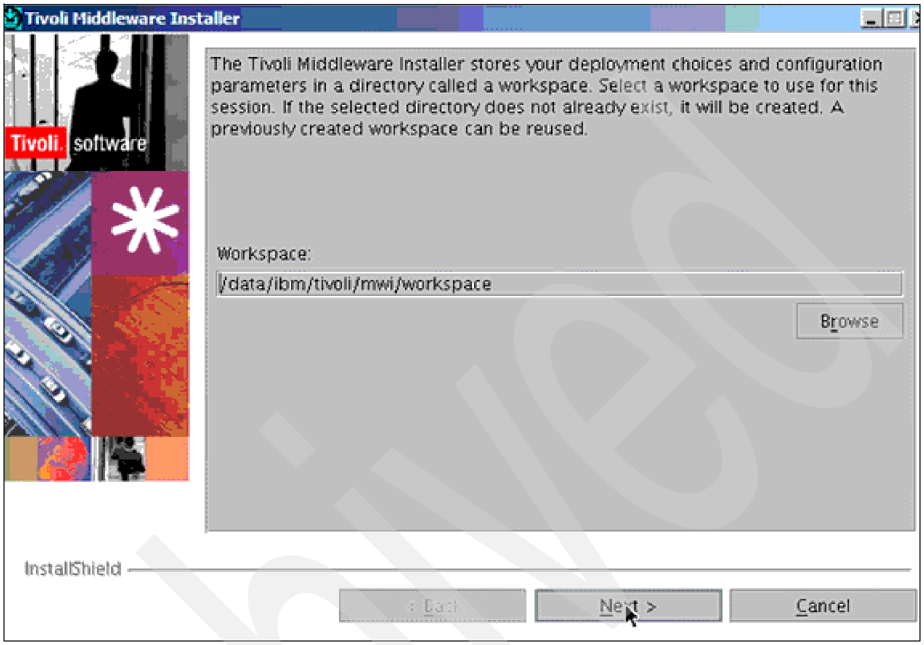


Figure 4-56 Workspace

11. Click **Next** (Figure 4-57) to continue with the installation of the IBM Autonomic Deployment Engine.

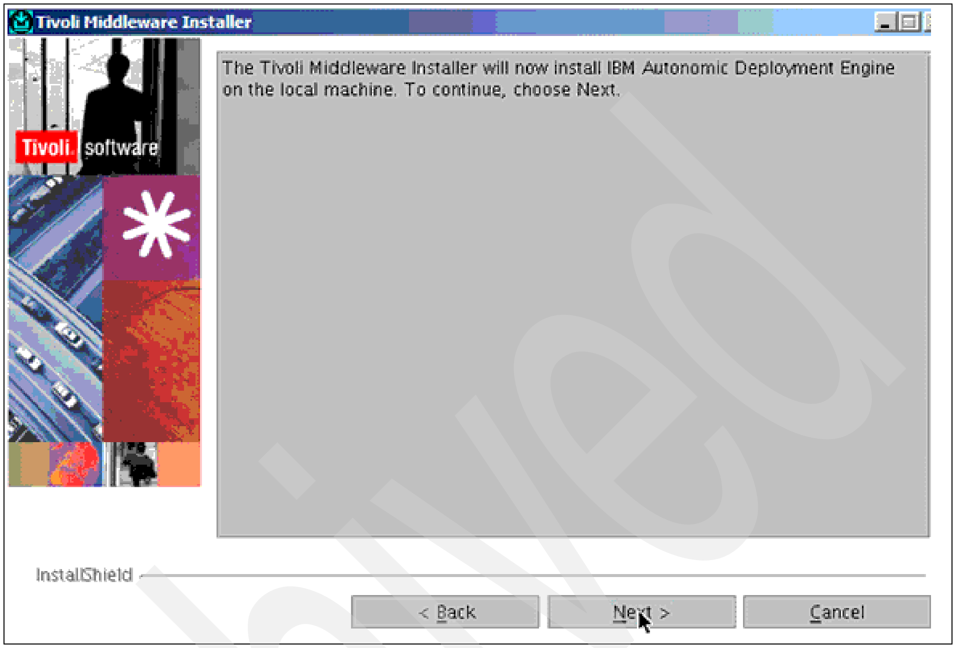


Figure 4-57 IBM Autonomic Deployment Engine

12. The installation program now looks for software instances that can be used for the current installation. Click **Next** (Figure 4-58).

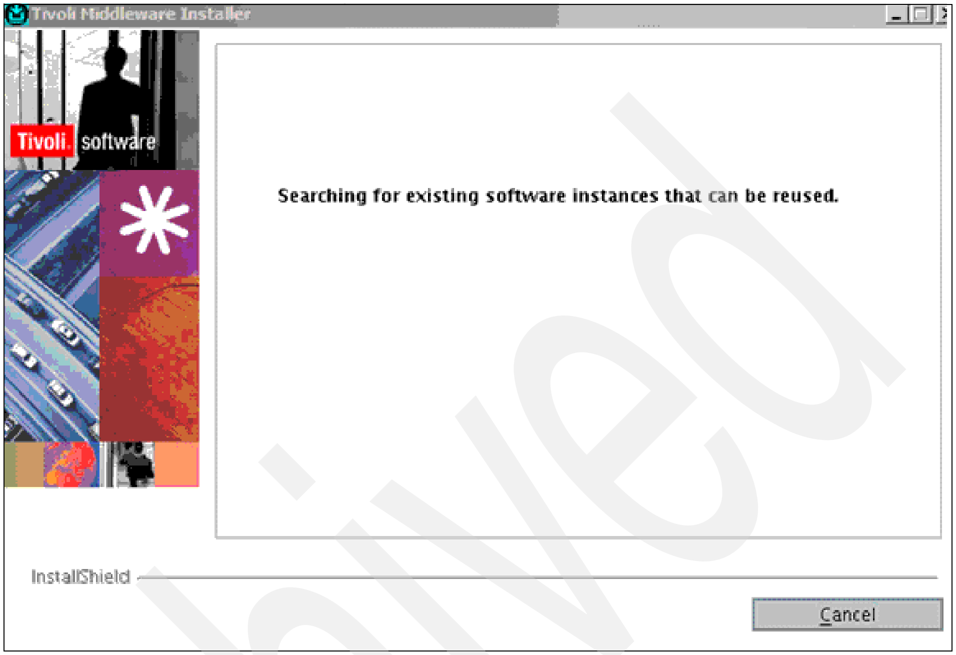


Figure 4-58 Software instances

13.Select all the options and click **Next** (Figure 4-59) to continue.

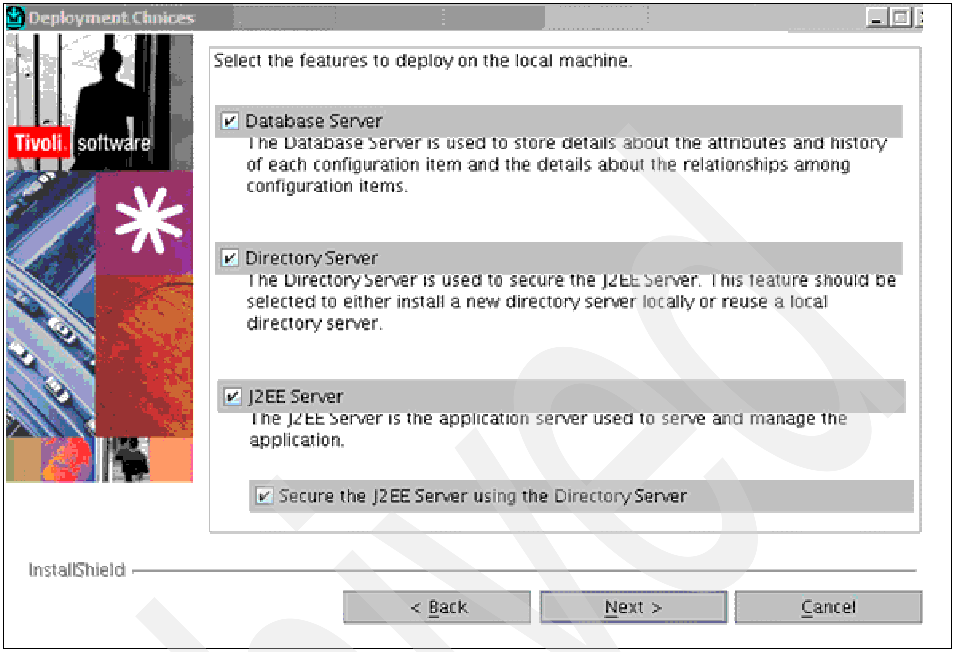


Figure 4-59 Deployment Choices

The installation program will now generate the deployment plan, as shown in Figure 4-60.

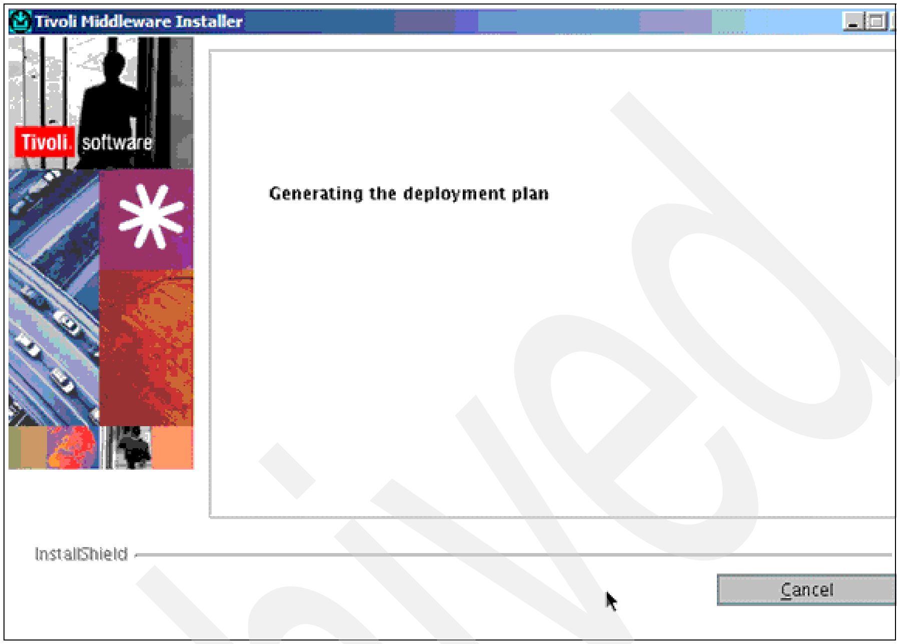


Figure 4-60 Deployment plan

14. Click **Next** to proceed with the configuration parameters for this plan (Figure 4-61).

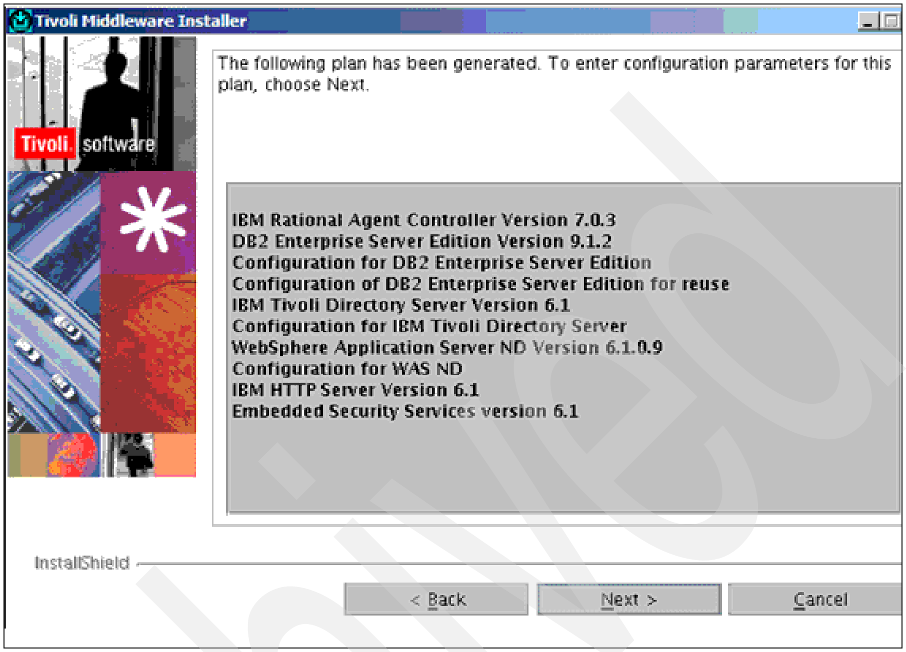


Figure 4-61 Plan to be generated

15. You are now presented with the option of using the same password for the subsequent password options. Should you need to use different passwords, you will not select **Use this password as the value for all subsequent passwords**. Click **Next** (Figure 4-62) to continue.

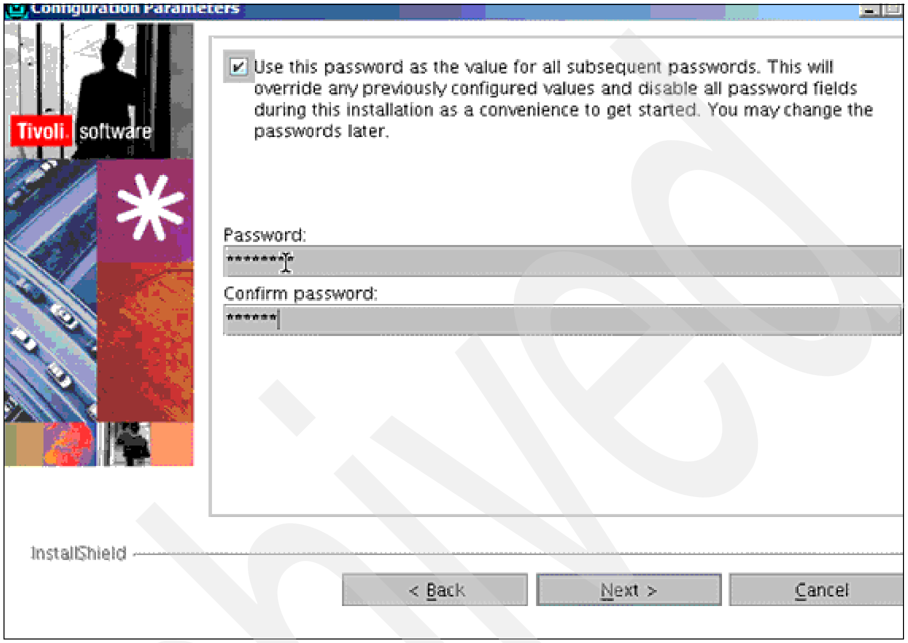


Figure 4-62 Configuration Parameters

16. The installation program provides the parameters by default. Fill in the requested parameters if they differs from the listed parameters. If you have previously selected to use the password as the value for all subsequent passwords, then this field will be already filled. Click **Next** (Figure 4-63).

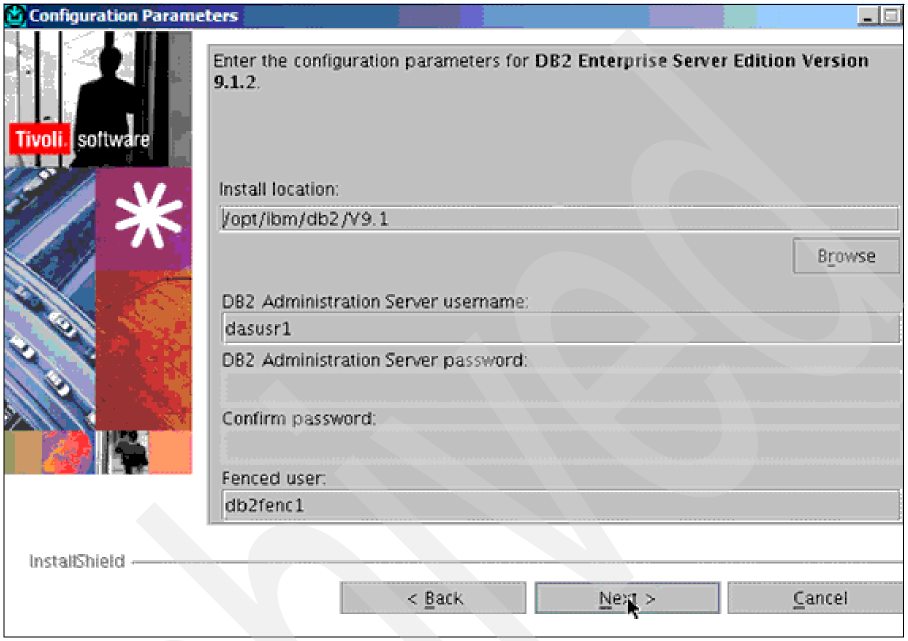


Figure 4-63 Configuration Parameters

17. Accept the default values and click **Next** (Figure 4-64).

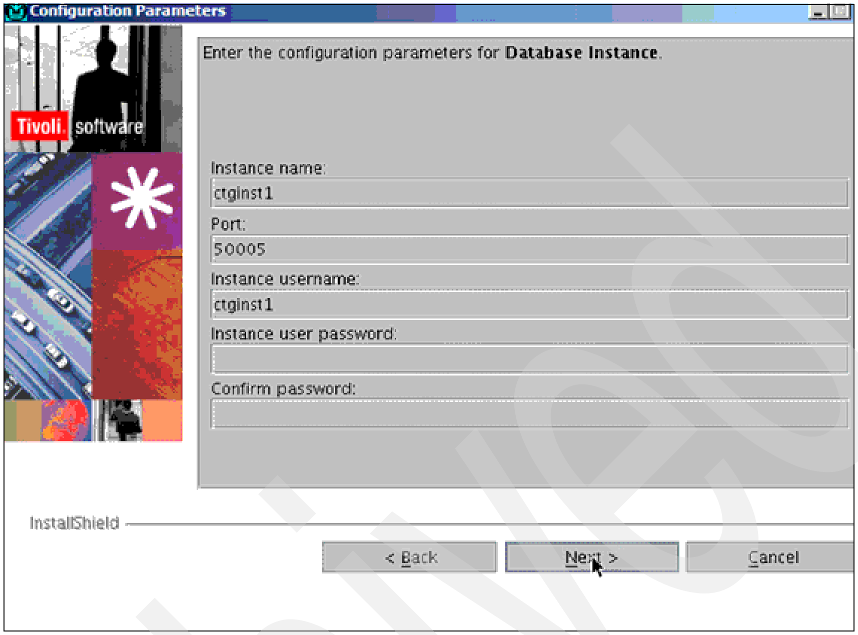


Figure 4-64 Configuration Parameters

18. Accept the default values and click **Next** (Figure 4-65).

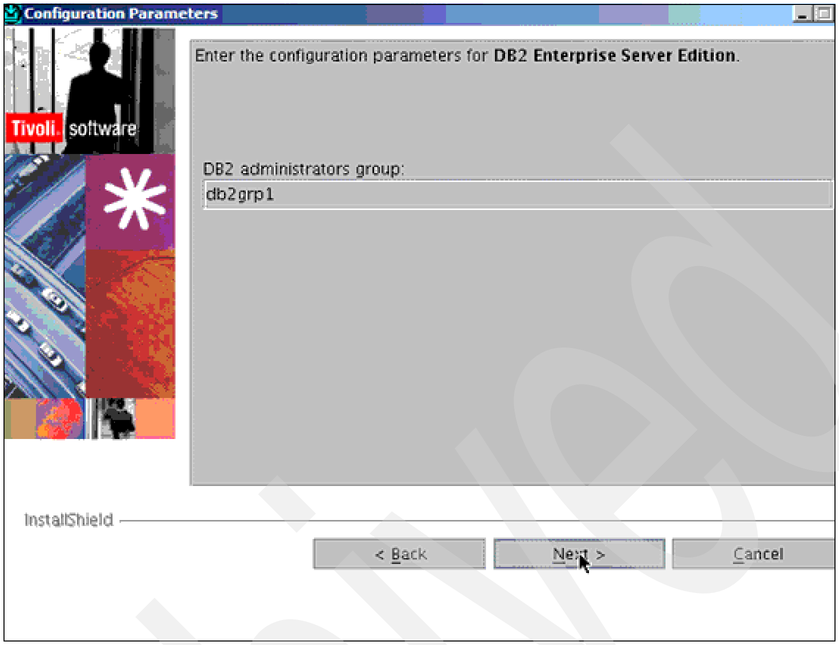


Figure 4-65 Configuration Parameters

19. Accept the default values and click **Next** (Figure 4-66).

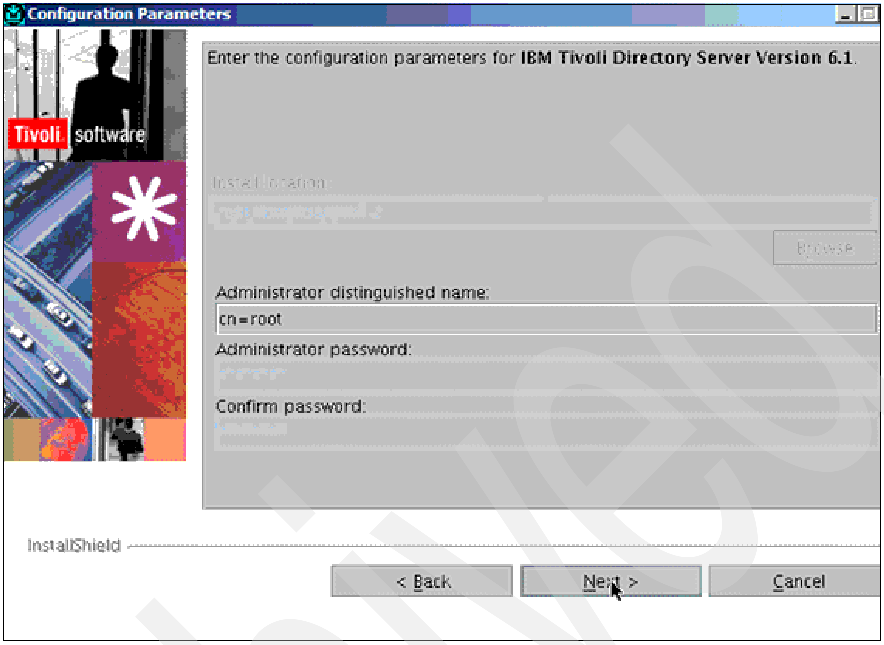


Figure 4-66 Configuration Parameters

20.Accept the default values and click **Next** (Figure 4-67).

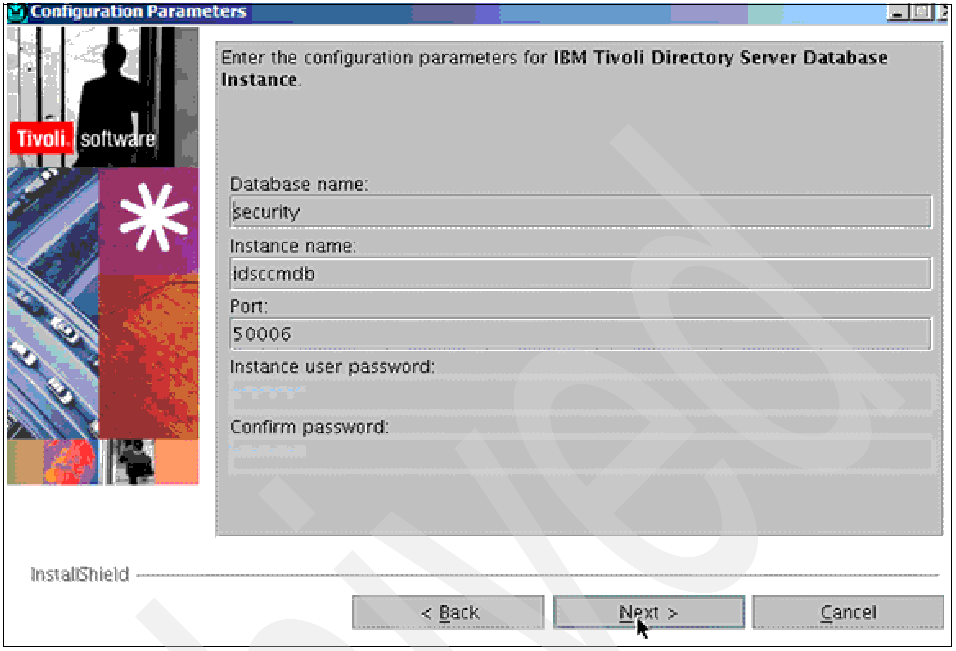


Figure 4-67 Configuration Parameters

21. Accept the default values and click **Next** (Figure 4-68).

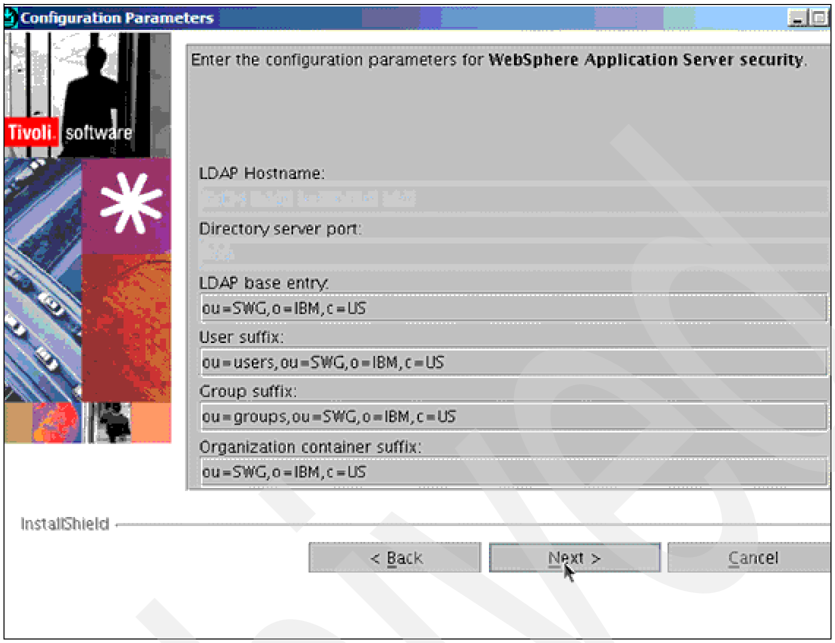


Figure 4-68 Configuration Parameters

22. Accept the default values and click **Next** (Figure 4-69).

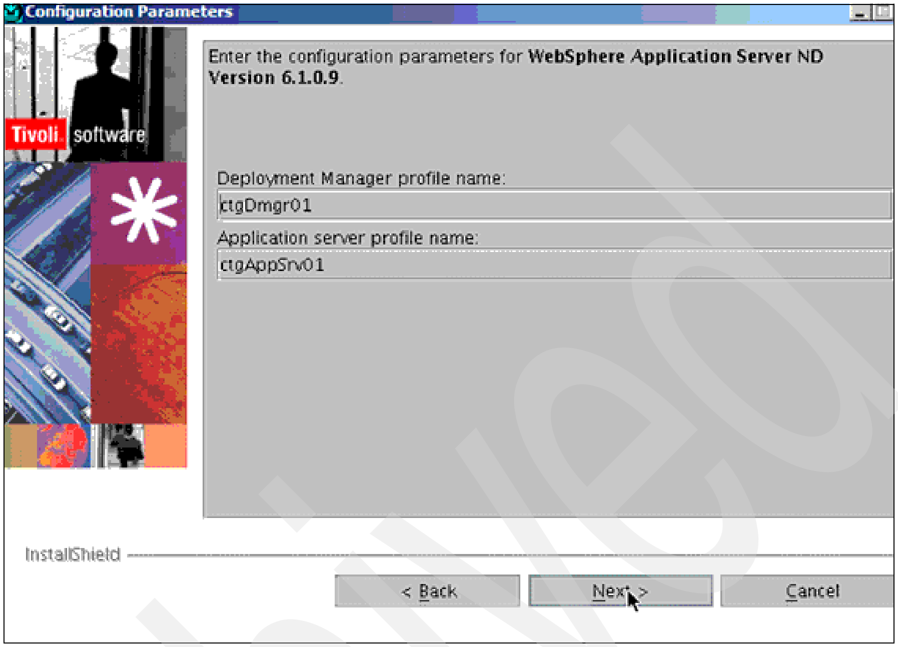


Figure 4-69 Configuration Parameters

23. Accept the default values and click **Next** (Figure 4-70).

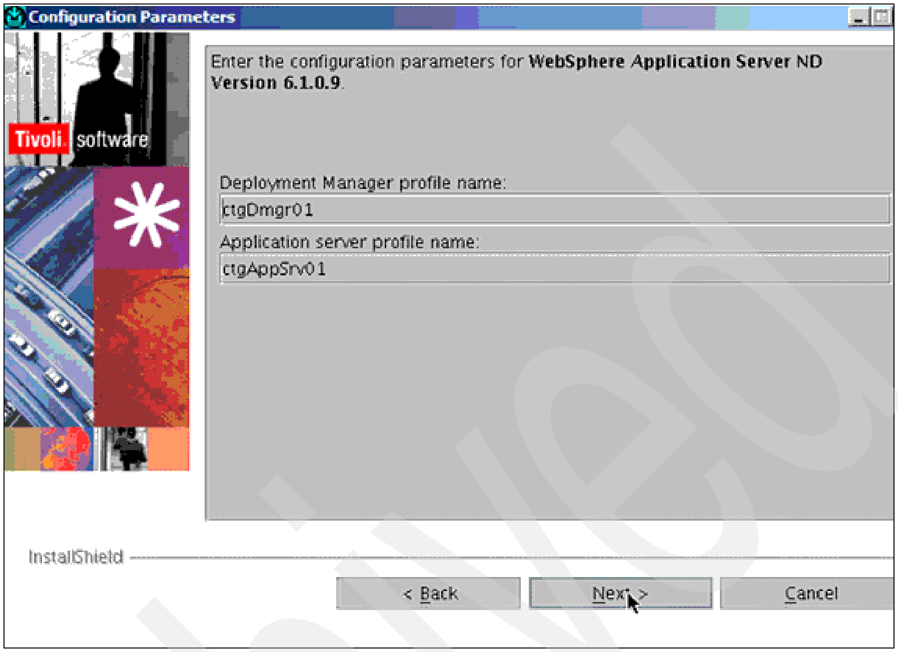


Figure 4-70 Configuration Parameters

24. Accept the default values and click **Next** (Figure 4-71).

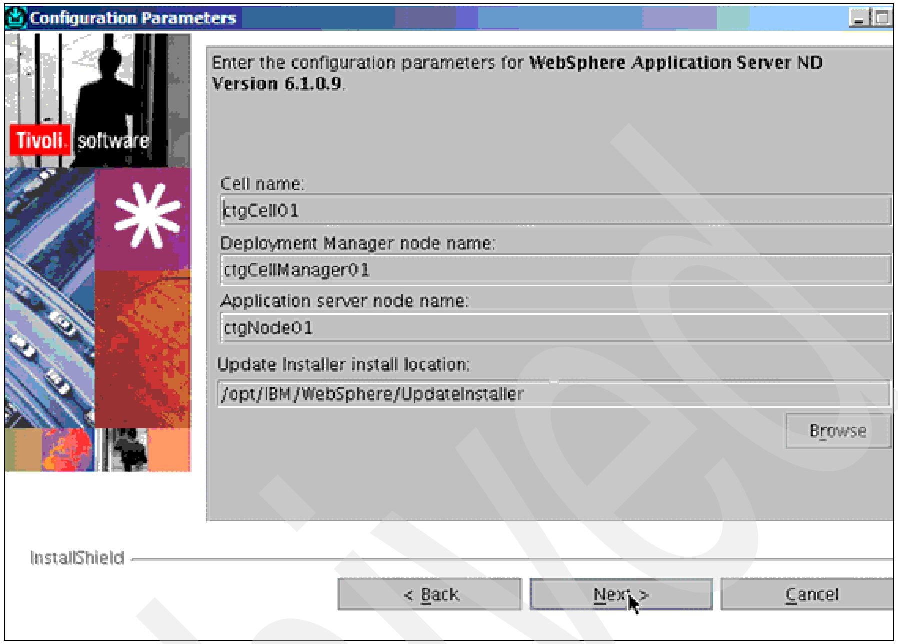


Figure 4-71 Configuration Parameters

25. Accept the default values and click **Next** (Figure 4-72).

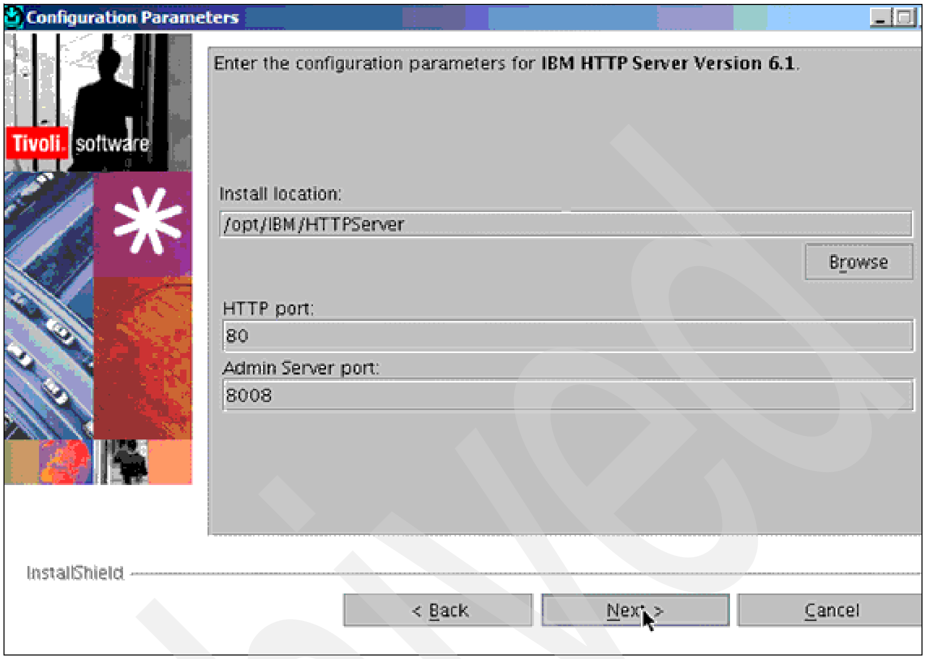


Figure 4-72 Configuration Parameters

26. Accept the default values and click **Next** (Figure 4-73 on page 183).

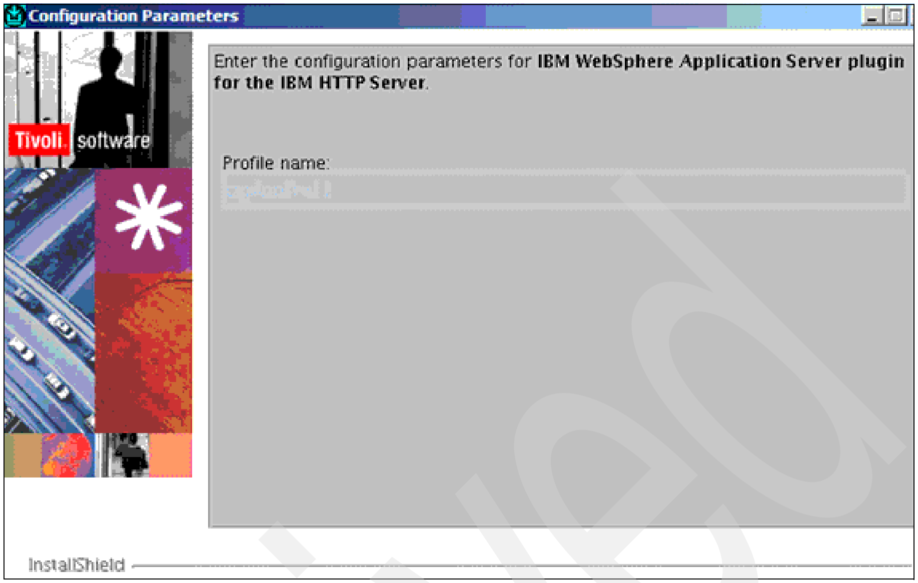


Figure 4-73 Configuration Parameters

27. Accept the default values and click **Next** (Figure 4-74).

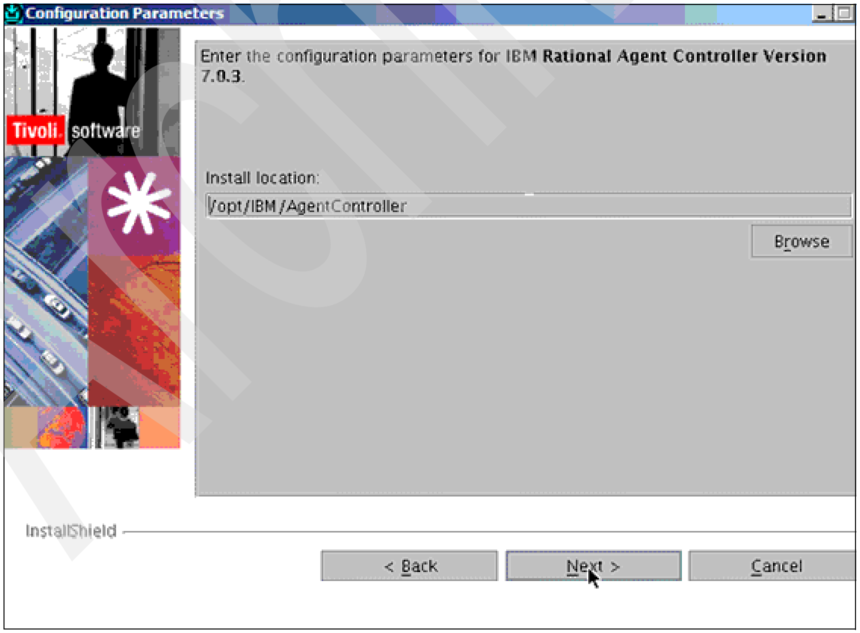


Figure 4-74 Configuration Parameters

28. You must locate the directory containing all the software images to proceed with the installation. You can either copy these files from a source, such as a CD, or just point to an existing directory containing the images. In our case, we specify an existing directory containing all the images. Click **Next** (Figure 4-75).

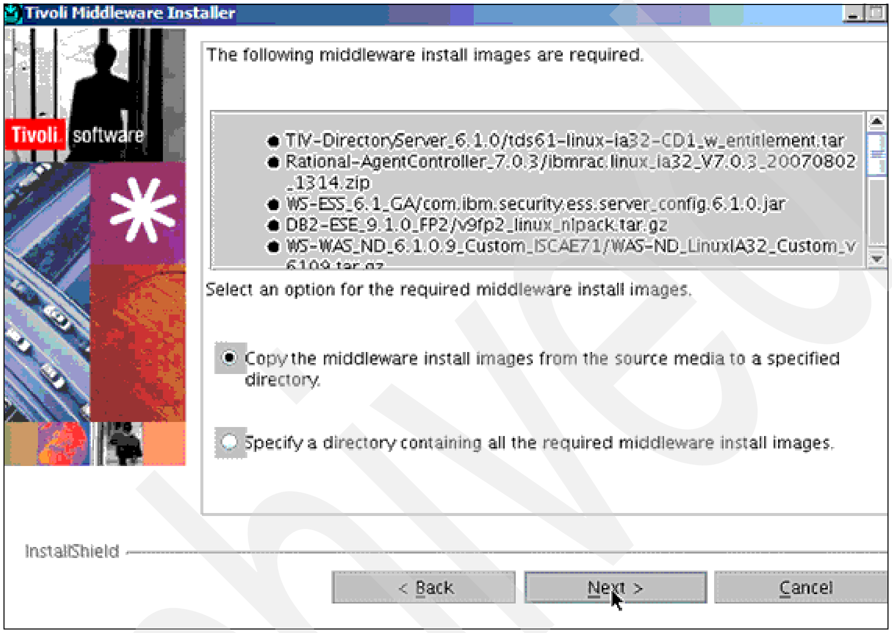


Figure 4-75 Tivoli Middleware Installer

29. Select the directory for the directory containing the middleware images. Click **Next** to continue (Figure 4-76 on page 185).

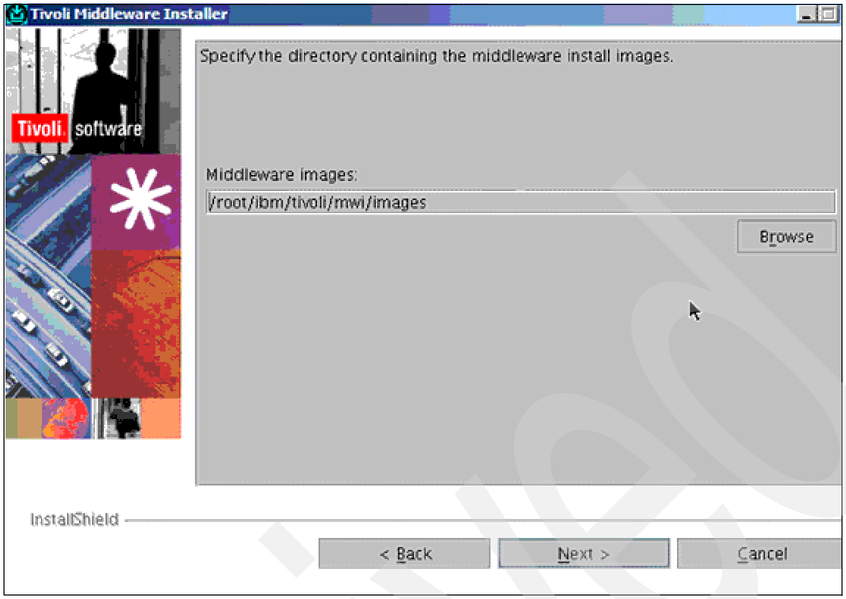


Figure 4-76 Middleware install images

30. Browse through directory containing the middleware images and locate the middleware installation images. Click **Next** (Figure 4-77).

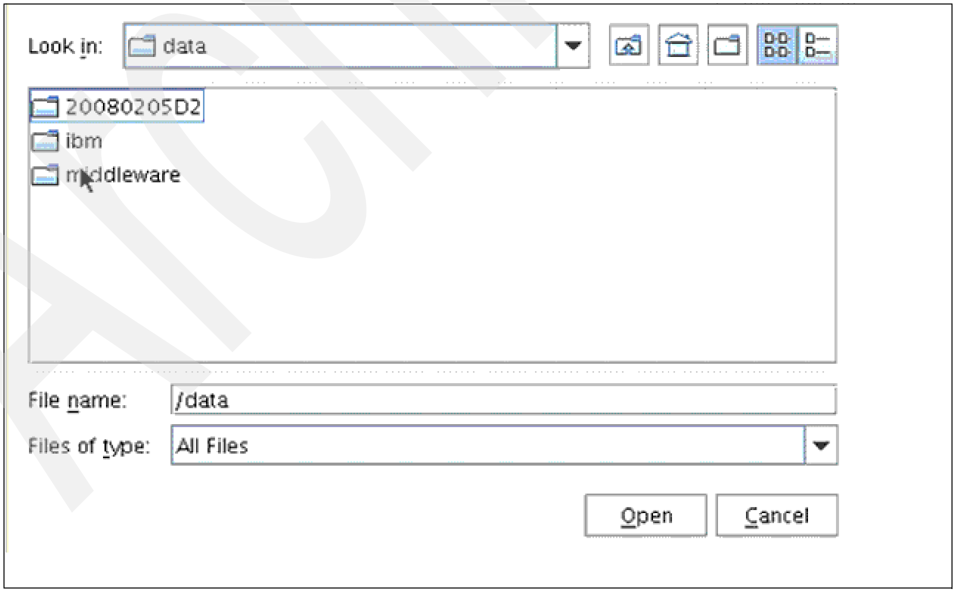


Figure 4-77 Middleware install images

31.Click **Next** to select /data/middleware (your images might be in a different location). See Figure 4-78.

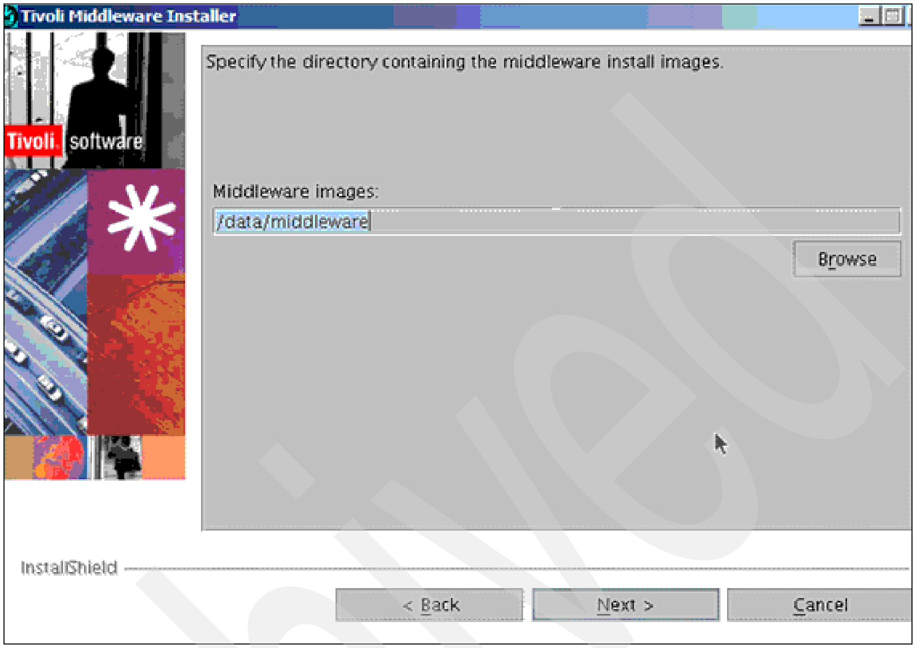


Figure 4-78 Middleware install images

32.A temporary directory will be used for the installation program. You can keep the default tmp directory or select an alternate temporary directory and then click **Next** to continue with the installation (Figure 4-79 on page 187).

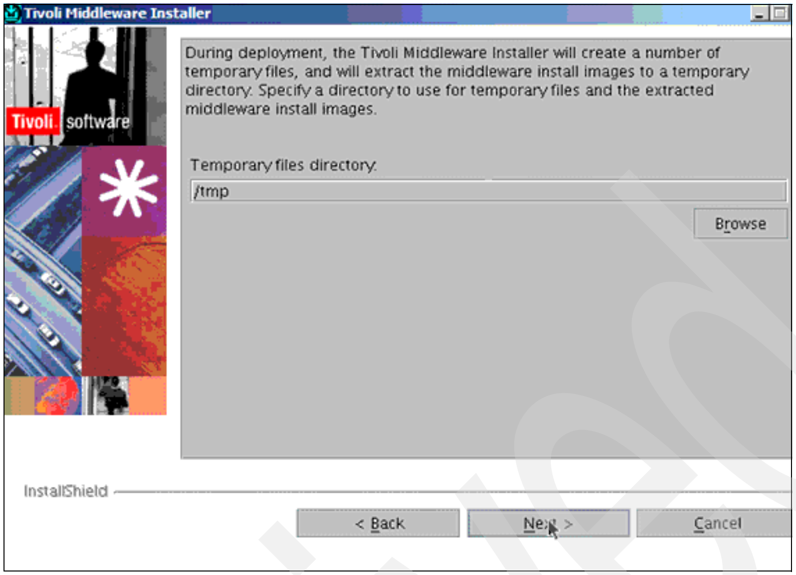


Figure 4-79 Temporary directory

33. Click **Next** to continue with deploying the plan (Figure 4-80).

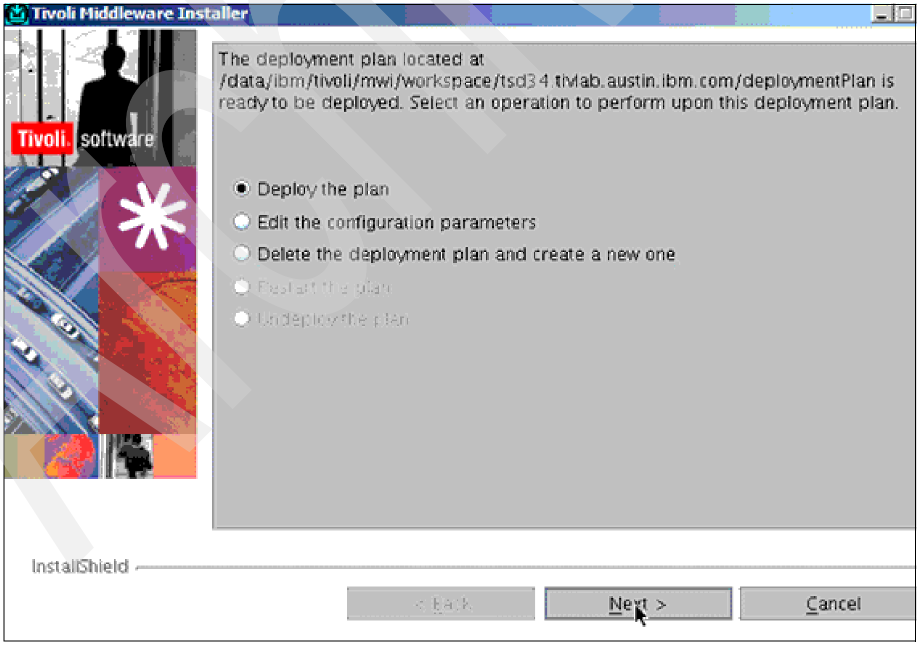


Figure 4-80 Deploy the plan

Note: The first time you run the installation, the default option **Deploy the plan** will be selected, while during subsequent installations you will be allowed to select the other options, such as for deleting the plan or editing the plan.

34. The installation program will now check that you have enough disk space for the software installation. Click **Next** (Figure 4-81).

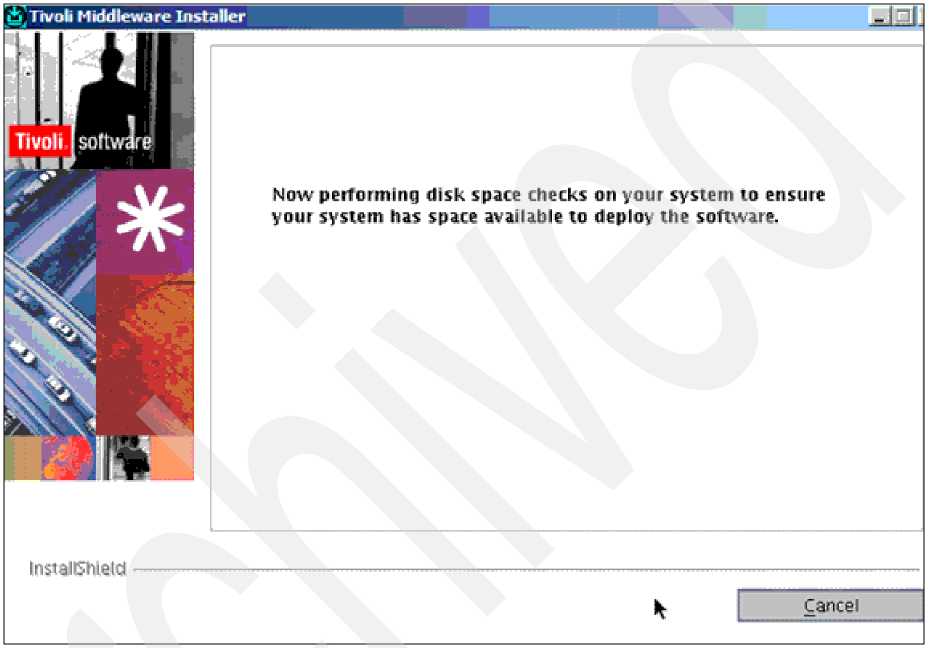


Figure 4-81 Disk space checks

35. The plan is displayed and lists the performed checks. Click **Next** (Figure 4-82).

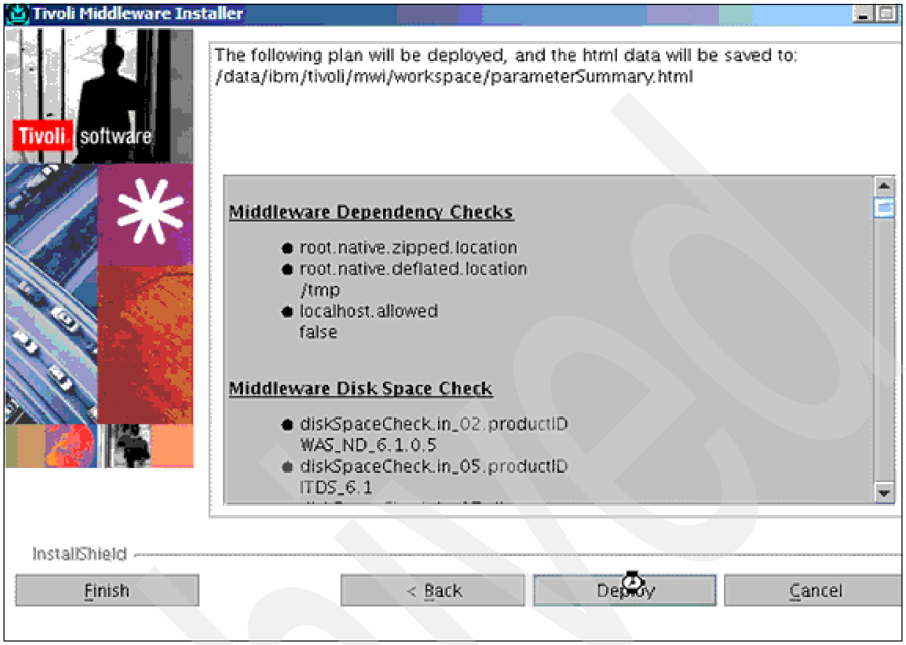


Figure 4-82 Disk space checks

36. As the installation proceeds, you are notified about the progress and the components being installed. Click **Next** (Figure 4-83).

Figure 4-83 Deploying machine plan

> **Note:** The indicated time for the installation might differ depending on your system performance and configuration.

37. Once the installation is complete, click **Finish** to close the window. Click **Next** (Figure 4-84 on page 191).

Figure 4-84 Deploying machine plan - successful completion

38. Once the installation is complete, click **Finish** to close the window.

4.3.3 Log files

The following are the installation log files.

Deployment plan log

This log is found in <Workspace Directory>/histamine/deploymentPlan.

This log file is generated each time the deployment plan is used to install or uninstall middleware products.

Machine plan log

This log is found in <Workspace Directory>/hostname/deploymentPlan/MachinePlan_hostname_processID.

In this log file, you will find the output after the machine plan has been ran.

Deployment plan log

This log is found in <Workspace Directory> /hostname/deploymentPlan/MachinePlan_hostname/stepNum_stepID.

The stepNum is the sequence number of this step in the installation process order.

4.3.4 Base services installation and configuration considerations

The following summarizes base services installation and configuration considerations.

Pre-installation considerations

The Service Request Manager installation program provides an interface for installing and deploying base services. The Service Request Manager installation program records choices you make about your base services deployment and configuration parameters associated with those choices, and then installs and deploys base services based upon the information you entered.

There are two installation paths available to you when installing base services.

Simple

A simple deployment consists of installing all middleware on one system. You will not have the option of using existing middleware within your organization with Service Request Manager. All middleware used in conjunction with Service Request Manager must have been installed on the system using the Tivoli middleware installer using default values. Base services will be installed using default values provided by the Tivoli middleware and Service Request Manager installation programs.

Custom

A custom deployment typically involves deploying Service Request Manager across several systems, some of which might already host middleware products that you wish to use with your Service Request Manager deployment. Deploying through the custom installation path also allows you to modify default installation values.

Note that this deployment option does not require you to spread the Service Request Manager deployment across several systems. You can enter the name of the local host as the destination for all base services components that will be installed using the Tivoli middleware installer and the Service Request Manager installation program.

The Tivoli Service Request Manager installation program can automate the configuration of middleware for use with Tivoli Service Request Manager. If you choose not to have the Tivoli Service Request Manager installation program automatically configure middleware, you will have to configure that piece of middleware manually *prior* to the installation of base services. Note that while you can deploy base services in a distributed environment consisting of predominately AIX and Linux systems, the Service installation program information that you input into the Tivoli Service Request Manager installation program is stored in the maximo.properties file and the Maximo database. These values will be populated into the window fields of the Tivoli Service Request Manager installation program upon subsequent uses of the program. So, if you cancel the installation program after entering values across several installation panels, the installation program will recall the values the next time you start up the Tivoli Service Request Manager installation program. (with the exception of the Tivoli Service Request Manager install directory and the shortcut option chosen). You can restore the default values in the Tivoli Service Request Manager installation program by deleting <Maximo_Home>/applications/maximo/properties/maximo.properties.

Base services install path values

The Simple install path provides you with the values listed in Table 4-2. Where indicated, you will provide the corresponding values.

Table 4-2 Base services install path values

Category	Field	Value	Provided by User?
Deployment Option	Deployment	simple	
Database Configuration	Database Type	DB2	
Database Configuration	Host Name		Yes
Database Configuration	Port	50005	
Database Configuration	Database Name	maxdb71	
Database Configuration	Instance	ctginst1	
Database Configuration	User ID		Yes

Category	Field	Value	Provided by User?
Database Configuration	Automate Database Configuration	yes	
Database Configuration	Remote Access User ID		Yes
Database Configuration	Database Install Directory	► Windows: C:\Program Files\IBM\SQLLIB ► AIX and Linux: /opt/ibm/db2/V9.1	
Database Configuration	Instance Administrative User ID	► Windows: db2admin ► AIX and Linux: ctginst1	Yes
Database Configuration	Windows Service User ID	db2admin	
Database Configuration	Data table space name	maxdata	
Database Configuration	Data table space size	medium	
Database Configuration	Temporary table space name	maxtemp	
Database Configuration	Temporary table space size	1000	
WebSphere Connectivity	Host name		Yes
WebSphere Connectivity	SOAP Port	8879	

Category	Field	Value	Provided by User?
WebSphere Configuration	WebSphere server home directory	► Windows: C:\Program Files\IBM\WebSphere\AppServer ► AIX and Linux: /opt/IBM/WebSphere/AppServer	
WebSphere Configuration	User Name	wasadmin	Yes
WebSphere Configuration	Profile Name	ctgDmgr01	
WebSphere Configuration	Automate WebSphere Configuration	yes	
WebSphere Configuration	Remote Access User ID		Yes
WebSphere Application Server Configuration	Web server port	80	
WebSphere Application Server Configuration	Web server name	webserver1	
WebSphere Application Server Configuration	Node Name	ctgNode01	
WebSphere Application Server Configuration	Cluster name	MAXIMOCLUSTER	
WebSphere Application Server Configuration	Application server	MXServer (This value cannot be changed.)	
Integration Adapter JMS Configuration	JMS DataSource name	meajmsds	
Integration Adapter JMS Configuration	Persist JMS message	no	

Category	Field	Value	Provided by User?
Security Configuration	Use Default Schema	yes	
Security Configuration	Create Users Automatically	yes	
Maximo Configuration	Install Directory	C:\IBM\maximo	Yes
Configuration Step	Run Configuration Now	yes	

This procedure explains how to use the base services installation program to install base services.

In addition to configuring new instances of middleware products installed by the Tivoli middleware installer, the base services installation program can configure existing instances of prerequisite products, including those from other vendors, that you wish to use with base services.

The instructions provided are for a *multiple machine* installation using default values and assume that you choose to have the installation program automatically configure middleware across multiple machines to work with base services. If you do not allow the installation program to automatically configure middleware, it will still perform programmatic checks to verify that the documented manual steps were performed properly. If any errors are encountered, a dialog box detailing the error will appear. You will not be permitted to continue in the base services installation task until the errors are resolved.

The base services installation program can only be run from a Windows-based system.

Note: Avoid using localhost for host name values in the install program. Specify the actual fully qualified host name of the system for all host name values.

4.3.5 Installing the base services on Windows

To install base services, complete the following steps:

1. If you have not started the Tivoli Service Request Manager launchpad, then:
   a. Log on as an administrator.
   b. Insert the Tivoli Service Request Manager CD into the CD drive.
   c. Start the launchpad.
2. The launchpad starts automatically. Click **Next** (Figure 4-85).

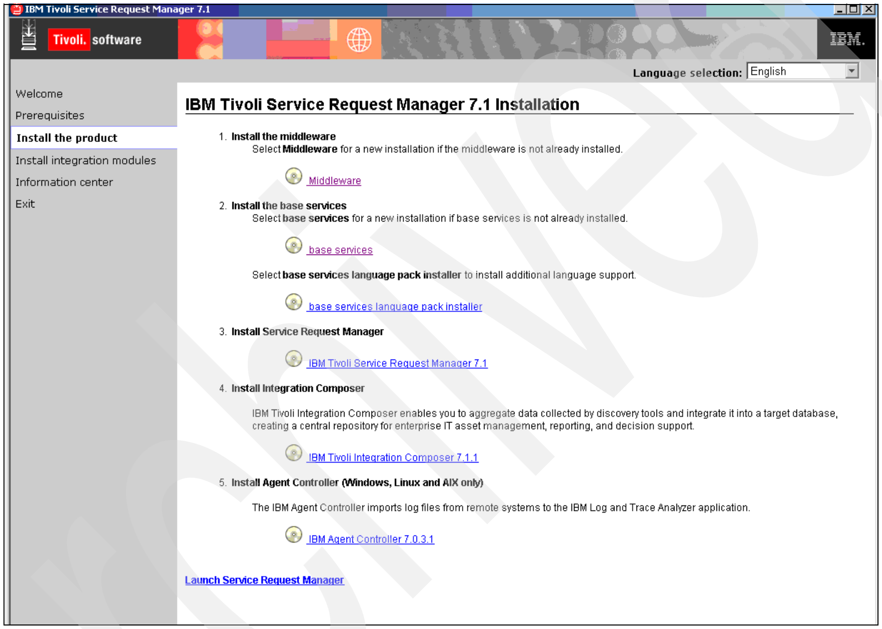


Figure 4-85 Launchpad

If it does not launch automatically, run the launchpad program by navigating to the root directory, or the downloaded installation image, and run the following command at a command prompt (or by double-clicking it in Windows Explorer):

```
launchpad.exe
```

3. In the IBM Tivoli Service Request Manager window, click **Base Services** (Figure 4-86).

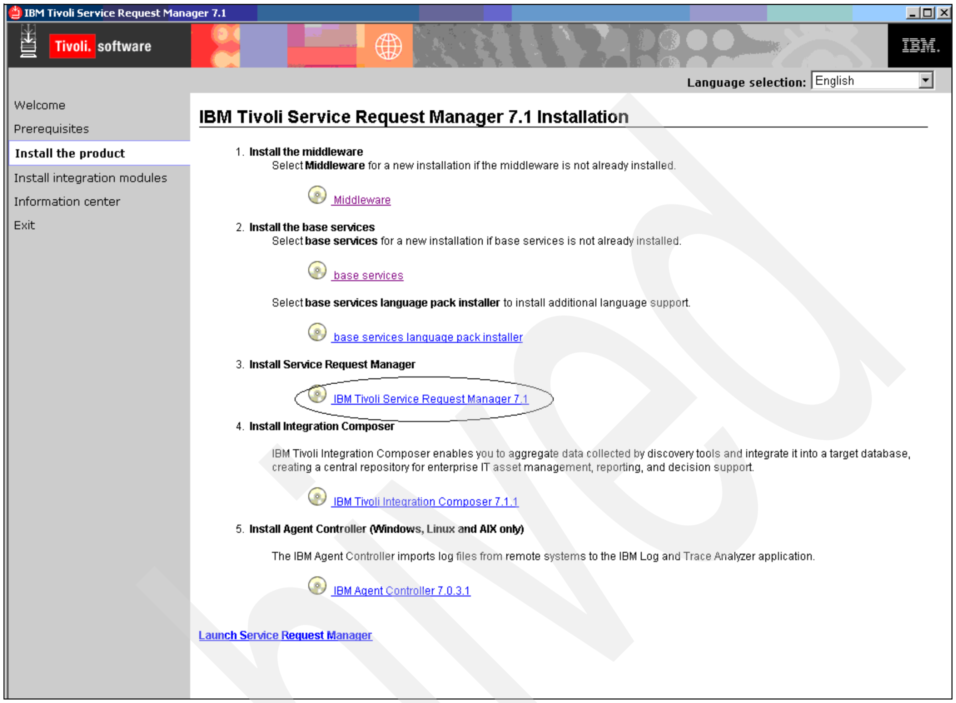


Figure 4-86 Base Services installation

You will be seeing the following window as the option is being launched. Click **Next** (Figure 4-87).

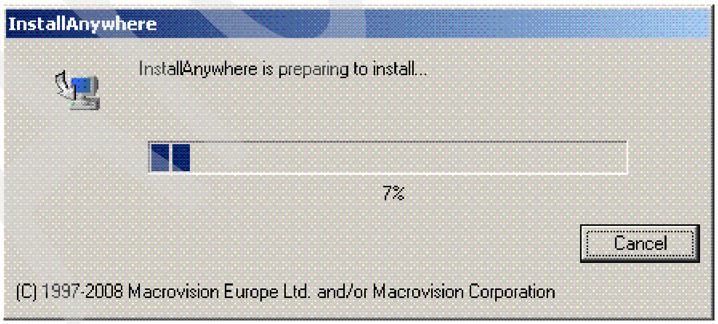


Figure 4-87 Base Services installation

4. In the base services (Figure 4-88 on page 199) welcome window, select the installer language and click **OK**.

Figure 4-88 Select the language

5. In the IBM Tivoli base services Introduction window, click **Next** (Figure 4-89).

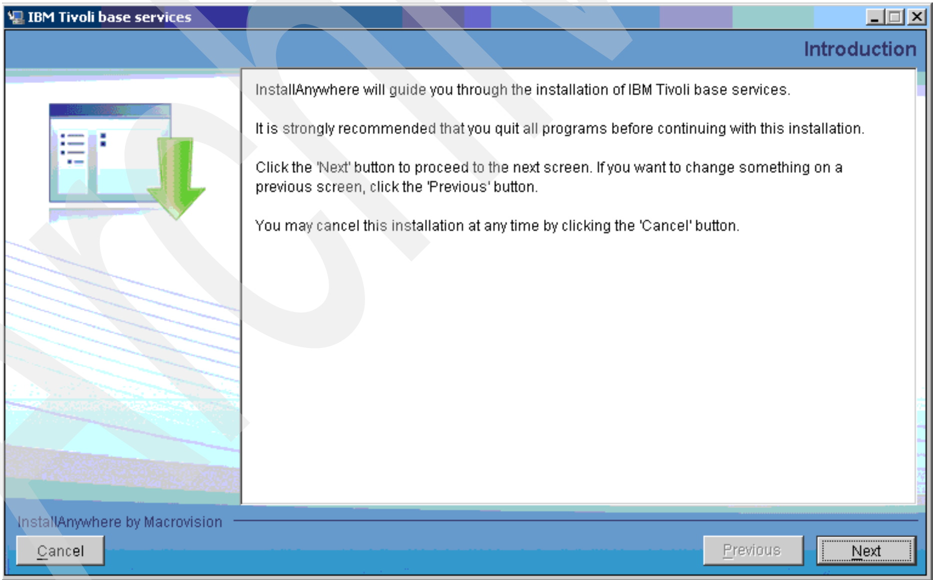


Figure 4-89 IBM Tivoli base services Introduction window

6. In the software License agreement window (Figure 4-90), select **Accept**.

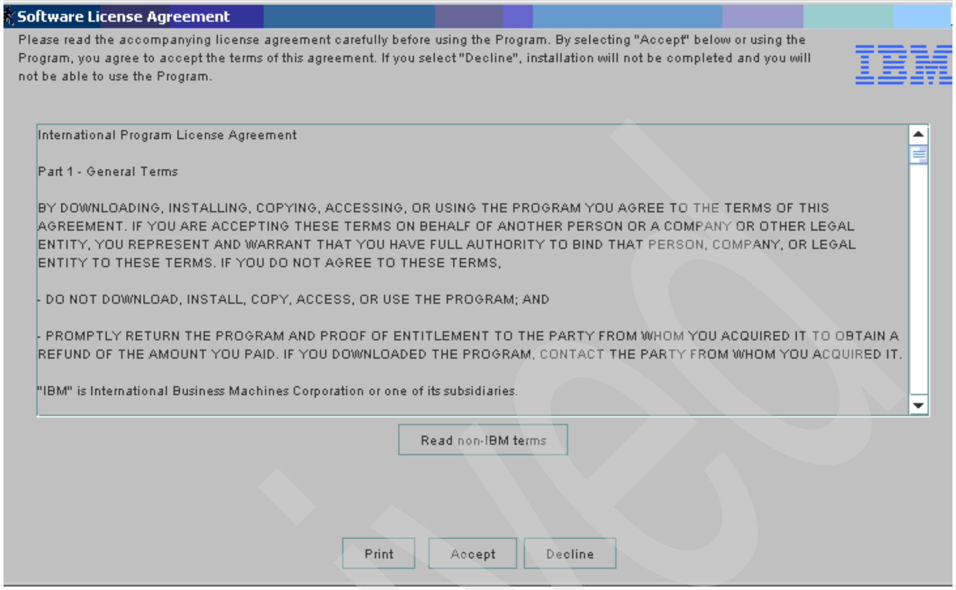


Figure 4-90 License Agreement

7. In the Import Middleware Configuration Information window, specify that you want to use the field values you input into the Tivoli middleware installer as the default values for those same fields in the base services installation program, and then click **Next** (Figure 4-91 on page 201).

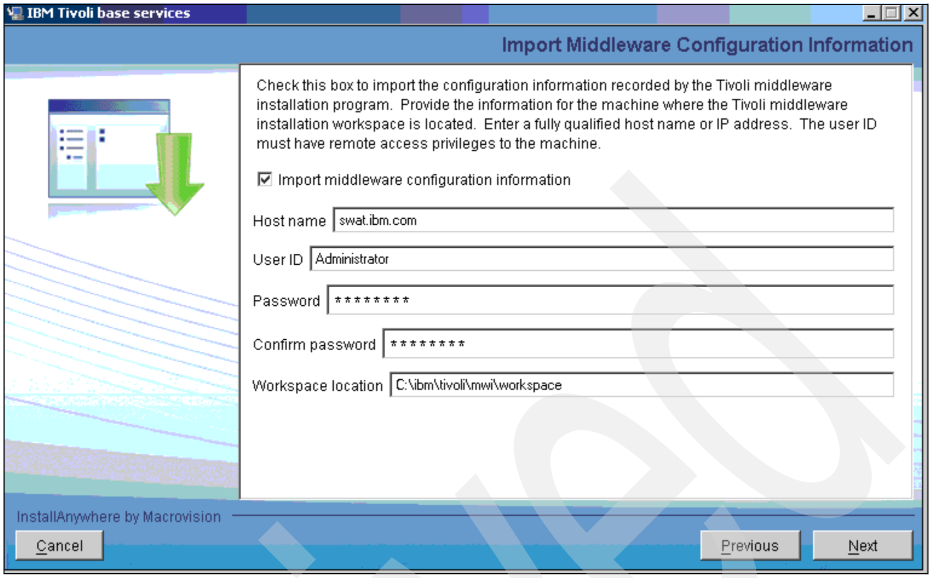


Figure 4-91 Import Middleware Configuration Information window

– Import middleware configuration information

  Select this check box if you want to allow the base services installation program to reuse values entered in the Tivoli middleware installer.

  > **Note:** If you select this feature while installing base services by way of RXA, the Workspace Location that you specify cannot be located on a networked drive of the remote system. It must reside locally on the remote system.

  The middleware default information will not be used if you select the Simple deployment path.

– Host name

  Enter the fully-qualified host name of the system where the Tivoli middleware installer was run.

– User ID

  Enter the User ID that was used to run the Tivoli middleware installer.

– Password

  Enter the password of the User ID that was used to run the Tivoli middleware installer.

– Confirm password

Enter the password of the User ID that was used to run the Tivoli middleware installer again.

– Workspace Location

Enter the location of the topology file that contains the values entered for the Tivoli middleware installer. This file is found in the workspace that was defined during the middleware installation task, for example, C:\ibm\tivoli\mwi\workspace.

At this stage, your input will be validated by the installation program. Click **Next** (Figure 4-92).

Figure 4-92 Validating inputs

8. From the Choose Deployment window, select the **Simple** deployment topology, and then click **Next** (Figure 4-93 on page 203).

- Simple

  Select **Simple** if you want to deploy all base services components on a single system. This deployment option is typically only used for demonstration, proof-of-concept, or training purposes.

- Custom

  Select **Custom** if you want to deploy base services components across several systems. This deployment option is typically used in a production environment.

> **Note:** Using the Simple installation will automatically select the DB2 database. Using the Custom installation will provide you with the option to select Oracle.

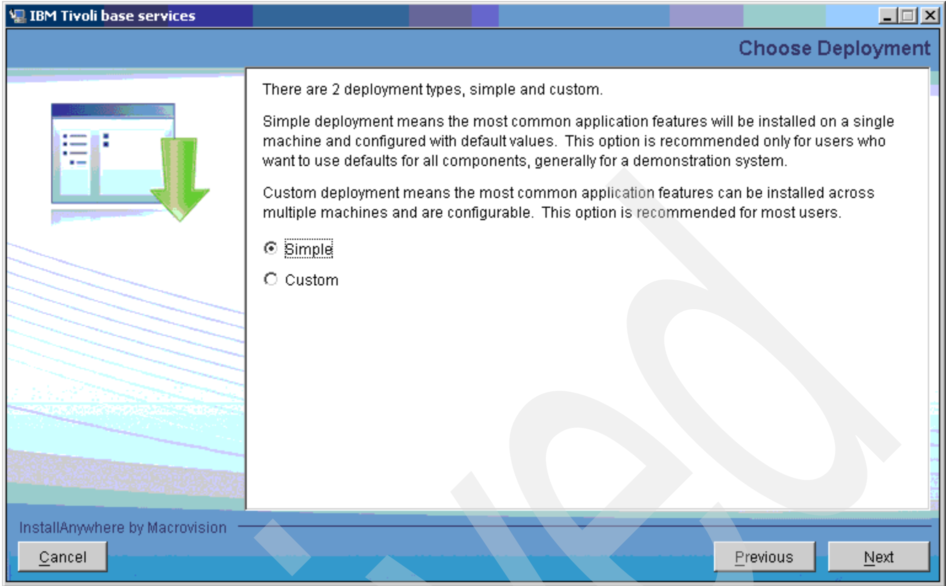


Figure 4-93 Simple installation

9. Click **Next** (Figure 4-94) to continue. The installation program then configures your system with the parameters provided above.

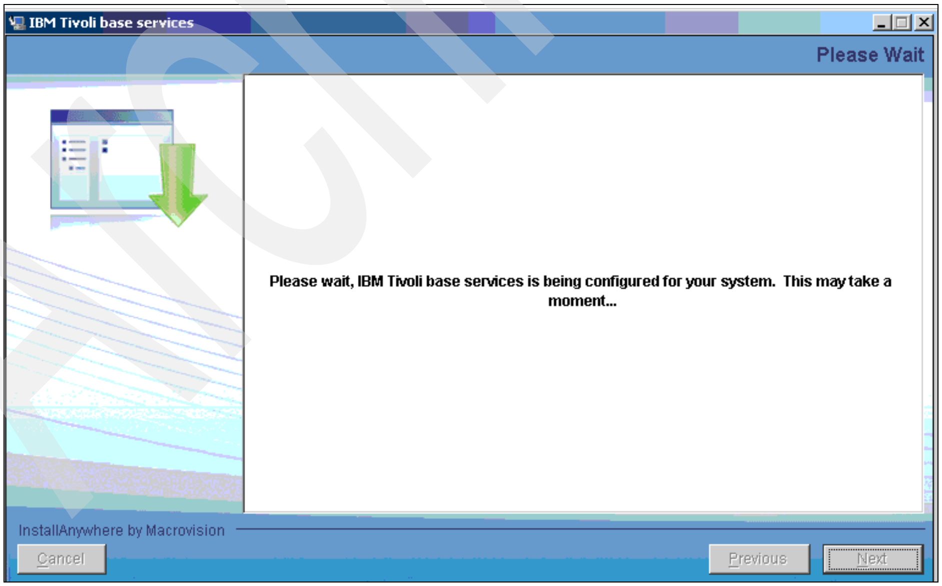


Figure 4-94 Configuring your system

10. From the Choose Install Folder window, specify the directory you will use to install base services, and then click **Next** (Figure 4-95). By default, this value is C:\IBM\SMP. The path you specify must not contain spaces.

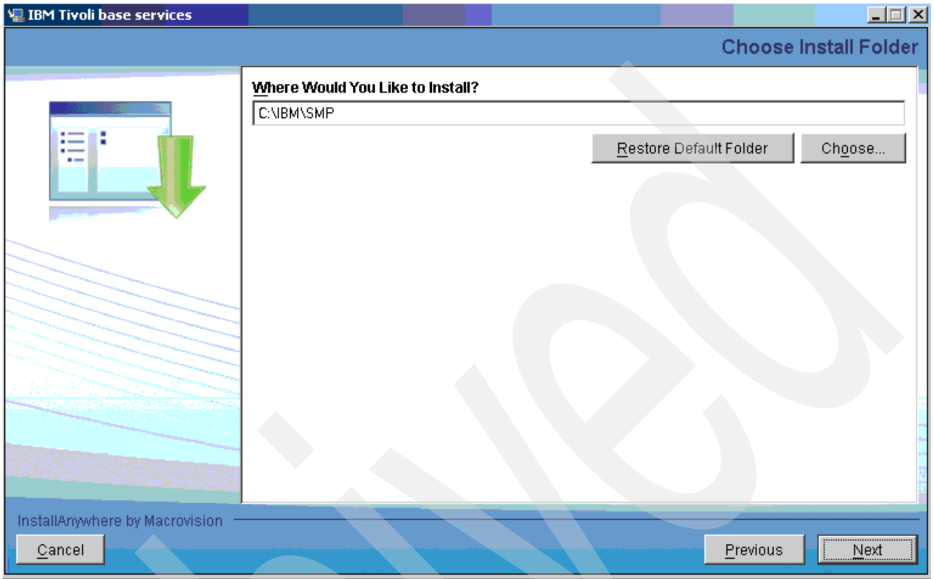


Figure 4-95 Installation location

11. Click **Next** to continue. In Figure 4-96 on page 205, you need to enter the middleware server information.

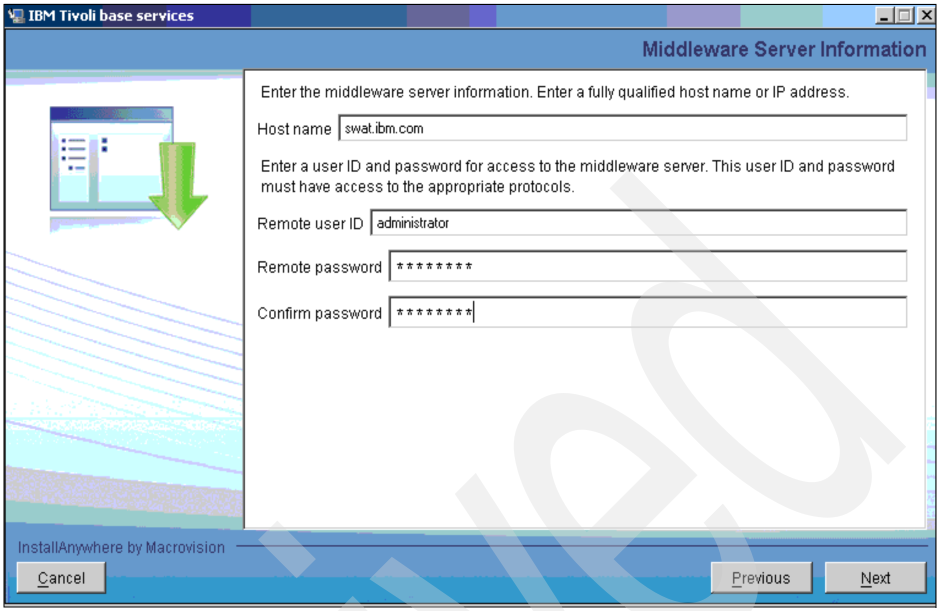


Figure 4-96 Middleware server information

Use the following information to provide values in this window:

– DB2 Host name

  Enter the host name of the machine hosting DB2. The host name must be fully qualified.

– Remote User ID

  Enter the user ID used to access the server. The default for all platforms is maximo. This user ID will be created if it does not already exist. This user ID cannot be the same one used as the instance administrator user ID.

– Password

  Enter the password for the Administrator ID.

– Confirm password

  Enter the password for the Administrator ID.

12. Click **Next** to continue. Your input will be validated by the installation program. The next window is the Database information, as shown in Figure 4-97.

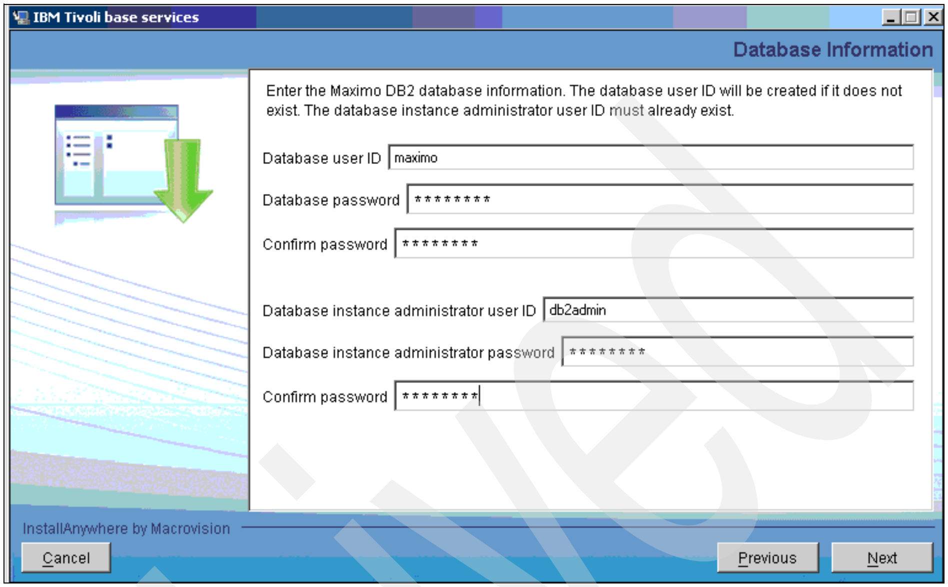


Figure 4-97 Database information

13. Type in the DB2 database information and click **Next** to continue.

 – *Database User ID*

 Enter the user ID used by Maximo to access DB2. The default for all platforms is maximo. This user ID will be created if it does not already exist. This user ID cannot be the same one used as the instance administrator user ID.

 – Password

 Enter the password for the user ID used to access DB2.

 – Confirm password

 Enter the password for the user ID used to access DB2 again.

14. In the WebSphere Server Information window, enter the WebSphere Administration parameters and click **Next** (Figure 4-98 on page 207) to continue.

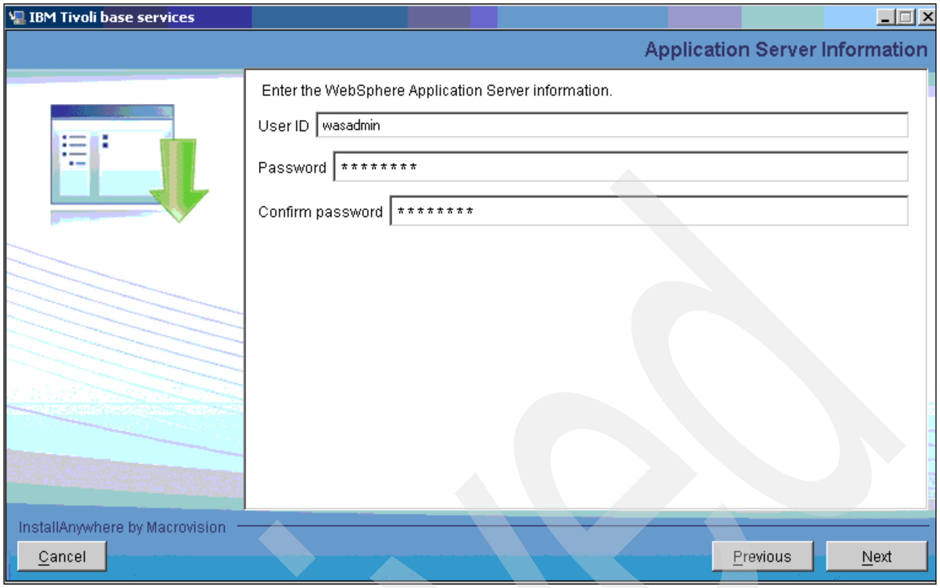


Figure 4-98 WebSphere Administration information

The base services is now configured for your system. Next, the WebSphere thin client is then installed, as shown in Figure 4-99.

Figure 4-99 WebSphere thin client installation

> **Note:** The base services installation program automatically configures the database for use by base services. Examples of automated tasks include creating table spaces, creating database tables, creating database schemas, creating users, and so on.

The installation program will now ask for your choice of shortcut icons (Figure 4-100).

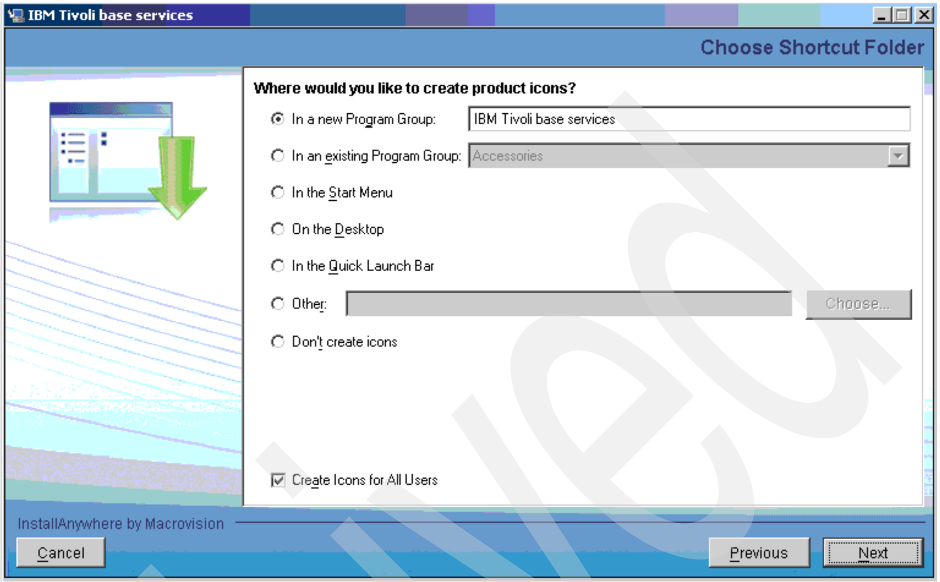


Figure 4-100 Choose Shortcut Folder

– In a new Program Group

  Select this option and enter the name of a new program group if you would like to create base services shortcuts in a new program group.

– In an existing Program Group

  Select this option and choose the name of an existing program group to store base services shortcuts.

– In the Start Menu

  Select this option to create shortcuts for base services in the Start menu. In order to use the Start Menu shortcut in conjunction with Internet Explorer, ensure that you have added the base services URL to the trusted sites Web content zone and disable the option of requiring server verification for all sites in the zone.

– On the Desktop

  Select this option to create shortcuts for base services on the desktop.

– In the Quick Launch Bar

  This option should not be used. Do not select this option. Selecting this option will not create a shortcut in the Quick Launch bar.

– Other

Select this option and use the **Choose...** button to select another location to create base services shortcuts.

– Don't create icons

Select this option if you do not want any base services shortcuts created.

– Create Icons for All Users

Select this option if you would like base services desktop icons to appear on the desktop for all system users. Click **Next** when you are done.

15. In the Input Summary window, review the information you have provided to the base services installation program and then click **Next** (Figure 4-101).

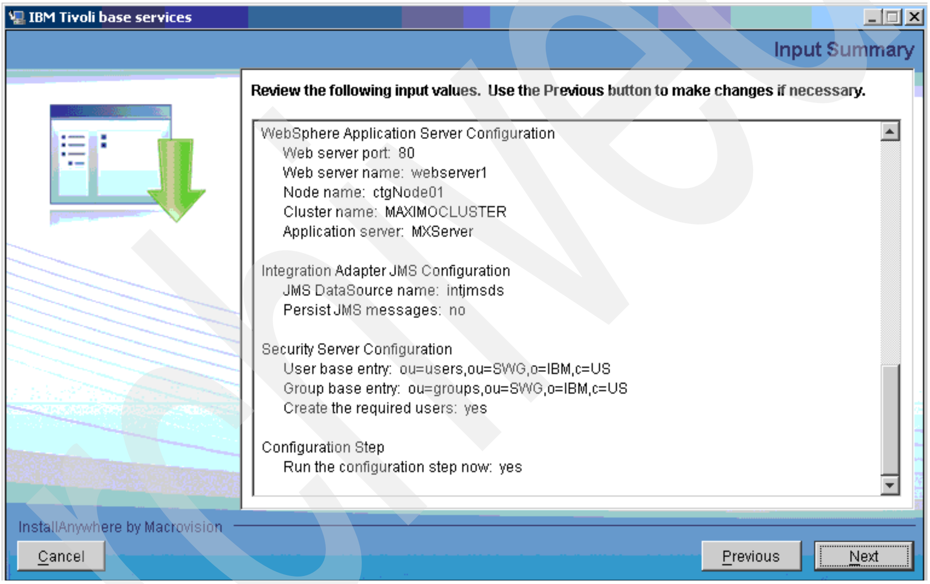


Figure 4-101 Summary information

Use the **Previous** button to return to previous windows to make any changes.

16.In the Pre-Installation Summary window, review the installation information presented and then click **Install** (Figure 4-102). The installation task will begin. Progress can be monitored by viewing the messages displayed above the progress bar.

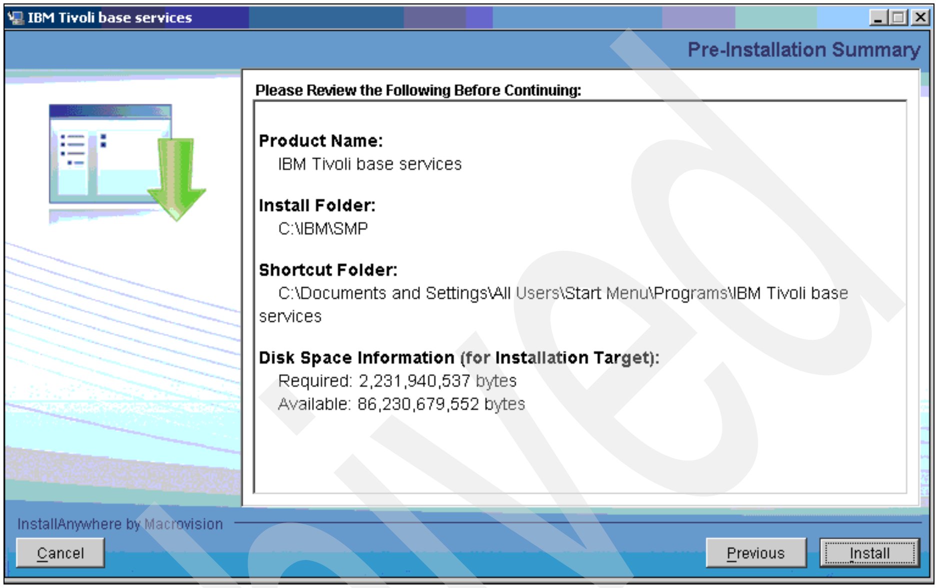


Figure 4-102 Pre-Installation Summary

17.The installation program proceeds while notifying you of the different stages of the installation progress. First, the Java Runtime Environment will be installed. Click **Next** (Figure 4-103 on page 211).

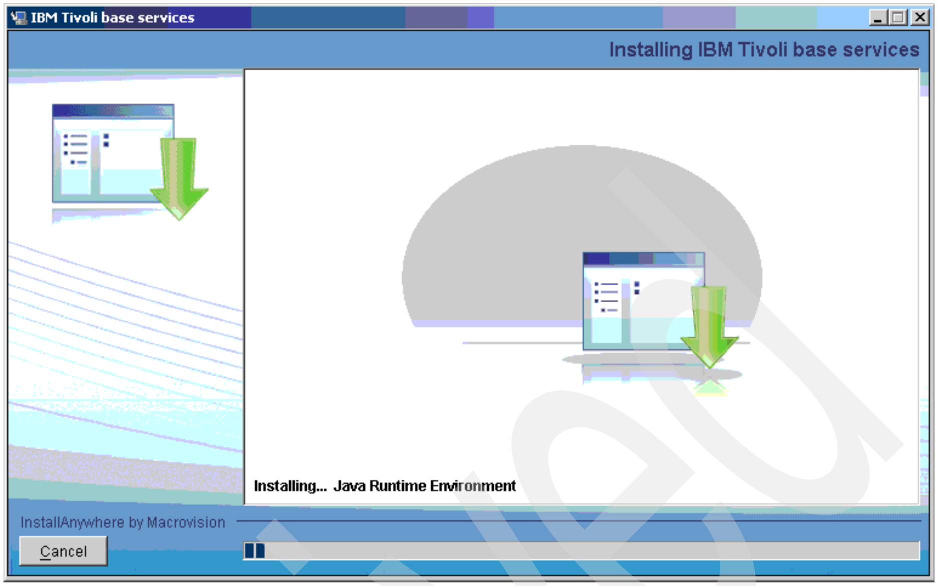


Figure 4-103 Java Runtime Environment

18. In the next step, applications will be installed. Click **Next** (Figure 4-104).

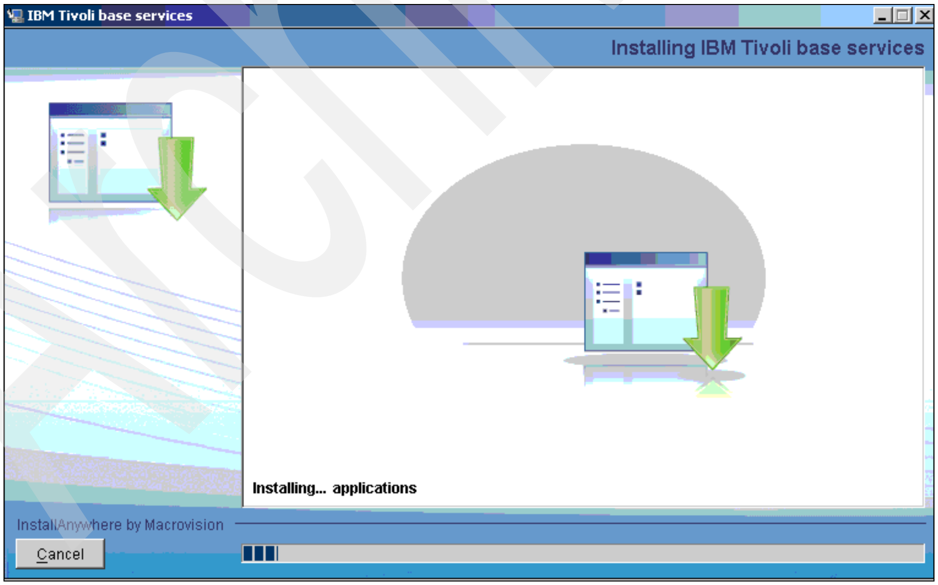


Figure 4-104 Installing the applications

19. Tools will be installed. Click **Next** (Figure 4-105).

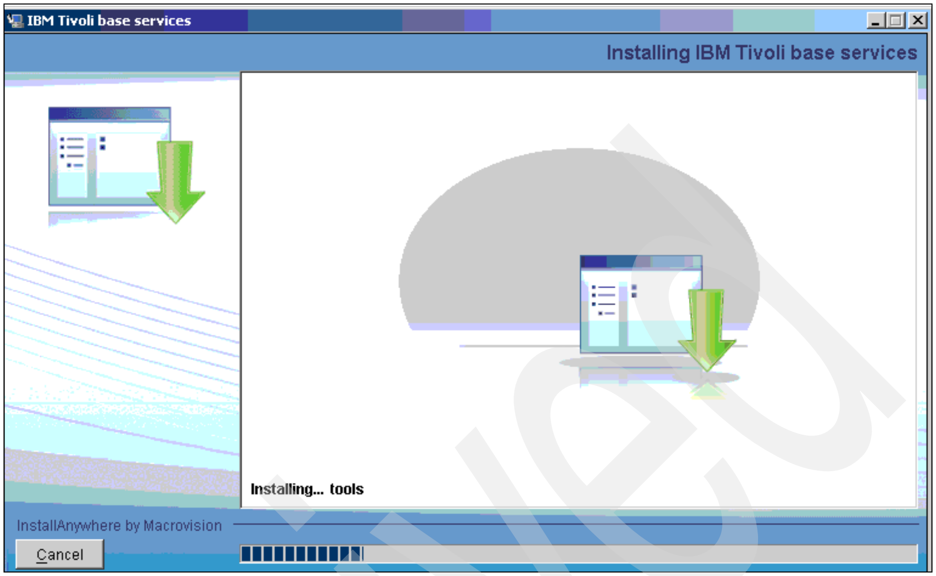


Figure 4-105 Installing tools

20. Process Managers products will be installed. Click **Next** (Figure 4-106).

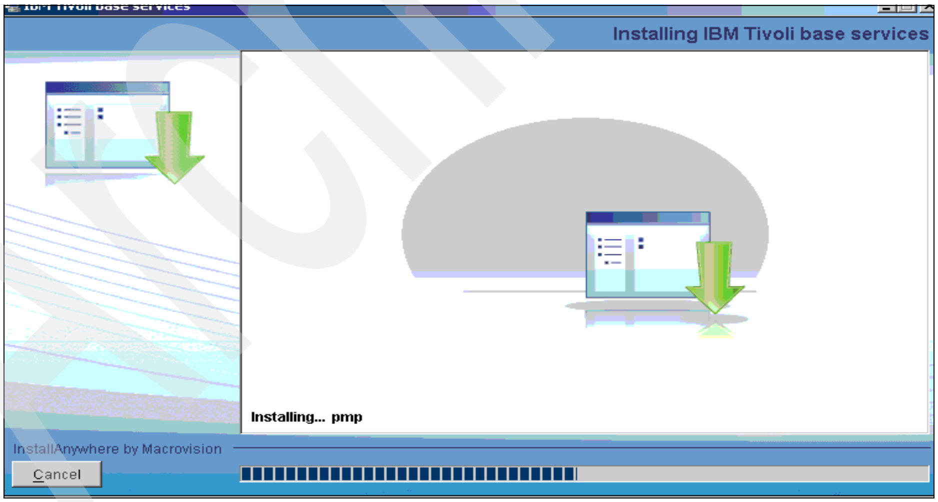


Figure 4-106 Installing Process Managers Products

21. DB2 will be configured. Click **Next** (Figure 4-107 on page 213).

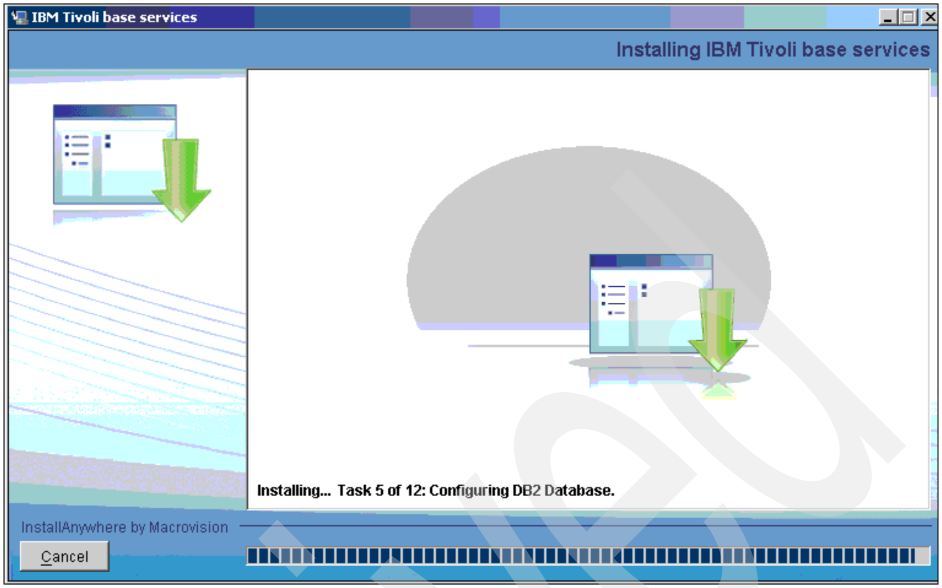


Figure 4-107 Configuring the DB2 database

22. The installation program will verify the DB2 database configuration. Click **Next** (Figure 4-108).

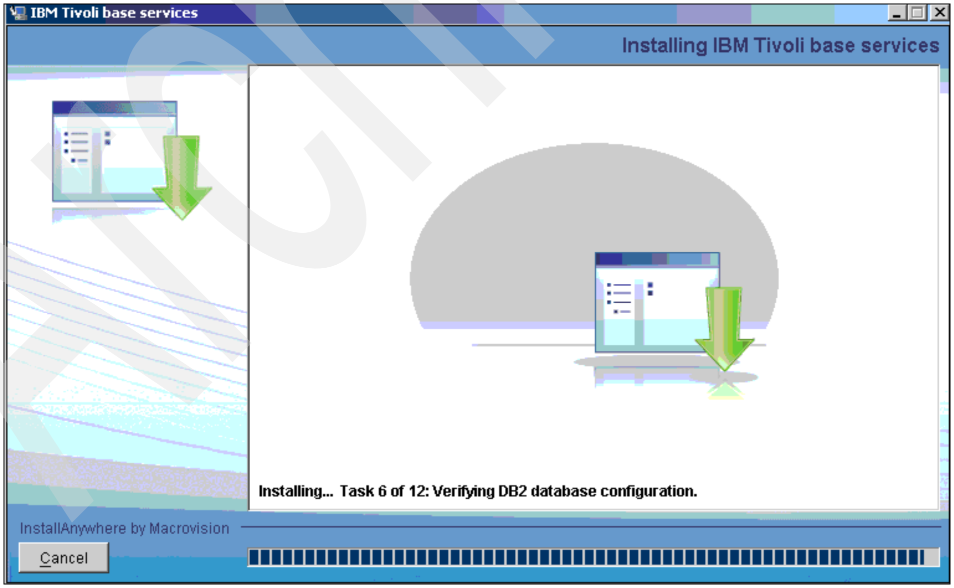


Figure 4-108 Verifying the DB2 database configuration

23.The installation program will validate the WebSphere configuration input parameters and configuration. Click **Next** (Figure 4-109).

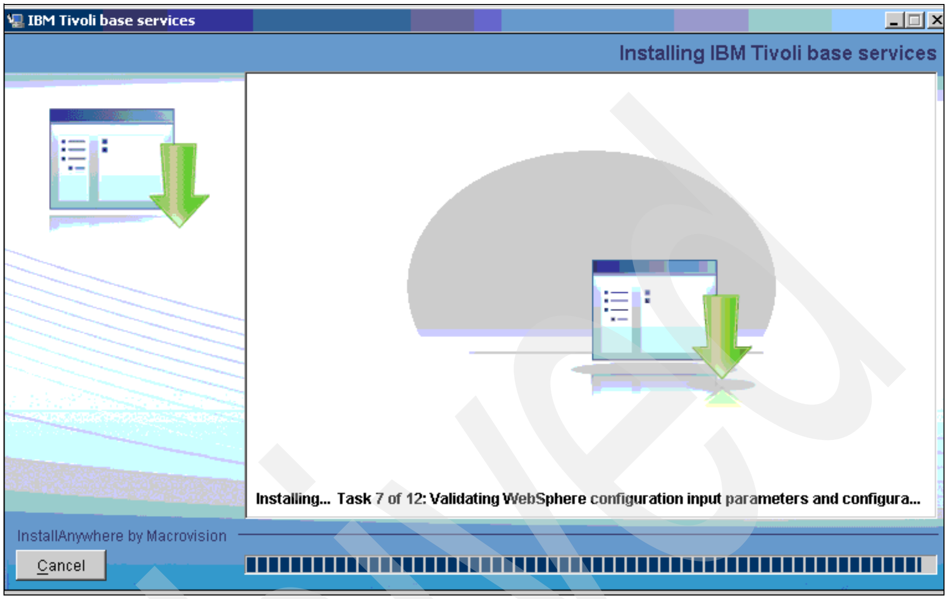


Figure 4-109 Validating the WebSphere configuration input parameters and configuration

24.WebSphere Application Server will be configured. Click **Next** (Figure 4-110 on page 215).

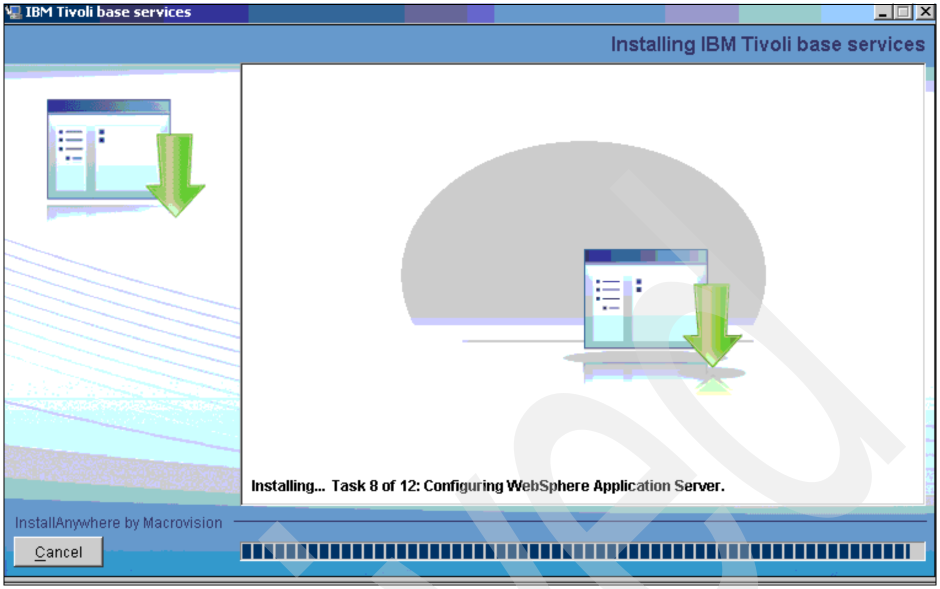


Figure 4-110 Configuring the WebSphere Application Server

25. The installation program will validate the process manager input parameters and configuration. Click **Next** (Figure 4-111).

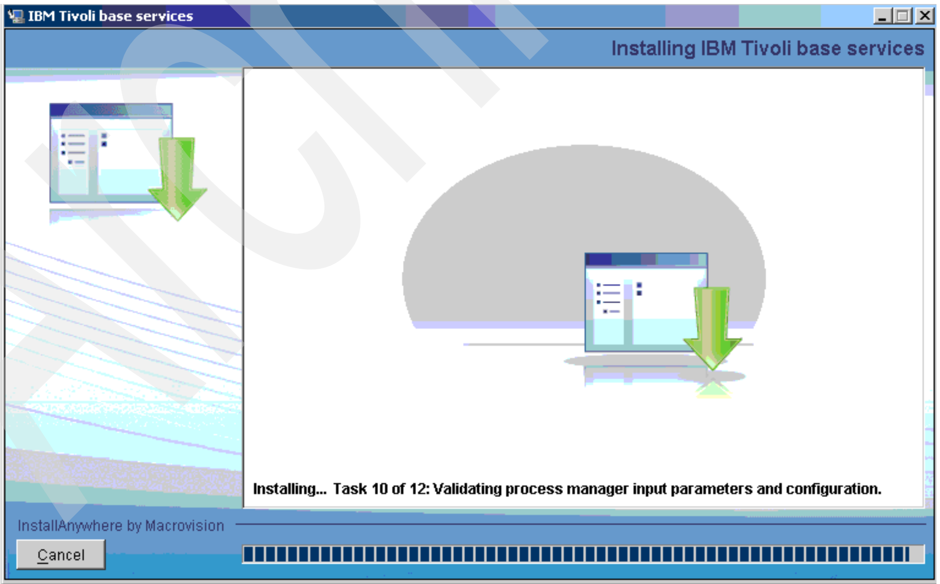


Figure 4-111 Validating the process manager input parameters and configuration

26. Next, the Process Managers will be configured and validated. Click **Next** (Figure 4-112).

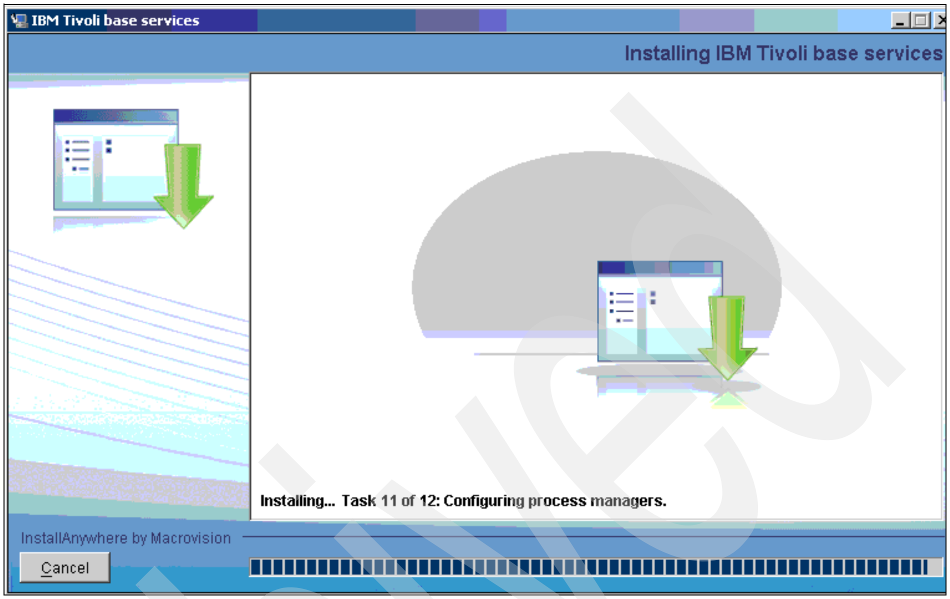


Figure 4-112 Configuring the process managers

27. You are then presented with a window that will prompt you for information about installing a new base language or additional language. Select **Yes** or **No** and click **Next** (Figure 4-113) to continue.

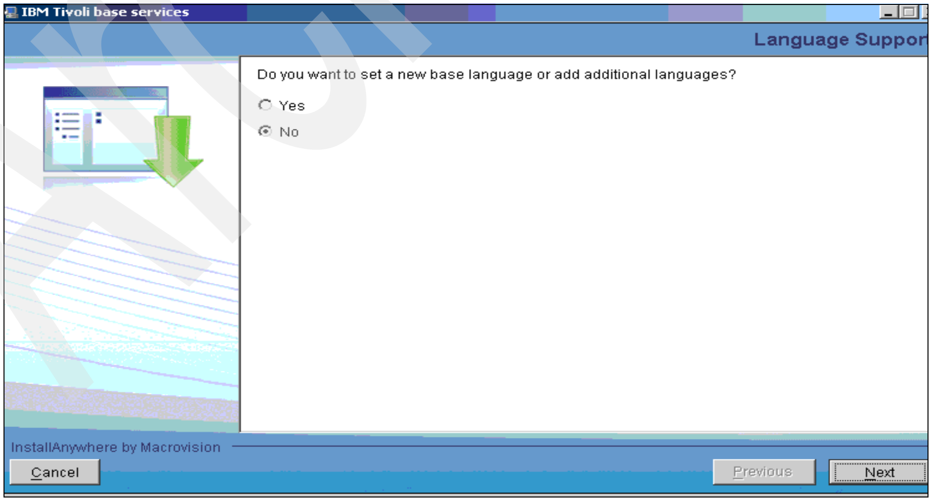


Figure 4-113 Language support

28.Click **Next** (Figure 4-114).

Note: For the purpose of this installation, we select **Yes**.

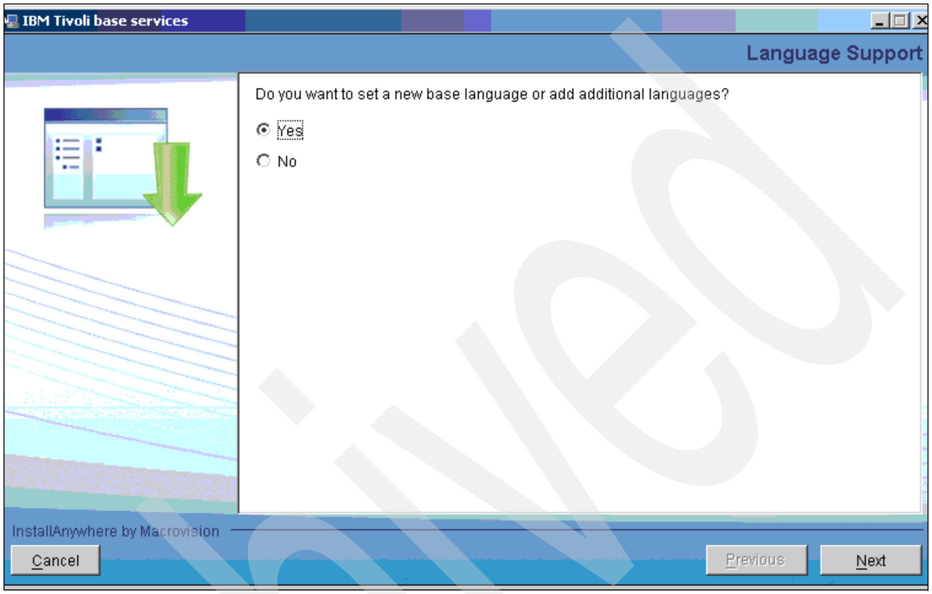


Figure 4-114 Language support

29.The Language installation is initiated. Click **Next** (Figure 4-115).

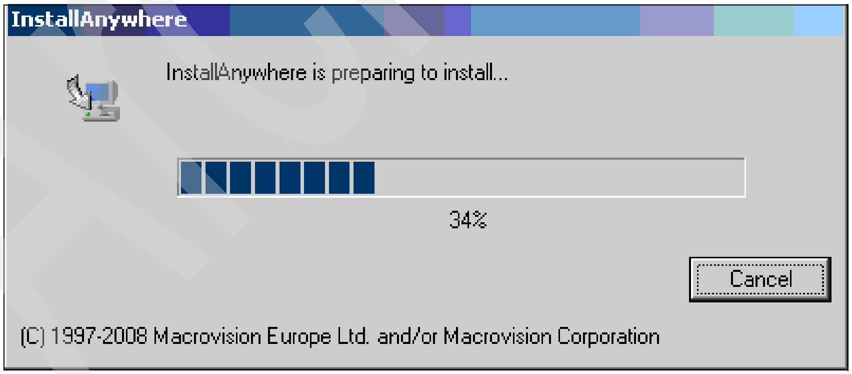


Figure 4-115 Language support

30.From the Install Complete window, click **Done**.

Once the base services installation program has completed the installation and configuration tasks, it will exit. The logs can be found at <CCMDB_Home>/logs.

> **Note:** To install the base services on Linux server, proceed as you would with the Windows platform, with the exception that a second machine with a Windows platform is used to perform the installation.

4.3.6 Language pack installation for Windows

1. To perform the service pack installation, use the launchpad command (`Launchpad.exe`) and click **Next**.
2. At the Tivoli Service Request Manager V7.1 Welcome window (Figure 4-116), select **Install the product** on the left hand side of the window and then choose **Base Services Language Pack Installer**.

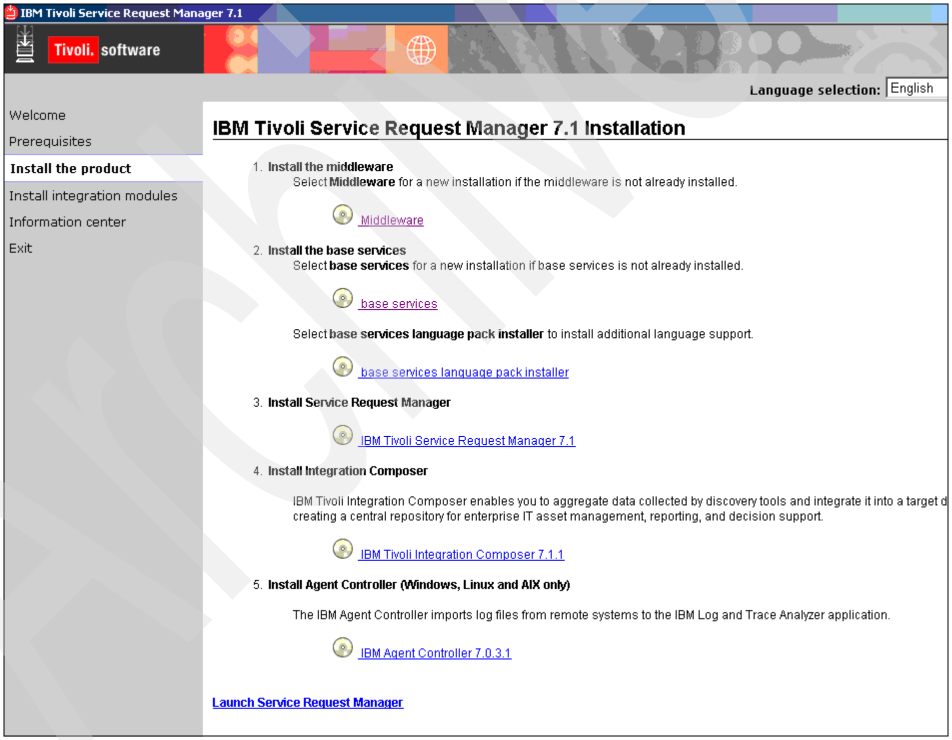


Figure 4-116 Launchpad

Note: To install additional language support for base services for Linux, proceed as for Windows by using a second machine with a Windows platform.

3. Click **Base Services Language Pack Installer** and click **Next** (Figure 4-117).

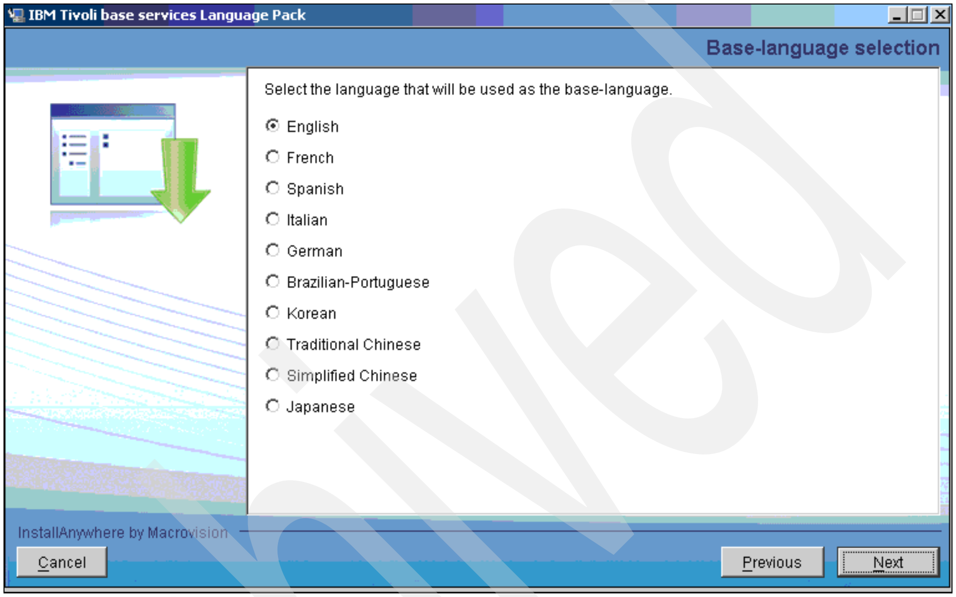


Figure 4-117 Base services Language Pack installer

4. Select the base language and click **Next** to continue (Figure 4-118).

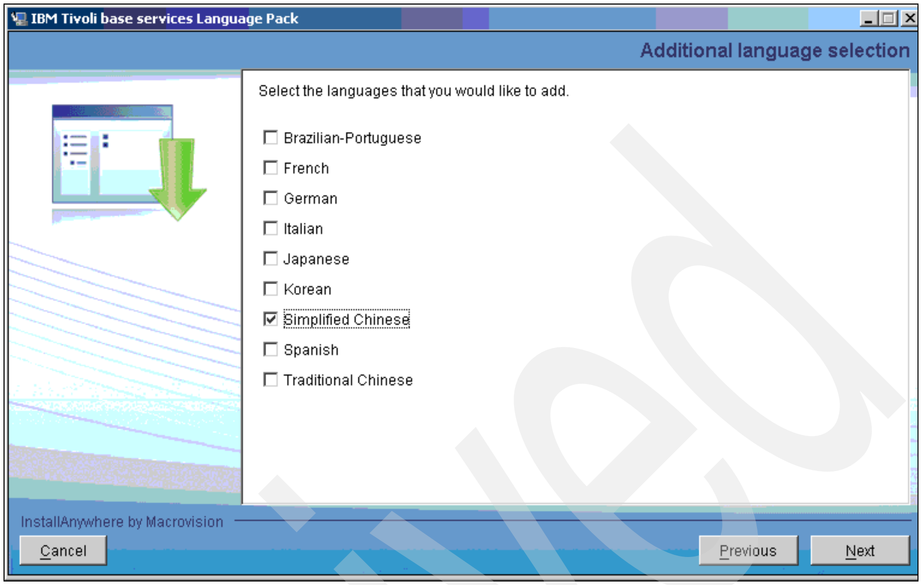


Figure 4-118 Base services Language Pack installer

5. Once your selection is complete, click **Next** to continue (Figure 4-119).

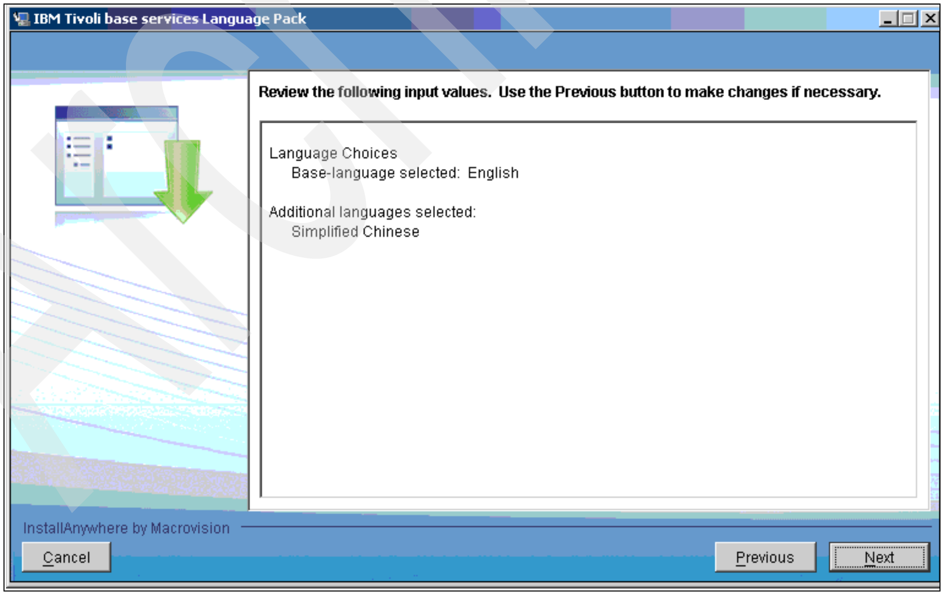


Figure 4-119 Base services Language Pack installer

6. Click **Install** (Figure 4-120).

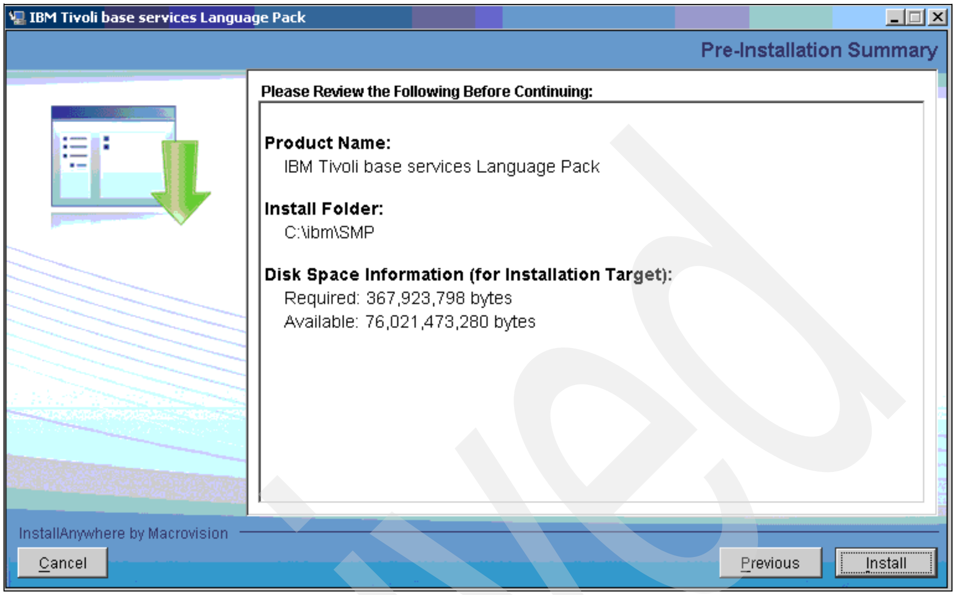


Figure 4-120 Base services Language Pack installer

The installation program is initiated. The language is then installed (Figure 4-121).

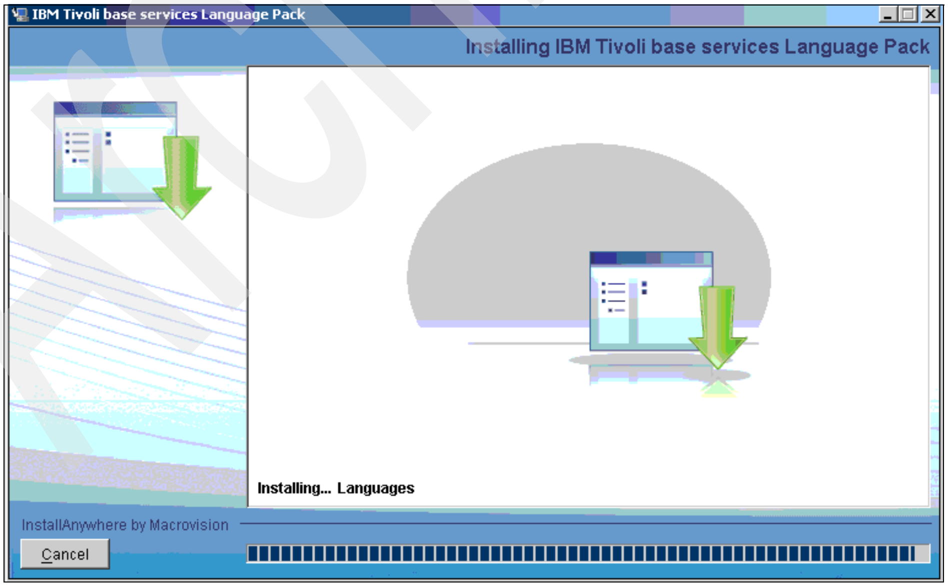


Figure 4-121 Base services Language Pack installer

The EARs (Enterprise Application Archives) are then rebuilt (Figure 4-122).

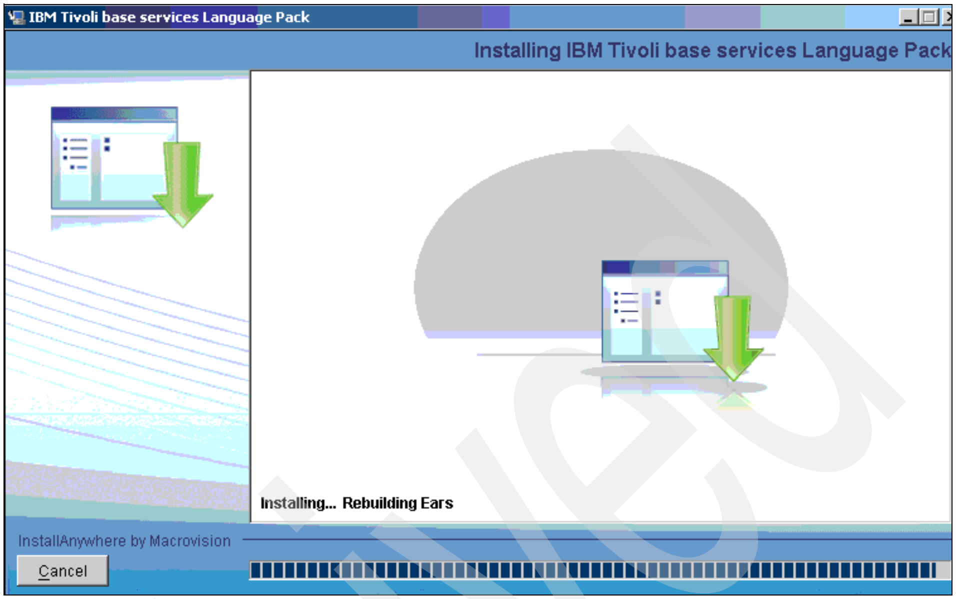


Figure 4-122 Enterprise Application Archive built

Note: *Maximo application* refers to an instance of Maximo. Enterprise Application Archive (EAR) files define what constitutes a Maximo application. You can have multiple Maximo applications, that is, multiple deployed EAR files on one application server.

The EARs are deployed once the built process is complete (Figure 4-123 on page 223).

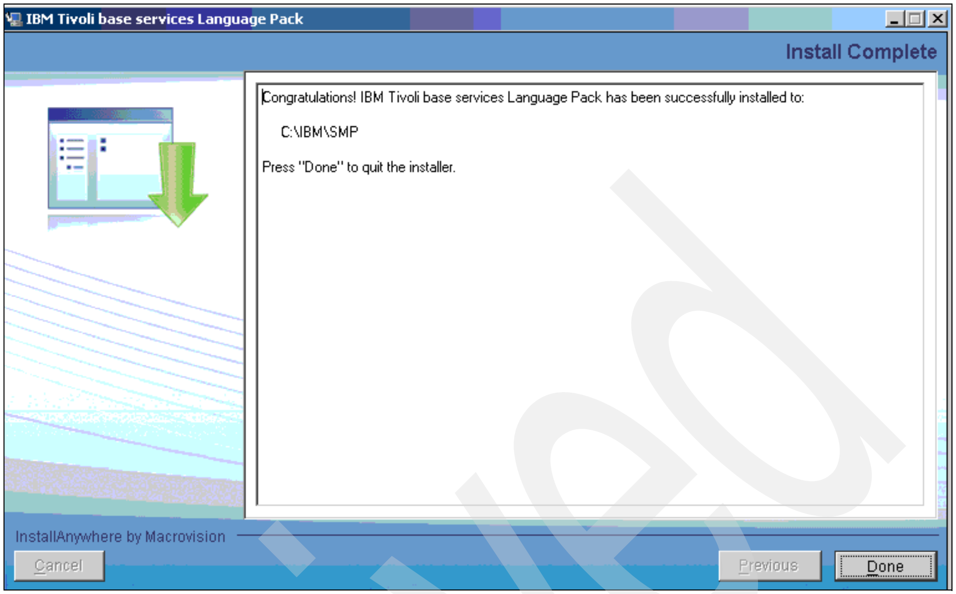


Figure 4-123 Built complete

7. Once the installation is complete, press **Done** to close the window.

4.4 Installing Tivoli Service Request Manager

This section contains information to install Tivoli Service Request Manager.

The Tivoli Service Request Manager installation consists of installing the following components:

- Service Desk
  - Service Desk Content - Best Practices
  - Service Desk Content - Classification
  - Service Desk Demo Data
- Service Catalog
  - Service Catalog Content - User
- Tivoli Service Request Manager V7.1 Language Support
  - Service Request Manager Search Language Support
  - Service Desk Language Support

- Service Catalog Language Support
- Service Desk Content - Best Practices Language Support
- Service Desk Content - Classification Language Support
- Manage Survey Applications Language Support
- Manage Survey Application for Service Request Manager V7.1

Note: The above options can be partially or totally deployed depending on your installation requirements.

4.4.1 Installing Tivoli Service Request Manager Service Desk

Use this task to install the Tivoli Service Request Manager Service Desk.

The Tivoli Service Request Manager Service Desk installation consists of installing the Service Desk plus optional best practices content and sample data.

Note: All available options will be installed in this scenario. We will first install Tivoli Service Request Manager and then the Service Desk component (see Figure 4-128 on page 228). Optionally, you can install these together by selecting both the Service Request Manager V7.1 and Service Desk check boxes at the same time in the window shown in Figure 4-128.

To install the Service Desk component, complete the following steps:

1. If you have not started the Tivoli Service Request Manager launchpad, then log in as an administrator; otherwise, skip to step 4.
2. Insert the Service Request Manager CD into the CD drive.
3. Start the launchpad.

   The launchpad (Figure 4-124 on page 225) should start automatically. If it does not launch automatically, run the launchpad program by navigating to the root directory, or the downloaded installation image, and run the following command at a command prompt (or by double-clicking it in Windows Explorer):

   ```
   launchpad.exe
   ```

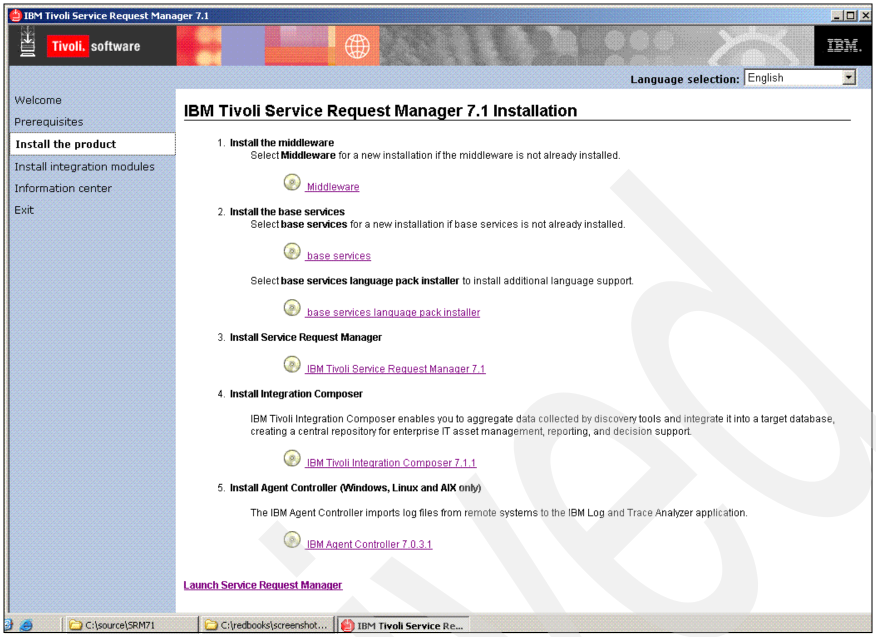


Figure 4-124 Launchpad

4. Click the **IBM Tivoli Service Request Manager 7.1** option.
5. Select the installation language and click **OK** (Figure 4-125).

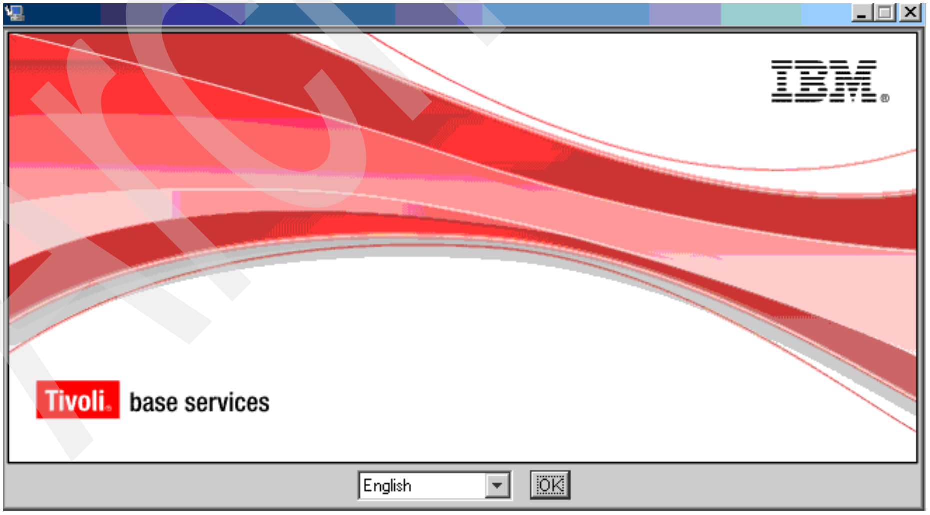


Figure 4-125 Language selection

6. In the Package Validation Results window, click **Next** (Figure 4-126).

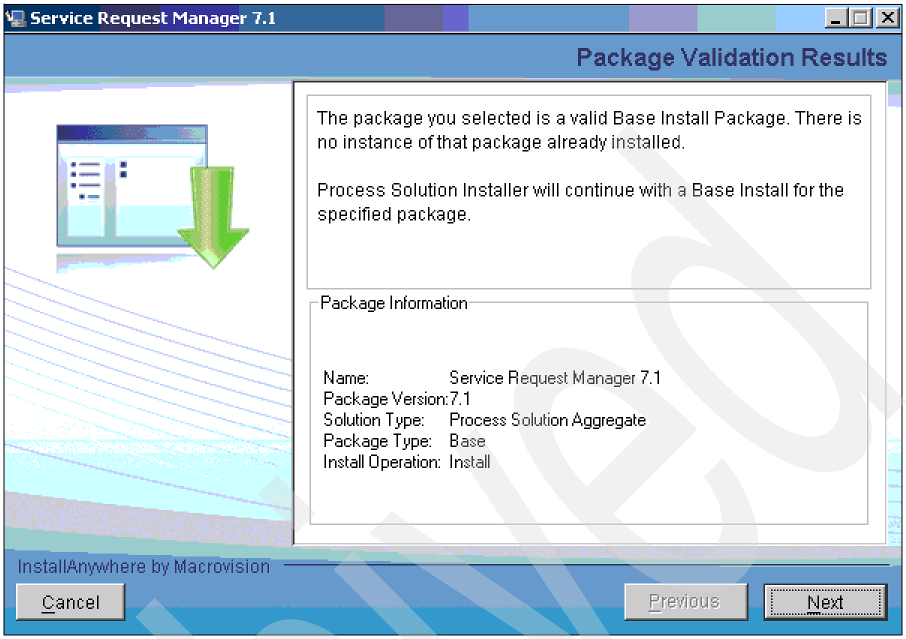


Figure 4-126 Package validation results

7. In the Software License Agreement, click **I accept both the IBM and non-IBM terms** and then click **Next** (Figure 4-127).

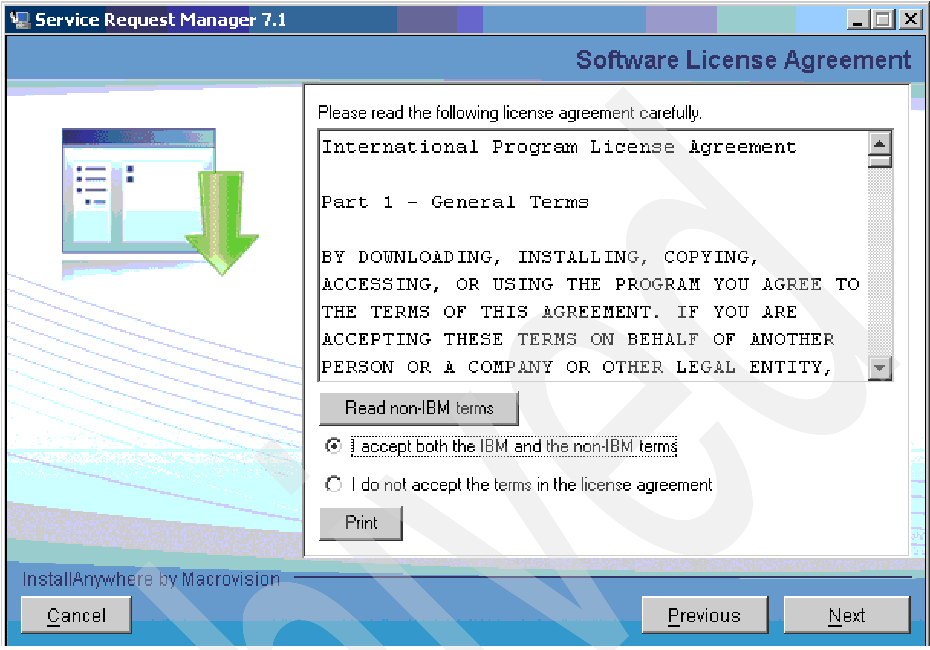


Figure 4-127 License Agreement

8. Select **Service Request Manager 7.1** and click **Next** (Figure 4-128) to continue.

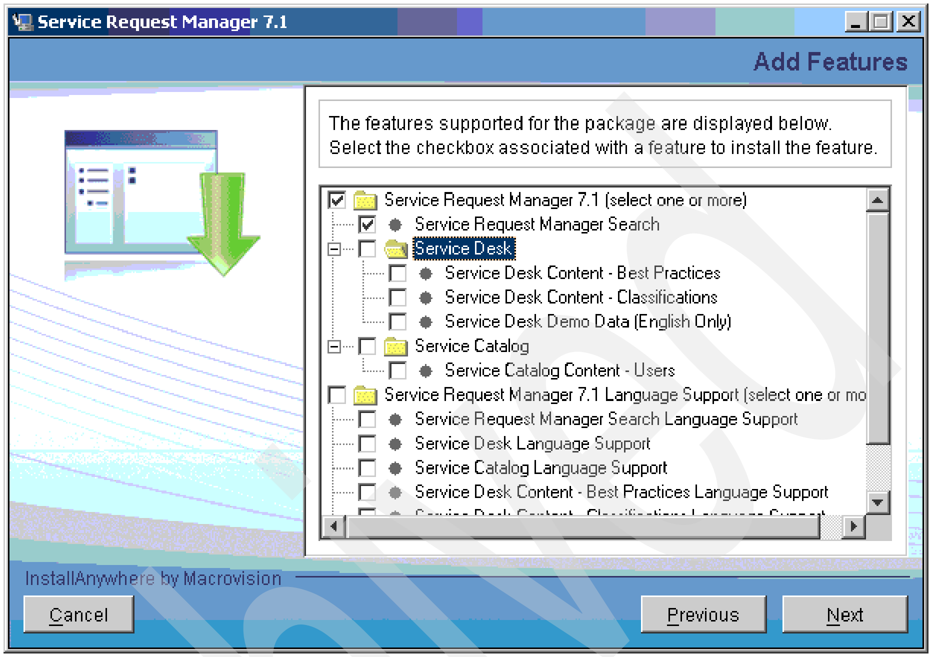


Figure 4-128   Add Features

9. The credentials are validated (Figure 4-129).

Figure 4-129   Credentials validated

10.Enter the password for the Maximo database as well as the WebSphere application and then click **Next** (Figure 4-130).

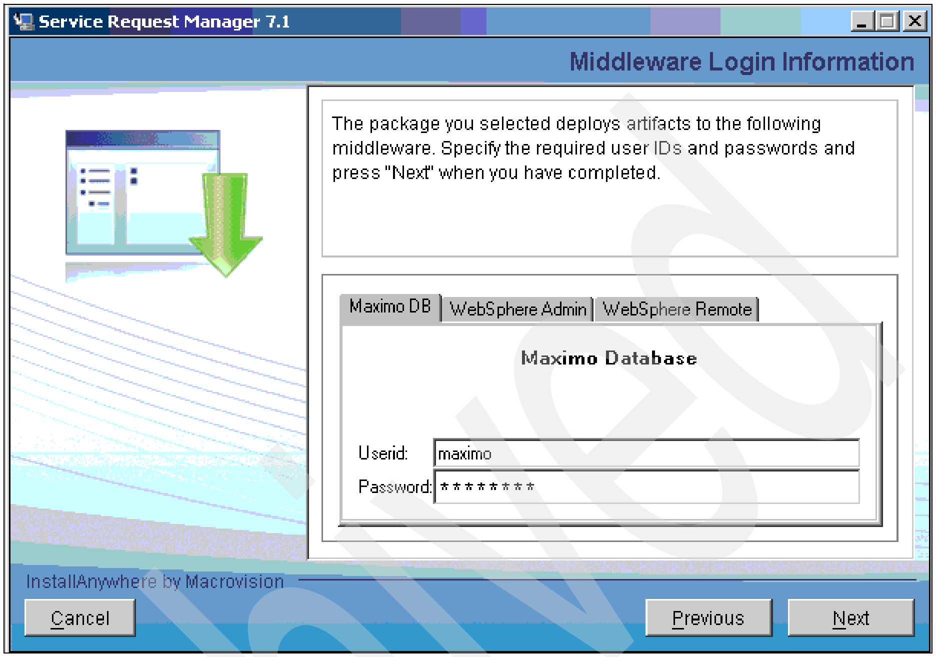


Figure 4-130 Middleware Login information

11. The system checks that the prerequisites are being met. Click **Next** (Figure 4-131).

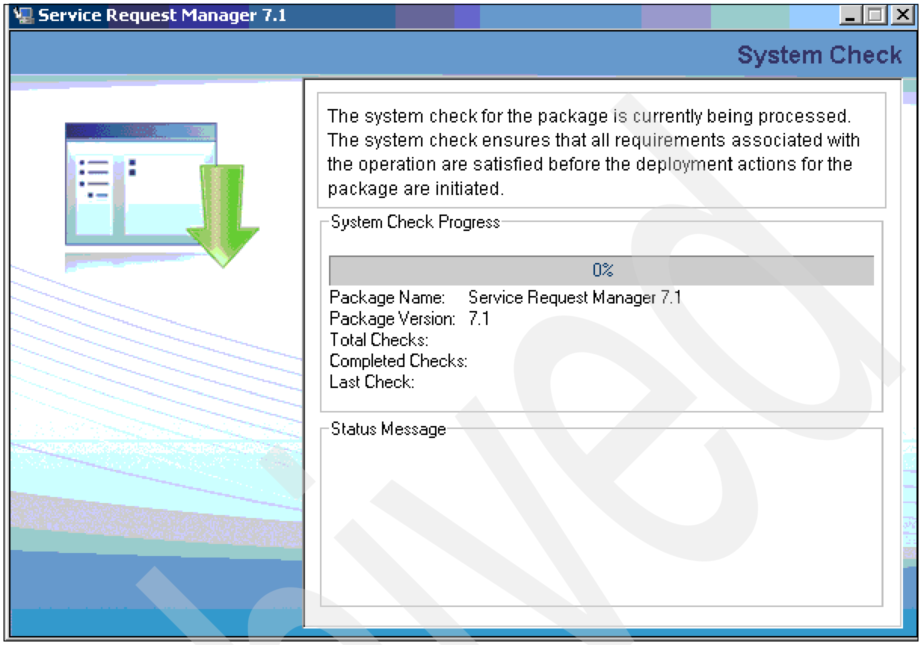


Figure 4-131 Prerequisites check

12. Click **Next** (Figure 4-132) to continue the installation.

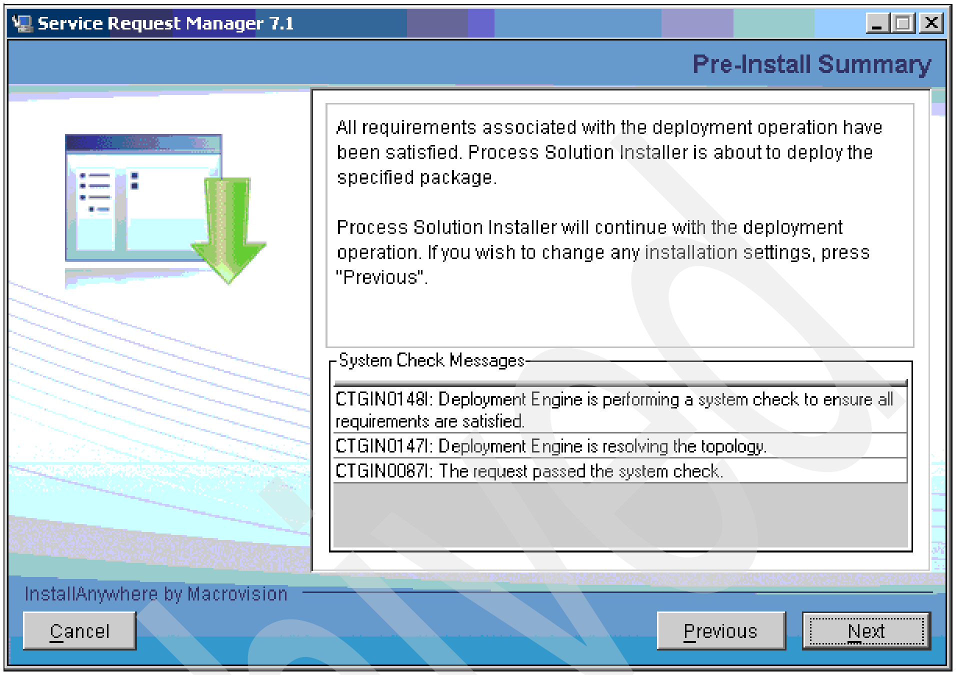


Figure 4-132 Pre-install summary

13. The application proceeds with the installation. Click **Next** (Figure 4-133).

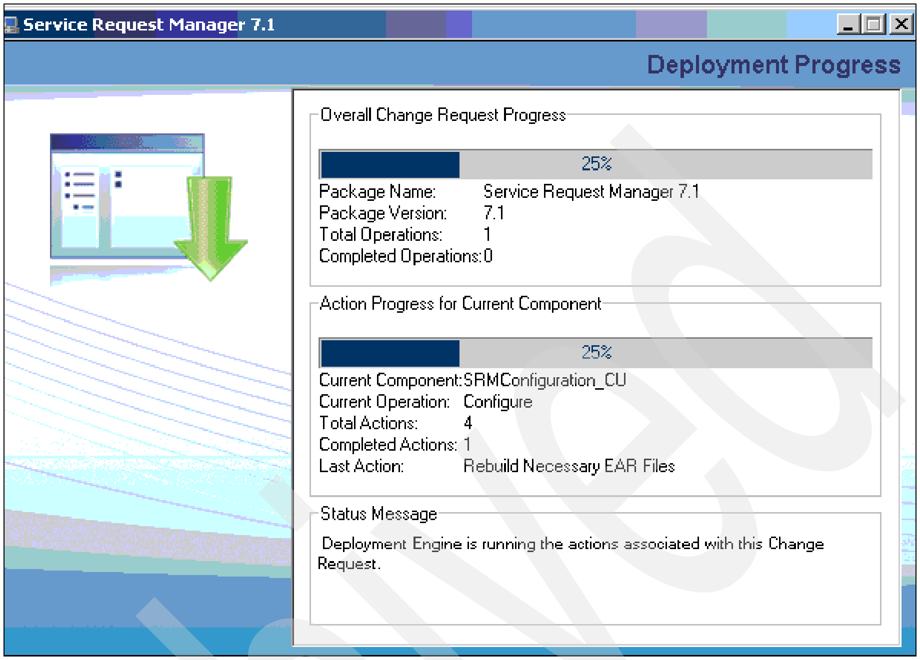


Figure 4-133 The application proceeds with the installation

14. Once the packages are deployed, click **Done** (Figure 4-134 on page 233).

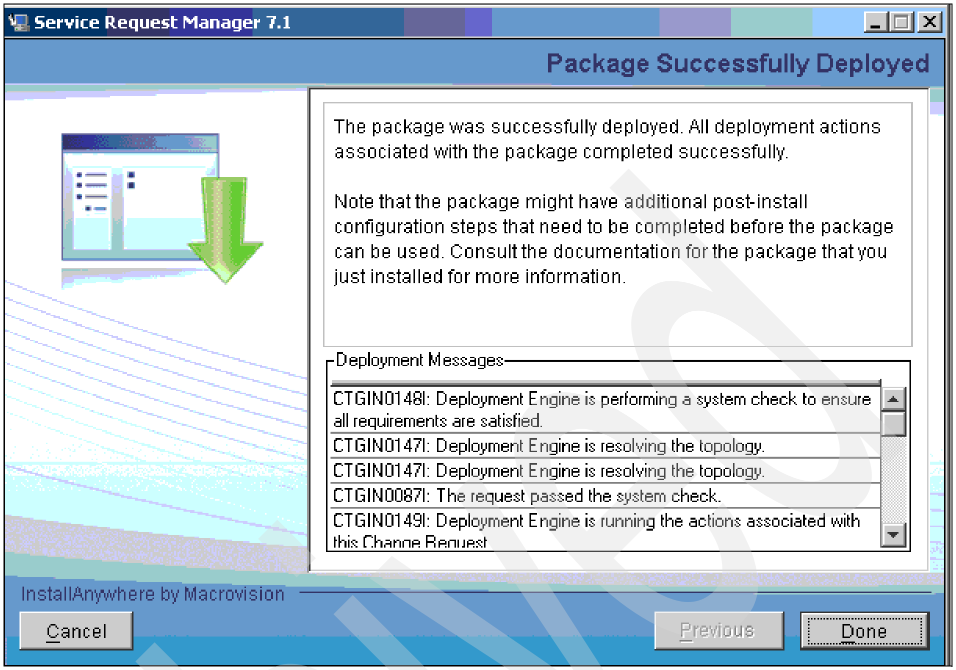


Figure 4-134 Packages are deployed

Installing the Service Desk

Do the following to install the Service Desk component:

1. Start the launchpad.

   The launchpad starts automatically. If it does not launch automatically, run the launchpad program by navigating to the root directory, or the downloaded installation image, and run the following command at a command prompt (or by double-clicking it in Windows Explorer):

   ```
   launchpad.exe
   ```

2. Select **IBM Tivoli Service Request Manager 7.1** to install the Service Desk along with other options listed in the installation window.

> **Note:** You can either select a particular option or chose all the options at once to avoid running the installation program again.

3. Select the language and click **OK** (Figure 4-135).

> **Note:** You can either select a particular option or chose all the options at once to avoid running the installation program again.

Figure 4-135 Language selection

4. Click **Next** (Figure 4-136) to continue in the Package Validation window.

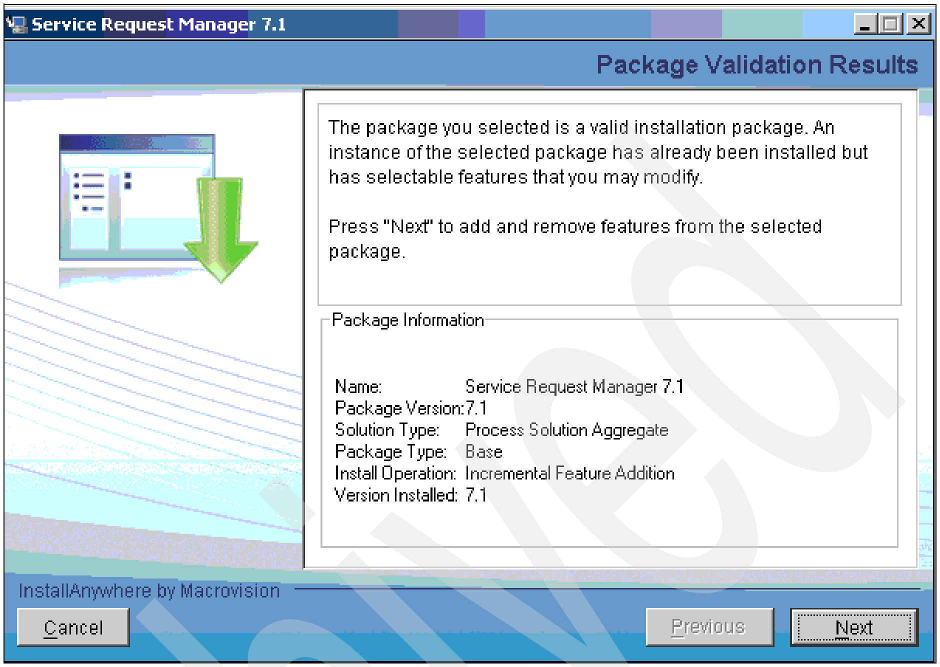


Figure 4-136 Package Validation

5. Select the required options and click **Next** (Figure 4-137) to continue.

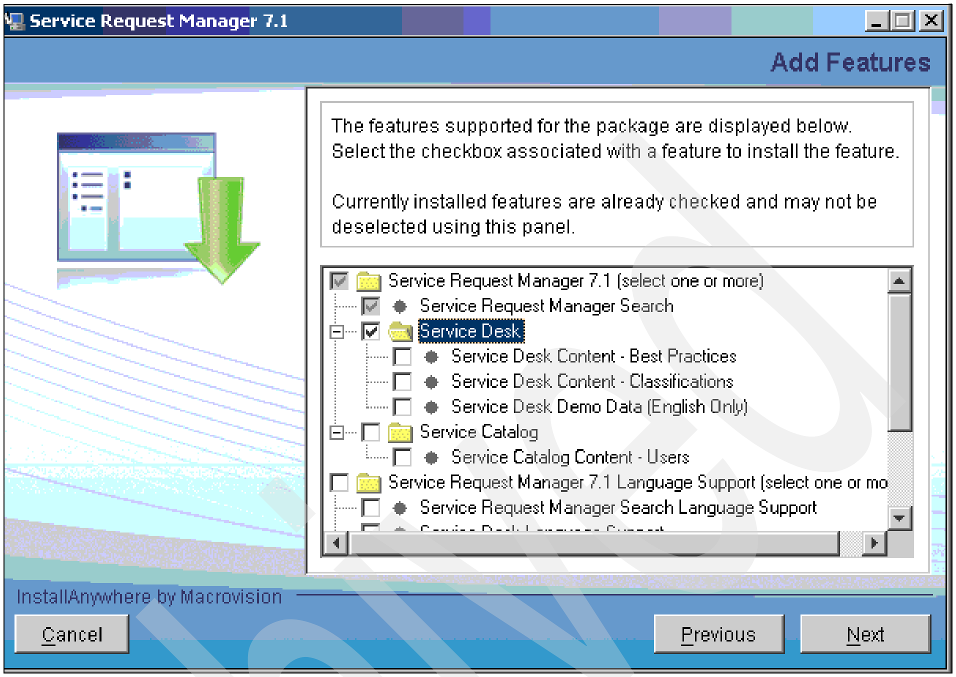


Figure 4-137 Select the required options

6. Type in the passwords and click **Next** (Figure 4-138) to continue.

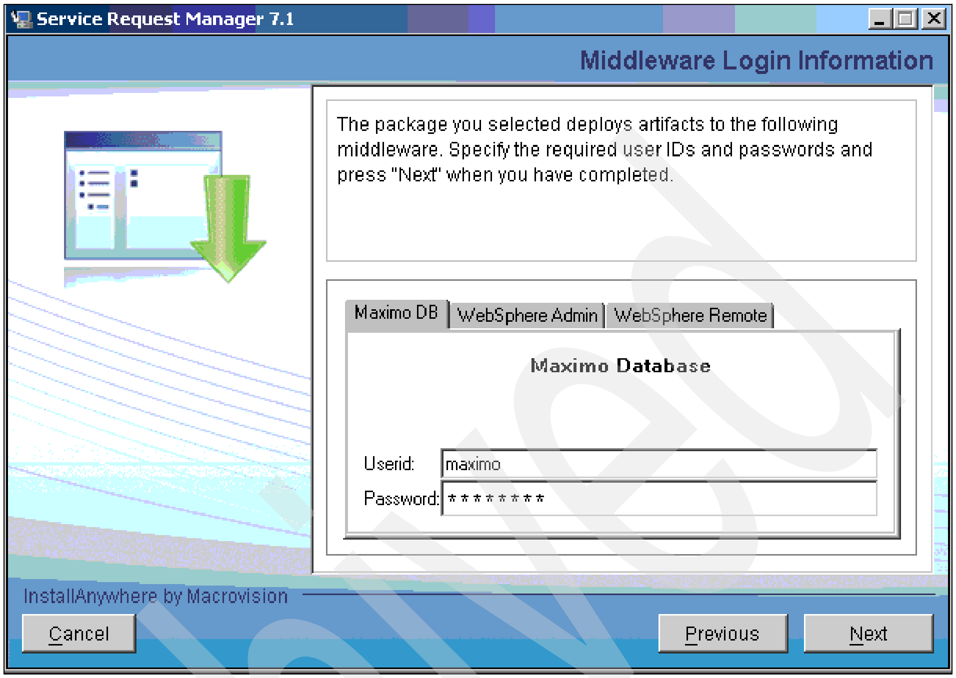


Figure 4-138 Login information

7. You are presented with a Pre-Install Summary. Click **Next** (Figure 4-139) to continue.

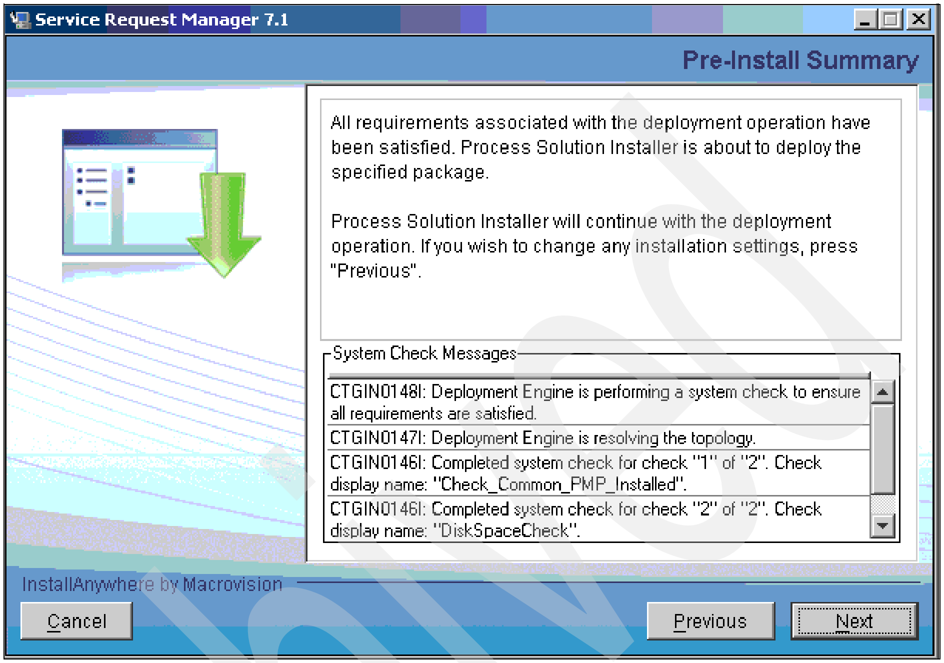


Figure 4-139 Pre-Installation summary

The installation proceeds (Figure 4-140).

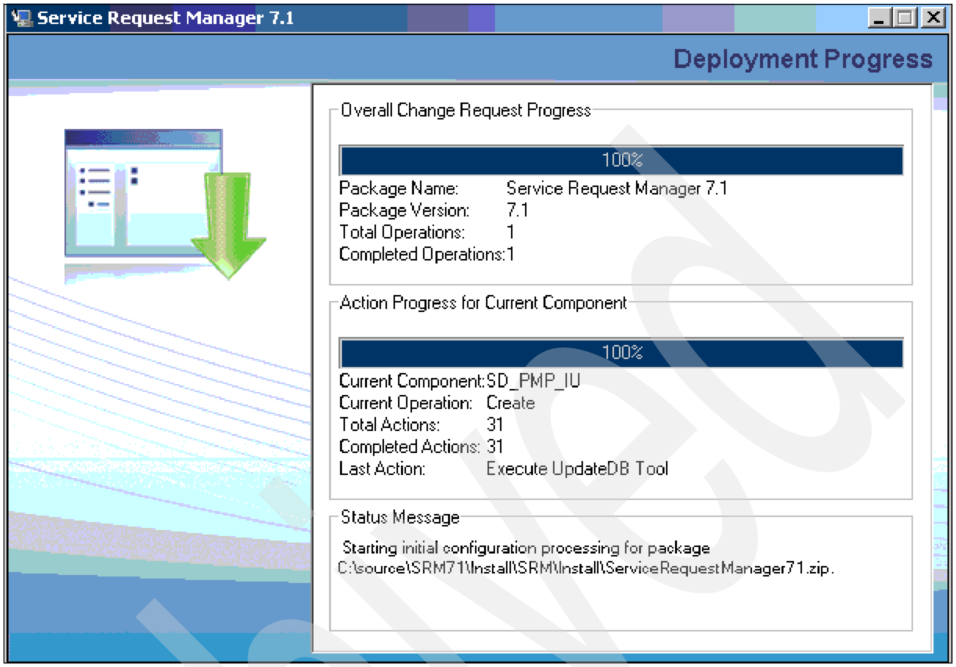


Figure 4-140 Installation proceeds

8. Once the installation is complete, press **Done** to close the window (Figure 4-141).

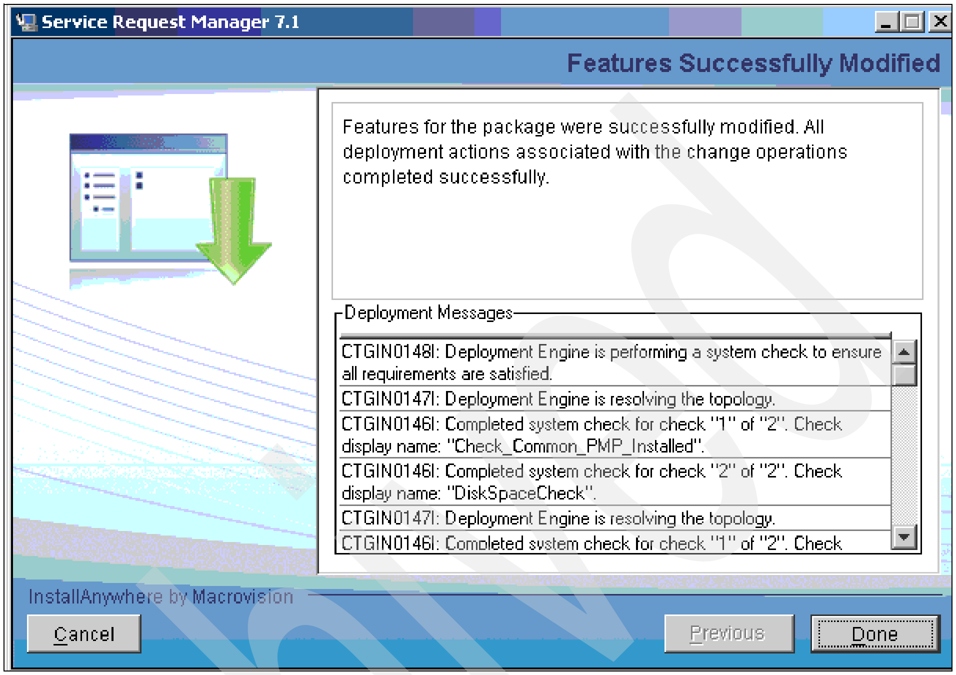


Figure 4-141 Installation complete

4.4.2 Installing Service Desk best practices content and sample data

Use this task to install Service Desk best practices content and sample data.

The installation of the following sample data is optional. You can install:

- Service Desk Content - Best Practices

  The Service Desk Content Best Practice will allow you to deploy, out of the box, many of the industry best practices methodologies (for example, escalation workflows and templates)

- Service Desk Content - Classification

  The Service Desk Classification data is a set of classifications derived from the United Nations Standard Product and Service Code (UNSPSC) classifications that can be installed in Tivoli Service Request Manager. This will allow you to use the Maximo Classifications application to create classifications. You also load classifications from an external source.

- Service Desk Demo Data

  This data allows you to start your Tivoli Service Request Manager application with all the data that one can expect for running a service desk, such as users, templates, workflows, sample incidents, sample problems, and so on.

1. Use the launchpad menu and select **Installing Best Practices Content for Service Desk**.
2. At the base services welcome window, select the installer language and then click **OK** (Figure 4-142).

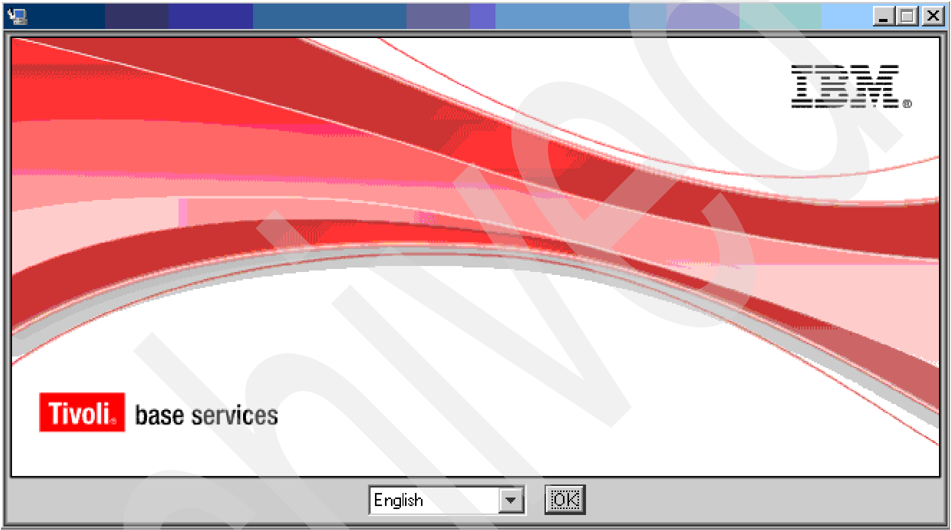


Figure 4-142 Language selection

3. At the IBM Tivoli Service Request Manager window, under Service Desk, click **Service Desk Content - Best Practice** (Figure 4-143).

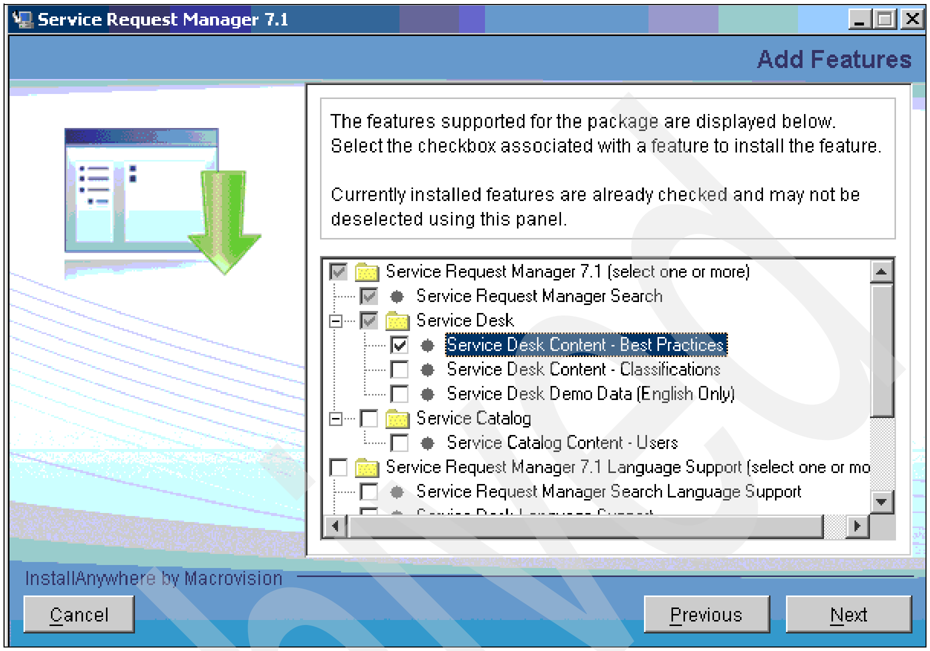


Figure 4-143 Service Desk Content - Best Practice

4. Type in the passwords and then click **Next** (Figure 4-144) to continue.

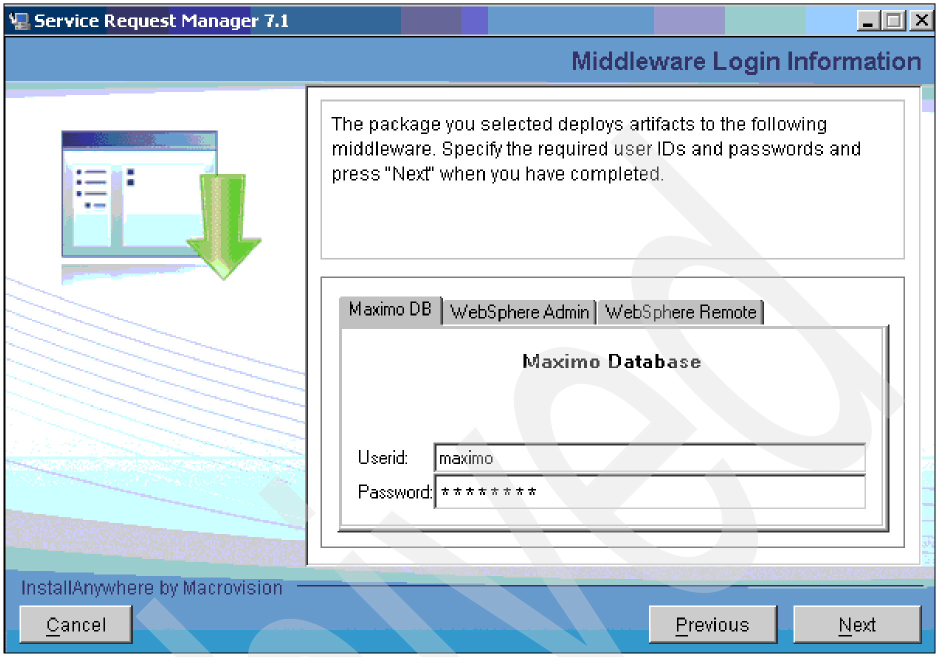


Figure 4-144 Login information

5. Click **Next** (Figure 4-145) to continue in the pre-install summary window.

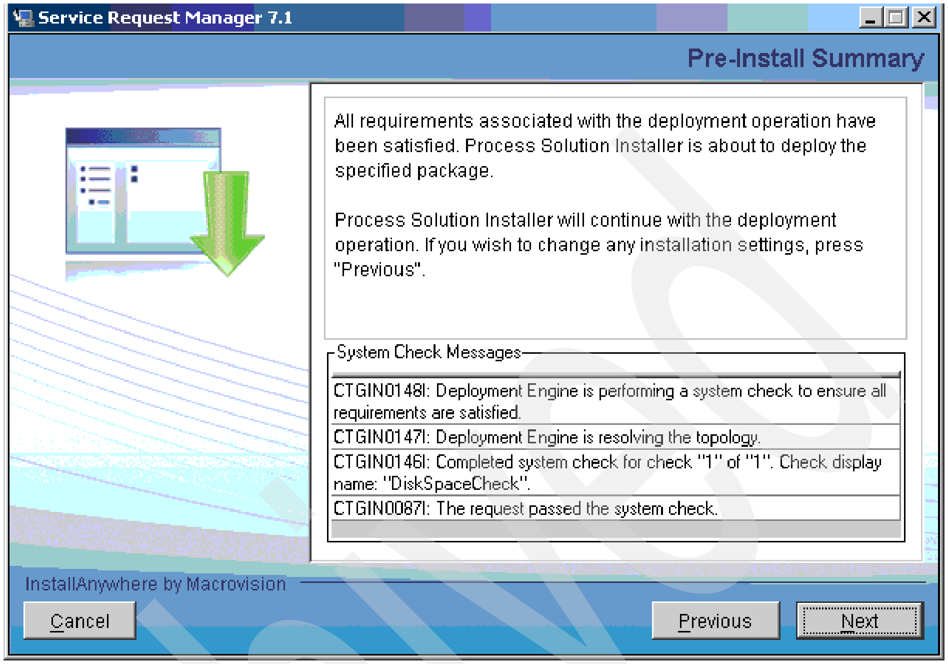


Figure 4-145 Pre-install summary

The installation is initiated (Figure 4-146).

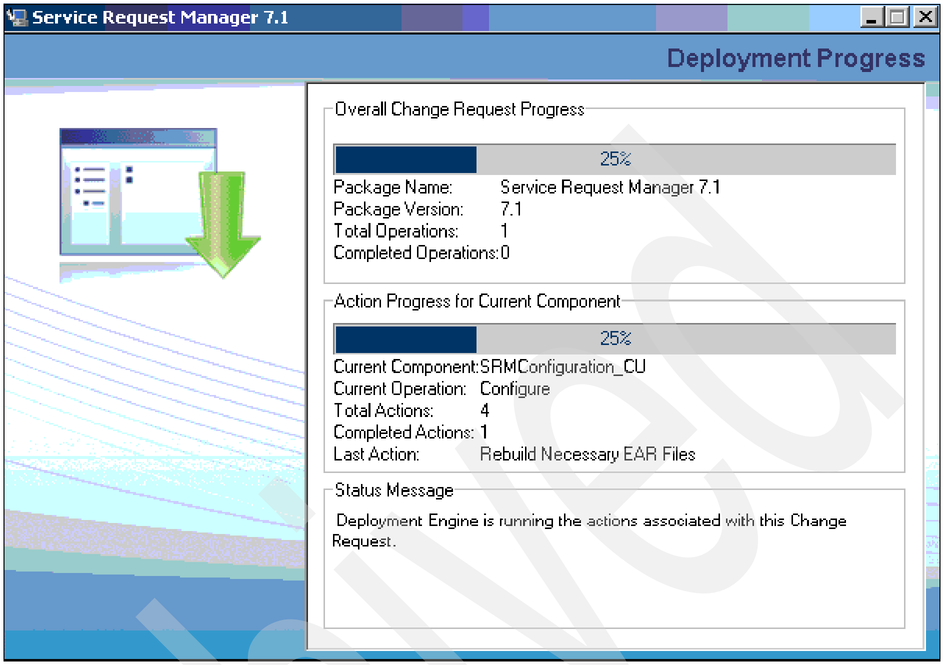


Figure 4-146 The installation is initiated

6. Once the application is complete, click **Done** to close the window (Figure 4-147).

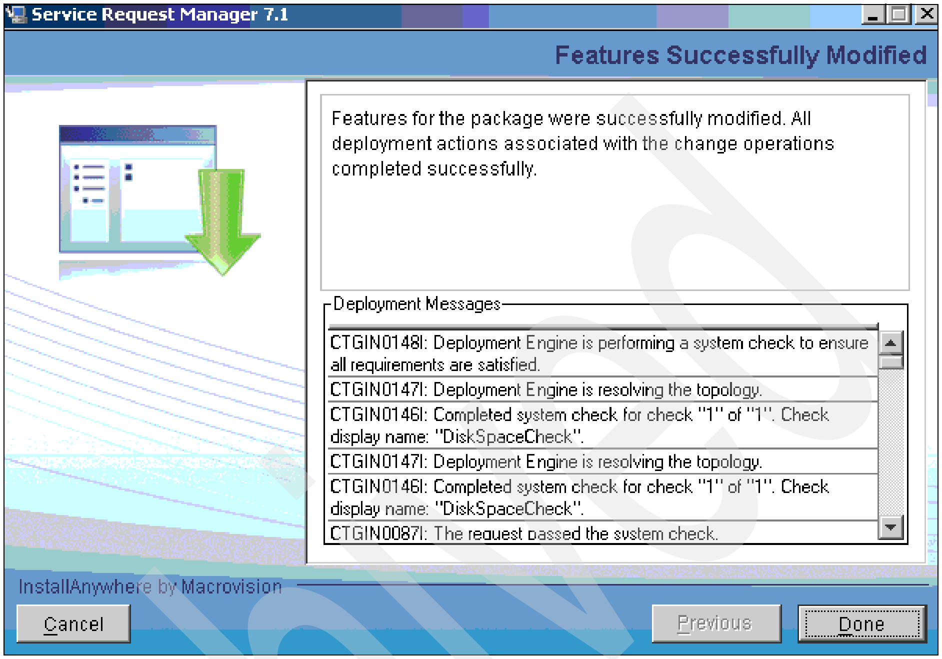


Figure 4-147 Features successfully modified

Service Desk Content-Classifications installation

Perform the following steps for Service Desk Content-Classifications installation:

1. Start the launchpad to start the installation. Select the language and click **OK** to continue (Figure 4-148 on page 247).

Figure 4-148 Select the language

2. Select the **Service Desk Content-Classifications** option and then click **Next** (Figure 4-149).

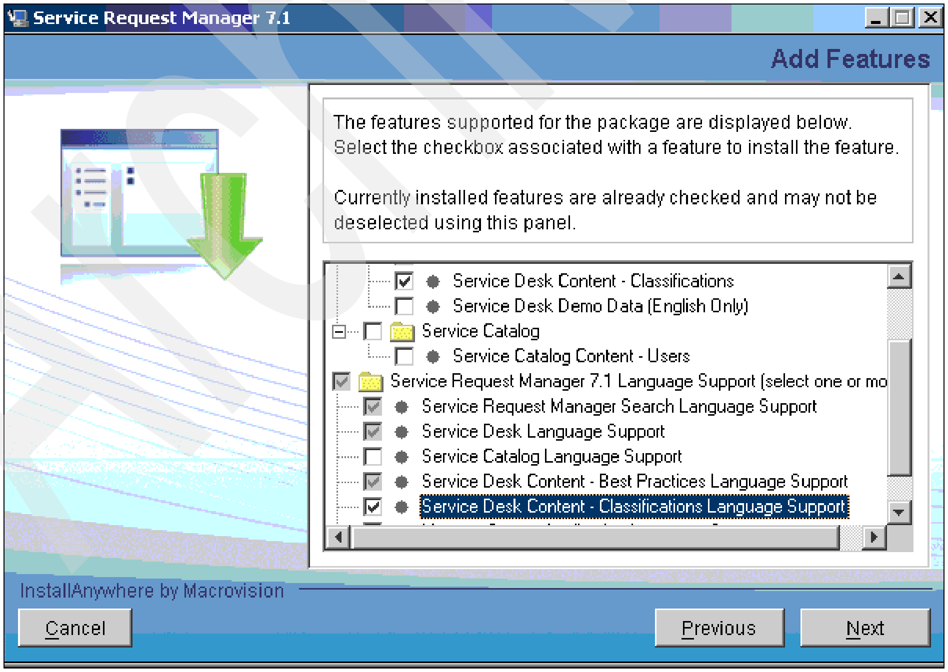


Figure 4-149 Service Desk Content-Classifications option

3. Enter the passwords and click **Next** (Figure 4-150) to continue. The installation is initiated.

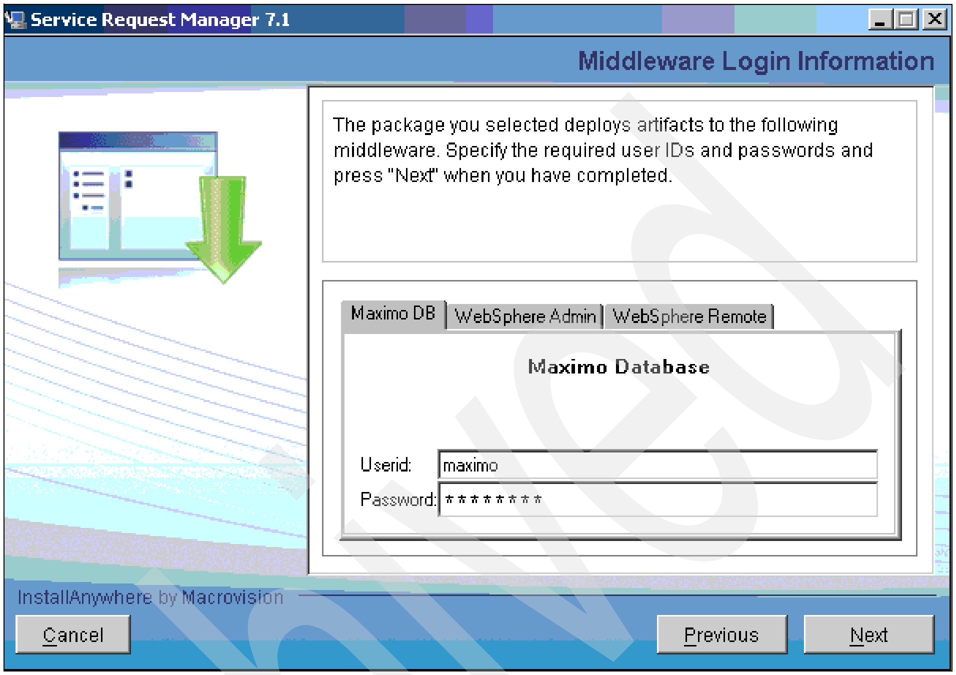


Figure 4-150 Login information

4. Enter the passwords and click **Next** (Figure 4-151) to continue. The installation is initiated.

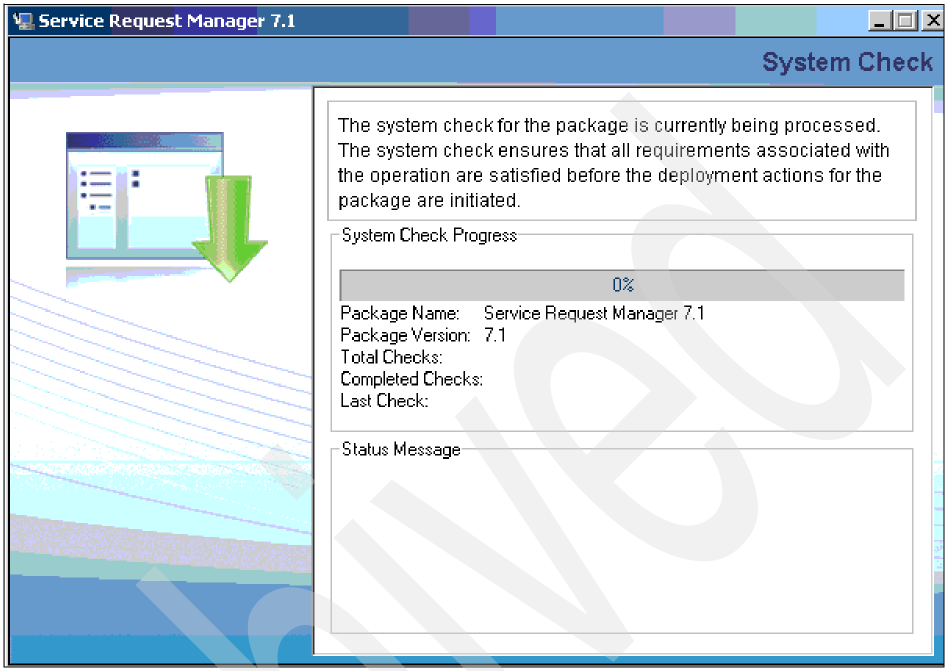


Figure 4-151 Installation is initiated

5. You are presented with a pre-installation summary. Click **Next** (Figure 4-152).

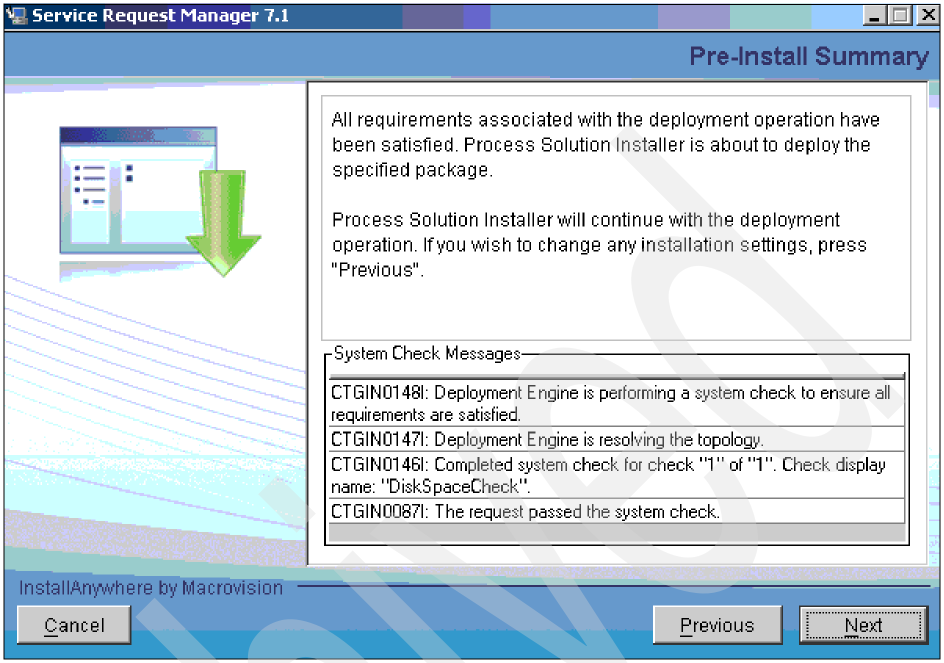


Figure 4-152 Pre-installation summary

You will see the Deployment Progress window (Figure 4-153).

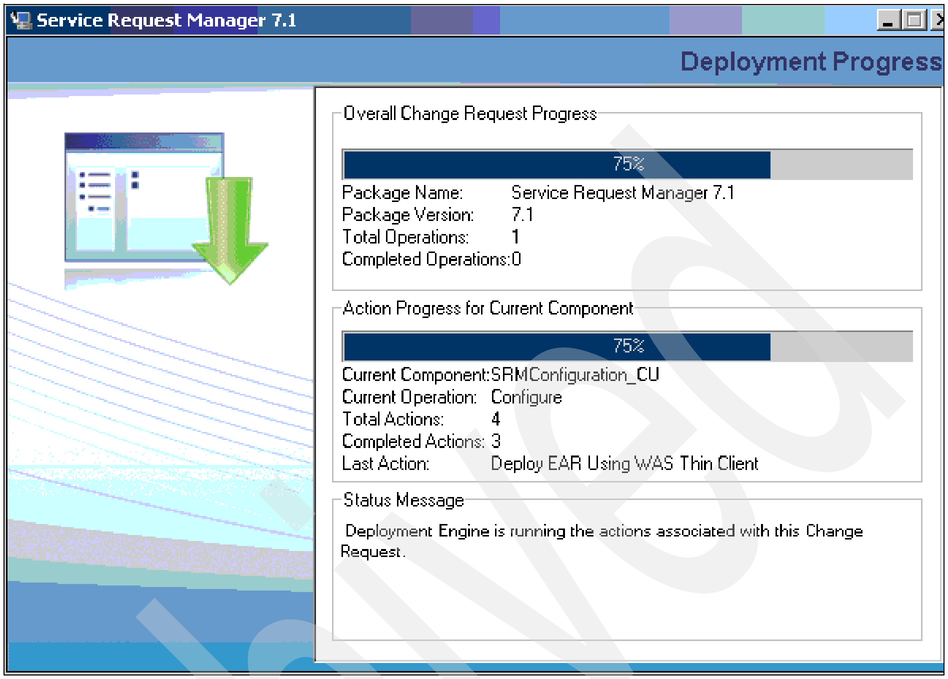


Figure 4-153 Deployment Progress

6. At the Package Successfully Deployed window, click **Done** (Figure 4-154). The Installing Process Solution Package window displays and then returns to the IBM Tivoli Service Request Manager window.

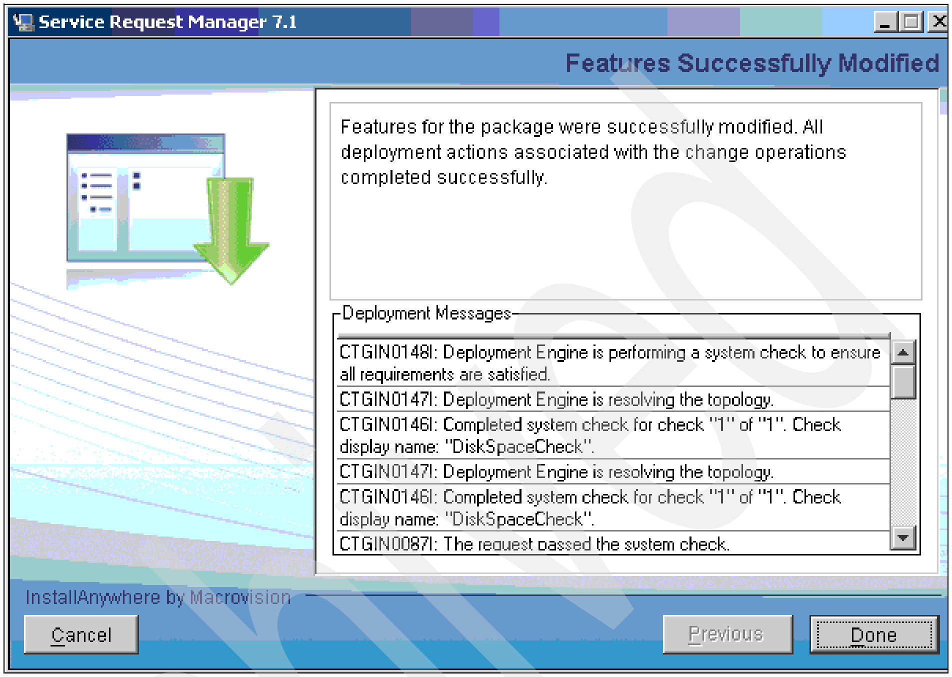


Figure 4-154 Features Successfully Modified

7. At the Package Successfully Deployed window, click **Done**.

> **Note:** The installation of the remaining options (for example, Service Catalog and Survey) are similar to the above installation. As previously mentioned, these can be installed all at once or on a one by one basis.

4.5 Using Oracle Database

Tivoli Service Request Manager V7.1, by default, is shipped with the DB2 database. Oracle Relational Database is also supported. The supported version of the Oracle Database is Oracle 10g.

To use Oracle with Tivoli Service Request Manager V7.1, you have to procure a separate Oracle Database and install the database as per the instructions provided in this section.

Note: Oracle Instance must be prepared before installing Tivoli Service Request Manager V7.1.

4.5.1 Oracle Database installation

To install Oracle on a Windows 2003 SP2 server, perform the following steps:

1. Run `Setup.exe` from the Oracle10g Software CD.
2. Select Installation Type **Standard Edition**, enter the password, and click **Next**, as shown in Figure 4-155.

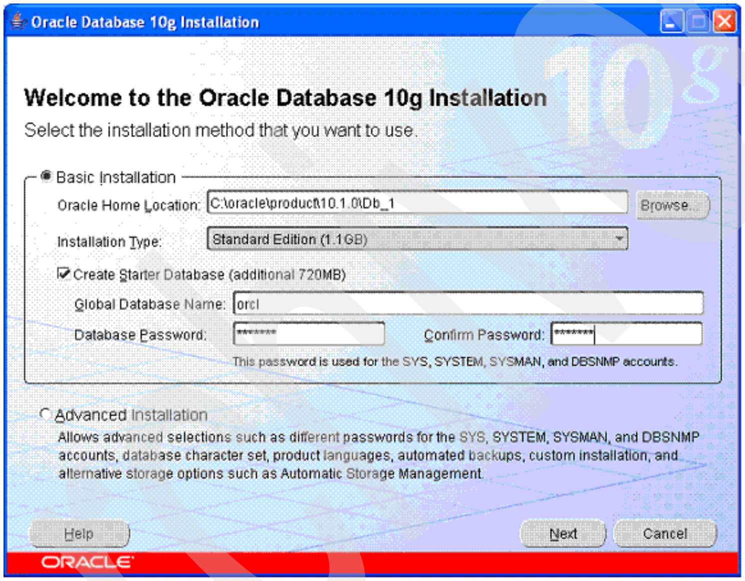


Figure 4-155 Oracle 10 g Standard installation

You may get an error window due to a bug in the installation process; always click **Continue** when ever you get this window, as shown in Figure 4-156.

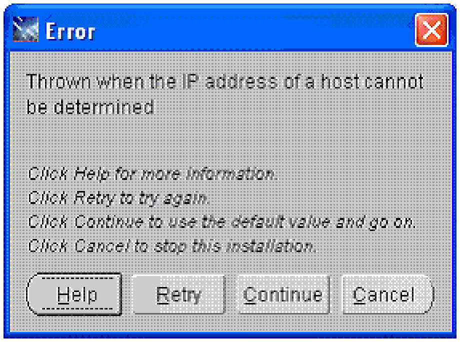


Figure 4-156 Install error message

3. Read the Installer summary and click **Install**, as shown in Figure 4-157.

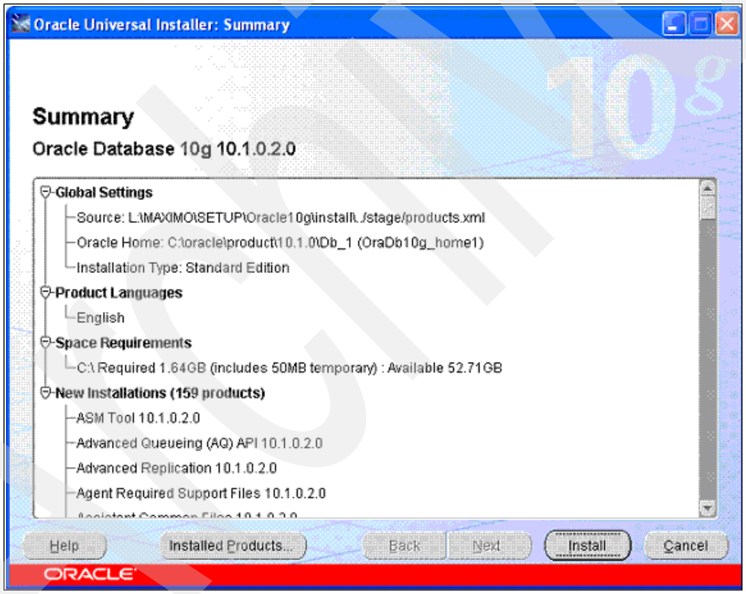


Figure 4-157 Install summary

The installation proceeds as shown in Figure 4-158 on page 255.

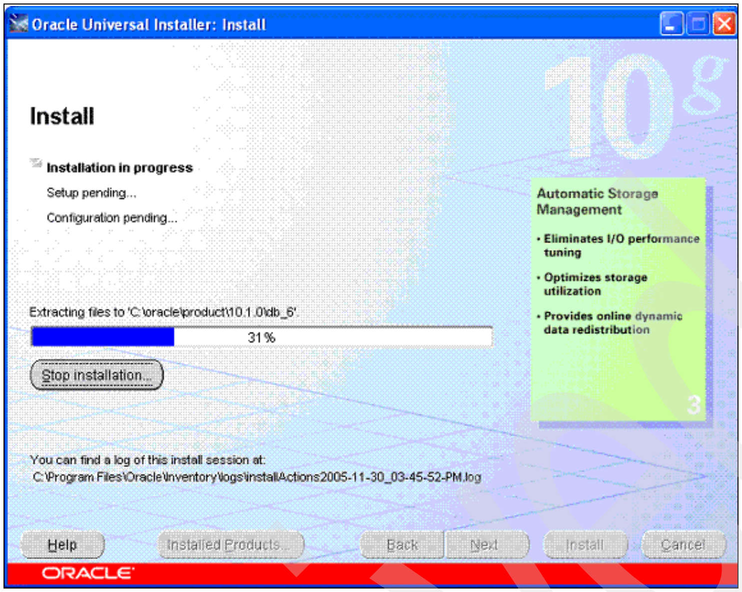


Figure 4-158 Installation progress

The installation will continue and will take some time. If you are installing from CDs, you will be asked to insert additional CDs.

4. On the Database Configuration Assistant window, click **Password Management** to enable the CTXSYS account, as shown in Figure 4-159. This is required for Text search.

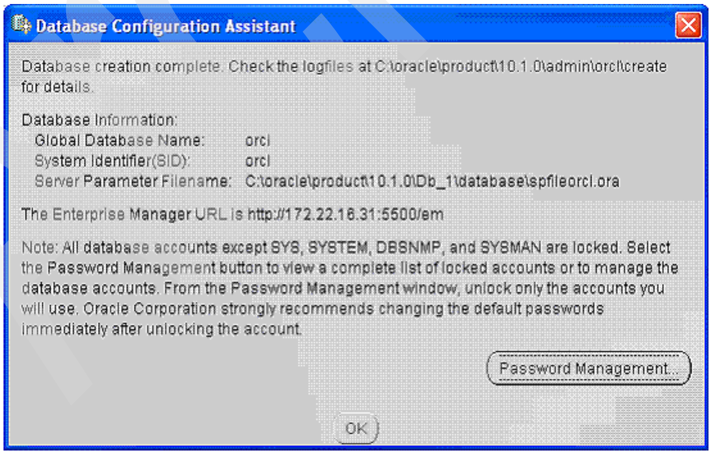


Figure 4-159 Password management

5. Unlock the CTXSYS account by unchecking the **Lock Account** check mark. By default, the account is locked, as shown in Figure 4-160. Enter a password, preferably CTXSYS/CTXSYS; otherwise, make a note of the password for later use.

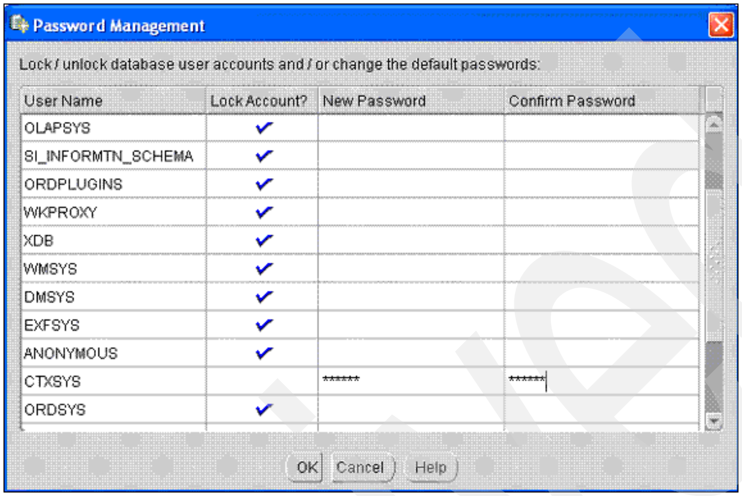


Figure 4-160 Password Management CTXSYS

6. Click **OK** to complete.
7. At this point the database installation and instance creation is completed. Click **Exit**, as shown in **Figure 4-161 on page 257**.

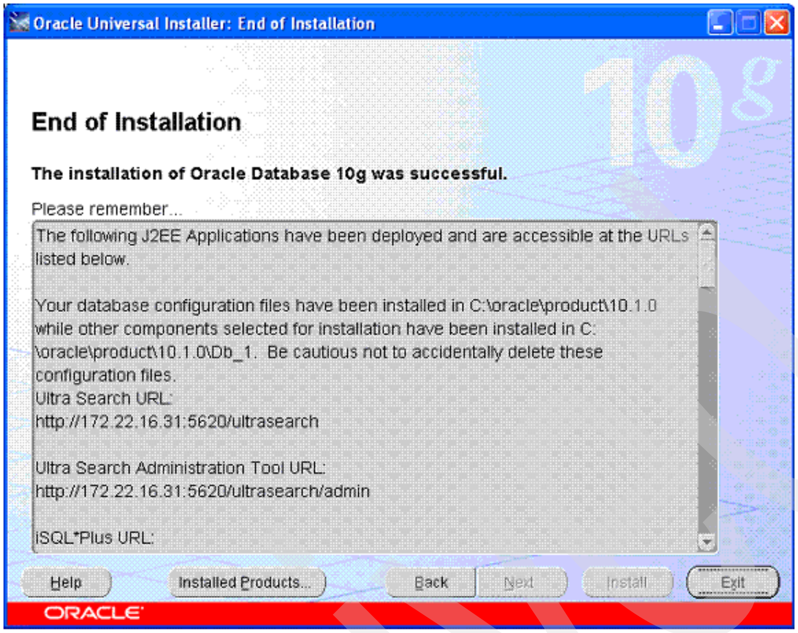


Figure 4-161 End of installation

At this point in the installation, you will create the Tivoli Service Request Manager database.

The default database (Orcl) created by Oracle may not be suitable for Tivoli Service Request Manager. Go to the Database Configuration Assistant and create a new database for Tivoli Service Request Manager. These are the steps you need to follow if you need to create additional databases after installing Oracle:

1. Select **Start** → **Programs** → **Oracle** → **Configuration and migration tools** to access the Database Configuration Assistant.

2. Go to Database Configuration Assistant, shown in Figure 4-162. Click **Next** to select from the options to create a database.

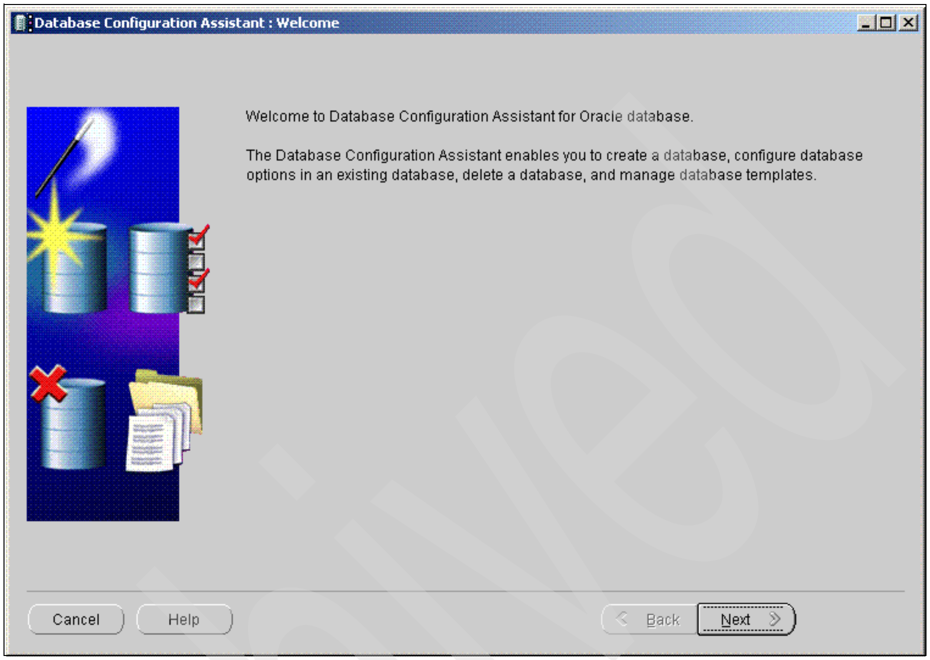


Figure 4-162 Database Configuration Assistant

3. Select the **Create a Database** radio button option, as shown in Figure 4-163, and click **Next**.

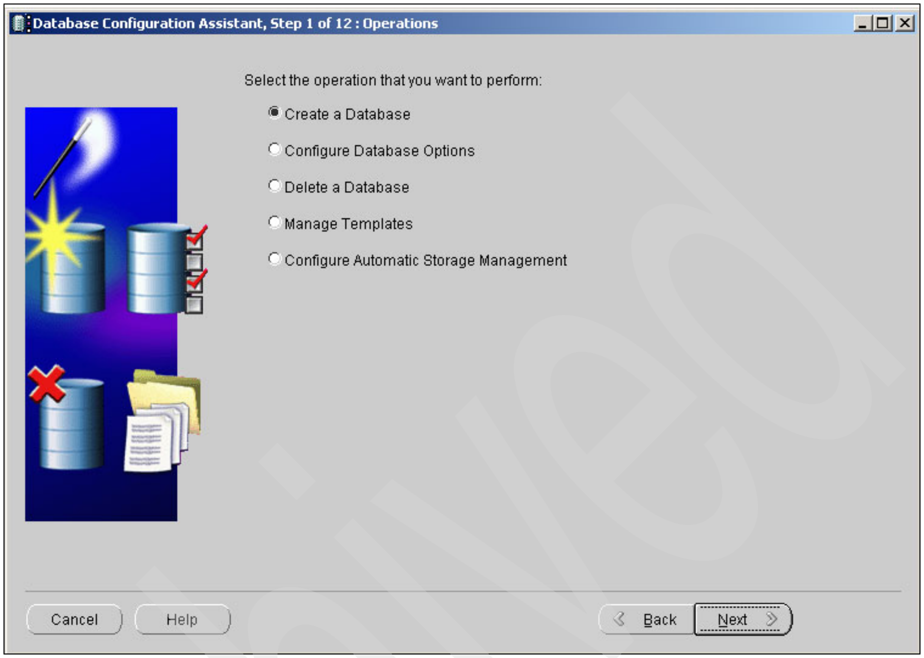


Figure 4-163 Step 1of 12 Operations

4. Select the **General Purpose** database template radio button option, as shown in Figure 4-164. Click **Next**.

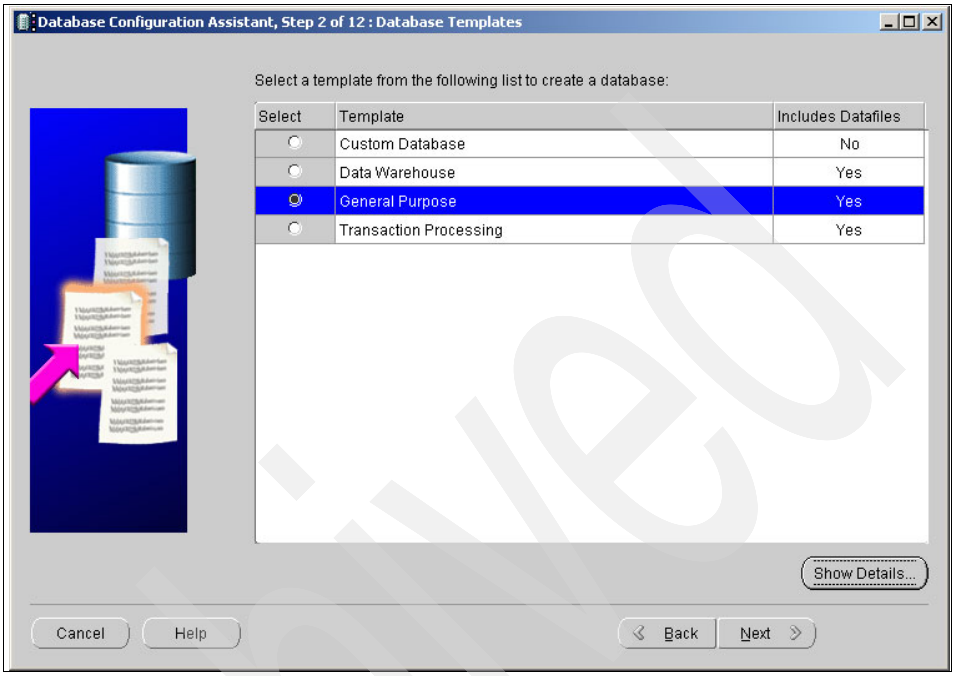


Figure 4-164 Step 2 of 12 Database Templates

5. Provide a database name, for example, maximo, as shown in Figure 4-165 on page 261.

> **Note:** Any database name can be provided for the database. It does not need to be Maximo.

6. Click **Next**.

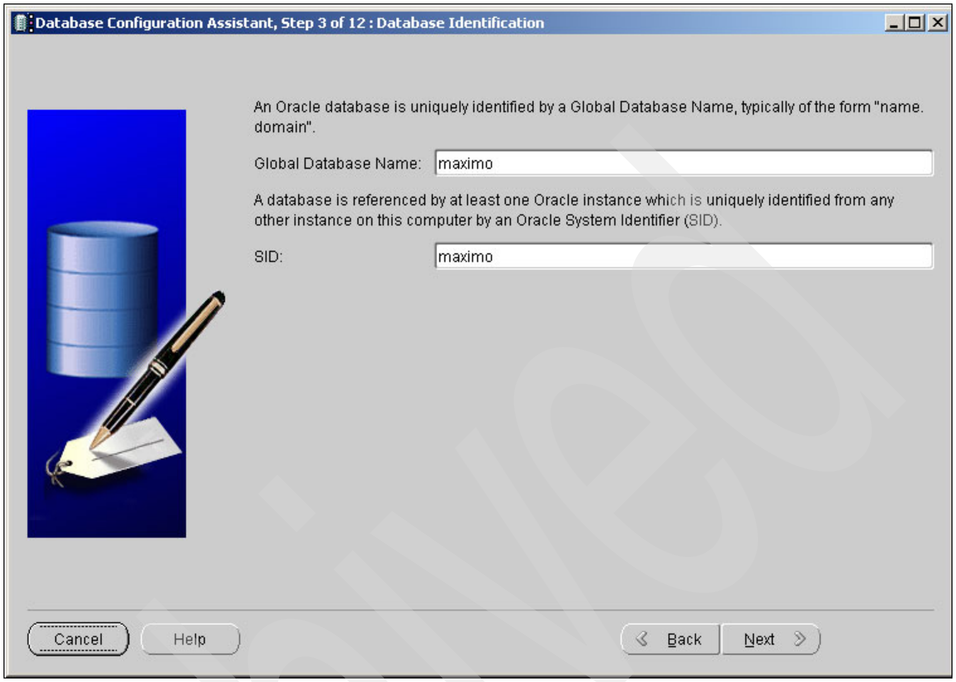


Figure 4-165 Step 3 of 12 Database Identification

7. In Figure 4-166, choose the default option (the check box **Configure the Database with Enterprise Manager** is checked). Click **Next**.

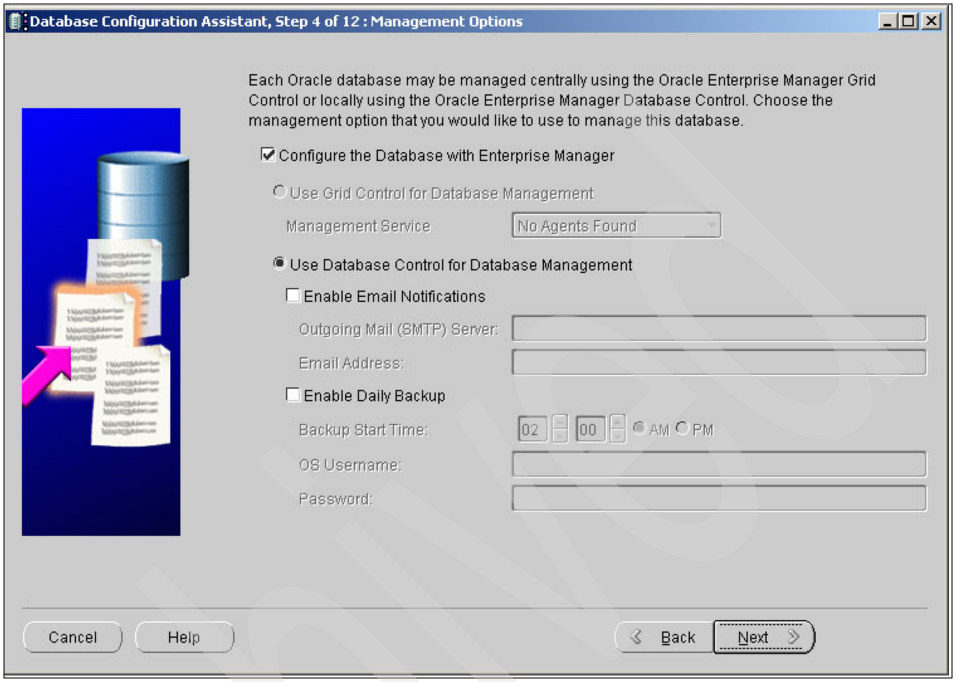


Figure 4-166 Step 4 of 12 Management Options

8. In Figure 4-167, provide a password. Please make note of the password for future reference purposes. Click **Next.**

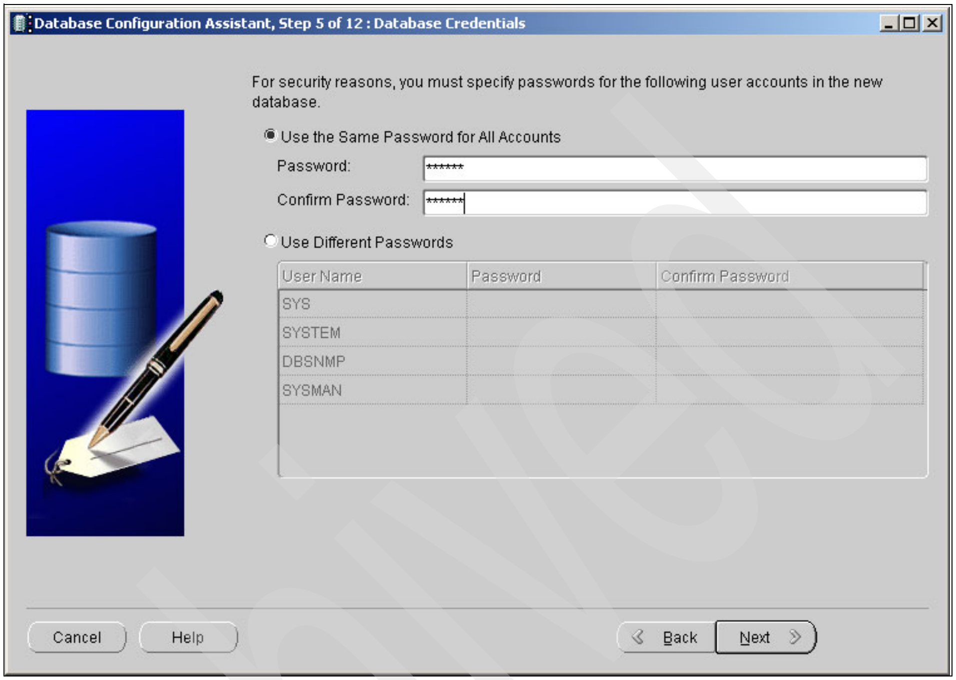


Figure 4-167 Step 5 of 12 Database Credentials

a. In Figure 4-168, choose the default option **File System**. Click **Next**.

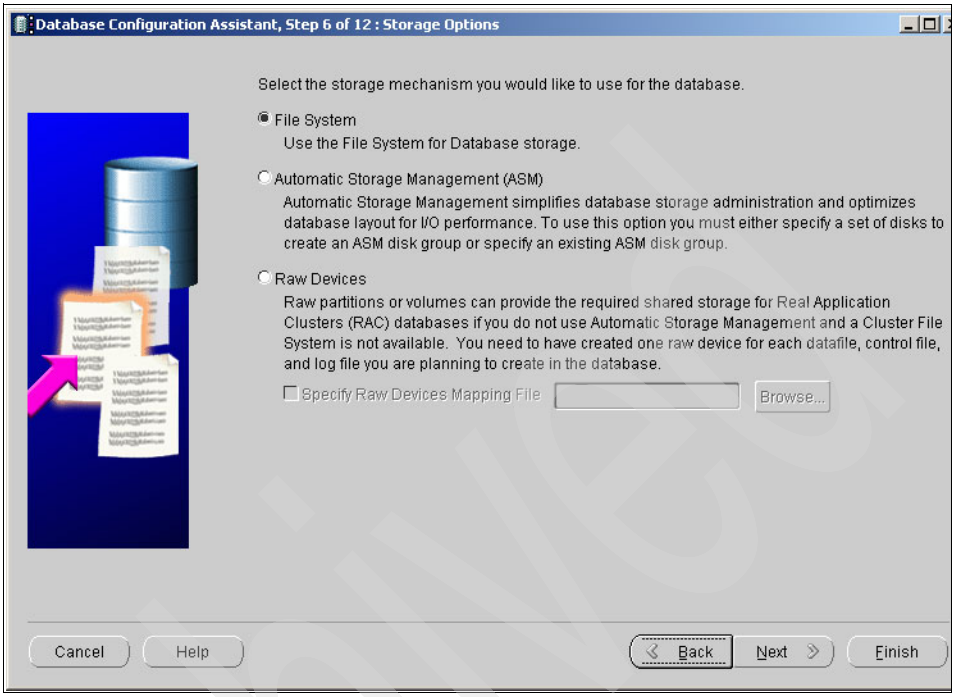


Figure 4-168 Step 6 of 12 Storage Options

9. In Figure 4-169, choose the default option (**Use Database file location from Template**). Click **Next**.

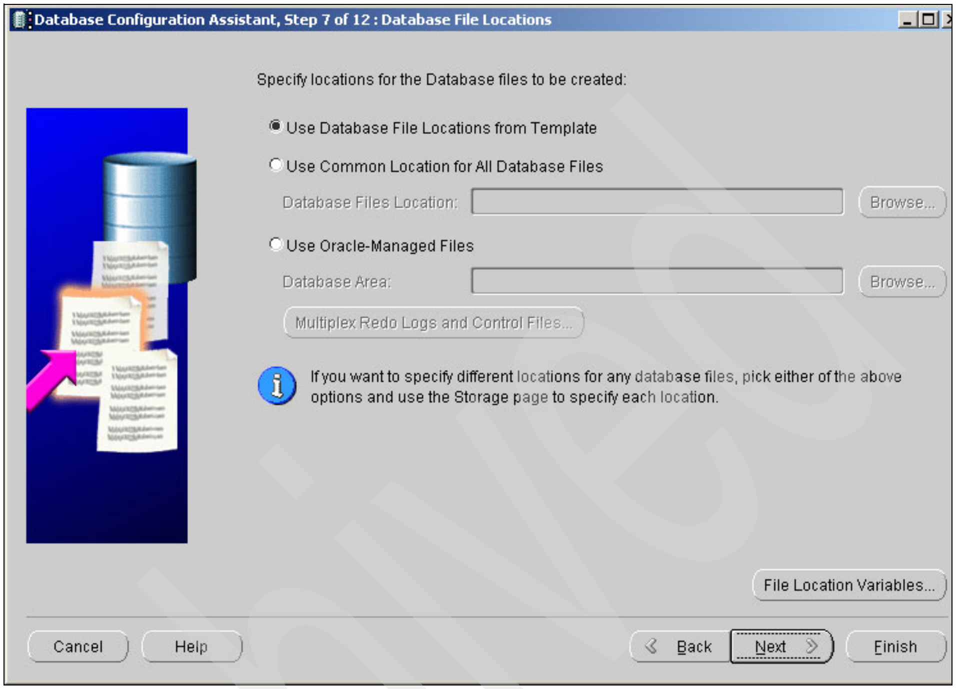


Figure 4-169 Step 7 of 12 Database File Location

10. In Figure 4-170, choose the default option (select **Specify Flash Recovery Area and** use the default values for this option). Click **Next**.

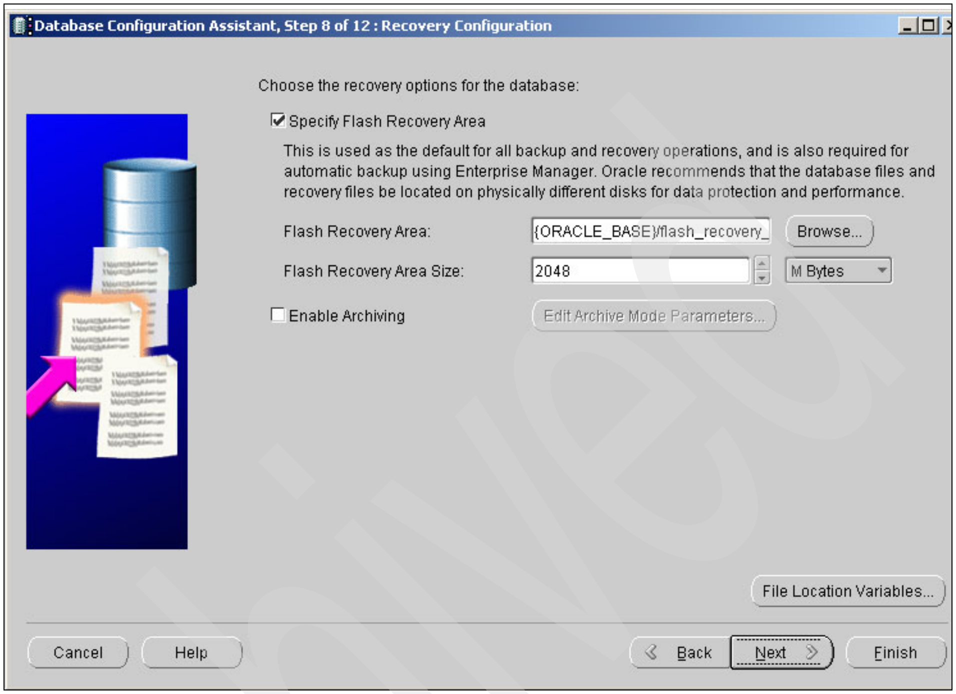


Figure 4-170 Step 8 of 12 Recovery Configuration

11. Select the **Custom Scripts** tab, as shown in Figure 4-171.

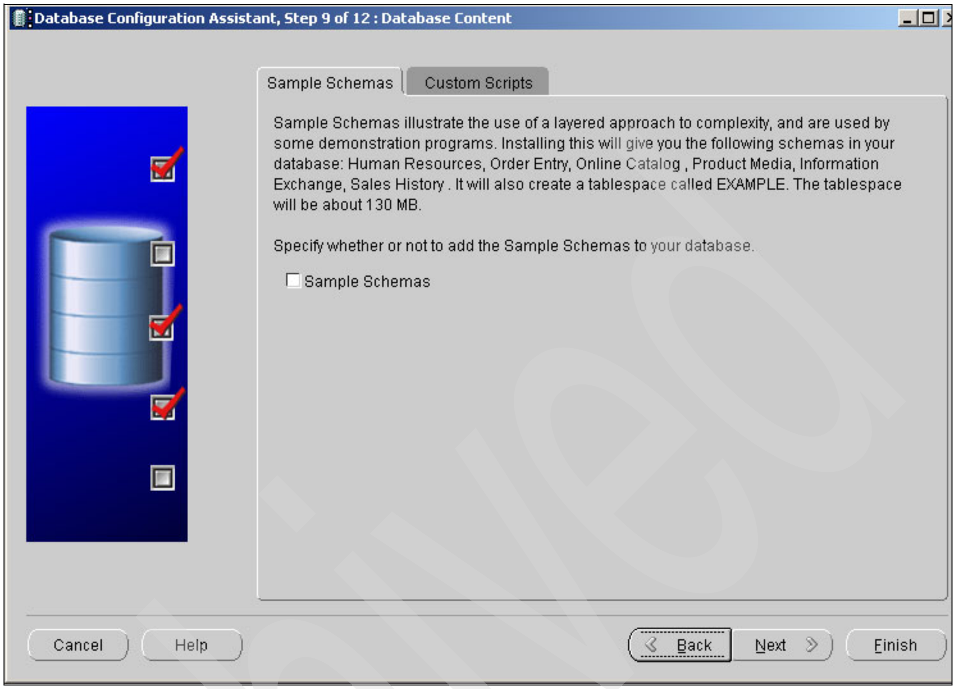


Figure 4-171 Step 9 of 12 Database Content

12. The window shown in Figure 4-172 appears.

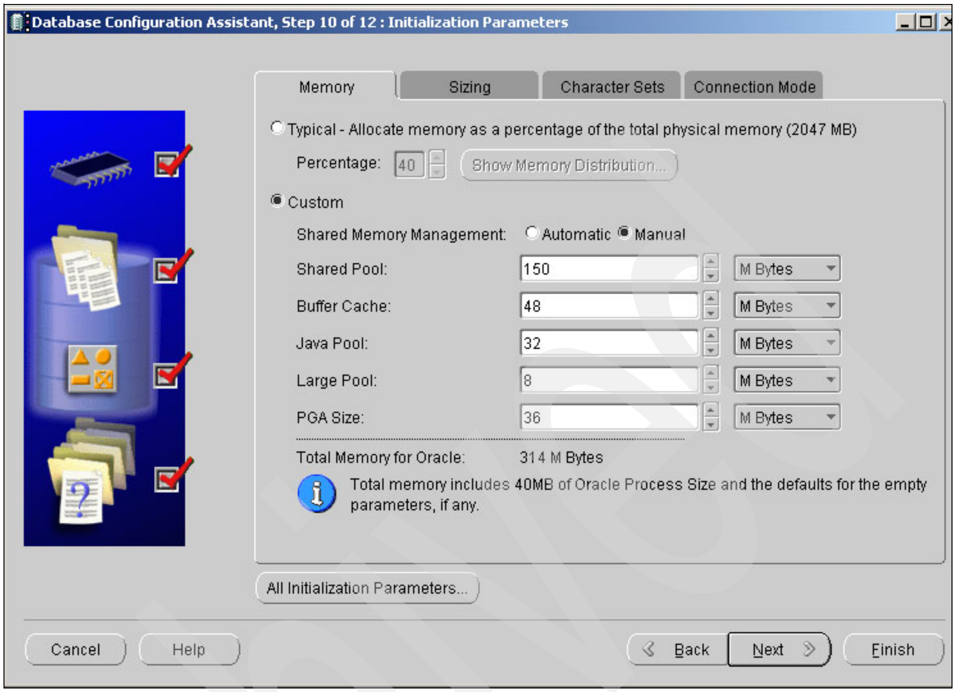


Figure 4-172 Step 10 of 12 Installation Parameters

Click the **Character Set** tab, as shown in Figure 4-173. Use Unicode if you want to enable Multilanguage. Click **All Initialization Parameters**.

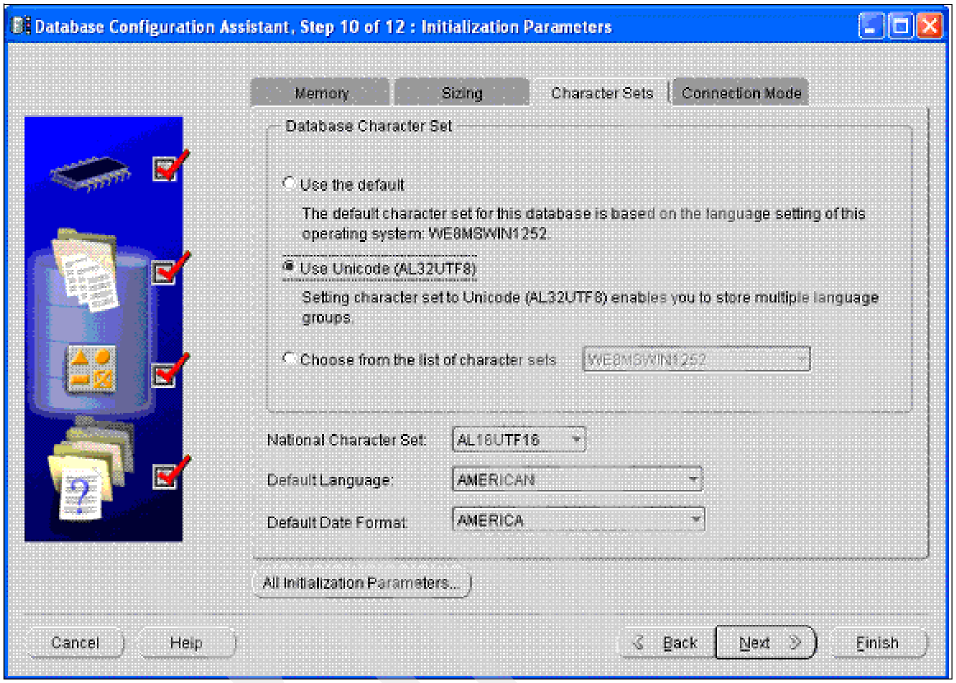


Figure 4-173 Character Sets

13. Change the open_cursors value to 1000 and change nls_length_semantics to CHAR, as shown in Figure 4-174. Click **Close**.

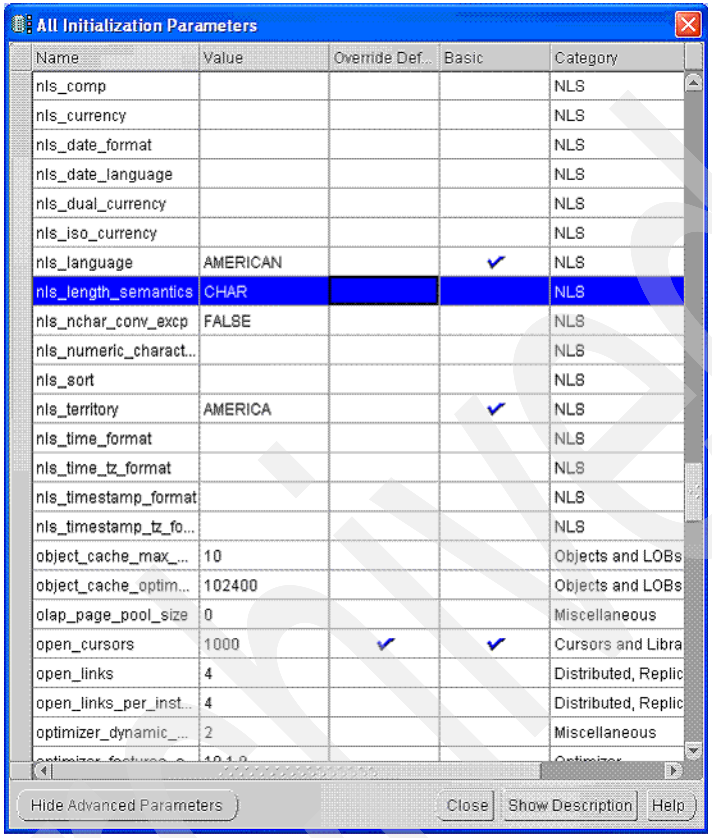


Figure 4-174 All Installation Parameters

14. Click **Next** in the Initialization parameters window to proceed.

15. Click **Next** in the Database Storage window on Figure 4-175.

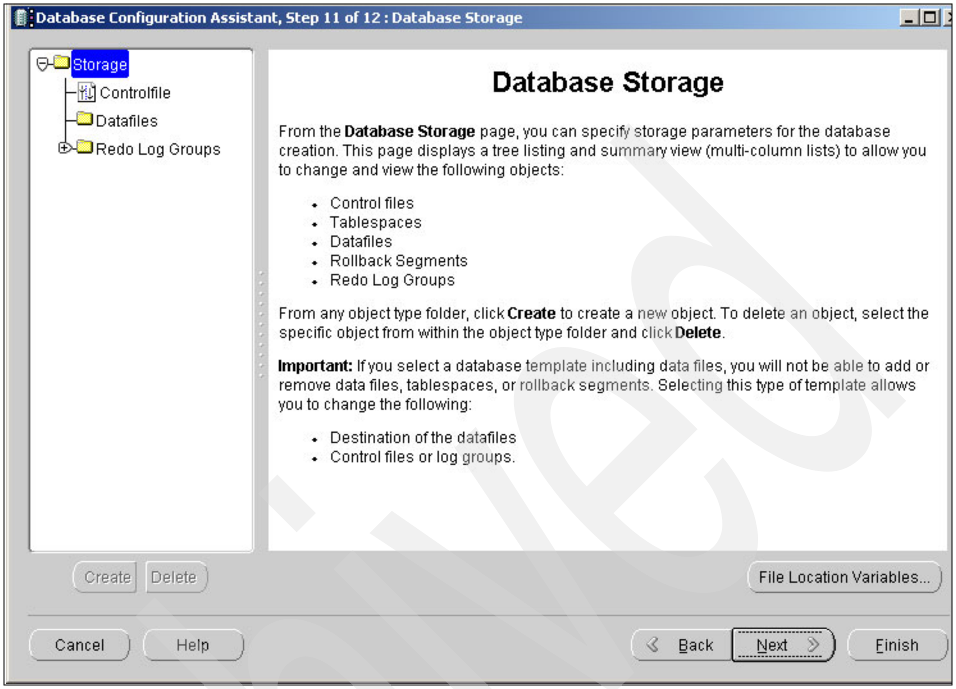


Figure 4-175 Steps 11 of 12 Database Storage

Make sure that Create Database is selected in Figure 4-176.

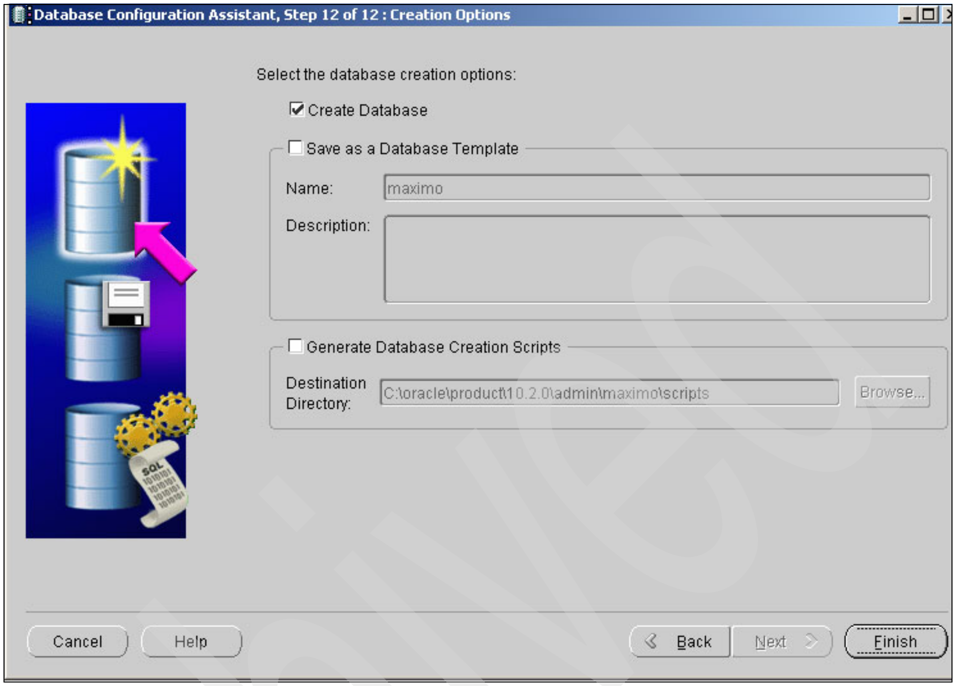


Figure 4-176 Step 12 of 12 Creation Options

16. Click **OK** in the confirmation window shown in Figure 4-177 on page 273 and the database will be created. The creation of the database proceeds, as shown in Figure 4-178 on page 273.

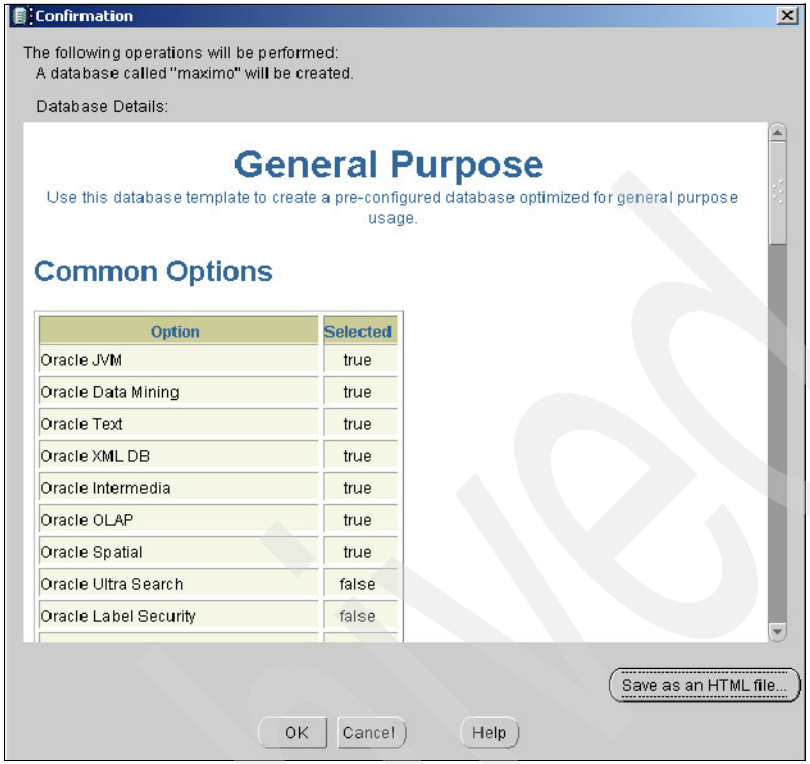


Figure 4-177 Confirmation

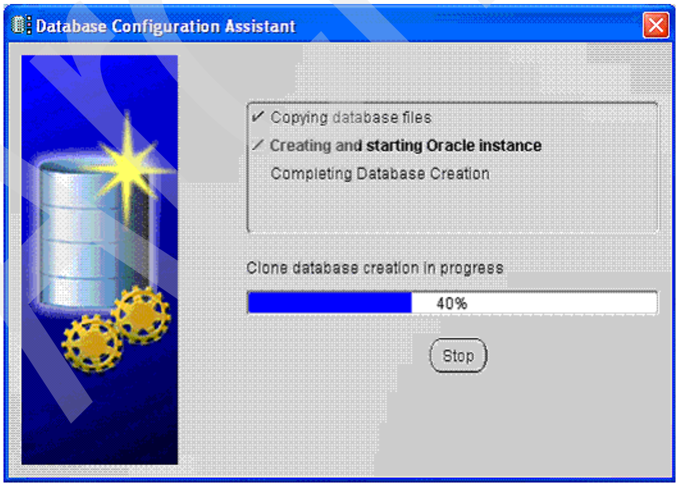


Figure 4-178 Database creation in progress

17. In the Password management window, select **Password Management** to enable the CTXSYS account. This is required for Text search. Unlock the CTXSYS account (uncheck the check mark). By default, the account is locked. Enter a password, preferably CTXSYS/CTXSYS; otherwise, make a note of the password for later use. Click **OK** and then **Exit**.
18. Now that the new database is created, you need to create a table space to hold maximo tables and create a maximo account. Log in to the database using sqlplus:

    ```
    Sqlplus system/<password you entered while installing> as sysdba
    ```

    If you have multiple instances, then use:

    ```
    sqlplus username/password@instancename
    ```

19. Use the following command to create the maximo table space. The table space name can be anything; we use maximo as an example table space name:

    ```
    Create tablespace maximo datafile
    'C:\oracle\product\10.1.0\oradata\merlin\maximo.dbf' size 1000M
    autoextend on;
    ```

> **Note:** The directory of the data file needs to be explicitly specified as defined above. The directory used here (C:\oracle\product\10.1.0\oradata\merlin\maximo.dbf) is an example. You should use the value from your installation.

Example 4-1 shows the output of this command.

Example 4-1 Table space creation

```
SQL*Plus: Release 10.2.0.1.0 - Production on Tue Mar 18 16:09:34 2008
Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> Create tablespace maximo datafile
'c:\oracle\product\10.2.0\oradata\maximo\maximo.dbf' size 100
0M autoextend on;
Tablespace created.
SQL> create user maximo identified by maximo default tablespace maximo
temporary tablespace temp;
User created.
SQL> Grant connect, dba to maximo;
Grant succeeded.
```

```
SQL> grant create job to maximo;
Grant succeeded.
SQL>
```

20. Use the following commands to create user maximo and provide access grants:

    ```
    Create user maximo identified by maximo default tablespace maximo
    temporary tablespace temp;
    Grant connect, dba to maximo;
    Grant create job to maximo;
    ```

21. Next, you need to grant permission for text search procedures. To do this, connect to the database as ctxsys from Oracle SQLplus and run the following commands:

    ```
    Sqlplus ctxsys/ctxsys@<db alias>
    Grant all on ctx_ddl to public;
    ```

Example 4-2 shows the output of these commands.

Example 4-2 Grant permission for text search procedure

```
SQL*Plus: Release 10.2.0.1.0 - Production on Tue Mar 18 16:47:48 2008

Copyright (c) 1982, 2005, Oracle.  All rights reserved.


Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> connect ctxsys/ctxsys@maximo as sysdba
Connected.
SQL> Grant all on ctx_ddl to public;

Grant succeeded.

SQL>
```

At this point, you will need to install additional middleware components (directory server and J2EE server) as required. In the Deployment Choices window (Figure 4-12 on page 125), you have to unselect the **Database Server** option and proceed with the installation of the other middleware components.

Next, you proceed with installing the base services and Tivoli Service Request Manager, as described in 4.3.5, “Installing the base services on Windows” on page 197 and 4.4, “Installing Tivoli Service Request Manager” on page 223. There is one additional step for Oracle Database, after installing Tivoli Service Request Manager, which is described in the following section.

4.5.2 Tivoli Service Request Manager V7.1 configuration steps

Once you have installed Tivoli Service Request Manager, you need to configure Tivoli Service Request Manager V7.1 to talk to the Oracle Database.

1. The first thing you should do is to set up the Maximo properties. Tivoli Service Request Manager application server connects using one account that is specified in the maximo.properties file. All Tivoli Service Request Manager user accounts are stored and managed in the Maximo tables. So, after creating this database account, update maximo.properties with the database account name and the database URL connection information (Table 4-3), as described in the maximo.properties file.

Table 4-3 Maximo properties

Property	Description
mxe.db.driver oracle	Maximo driver for Oracle. By default, jdbc.driver.OracleDriver.
mxe.db.schemaowner	Maximo Database schema name.
mxe.name	Maximo application server name.
mxe.db.url	Database connection URL: `jdbc:oracle:thin:@<database server name>:<port number>:<database SID>`.
mxe.db.user	Database user for database authentication.
mxe.db.password	Database User password.
mxe.rmi.port	Application RMI port.
mxe.encrypted	When set to True, the Maximo user password can be stored as an encrypted value in the Maximo.propertie file.
mxe.registry.port	Maximo Registry port.

Example 4-3 on page 277 show the maximo.properties file.

Example 4-3 Maximo properties

```
mxe.db.driver=oracle.jdbc.driver.OracleDriver
mxe.db.schemaowner=maximo
mxe.name=MXServer
mxe.db.url=jdbc:oracle:thin:@belgrade:1521:maximo
mxe.db.user=maximo
mxe.db.password=maximo
mxe.rmi.port=0
mxe.encrypted=true
mxe.registry.port=13400
mxe.encrypted=true
```

When you use the database installation and configuration wizard to create the database, a database alias entry is created in tnsnames.ora. The alias created is the global name you specified. As in our example, Maximo is created in tnsnames.ora and you can use Maximo as the SID in the database URL.

At this point, you have everything you need to run **maxinst** and create the Maximo merlin database.

2. Run **maxinst** to create the Maximo database.
   a. Go to the Tivoli Service Request Manager root directory and change the directory to <drive letter>:\..IBM\maximo\tools\maximo.
   b. Copy maxdemo.ora or maximo.ora as appropriate to the local directory and run **maxinst**. Make sure maximo.properties is set up correctly.
   c. The **maxinst** command will run and create the database tables and necessary database objects in the database instance specified in maximo.properties. It also writes a log file under the log directory. You may check the log files to see if there are any errors or to check the status of Maximo database creation. The **maxinst** command will stop running if it encounters any errors. If so, you need to fix the error, clean the database, and re-run **maxinst**.

4.6 Post installation steps

This section describes tasks that can be completed after the installation of Service Request Manager.

4.6.1 Initial Data configuration

Once you have successfully installed and configured the Tivoli Service Request Manager components, there are several data configuration tasks you must complete prior to using Tivoli Service Request Manager.

Signing a default user ID

User management is managed though the directory server you have configured to use with base services. When first installed, base services contain the default user IDs shown in Table 4-4, which are members of the specified security group.

> **Important:** Before you begin this procedure, ensure you have the following users and groups created in your LDAP repository.

Table 4-4 Users and groups

User	Groups
wasadmin	
maxadmin (maxadminusr for MS Active Directory)	maxadmin
mxintadm	maxadmin
maxreg	

The default password for each user ID is the same as the User Name (for example, maxadmin is both the user name and default password).

> **Note:** User names and passwords are case sensitive. The default user names and passwords are lowercase.

> **Note:** The default password for each user ID is the same as the User Name (for example, maxadmin is both the user name and default password).

To sign in, complete the following steps:

1. Open a browser window.
2. Navigate to the base services log in URL. For example:

```
http://hostname:<port>/maximo
```

3. Enter the user name maxadmin (lower case).
4. Enter the password maxadmin (lower case) and press Enter. The software displays an empty Start Center.

Create currency codes

You must define a currency code for an organization.

To define a currency code for an organization, complete the following steps:

1. Open the Currency Code application for Users by selecting **Goto** → **Financial** → **Currency Code**.
2. Click **New Row**.
3. Enter a currency name, for example, USD.
4. Click **Save**.

Create item and company sets

You must define item and company sets for an organization.

To define a currency code for an organization, complete the following steps:

1. Open the Sets application for Users by selecting **Goto** → **Administration** → **Sets**
2. Click **New Row**.
3. Enter a company set name, for example, IT Items.
4. Enter ITEM in the Type field.
5. Click **New Row**.
6. Enter an item set name, for example, IT Comps.
7. Enter COMPANY in the Type field.
8. Click **Save**.

Create an organization

You must define at least one organization for Service Request Manager.

To define an organization, complete the following steps:

1. Open the Organizations application by selecting **Goto** → **Administration** → **Organizations**.
2. Click the **New Organization** icon in the toolbar.
3. Enter an organization name in the Organization field. For example, ENGLENA.
4. Enter the base currency you defined in the Base Currency 1 field. For example, USD.
5. Enter the item set you defined in the Item Set field. For example, IT Items.
6. Enter the company set you defined in the Company Set field. For example, IT Comps.
7. Enter the default item status of PENDING in the Default Item Status field.
8. Click the **Sites** tab.
9. Click **New Row**.
10. Enter a site name in the Site field. For example, B901.
11. Click **Save**.

Create a general ledger account component

You must create a general ledger account component for Service Request Manager.

To create a general ledger account component, complete the following steps:

1. Open the Database Configuration application by selecting **Goto** → **System Configuration** → **Platform Configuration** → **Database Configuration**.
2. Select **GL Account Configuration** from the Select Action drop-down menu.
3. Click **New Row**.
4. Enter a component name in the Component field. For example, MYCOMPONENT.
5. Enter a numerical length for the component. For example, 5.
6. Enter a type for the component. For example, ALN.
7. Click **OK**.

Create a general ledger account

You must create a general ledger account for Service Request Manager.

To create a general ledger account, complete the following steps:

1. Open the Chart of Accounts application by selecting **Goto** → **Financials** → **Chart of Accounts**.
2. Click the name of your organization to select it. For example, click ENGLENA.
3. Select **GL Component Maintenance** from the Select Action drop-down menu.
4. Click **New Row**.
5. Add a GL Component value and then click **OK**. For example, 1234.
6. Click **New Row**.
7. Select your General Ledger Account.
8. Click **Save**.
9. Open the Organizations application by selecting **Goto** → **Administration** → **Organizations.**
10. Click the organization name you created. For example, ENGLENA.
11. From the Clearing Account field, select the General Ledger Account you just created.
12. Select **Active**.
13. Click **Save**.

Create default insert site

You must create a default insert site for Service Request Manager. To create a default insert site, complete the following steps:

1. Open the Users application by selecting **Goto** → **Security** → **Users.**
2. Search for maxadmin and then select it to open the record for maxadmin.
3. Enter the site you created earlier in the Default Insert Site field. For example, B901.
4. Enter the site you created earlier in the Storeroom Site for Self-Service Requisitions field. For example, B901.
5. Click **Save**.
6. Open the WebSphere Admin Console and restart the MXServer application server. Note that if you encounter an error message that indicates that the record is being updated by another user, log out as MAXADMIN and then log back in.

Create a worktype

You must create a worktype for Service Request Manager.

To create a worktype, complete the following steps:

1. Open the Organizations application by selecting **Goto** → **Administration** → **Organizations**.
2. Search for the organization you created, for example, ENGLENA.
3. Click the name of the organization to open the record for that organization.
4. Select **Work Order Options** → **Work Type** from the Select Action drop-down menu.
5. Click **New Row**.
6. Select **WORKORDER** as the Work Order class.

> **Note:** If you have CCMDB installed as well, you can also select PMCFGWO instead of WORKORDER.

7. Set the Work Type as AR.
8. Set Start Status as INPRG.
9. Set Complete Status as COMP.
10. Click **New Row**.
11. Select **WORKORDER** as the Work Order class.
12. Set the Work Type as UR.
13. Set Start Status as INPRG.
14. Set Complete Status as COMP.
15. Click **OK**.
16. Click **Save**.

Signing out and signing in

When you make changes to a security group that your user ID is a member of, you must sign out and sign in again in order to see the changes. For example, even though you have granted the MAXADMIN group permission to create start center templates, the actions are not visible until you sign in again.

1. Sign out as MAXADMIN.
2. Sign in as MAXADMIN.

Part 3

Demonstration scenarios and best practices

In this part, we focus on demonstration scenarios using some of the new features of Tivoli Service Request Manager, such as reporting, survey, and search functions. We also provide best practices for fine tuning and high availability of Tivoli Service Request Manager components.

5

Reporting, survey, and search functions

This chapter provides an overview of some of the new functions available with Tivoli Service Request Manager V7.1. We cover the following topics:

5.1 Reporting

This section will describe how to use the new reporting tool called Business Intelligence and Reporting Tools (BIRT), introduced in Tivoli Service Request Manager V7.1. BIRT is a open source flexible tool, that enables you to execute existing out of box reports, or create custom reports to fit your individual business need.

5.1.1 Business Intelligence and Reporting Tools

Business Intelligence and Reporting Tools (BIRT) is an Eclipse-based open source reporting system for Web applications, especially those based on Java and J2EE. BIRT has two main components: a report designer based on Eclipse, and a runtime component that has been embedded within the V7 application. BIRT also offers a charting engine that lets you add charts to your own application.

5.1.2 BIRT Report Designer

In this section, we discuss the BIRT terminology used with the BIRT Designer Tools:

- Data Explorer

  Data Explorer organizes your data sources (connections) and data sets (queries). The data set editor allows you to test your data set to ensure the report receives the correct data. This view also is used to design report parameters.

- Layout View

  The Layout View is a WYSIWYG editor that provides drag and drop creation of the presentation portion of your report.

- Palette

  The Palette contains the standard BIRT report elements, such as labels, tables, and charts. It is used in conjunction with the Layout View to design reports.

- Property Editor

  The Property Editor presents the most commonly used properties in a convenient format that makes editing quick and easy. BIRT also integrates with the standard Eclipse property view to provide a detailed listing of all properties for an item.

- Report Preview

  Using Report Preview, you can test your report at any time with real data. The preview is a window directly within Eclipse.

- Script Editor

  Scripting adds business logic to reports during data access, during report generation, or during viewing. The code editor provides standard Eclipse features for editing your scripts: syntax coloring, auto-complete, and more.

- Outline

  BIRT reports are organized as a tree structure with the overall report as the root, and separate categories for styles, report content, data sources, data sets, report parameters, and more. The Outline view provides a compact overview of your entire report structure.

- Cheat Sheets

  Learning a new tool is always a challenge, but Eclipse offers an innovative solution: Cheat Sheets. These are brief bits of documentation that walk you through new tasks.

- Library Explorer

  BIRT allows the reuse of report objects, such as tables, data sources, and styles. Objects created for reuse are stored in a library file. BIRT supplies the Library Explorer view to allow you to browse the contents of report libraries.

- Chart Builder

  Adding Charts to BIRT designs is expedited with the Chart Builder. Chart creation is separated into three phases: Select Chart Type, Select Data, and Format Chart.

- Expression® Builder

  BIRT expressions are basically simple scripts that return a value. Expressions are used for assigning data values to report elements, building image locations, hyperlinks, parameter default values, and many other places. Expressions are constructed within BIRT using the Expression Builder.

For more information, refer to the following Web site:

http://www.eclipse.org/birt/phoenix/

5.1.3 BIRT Report Engine

The BIRT Report Engine is a Java based engine that enables you to run predefined reports and create custom ones inside Tivoli Service Release Manager.

The BIRT Report engine consists of several parts.

Design API

- Enables you to create a report on the fly
- Opens an existing report and identifies various design elements
- Modifies an existing report

Runtime API

- Used to run the report
- Provides various report output formats
- Analyzes the report output

Report Engine use

The Tivoli Service Request Manager report engine overview is shown in Figure 5-1.

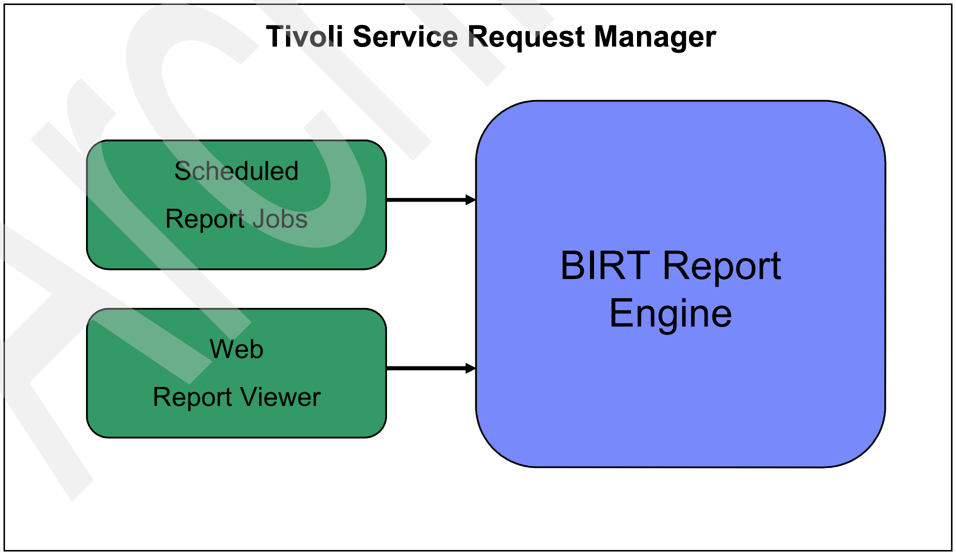


Figure 5-1 Report Engine use

Design and import process

In order to design reports for Tivoli Service Request Manager, you should use BIRT Report Designer.

All reports must be designed using the Scripted Data Source at this time to fully utilize the Tivoli Service Request Manager specific funcionality. You have the following functions available to you:

- Runtime Data Translation
- Time Zone Conversions

You can use the BIRT Report designer to test your reports. When reports are ready, import the report and all dependent files into Tivoli Service Request Manager. You have the following functions available to you:

- Report Library Files
- Report Lables (.properties files)
- Report resources (images and so on)

BIRT Report engine initialization

BIRT Report Engine is developed based on the OSGi platform (Eclipse 3.2 is also based on this) and is distributed as a set of OSGi bundles The OSGi platform is configured to initialize all the Report Engine services. The current implementation of the OSGi platform (Equinox) used by BIRT depends on a file system to load and initialize the platform classes.

The Platform classes are copied to a temporary location provided by the Application Server and that location is used to initialize the platform. The platform copy and initialization happens through a servlet.

Using BIRT Engine

BIRT Engine code is completely isolated and the BIRT Engine API is not accessible to the Tivoli Service Request Manager code. This isolation is required to run the BIRT Engine within a J2EE application.

The BIRT Engine functionality needed by Tivoli Service Request Manager is exposed through Tivoli Service Request Manager API interfaces and this API uses special mechanisms internally to access the BIRT Engine API. For example:

- Get Libraries used by the report.
- Get Parameters used by the report.

5.1.4 BIRT reporting process

In order to create or generate predefined BIRT reports on Tivoli Service Request Manager, you might consider three important tasks:

- Developing: Using BIRT Report Designer. This is recommended if you need to create another full report that is not available by default. Remember that BIRT Report Designer has to be downloaded and installed separately.
- Administration: Tivoli Service Request Manager V7.1 Report Administration Application.
- Executing: Tivoli Service Request Manager - Reports menu

5.1.5 BIRT Report Designer

The BIRT Report Designer is a visual tool provided by Eclipse, as a Rich Client Platform (RPT) application. The RCP is available as a set of plug-ins installed on an existing Eclipse server or as an all in one package including Eclipse. This tool makes it easier for developers to design reports. It has to be downloaded and installed separately. It is not part of the Tivoli Service Request Manager V7.1.1 installation.

You can find the tool at the following Web site:

http://download.eclipse.org/birt/downloads/

Design Engine

The Design Engine is responsible for creating and modifying report designs. The created report design is stored in the .rptdesign and .rptlibrary files. The Design Engine API (DEAPI) performs a wide range of low-level tasks:

- Reads and writes design files.
- Maintains the command history for undo/redo.
- Provides a rich semantic representation of the report design.
- Provides meta-data about the Report Object Model.
- Performs property value validation.
- Notifies the application when the model changes

BIRT Report Design files are XML files, noted by the extension .rptdesign. BIRT Reports can contain single or multiple files. The files are categorized as either library files or resource files.

BIRT library files are also XML files and have the extension.rptlibrary. BIRT library files can contain code that is used multiple times for items such as font type, size, page numbers, and time stamp.

Resource Files contain items such as images or external files. Resource files can be used by either report design files or library files.

The XML of the BIRT Report details which library files and resource files the report requires. In the XML file, a flag indicates whether the file is a library file.

5.1.6 Tivoli Service Request Manager V7.1 Report Administration

The Report Administration is integrated into the Tivoli Service Request Manager V7.1 interface. As the Report Administrator, you can specify:

- User reports availability and how they open, run, and print.
- Define parameters and Set Record Limits as Performance Enabler.
- Report security settings.

As an integrated application, the report application "look and feel" is like all other applications launched from Start Center. There are two ways to start Reporting Administration. The first is for initial administration, and the second can be used when reports are already defined:

Start the Report Administration application from the Start Center by selecting **Go To** → **Administration** → **Reporting** → **Report Administration**, as shown in Figure 5-2.

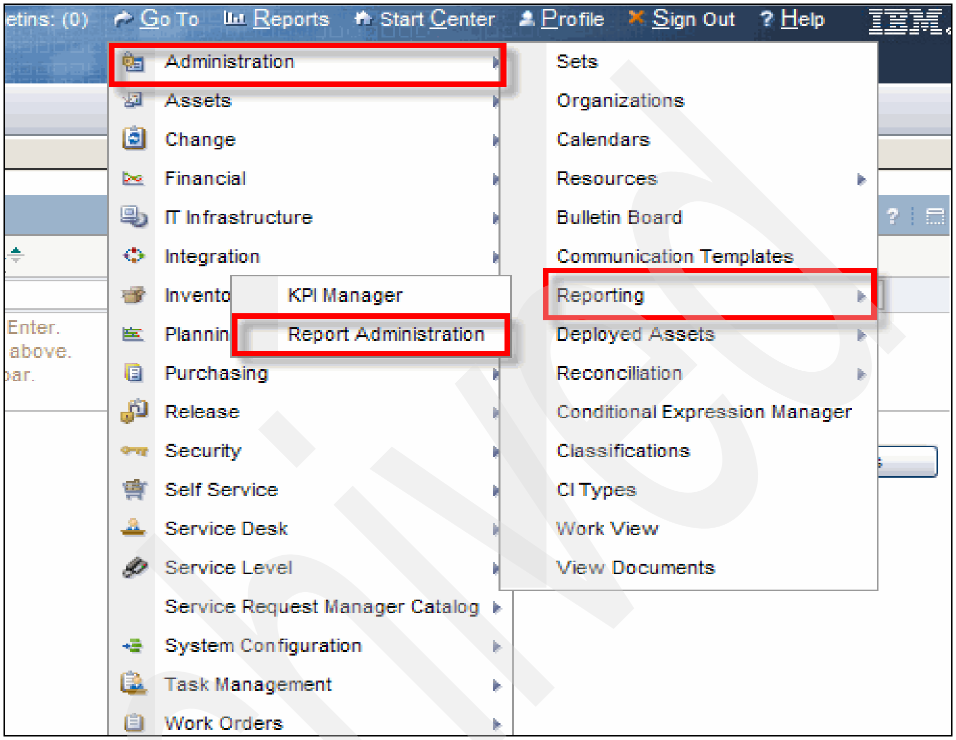


Figure 5-2 Report menu

Start the Report Administration application from the Start Center by selecting **Reports** → **Administration** → **Reporting** → **Report Administration**, as shown in Figure 5-3 on page 293.

From the Report Administration window, you can see the following tabs:

- List: List of all existing reports (Figure 5-4 on page 294)
- Reports: Details of a selected report (Figure 5-5 on page 295)
- Security: Set and view Report and Application Security (Figure 5-6 on page 295)

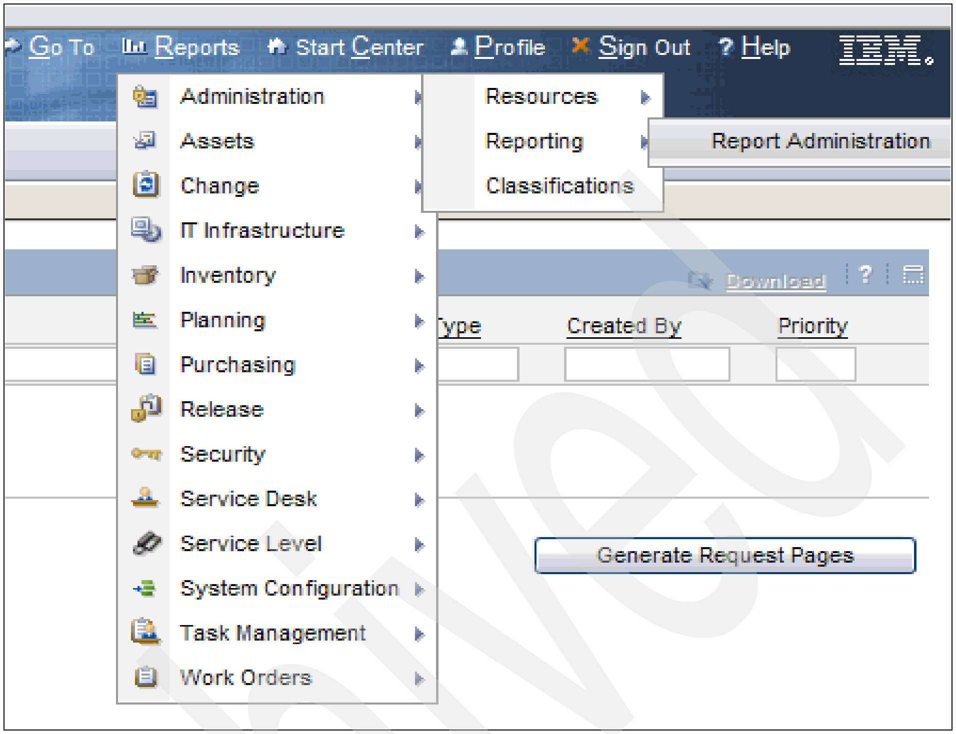


Figure 5-3 Entering the Report Administration

After selecting **Go To** → **Administration** → **Reporting** → **Report Administration**, you will see the Report Administration window, which has three tabs: List, Report, and Security.

In order to list only BIRT Service Request reports, enter “tsd” into the Report File Name field to filter this view for the desired reports, as shown in Figure 5-4.

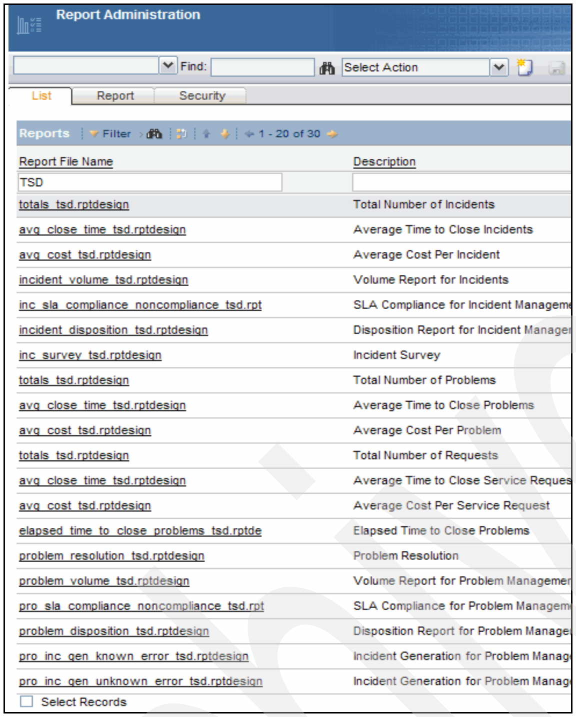


Figure 5-4 Report list for Service Request Manager

The middle tab shows the Report options described in Figure 5-5 on page 295, while Figure 5-6 on page 295 shows the Report Security.

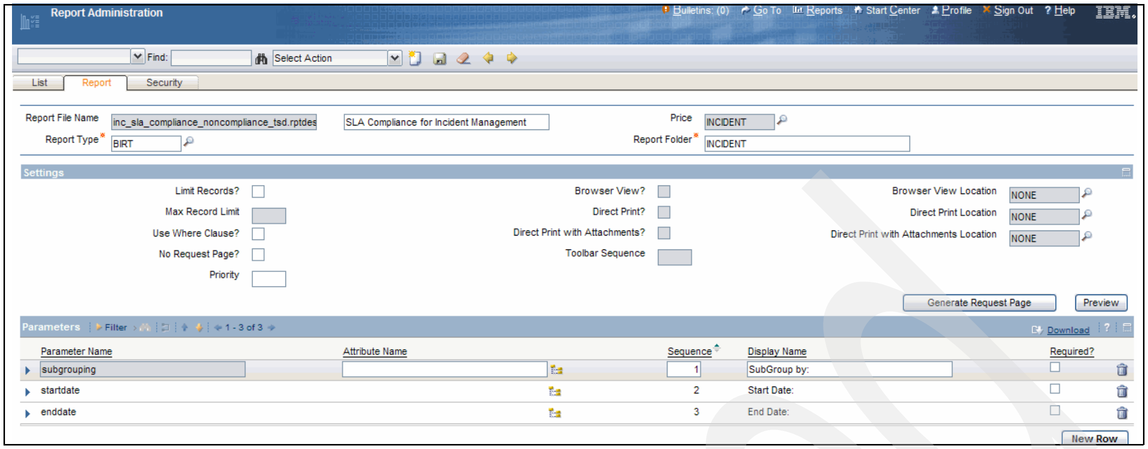


Figure 5-5 Report details

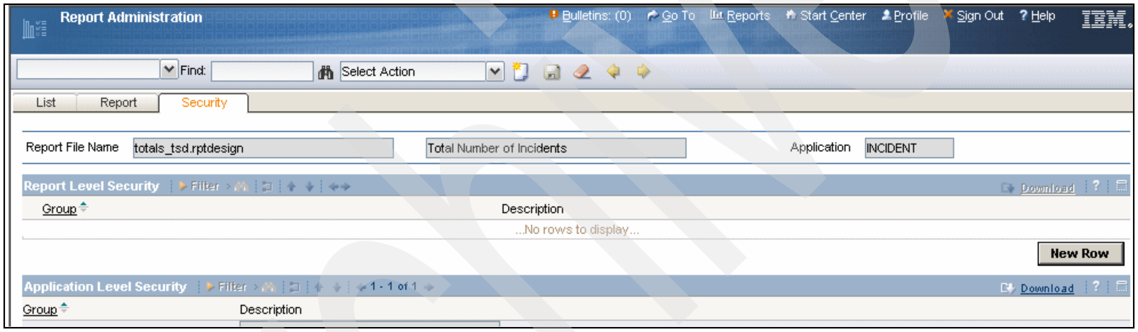


Figure 5-6 Report Security

The Select Action only offers general administration tasks for the List tab, such as:

- View Scheduled Reports
- Set Application Security
- View Group Security
- View Library Files
- Run Reports

Once a specific report is selected, additional Select Action items are available:

- Import Report
- Import Library File
- View Report Dependencies
- Add to Bookmark
- Duplicate Report
- Delete Report

The following sections cover the more complex Select Actions in more detail.

View Scheduled Reports

The View Scheduled Reports dialog box lets you manage scheduled report jobs, as shown in Figure 5-7. You can view the report load and delete scheduled report jobs as necessary.

Note: This functionality applies to BIRT reports only.

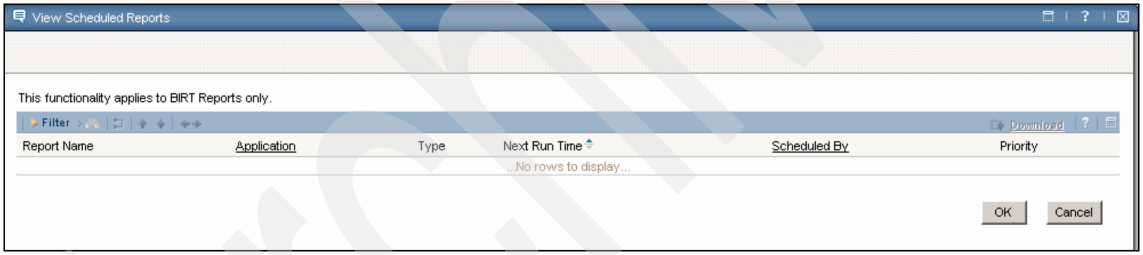


Figure 5-7 View Scheduled Reports

Set application security

Application security settings let you set group security for all reports in a selected application (Figure 5-8). The MAXADMIN group has access to all “out of the box” reports. You must set up group or report access to each individual application for new or customized reports.

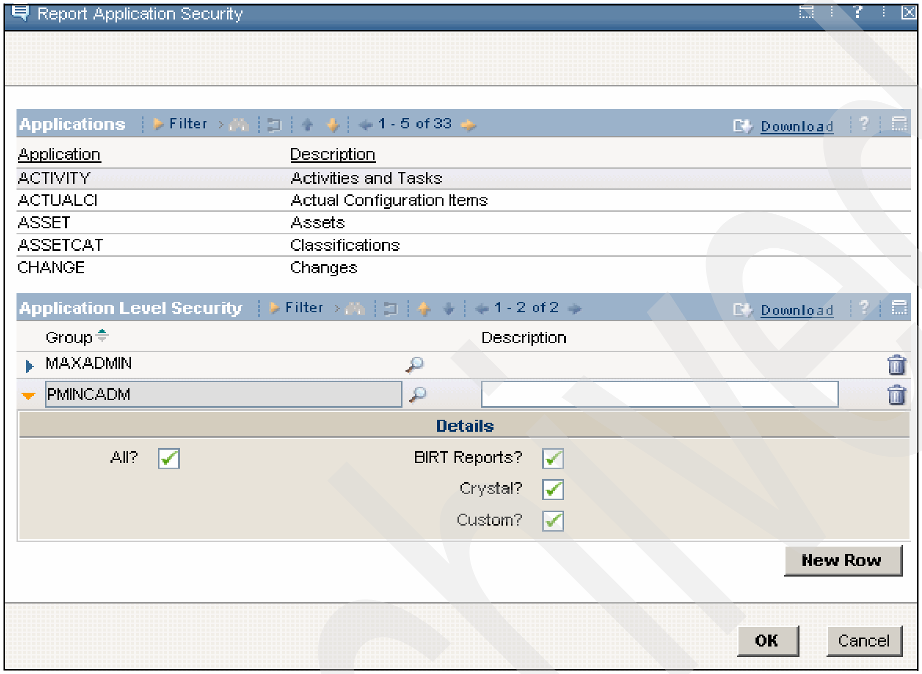


Figure 5-8 Set application security

The two different ways of setting security for all reports for an application and for individual reports are shown in Figure 5-9.

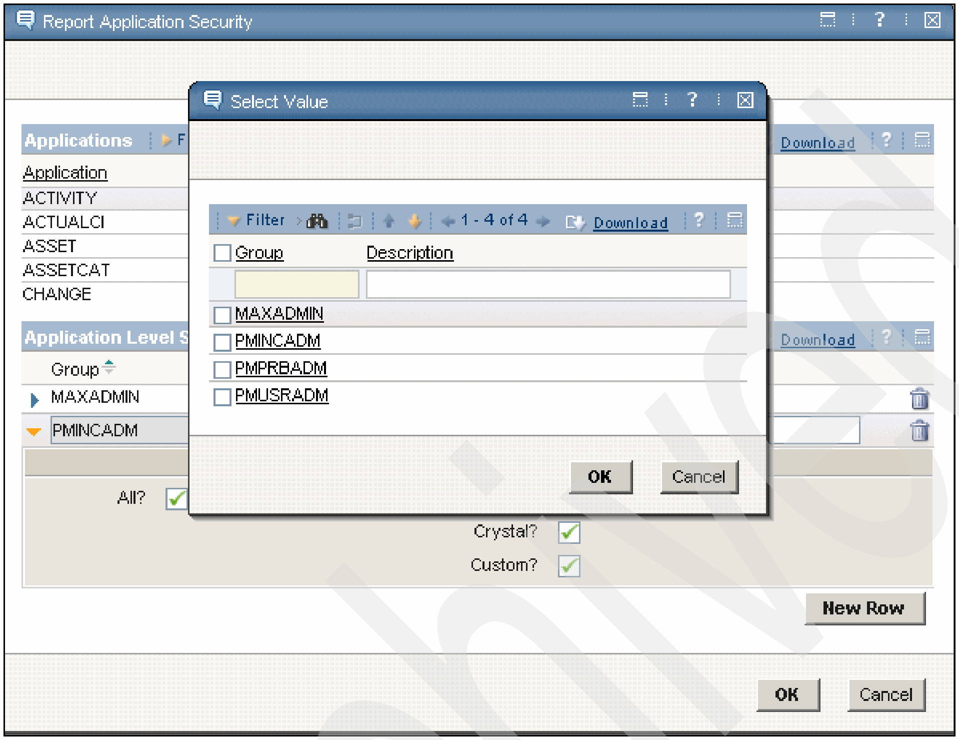


Figure 5-9 Report application security

The second way to set security is through the Security tab. In the Report Level Security window, click a new row and then click the magnifying glass (lookup) to select the security group, as shown in Figure 5-10.

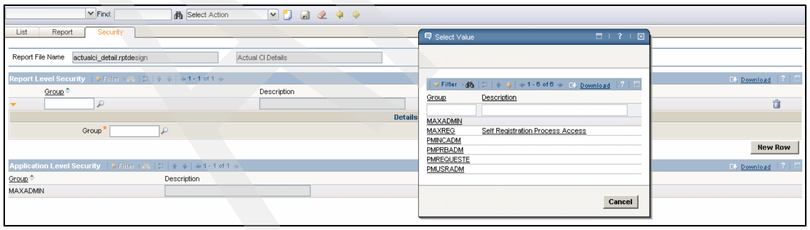


Figure 5-10 Individual security settings

View Group Security

You can manage report security for a group through the Report Administration application. The MAXADMIN group has access to registered “out of the box” reports. To access Group security select **Select Action** → **View Group Security**. Click the lookup symbol, and select the desired group. You must set up application access for other groups individually, as shown in Figure 5-11.

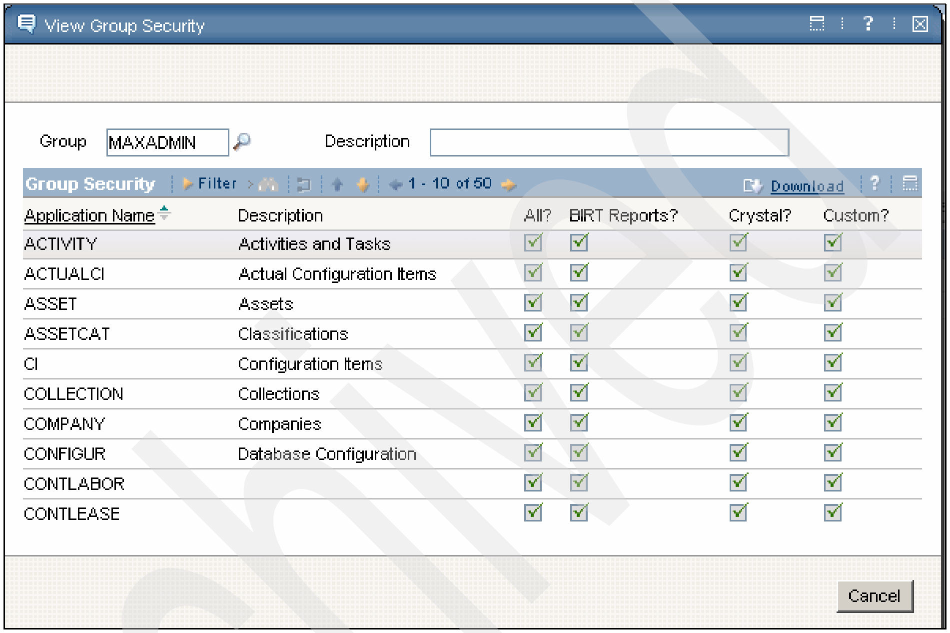


Figure 5-11 View Group Security

Import report

You perform this action to add a new report to your database or bring an updated version of an existing report into your database (Figure 5-12 on page 300).

Before you import the report design file, you must import any associated library files. This action is available from only the Report tab for the following reasons:

- If the report is new, you use the Report tab to add the report to the Report Administration application and then import the report to the database.
- If the report already exists, you import the report from the Report tab to be certain you choose a correct combination of Report Design file and Application name.

To add multiple design files, use the **`importreport.cmd`** command.

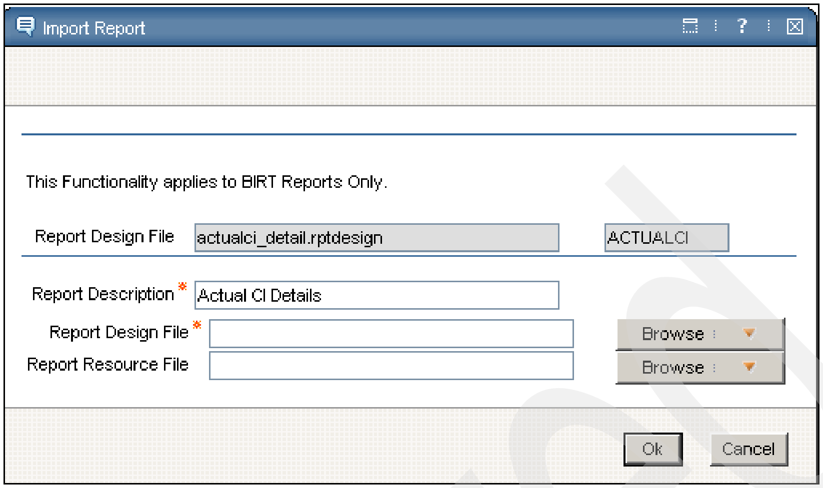


Figure 5-12 Import Report

Import Library File

Library files contain components that you can use in one or more report designs to provide consistent behavior and performance. Library files are useful when many reports use the same component multiple times. Use the Report Administration application to import a report library file into the database, as shown in Figure 5-13. You import a library file before you import the corresponding reports. In the Report Resources File field, enter the location of any resource files. Resource files contain items such as images or external files. This field is optional.

Figure 5-13 Import resources

View Report Dependencies

Use the View Report Dependencies action to view the libraries that a report design file requires. For each report library, you also can view dependent library files and check for any resource files. You can view report dependencies for BIRT reports, as shown in Figure 5-14.

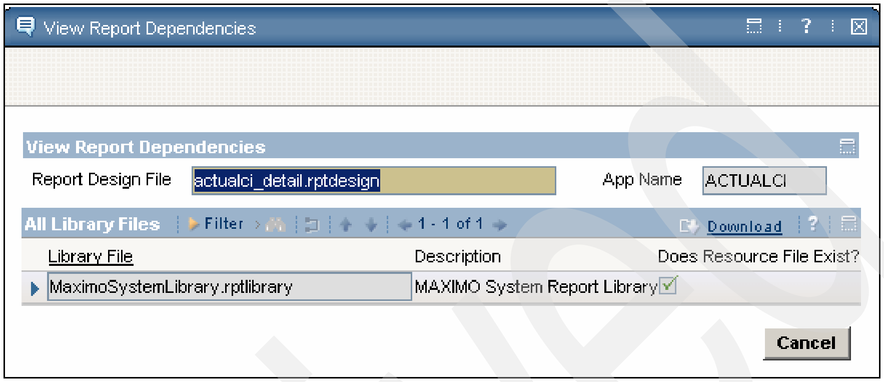


Figure 5-14 View report dependencies

Duplicate report

Among the reasons for duplicating a report are:

- You create a cloned application. You duplicate the report and save the duplicate to the cloned application.
- You want to register a report to multiple related applications. You duplicate the report and save it to the related applications.

Delete report

When you delete a report, you remove the report and its associated files from the database. You also remove any scheduled activities for the report.

5.1.7 Configure reports

There are several options to configure a report in the Report tab. Those with an (*) asterisk are required parameters (Figure 5-15).

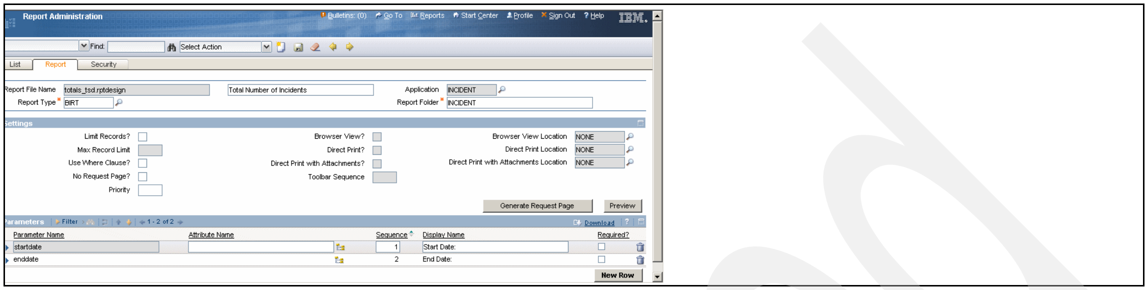


Figure 5-15 Report configuration

- Report Type

  BIRT, Crystal, or Custom. By determining the report type and settings, you register that report in the Tivoli Service Request Manager database.

- Limit Records

  This action limits the number of records against which a user can run a report. It prevents users from executing large queries, which can cause negative performance impacts. Use the Report Administration application to set record restrictions on reports. This feature applies to only reports without parameters. This parameter works in conjunction with Max Record Limit.

- Use Where Clause?

  Enables Current/Selected plus User Inputted parameters.

- No RequestPage

  Disables Request Page. Can be used for Reports Only available through hyperlinks, for reports updating the database, or accessible only through Self Service Applications.

- Priority

  A numeric field used in the report queuing process.

- Browser View and Browser View Location

  The Browser View feature lets you create a shortcut. With the shortcut, the user can click an icon once in the application toolbar to open a report directly in the browser. When Browser View is checked, you then enter a value other than None in the Browser View Location field. This field determines the application tabs that have an active Browser View icon.

The following options are available:

– All: The Browser View icon is available on all tabs for the selected application.

– List: The Browser View icon is only available on the List tab for the selected application.

– Main: The Browser View icon is available on all tabs, except the list tab.

– None: The Browser View icon does not appear in the selected application. None is the default.

Click **Save Report** to apply the changes.

- Direct Print and Direct Print Location

  The Direct Print feature lets you create a shortcut so a user can click an icon once in the application toolbar to print the report. The configuration is the same as for Browser View Location. The report will print to the user's default printer. The Direct Print Locations contain the same values as Browser View.

- Direct Print with Attachments and Direct Print with Attachments Location

  The Direct Print with Attached Documents feature lets you create a shortcut so that a user clicks an application icon once (and selects **Yes** in the Message dialog box) to print the report and any associated attached documents. The configuration is the same as for Browser View Location.

- Generate Request Page

  If you have added a report, or made any modifications to the settings or parameters of an existing report, you must generate the request page so that the report can be viewed. This option is available for all reports or at an individual report level.

- Preview Report:

  Enables the administrator to confirm the following items for a report:

  – The Request Page opens.

  – The correct parameters, if any, display on the request page.

  – The required fields are noted with an orange asterisk (*).

  – After entering the parameter values, the report executes with the correct data and format.

Figure 5-16 gives an overview of the report parameters window.

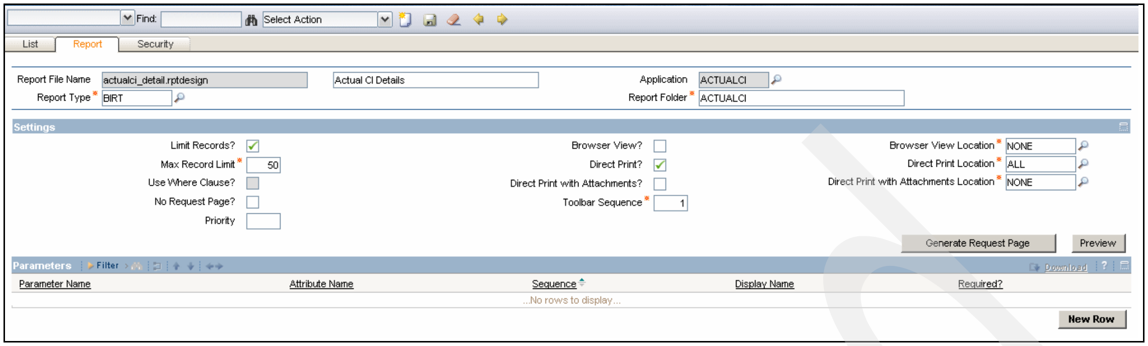


Figure 5-16 Report parameters

5.1.8 Run reports

Follow these instructions to run a report. After you run a report, you have the options to print, export data, and toggle the table of contents.

1. Open the Reports dialog box through one of the following methods:

   – From the Reports Menu in the application toolbar, select an application, for example, **Service Desk** → **Incidents**, as shown in Figure 5-17.

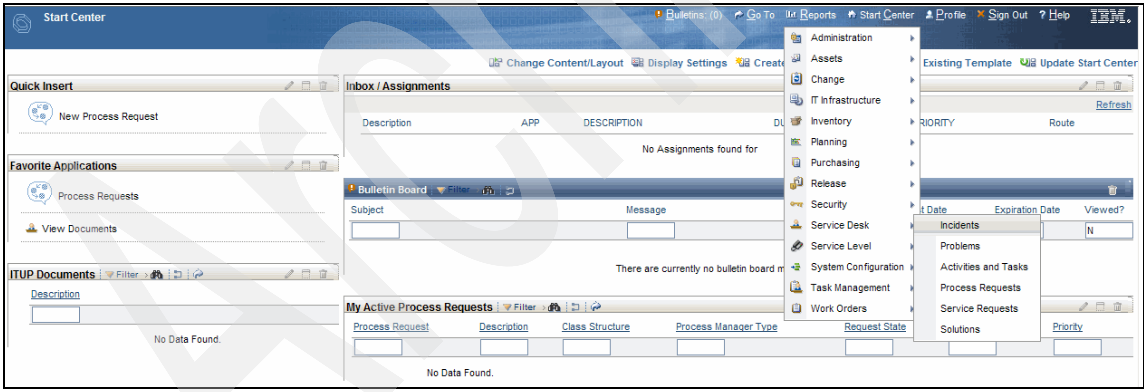


Figure 5-17 Report menu

– From the Select Action menu, select **Run Reports**. The On Demand Reports tab opens. The Reports to Run table window lists the available reports for the application that the users has security rights to. Click the report that you want to run, as shown in Figure 5-18.

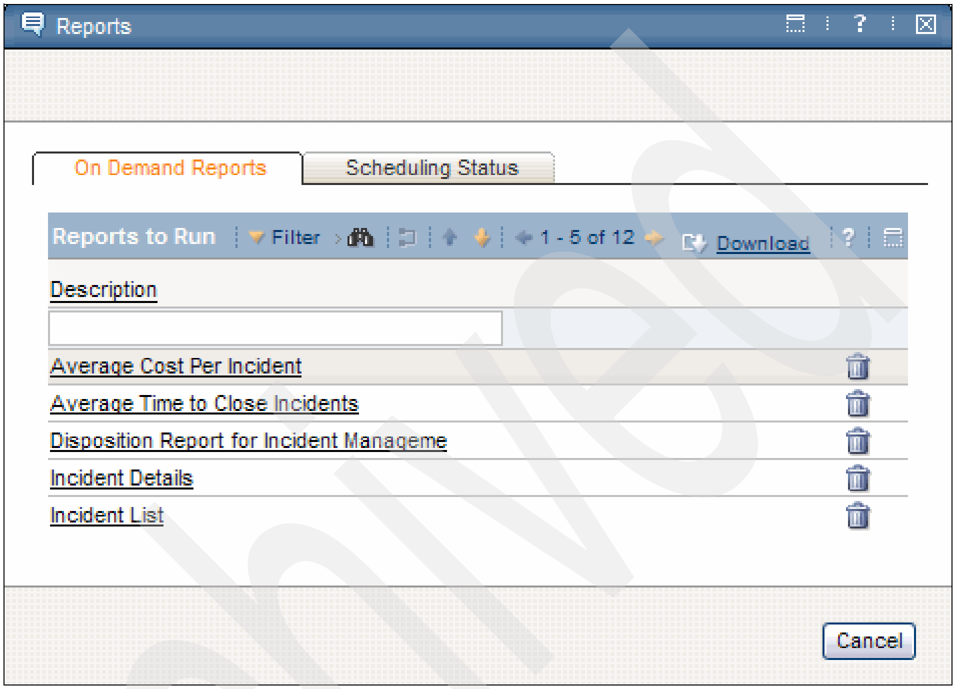


Figure 5-18 Selecting a report to run

2. Select the report you want to execute, for example, Total Number of Incidents. Enter the required parameters in the Request Page dialog box, as shown in Figure 5-19.

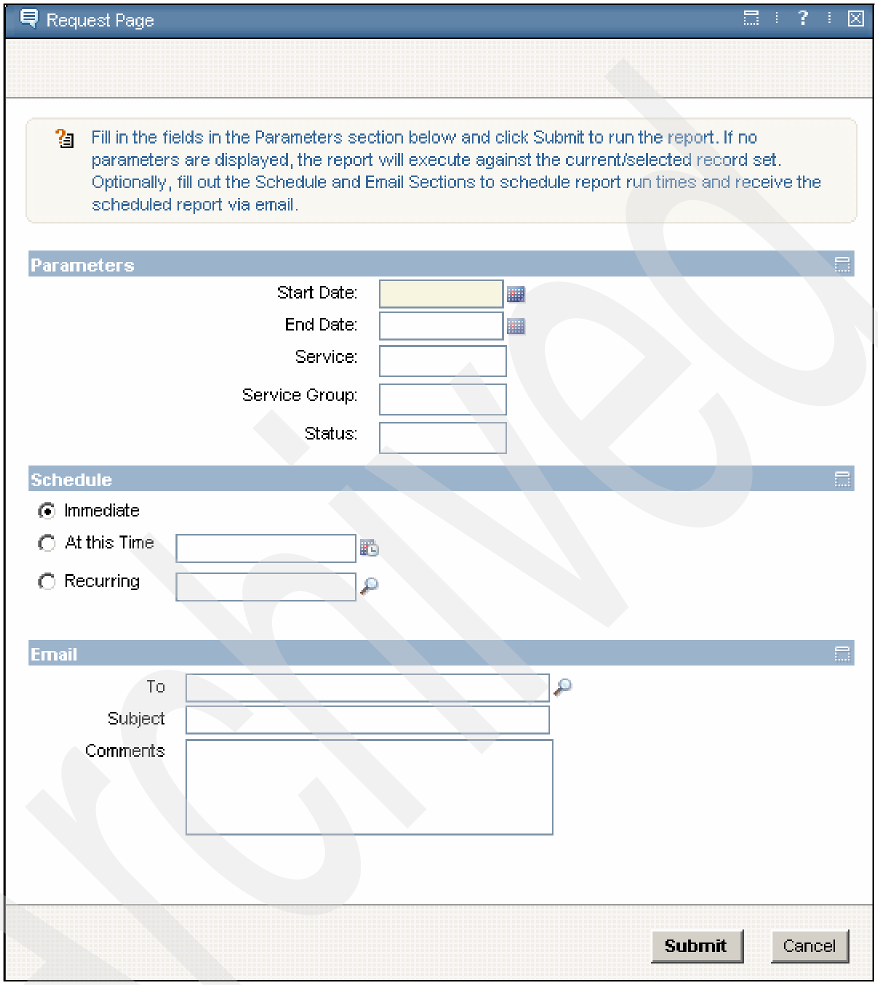
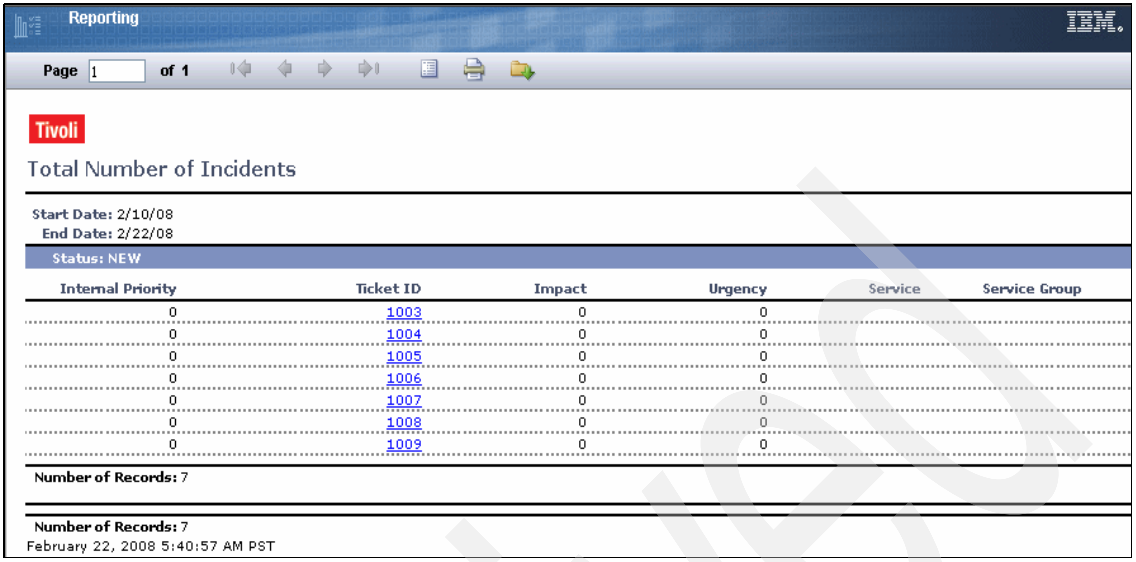


Figure 5-19 Run request

3. Click **Submit** to run the report. The report opens in your browser, as shown in Figure 5-20 on page 307.


Reporting
Page 1 of 1

Tivoli

Total Number of Incidents

Start Date: 2/10/08
End Date: 2/22/08

Status: NEW

Internal Priority	Ticket ID	Impact	Urgency	Service	Service Group
0	1003	0	0		
0	1004	0	0		
0	1005	0	0		
0	1006	0	0		
0	1007	0	0		
0	1008	0	0		
0	1009	0	0		

Number of Records: 7

Number of Records: 7
February 22, 2008 5:40:57 AM PST

Figure 5-20 Total Number of Incidents Report

4. On the Reporting toolbar, as shown in Figure 5-21, perform any of the following actions:
   - Click the **Print Report as PDF** icon to print the report.
   - Click the **Export Data** icon to export the data in .CSV format.
   - Click the **Toggle table of contents** icon to see the table of contents for your report. The report you select determines the table of contents.

To improve performance and to reduce load on the database server during working hours, it is possible to schedule report runs. Scheduled reports are then e-mailed when completed. The e-mail can be sent to a single user or a group including subject and comments. The report is e-mailed in PDF format.

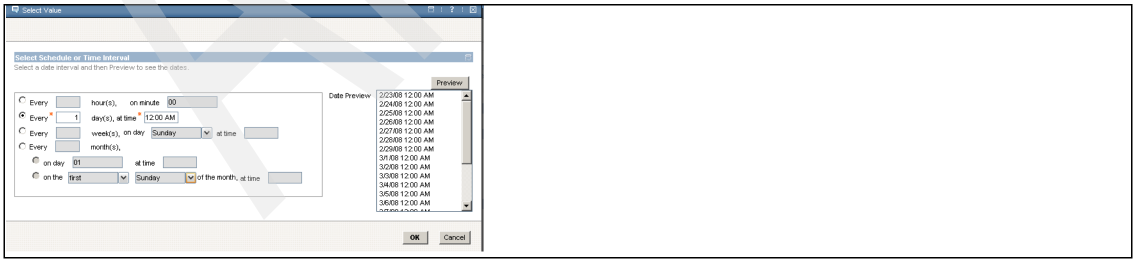


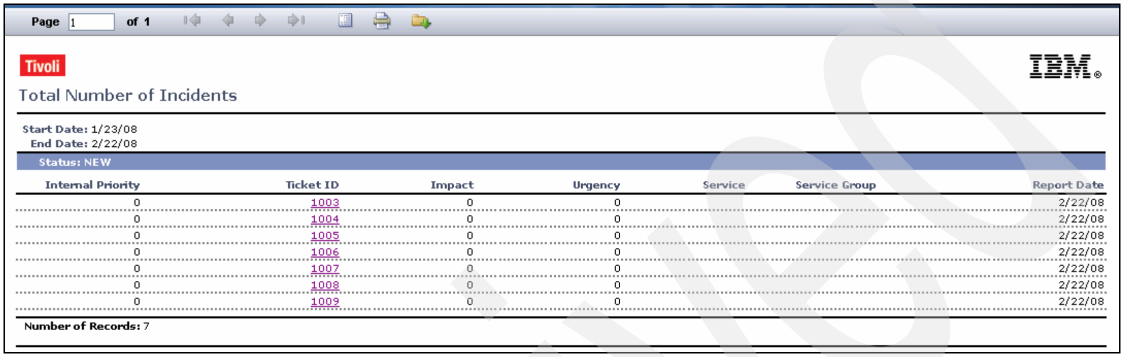
Figure 5-21 Report Scheduler

5.1.9 Built-in reports for Service Request Manager

Tivoli Service Request Manager comes with approximately 30 out of the box reports. This section shows some examples.

Total Number of Incidents

Total Number Of Incidents is shown in Figure 5-22.



Page 1 of 1

Tivoli

IBM.

Total Number of Incidents

Start Date: 1/23/08
End Date: 2/22/08

Status: NEW

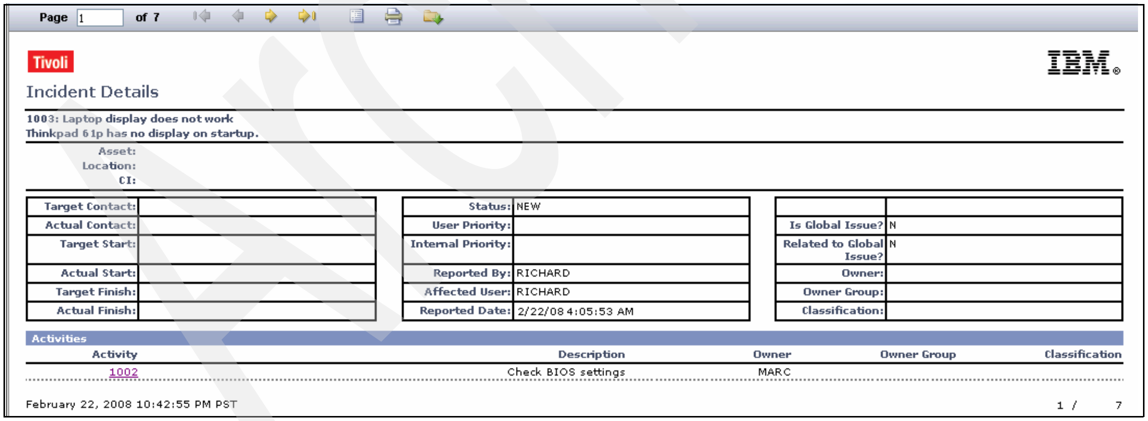
Internal Priority	Ticket ID	Impact	Urgency	Service	Service Group	Report Date
0	1003	0	0			2/22/08
0	1004	0	0			2/22/08
0	1005	0	0			2/22/08
0	1006	0	0			2/22/08
0	1007	0	0			2/22/08
0	1008	0	0			2/22/08
0	1009	0	0			2/22/08

Number of Records: 7

Figure 5-22 Total Number Of Incidents

Incident Details

Incident Details is a drill down from the Total Number Of Incidents, where you are able to understand what was reported on the incident, as shown inFigure 5-23.



Page 1 of 7

Tivoli

IBM.

Incident Details

1003: Laptop display does not work
Thinkpad 61p has no display on startup.

Asset:
Location:
CI:

Target Contact:		Status:	NEW		
Actual Contact:		User Priority:		Is Global Issue?	N
Target Start:		Internal Priority:		Related to Global Issue?	N
Actual Start:		Reported By:	RICHARD	Owner:	
Target Finish:		Affected User:	RICHARD	Owner Group:	
Actual Finish:		Reported Date:	2/22/08 4:05:53 AM	Classification:	

Activities

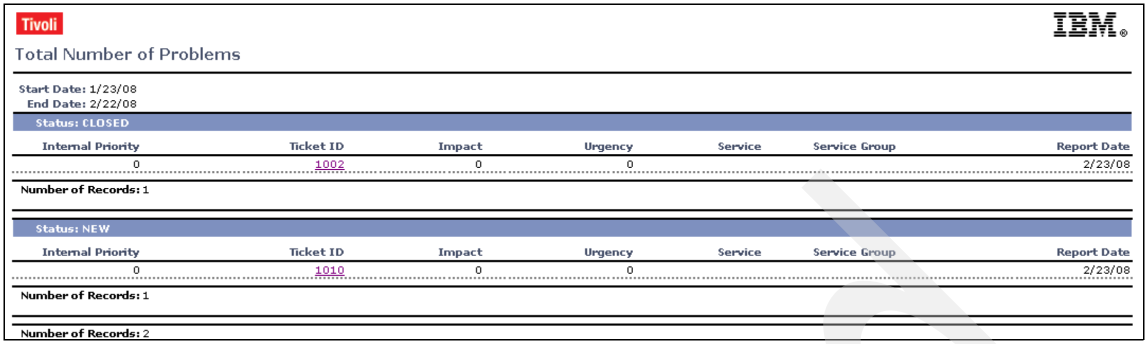
Activity	Description	Owner	Owner Group	Classification
1002	Check BIOS settings	MARC		

February 22, 2008 10:42:55 PM PST   1 / 7

Figure 5-23 Incident Details

Total Number of Problems

This report summarizes the total number of reported problems, as shown in Figure 5-24 on page 309.



Tivoli

IBM.

Total Number of Problems

Start Date: 1/23/08
End Date: 2/22/08

Status: CLOSED

Internal Priority	Ticket ID	Impact	Urgency	Service	Service Group	Report Date
0	1002	0	0			2/23/08

Number of Records: 1

Status: NEW

Internal Priority	Ticket ID	Impact	Urgency	Service	Service Group	Report Date
0	1010	0	0			2/23/08

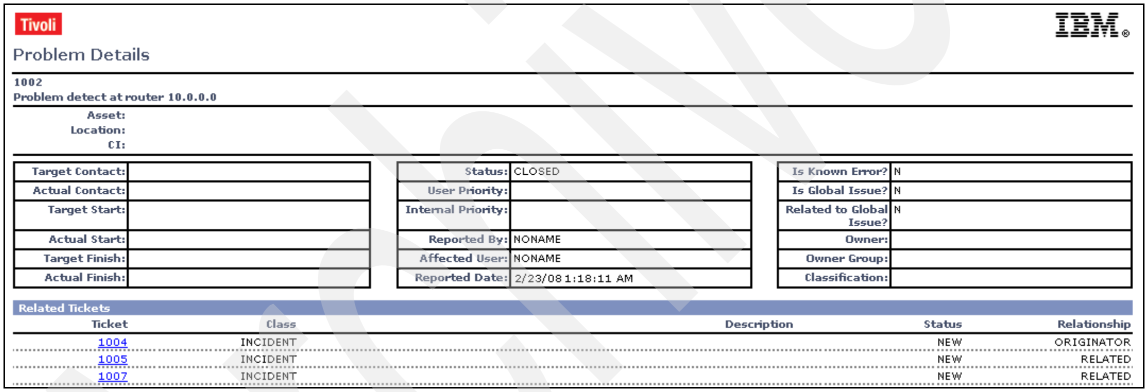
Number of Records: 1

Number of Records: 2

Figure 5-24 Total Number Of Problems

Problem Details

This report shows detailed problem information, including the related tickets for this problem, as shown in Figure 5-25.



Tivoli

IBM.

Problem Details

1002
Problem detect at router 10.0.0.0

Asset:
Location:
CI:

Target Contact:		Status:	CLOSED	Is Known Error?	N
Actual Contact:		User Priority:		Is Global Issue?	N
Target Start:		Internal Priority:		Related to Global Issue?	N
Actual Start:		Reported By:	NONAME	Owner:	
Target Finish:		Affected User:	NONAME	Owner Group:	
Actual Finish:		Reported Date:	2/23/08 1:18:11 AM	Classification:	

Related Tickets

Ticket	Class	Description	Status	Relationship
1004	INCIDENT		NEW	ORIGINATOR
1005	INCIDENT		NEW	RELATED
1007	INCIDENT		NEW	RELATED

Figure 5-25 Problem Details

5.1.10 Report management

Tivoli Service Request Manager V7.1 also provides the ability to monitor the usage and execution of BIRT reports. It shows what reports are used, how much time the execution took, whether the report was scheduled, and if it ran successfully.

Start the Report Administration application from the Start Center by selecting **Reports** → **Administration** → **Reporting** → **Report Usage**.

The report should be similar to the one shown in Figure 5-26.

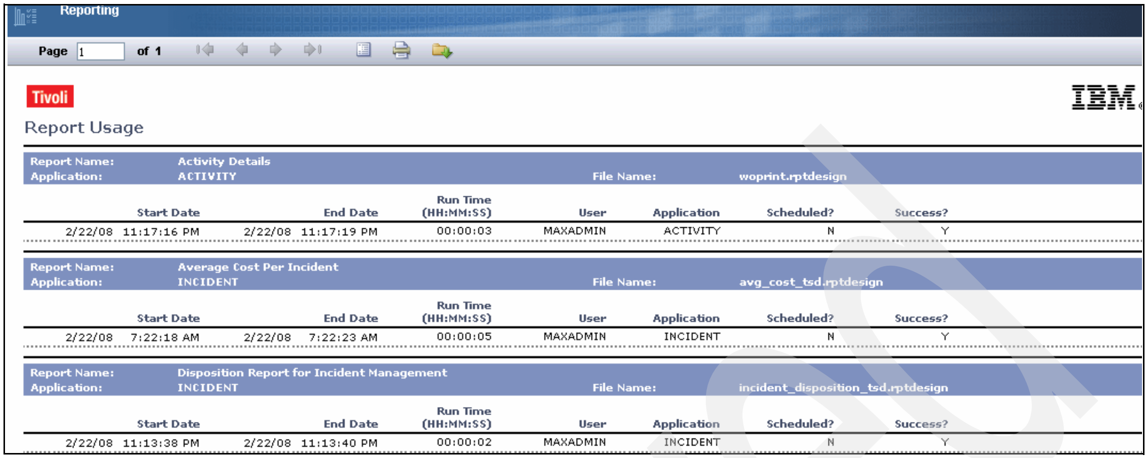


Figure 5-26 Report Usage

5.1.11 Report queue

The report queuing functionality limits the number of jobs that can be executed at any given time. Reports Jobs concurrently processed is configurable by the client. This helps distribute load on the server due to reports, and reduces the amount of wait time for report users who benefit from better performance.

5.1.12 Report Locale

In order to set up a report language other than English, you must enable reports for localization.

Two specific actions are required for Tivoli Service Request Manager V7.1 BIRT Reports: Both the language and the locale for the user must be set.

This can be set in the User's Default Profile Information, or in the Users Application. The request pages for each of the languages required by a client need to be generated in the report admin app for each of these different languages.

Client needs to generate a report for English, Spanish, and Portuguese

The report administrator's locale is set to English, as shown in Figure 5-27 on page 311.

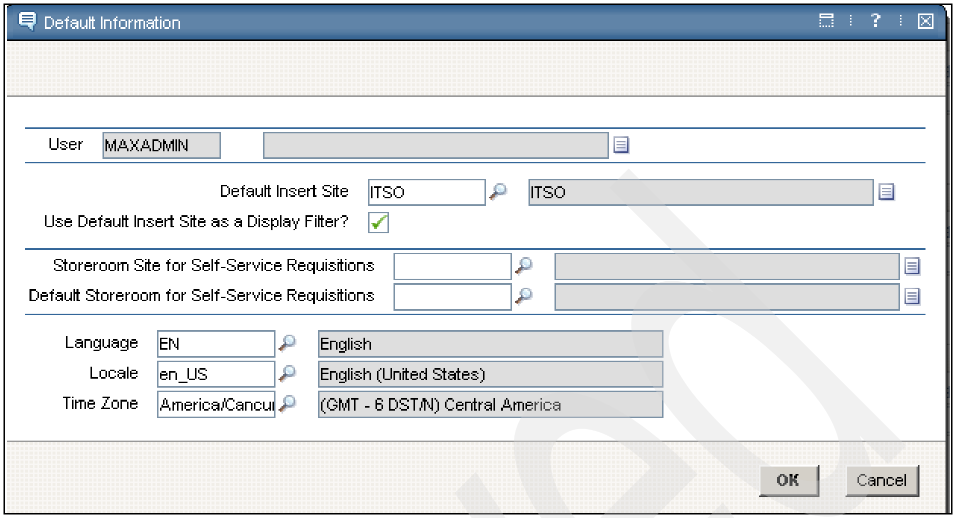


Figure 5-27 Default user information

1. First, access Report Admin, and generate the request pages' XML in English.
2. Sign out, sign back in, and change the locale to Spanish.
3. Access Report Admin and generate the report request pages in Spanish.
4. Sign out again, sign back in, change the locale to Portuguese, and generate the report request pages in Portuguese.

The request pages would then be available in all three languages: English, Spanish, and Portuguese.

5.2 Survey

Survey functionality addresses the Service Desk organizations necessity to gather ratings for the quality of service provided by the Service Desk and have a better understanding of how customers perceive the Service Desk's customer service. Customer Satisfaction Surveys are used to gather these qualitative evaluations and, in conjunction with a Service Desk's performance metrics, enhance management's ability to have an overall idea of Service Desk performance.

Survey application is built seamlessly using IBM Flexible Internet Evaluation Report Architecture (FIERA), which allows rapid creation of online surveys in a streamlined and efficient manner. A survey using FIERA enables a highly flexible presentation through a J2EE application, which makes use of a relational database back end to allow for the creation, display, and reporting of survey results for a user. The software provides the ability to generate online surveys and reporting of survey results for SR, Incident, and Problem processes in the Tivoli Service Request Manager V 7.1 Service Desk option.

5.2.1 Features

A Survey application provides feature rich Survey functionality for receiving innovative, independent, and flexible user feedback.

Distinctive Survey features that are now available in Tivoli Service Request Manager include:

- Survey provides the ability to prepare a survey questionnaire with multiple options, including check box options, radio buttons options, and free flowing comments space.
- There is a provision to archive Survey questions and their responses for reuse or research in the future.
- Automatic distribution of a Survey through e-mail to all users in the system over a cyclical time period using Survey along with Tivoli Service Request Manager Escalation triggered communication (for example, all users in the system are broadcast a Survey every six months).
- Automatic distribution of a Survey through e-mail to a specific user when a configured threshold has been reached across all Service Desk Tickets (for example, once the 50th Incident Ticket with a status of RESOLVED or CLOSED is generated in the system, an e-mail is sent to the user who created that Ticket).
- Agent access to a Survey from the Service Desk applications, allowing the Agent to conduct a telephone Survey with the user and record Survey responses.
- A Survey Administrator can make the survey results public or restrict them to be viewed only by the Survey Administrator.
- An Agent’s ability to manually send an e-mail to a user that contains a link to a Survey form that allows the e-mail recipient to answer Survey questions.

5.2.2 Survey usage scenarios

The following section discusses several Survey usage scenarios.

Survey administrator

Survey functionality in Tivoli Service Request Manager provides an administrator with the ability to easily compose, view, edit, and delete free form questions that become part of simple Customer Satisfaction Surveys. Survey questions can have both a short and long description. The administrator also has the ability to define and edit a response “scale” for each question (for example, admin designates that “1” = “poor” on a scale of 1-5).

Self service users and service desk agent

Both agents and self-service users can potentially have access to the Survey and can fill in Survey responses. In addition to the questions that include scale-based responses, Survey also includes a free-form comments box. Survey is easy to construct and easy to answer.

Self-service users can access the Survey at any time from the self-service application to provide comments/feedback on the Service Desk.

Self service user in workflow

In a workflow scenario, a Survey questionnaire can be published as a link in a communication template with predefined text. This communication template, when attached to a ticket workflow, sends out an e-mail to a user or a self-service user for gathering the feedback on the ticket. A self service user can provide the survey response by clicking the survey link from an e-mail.

5.2.3 Window flow

To access the Survey Manager application, clicking **Help** at the right top corner. If you have a pop-up blocker enabled for your browser, disable the pop-up blocker temporarily, as the Survey Manager application opens in a different window.

In the future, when the product will be generally available (GA version), the Survey Manager application will be available under the Administration Module, and in the Service Request, Incident Management, Problem Management, and Communication Template application Select Action menus.

In the Administration Module, Manage Survey can be accessed by selecting **Go to → System configuration → Platform configuration → Communication Templates → Select Action → Manage Survey**, as shown in Figure 5-28 and Figure 5-29.

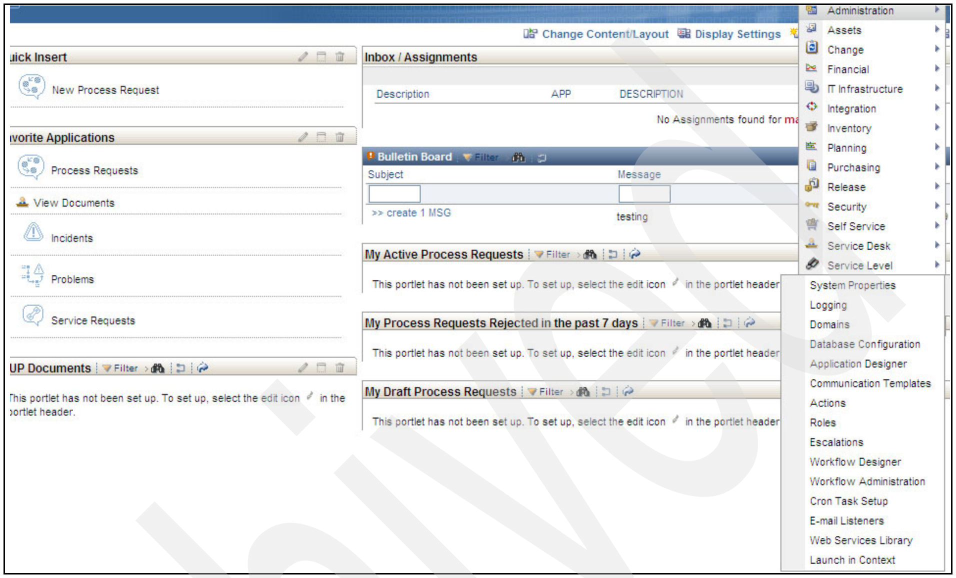


Figure 5-28 Communication Templates

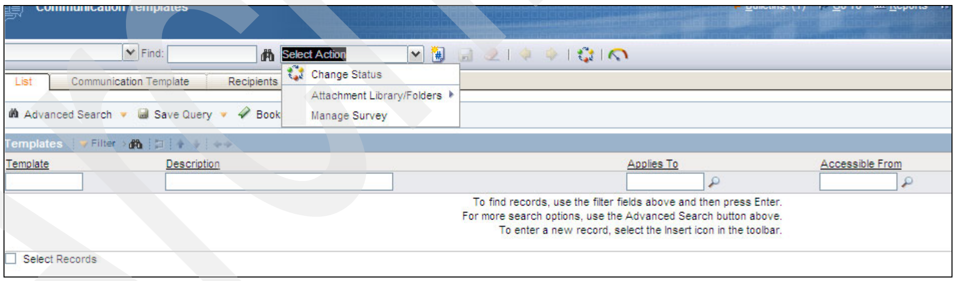


Figure 5-29 Communication Template Select Action menu

Manage Survey can also be accessed from within Incident Application, as shown in Figure 5-30 on page 315. Manage Survey can be accessed from Service Request and the Problem application Select Action menus.

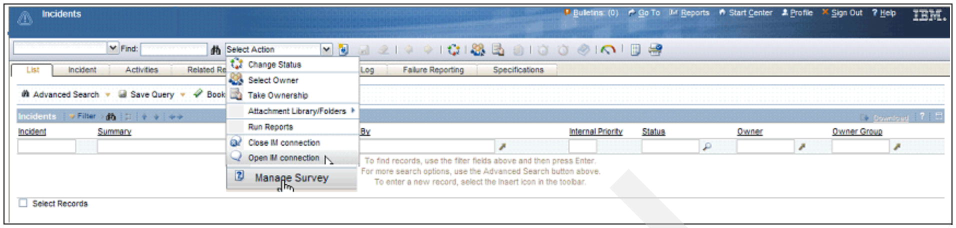


Figure 5-30 Incident Select Action menu

In the Accessing the Survey manager application, Survey supports the default tab List, which appears with three Modules (SR, Incident, and Problem), as shown in Figure 5-31. Currently, Survey permits publishing survey questions only for SR, Incident, and Problem tickets only.

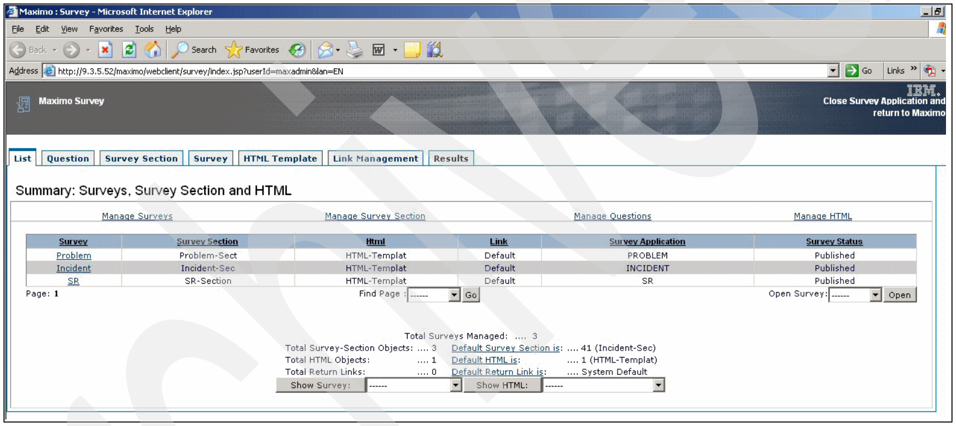


Figure 5-31 Survey List window

On the List tab, click the desired process for which you would like to enable Survey. The Result tab is displayed, as shown in Figure 5-32. This tab displays the current survey section that has been published. If a question is not published yet, pick a relevant Survey section for publishing.

The Results tab allows the Survey Administrator to publish and unpublish a survey.

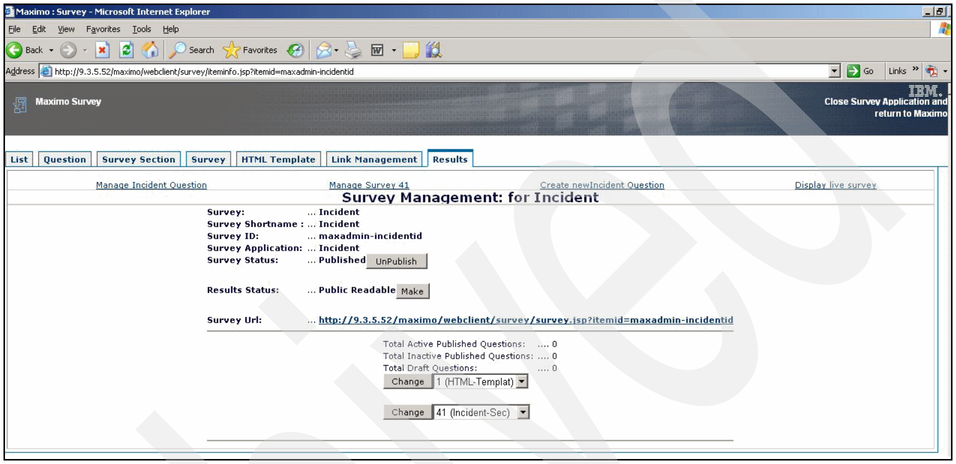


Figure 5-32 Survey results

The Questions tab allows the Survey Administrator role to prepare new questions for Survey. Click the Questions tab to add, modify, or archive the questions, as shown in Figure 5-33.

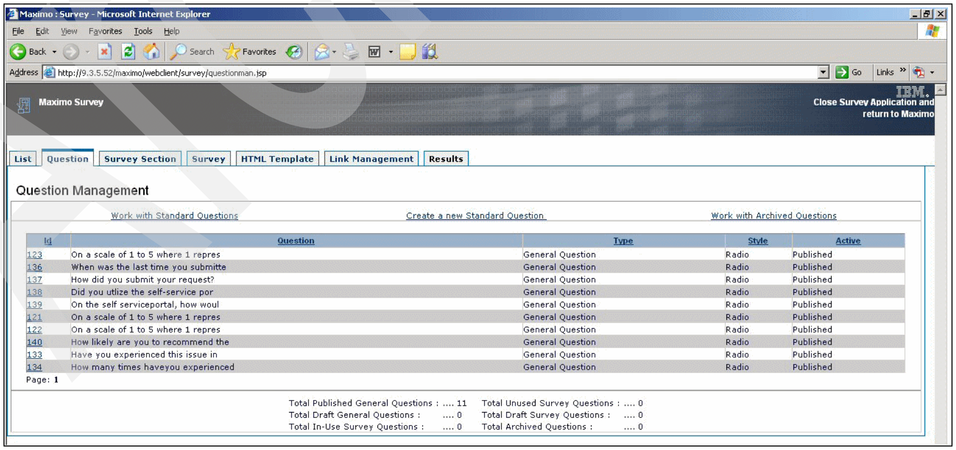


Figure 5-33 Survey questions

To create a new Question for Survey, click **Create new Standard Question**. A new window appears, as shown in Figure 5-34. Input the desired question text into the Question Text Box. Input the possible multiple answers for the question in the Answers section. A maximum of 15 answers can be provided for a single question. Configure the question style of the answers by selecting the style pattern from the **Question Style** drop-down menu, which includes standard styles like radio button, check box, single drop-down, multi drop-down menu, free form, statement, and header rule. The answers to the question appear to the user in the configured style.

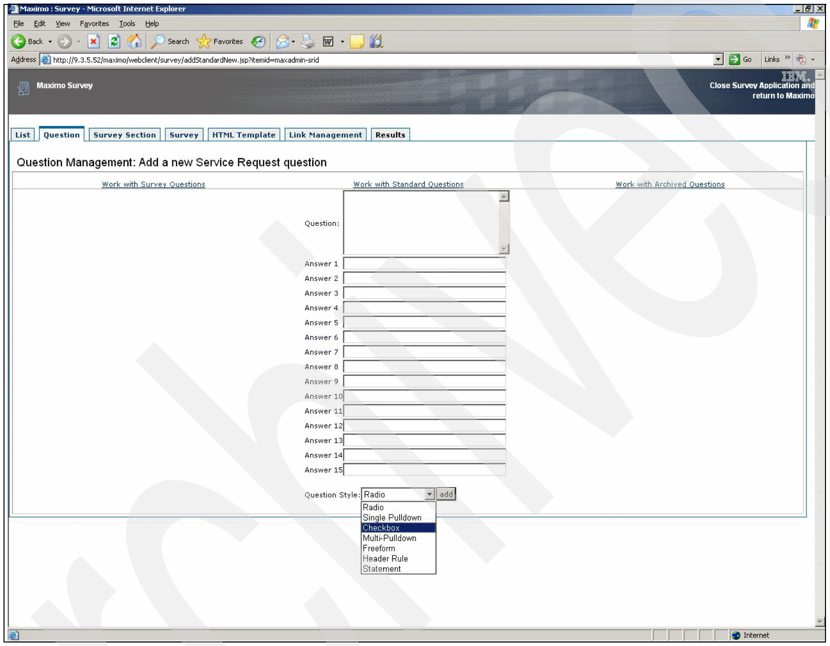


Figure 5-34 Survey create new questions

To modify the existing standard, click **Work with HTML** on the Questions tab, which appears as shown in Figure 5-35. This window provides the functionality to change the display style of the answer and to change the order of the answers. This window can also be used to make a clone of the standard question.

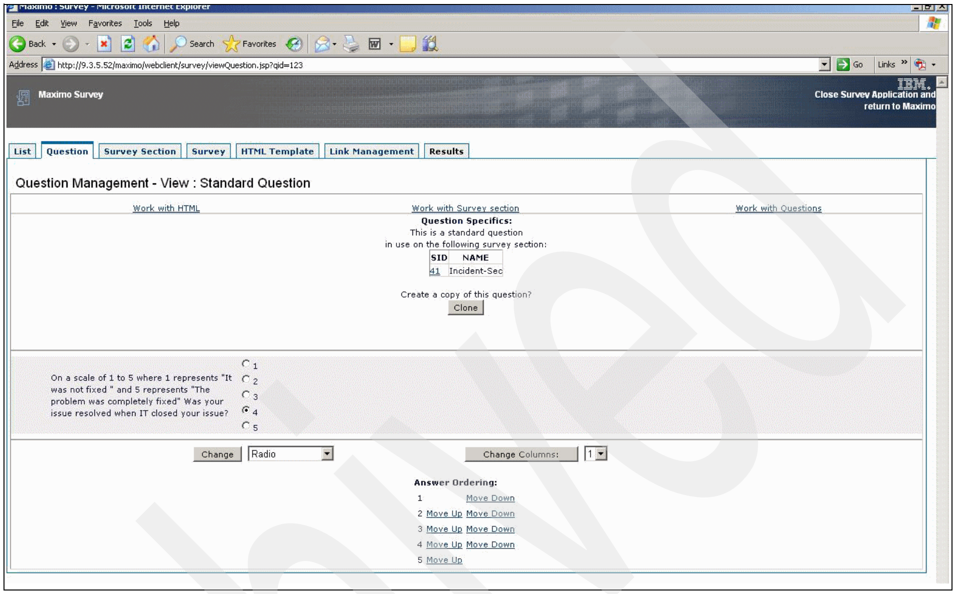


Figure 5-35 Survey work with HTML

The Survey Section tab is used for associating a survey section with supported process (like SR, Incident, and Problem), as shown in Figure 5-36. Click the **Edit** button in the Survey section to edit individual surveys that are associated with the process.

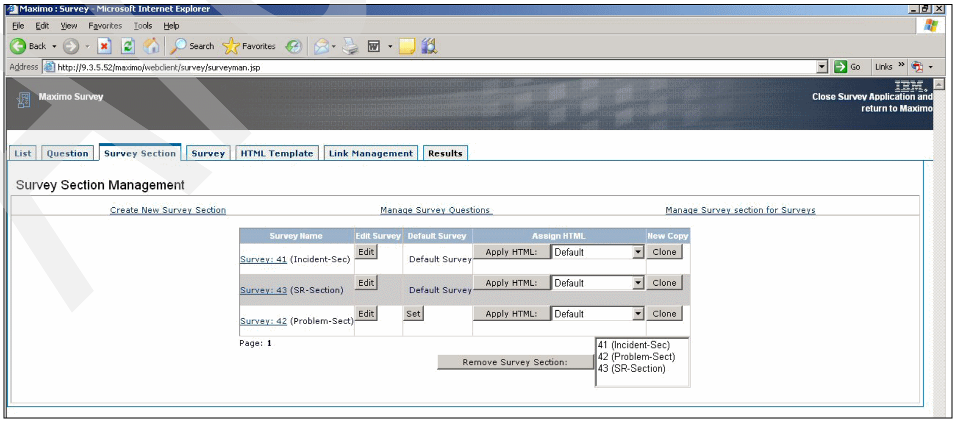


Figure 5-36 Survey section

The Survey tab is used for associating a survey section with the supported process (such as SR, Incident, and Problem), as shown in Figure 5-37. Click the **Edit** button in the Survey section to edit individual surveys that are associated with the process.

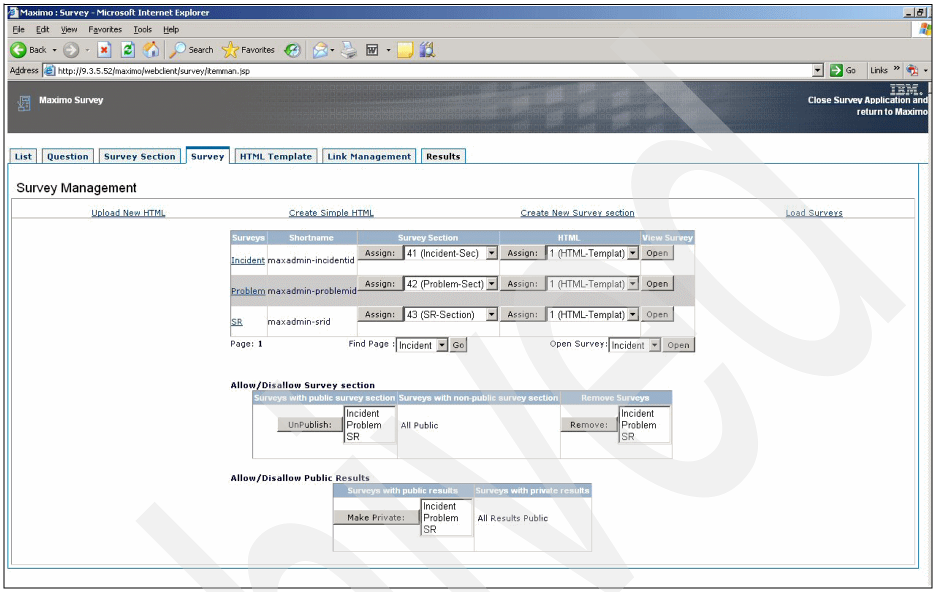


Figure 5-37 Survey tab

The HTML Template tab is used for creating or uploading a new HTML template for the survey. Once the HTML template is ready, the HTML tab can be used for associating a HTML template with the processes, as shown in Figure 5-38.

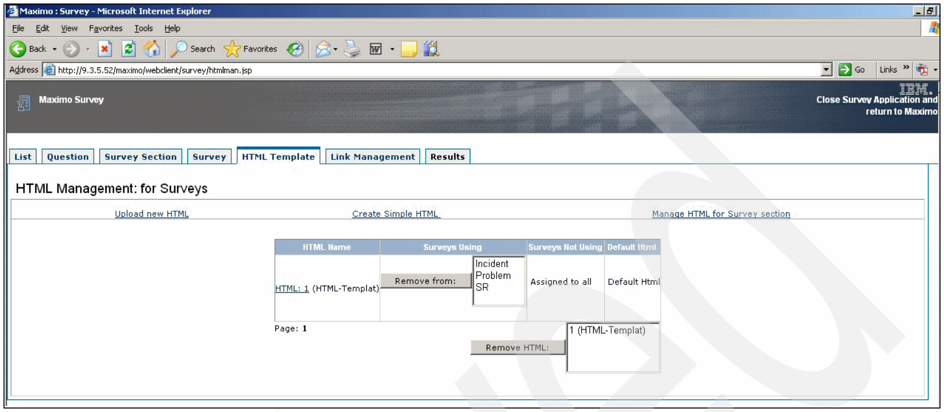


Figure 5-38 HTML Templates

The Link Management tab is used for capturing the return links for the survey, as shown in Figure 5-39.

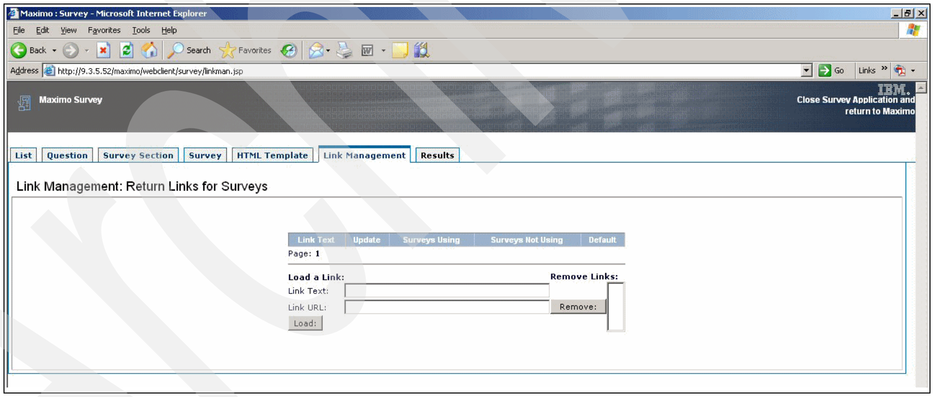


Figure 5-39 Link Management

Once the setup of Survey is done, the survey link from the Results tab is ready to be used in communication templates, as shown in Figure 5-41 on page 321, and in workflows when required.

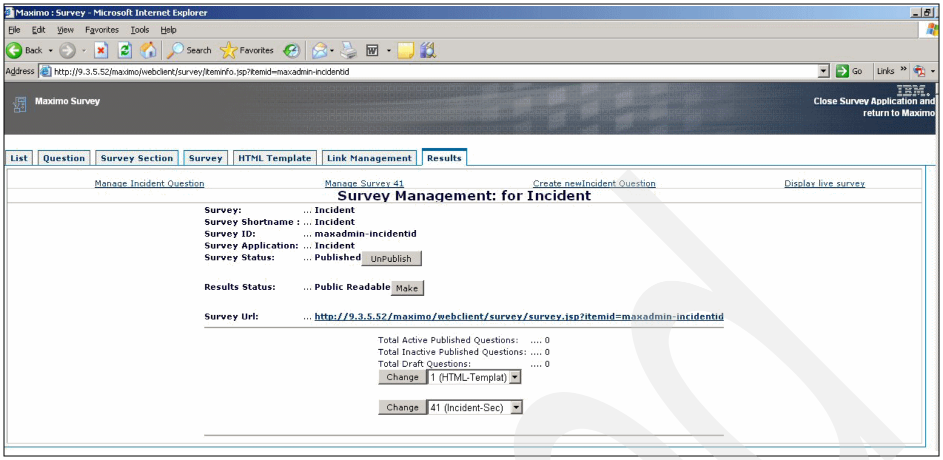


Figure 5-40 Results

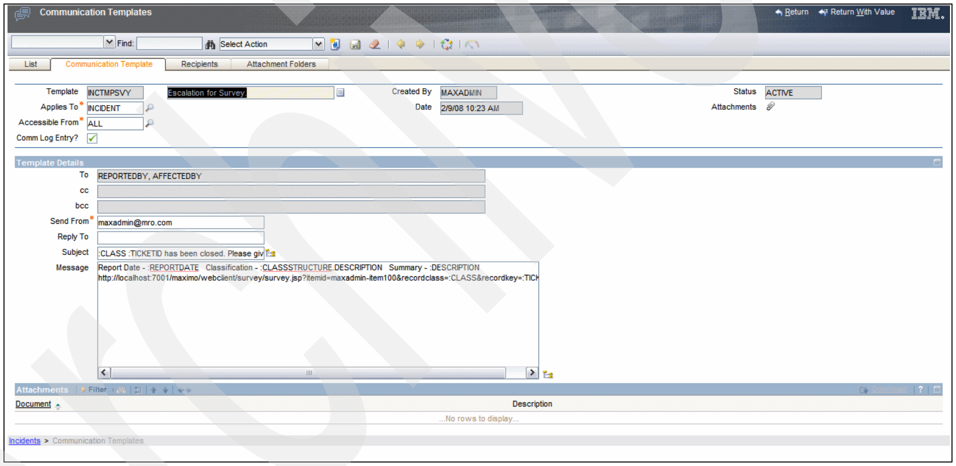


Figure 5-41 Survey link usage on Communication Template

Once a survey is sent to a customer through e-mail, a customer or user can access the Survey form from the link in the e-mail and provide survey inputs. Once the user submits the survey form, the response is collected.

In the scenario in which the help desk personnel calls the user randomly to obtain a survey, the help desk personnel can access the Survey link on behalf of the user and capture the details obtained during the call.

Once survey inputs are captured using the Survey forms, the cumulative performance report can be viewed from the Reports menu. For example, an Incident Survey report can be found by selecting **Go To** → **Reports** → **Service Desk** → **Incident Survey**. In a similar fashion, Problem reports can be found under its respective report section by selecting **Go To** → **Reports** → **Service Desk**

The reports that are available out of the box for analysis of Survey results are:

- The questions for SR/Incident/Problem report, with a percentage of the answers chosen
- The report on the number of surveys sent and surveys received from users

5.3 Search

Tivoli Service Request Manager search provides application wide indexing and searching capability. It uses the Lucene search engine, which has been recognized for its utility in the implementation of search engines and local, single-site searching. Lucene is an indexing and search library that can search throughout Service Desk and Service Catalog objects data, including attachments. The contents of each page are then analyzed to determine how it should be indexed (for example, words are extracted from the titles, headings, or special fields called meta tags). These index records include Tivoli Service Request Manager Application Module data and can search within the attachments (these are specific, compatible attachments, such as text or document files).

5.3.1 Features

The global search is based on a configurable, indexed search that differentiates it from the traditional resource consuming query based search. The indexed search can be accessed from three application windows, and the enhanced Tivoli Service Request Manager search capability permits three different searches scenarios:

1. Global Search – Service Desk

   The Global Search application is available to the Service Desk analyst for searching for Service Desk and Service Catalog objects in Tivoli Service Request Manager. Global Search allows a system administrator to perform a faster, wider search within the Tivoli Service Request Manager Service Desk and catalog. This search helps an analyst perform faster searches on Service Desk and Service Catalog objects from a single window to find past requests, incidents, problems, and solutions along with the attachments. On finding the result, the analyst can quickly refer to a solution from older tickets. On the Service Catalog side, the Global search can perform searches for Service Catalog objects like Offer, Catalog, Catalog Requests, and so on. The search also be used to intelligently identify dependencies between tickets and link tickets to a global ticket.

2. Service Request Manager Search

   The Service Request Manager Search application is available for the Self-Service Users to search for Service Desk and Service Catalog objects in Tivoli Service Request Manager. Service Request Manager allows the Self Service user to search only for records created by the Self Service user within Service Desk and Service Catalog and permits search for data that is exposed to the Self Service user in the Tivoli Service Request Manager, for example, Self Service Solutions.

   This search helps a Self Service User to perform faster searches on Self Service solutions, Service Desk, and Service Catalog objects from a single window. After finding the result, a Self Service User can quickly refer to the Self Service solution from the solutions application and refer to older tickets that were reported that resolved the issues reported earlier.

3. Attachment Search – Solution, SR, Incident, and Problem

   The Global search not only indexes data from the database, but it can also search within different file types that will be indexed as part of the attachment content index. By default, Lucene search supports .doc, .txt, .html, and .xml file types. All attachments with the specified file types will be indexed, but note that the search will only work on text based files. It will not work on any audio/video, .ppt, or .xls files.

5.3.2 Prerequisite configuration for Global Search

For Global search to function properly, certain setup activities have to be performed to enable the indexing mechanism.

The setup verification activities are:

1. Cron task setup.

   Pmobsearchcronj is the cron task responsible for creating and updating index files for Lucene. By default, PmObjSearchCron is active out of box and is set to a 24 hours schedule. This setup does not need to normally be altered or rescheduled.

If the cron task needs to be set up from scratch or for a different schedule, perform the following steps:

a. Select **Go To** → **System Configuration** → **Platform Configuration** → **Cron Task Setup**, as shown in Figure 5-42.

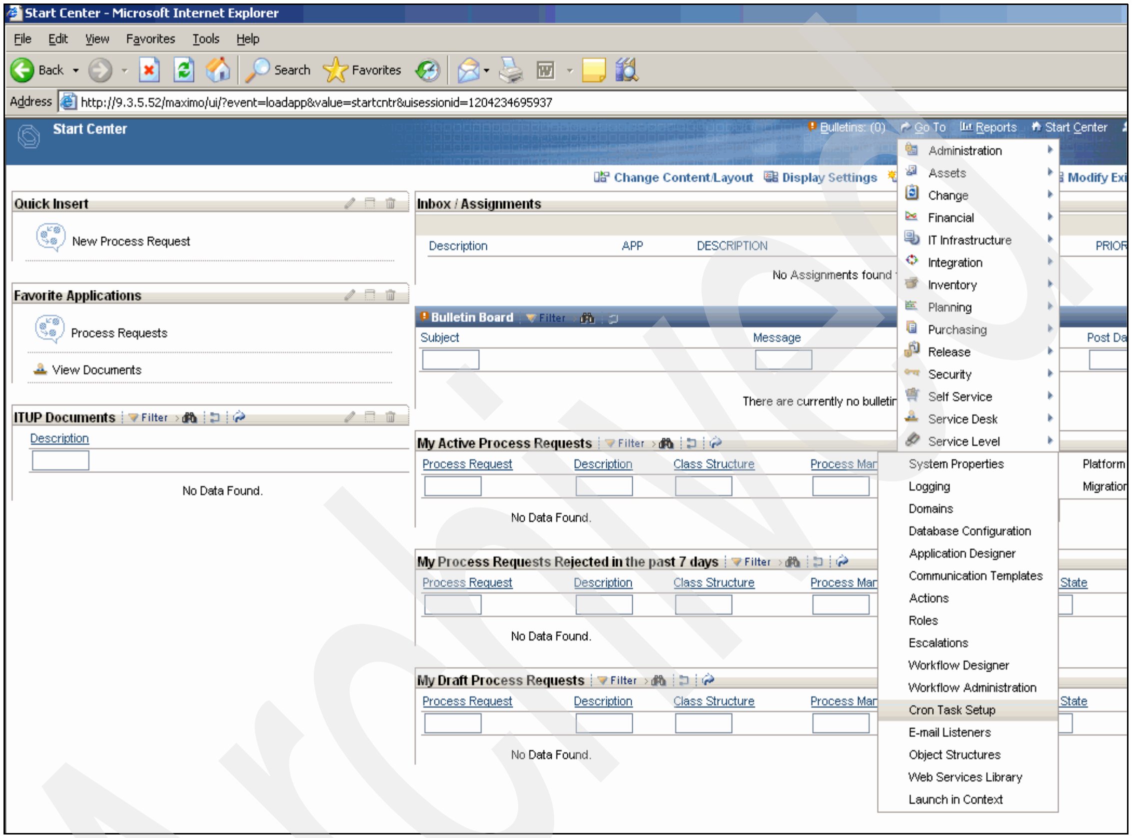


Figure 5-42 Cron Task on Go To Menu

b. Search for "Pmobsearchcronj" under the Cron Task column, as shown in Figure 5-43 on page 325.

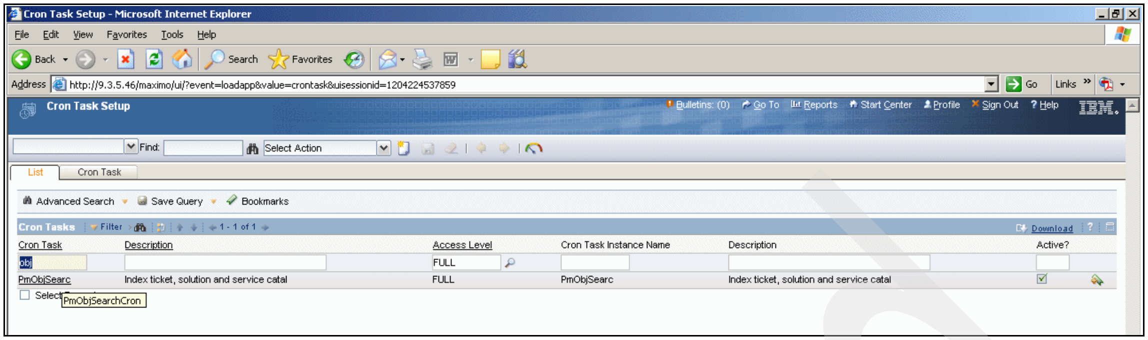


Figure 5-43 Cron Task Setup List Tab

c. Click the link and check the **Active** check box to make the cron task active and set the schedule time. Save the record, as shown in Figure 5-44. The time will determine how frequently the cron task will execute.

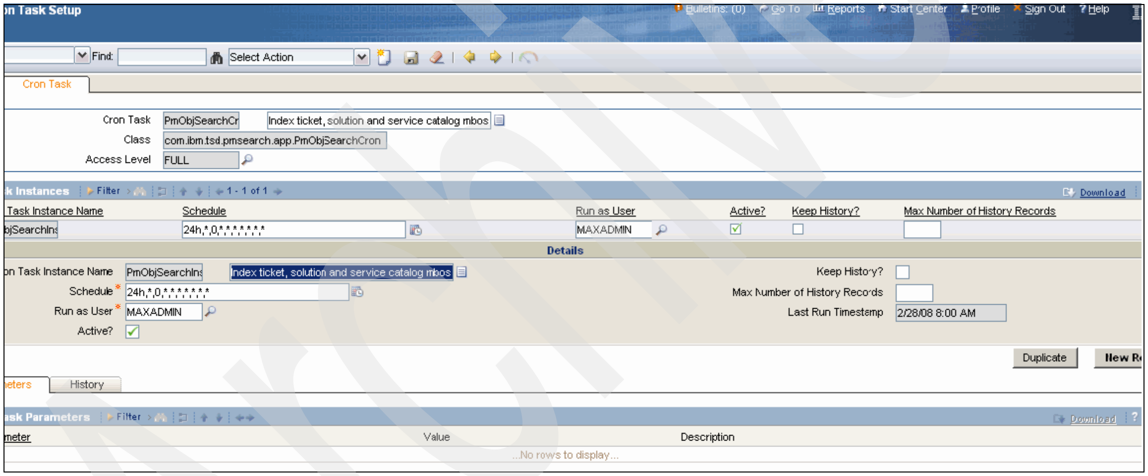


Figure 5-44 Cron Task Detail Tab

d. Wait for some time for the Lucene index to be created. By default, the index will be created under the following folders in the WebSphere environment:

<driveletter>:\...\IBM\WebSphere\AppServer\profiles\ctgAppSrv01\objsearchindex_<baselanguage>

<driveletter>:\...\IBM\WebSphere\AppServer\profiles\ctgAppSrv01\objsearchindex_<additional_lang1>

<driveletter>:\...\IBM\WebSphere\AppServer\profiles\ctgAppSrv01\objsearchindex_<additional_lang2>

For example:

<driveletter>:\...\IBM\WebSphere\AppServer\profiles\ctgAppSrv01\objsearchindex_EN

e. The drive (c: or d:) will change based on the installation directory. The “objsearchindex” is the default folder under which the index file will be created. In this case, as the base language is English, the Objsearchindex_EN folder is created as shown in Figure 5-45. Once this set of files has been created, you can start testing the search functionality.

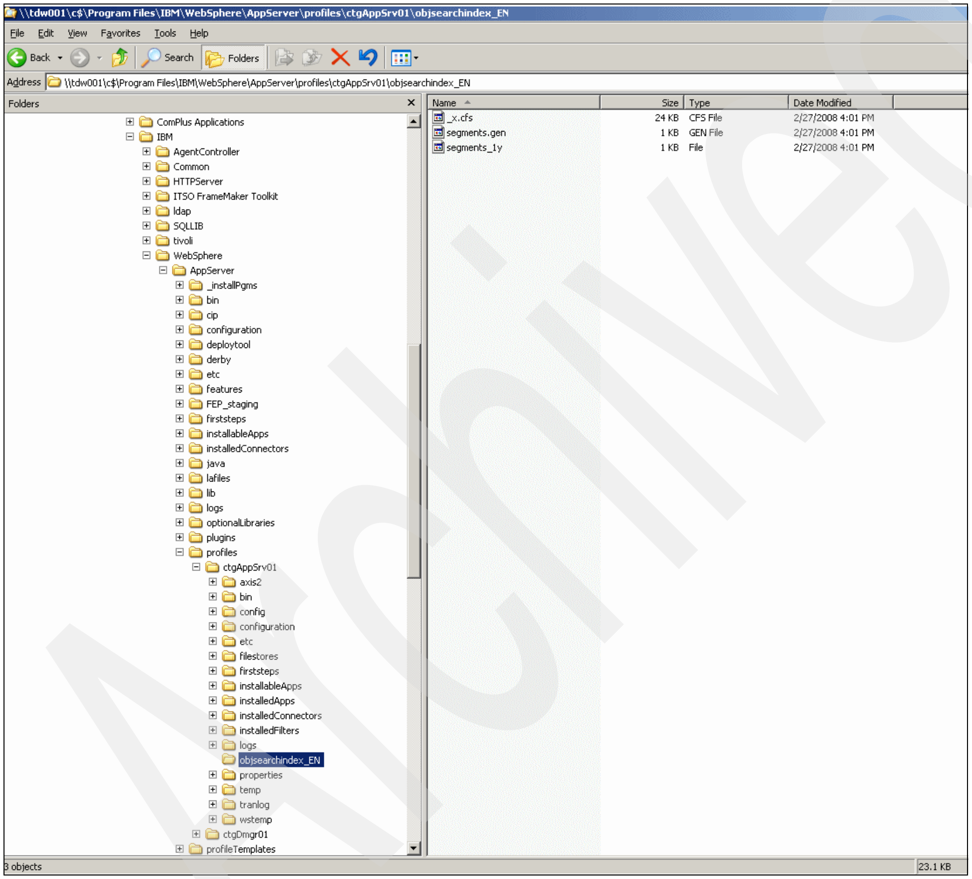


Figure 5-45 Objsearchindex_EN folder

2. Verify System Properties: LWDICT, LUCENEOBJINDEX, and ATTEXT.

   The user has the option to set the following three properties to enable the search. However, these are preset to the default values out of the box.

a. LWDICT: This property holds the folder path where the LanguageWare® dictionary files are stored. This property does not require frequent changes; however, Maximo Administrator can use the following information to change the location of the dictionary (.dic) files. Verify the property and proceed to the next property.

   i. To set this property, select **Go To** → **System Configuration** → **Platform Configuration** → **System Properties**, as shown in Figure 5-46.

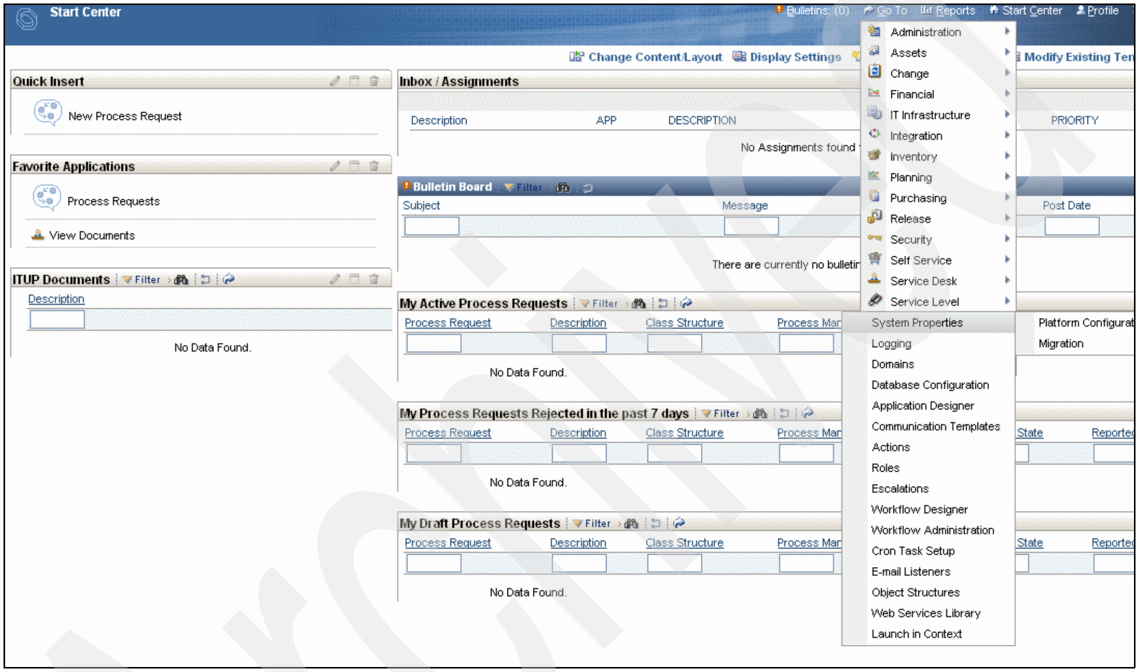


Figure 5-46 Go to System Properties

ii. Search for LWDICT in the Global Properties Table, as shown in Figure 5-47.

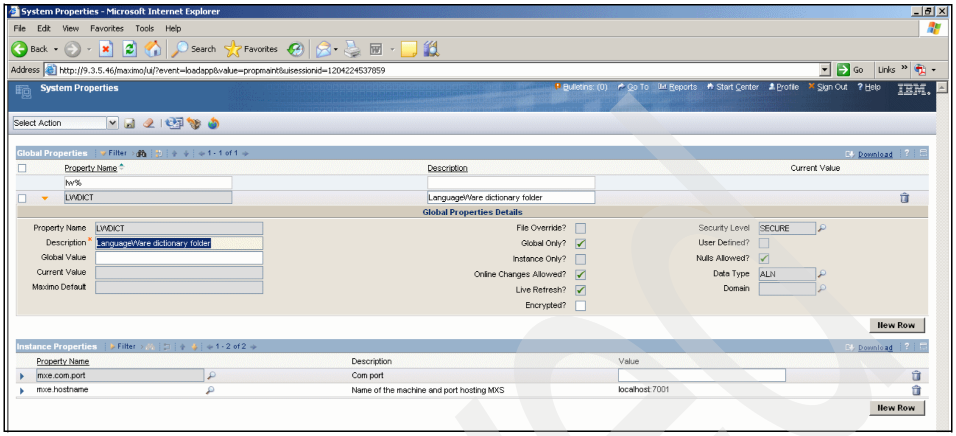


Figure 5-47 LWDICT GLobal properties

iii. By default the dictionary files will be under the <maximo install>/applications/maximo/lib folder. Check if you see any .dic files under this directory structure. Verify the folder name and look for .dic files, as shown in Figure 5-48.

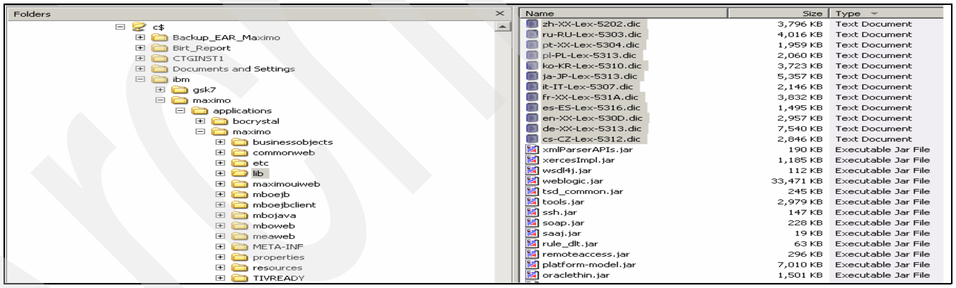


Figure 5-48 Folder location for .dic files

iv. Enter the above folder path into the Global value of the system property and save the record. Then check the check box next to the LWDICT property and select **Live Refresh** under the Select Action drop-down menu.

v. Click **OK**. You will see that the folder path will be set to both the Global and Current value fields, as shown in Figure 5-49 on page 329.

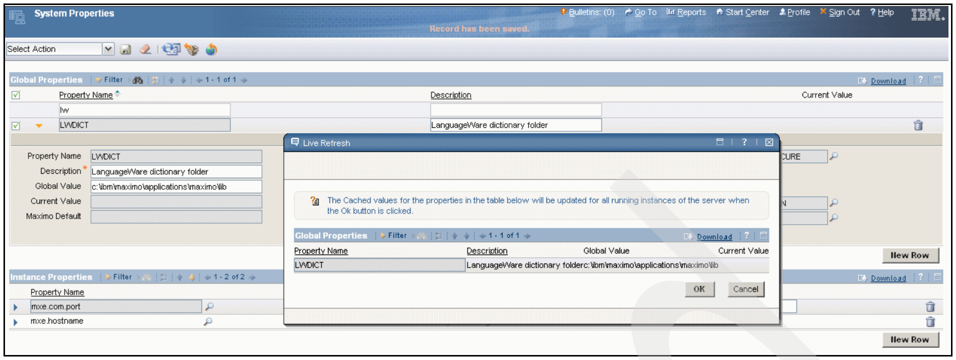


Figure 5-49 Live Refresh LWDICT properties

The LWDICT property has been set.

b. LUCENEOBJINDEX: This property determines where the Lucene index file has to be stored. Verify the property by checking the following:

   i. To set this property, select **Go To** → **System Configuration** → **Platform Configuration** → **System Properties**.

   ii. Search for LUCENEOBJINDEX in the Global Properties Table. By default, it is stored under the WebSphere installation directory structure under the objsearchindex folder, as shown in Figure 5-45 on page 326.

   iii. To set up the property, enter the above folder path into the Global value of the system property and save the record. Then check the check box next to the LWDICT property and select **Live Refresh** under the Select Action menu.

c. ATTEXT: This system property is used to specify the file types that will be indexed as part of the attachment content index. This is used for the Attachment Search functionality. By default, the .doc, .txt, .html, and .xml file types are supported, and the .pdf file type has to be added to the non-supported list, as it is not supported by Lucene search. This is an optional setting only and any change to this property will override the default settings for .htms, .doc, .txt, .xml, and .html.

   i. To set this property, select **Go To** → **System Configuration** → **Platform Configuration** → **System Properties**.

   ii. Search for ATTEXT in the Global Properties Table.

   iii. The value for this property must be specified as a string of file types separated by a comma, for example, ".properties,.txt,.html". This means that all attachments with the specified file types will be indexed, but the search will only work on text based files. It will not work on any audio/video, .ppt, or .xls files.

5.3.3 Screen flow

In this section, we will provide examples of different search options.

Global Search

This search is meant for Service Desk objects only. It is available for the SD analyst. This is available under **Go To** → **Service Desk** → **Global Search**, as shown in Figure 5-50.

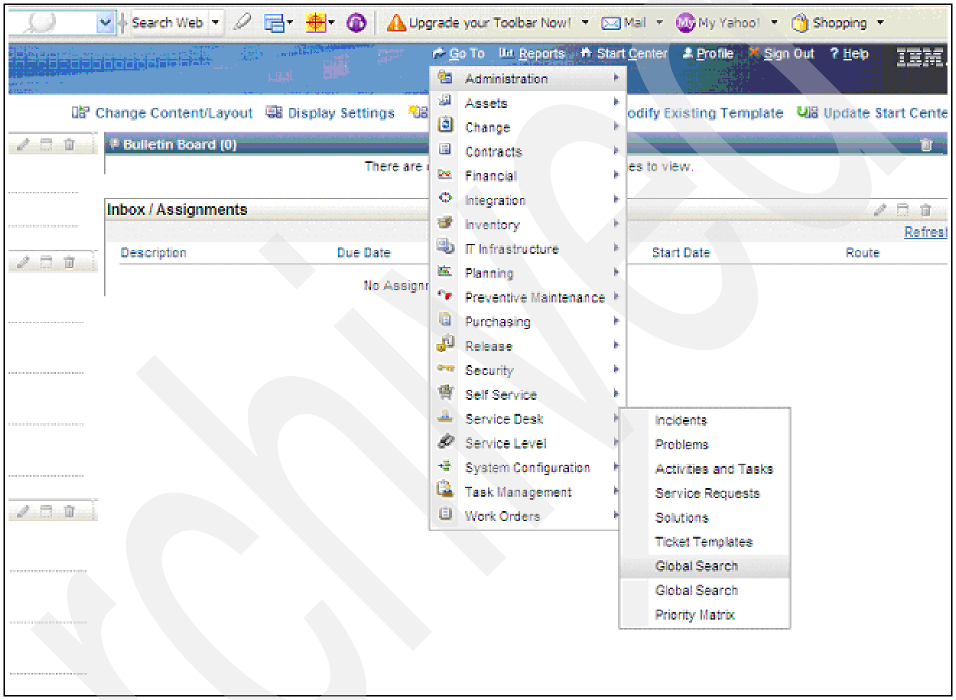


Figure 5-50 Go To Global search

The application is a Self Service application and looks like the window shown in Figure 5-51.

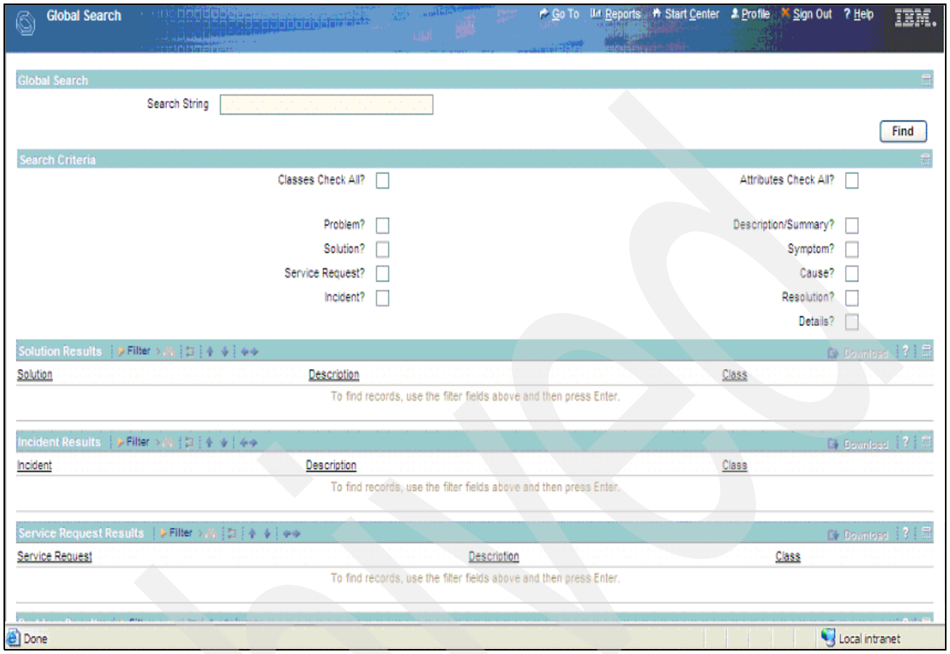


Figure 5-51 Global Search

Enter a search string and select the classes from the left side and the attributes from the right side for which the search will be performed. Click the **Find** button to start the search, as shown in Figure 5-51 on page 331. The results will be displayed section wise in the tables. To launch an individual record from the list, click the corresponding record ID, as shown in Figure 5-52.

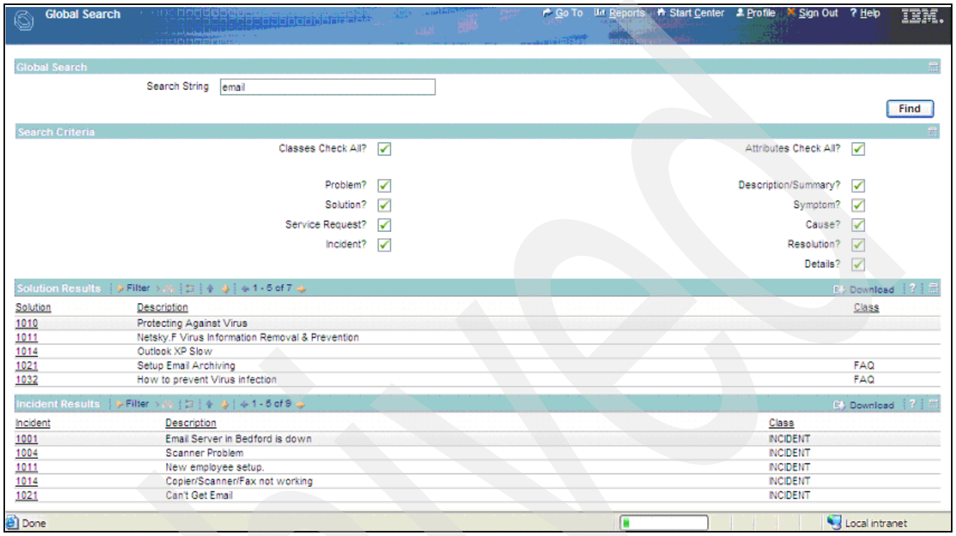


Figure 5-52 Global Search Results

Global SRM search

This search supports both Service Desk and Service Catalog objects. It is available for the Self Service user. It can be accessed by selecting **Go To** → **Self Service** → **Global SRM Search**, as shown in Figure 5-53.

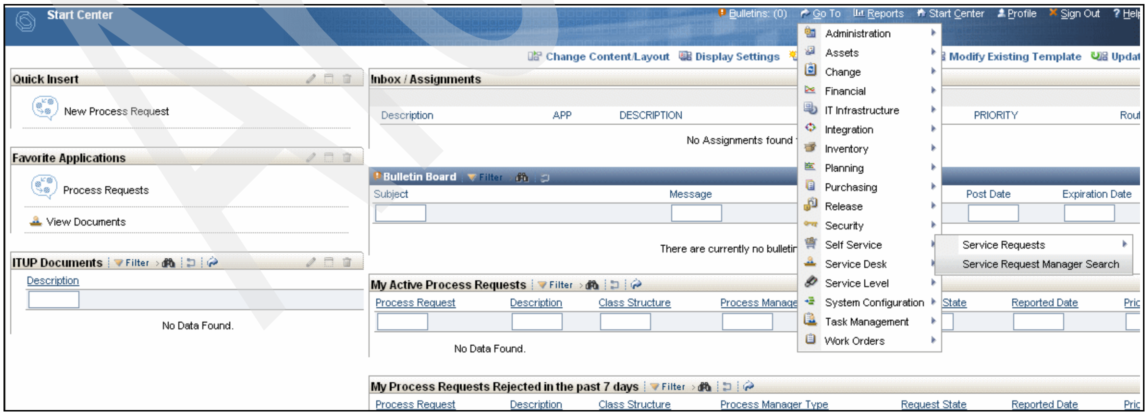


Figure 5-53 Go To Global SRM Search

When you click the link, you will be taken to the SRM Search window, as shown in Figure 5-54, which is similar to the Global Search window.

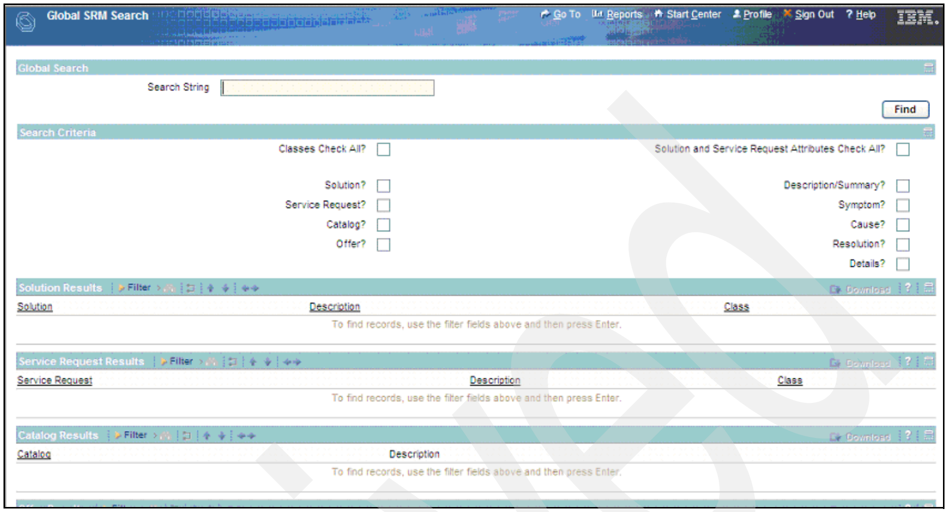


Figure 5-54 Global SRM Search

The rest of Global SRM Search's functionality is just like the functionality of Global Search.

Attachment Search: .doc, .txt, .html, and .xml file types

This search will search for contents in files attached to a Solution, SR, Problem, or Incident.

In this example, we will use the Solution application:

1. A document has to be attached to the Solution. Select **Go To** → **Service Desk** → **Solutions**, as shown in Figure 5-55.

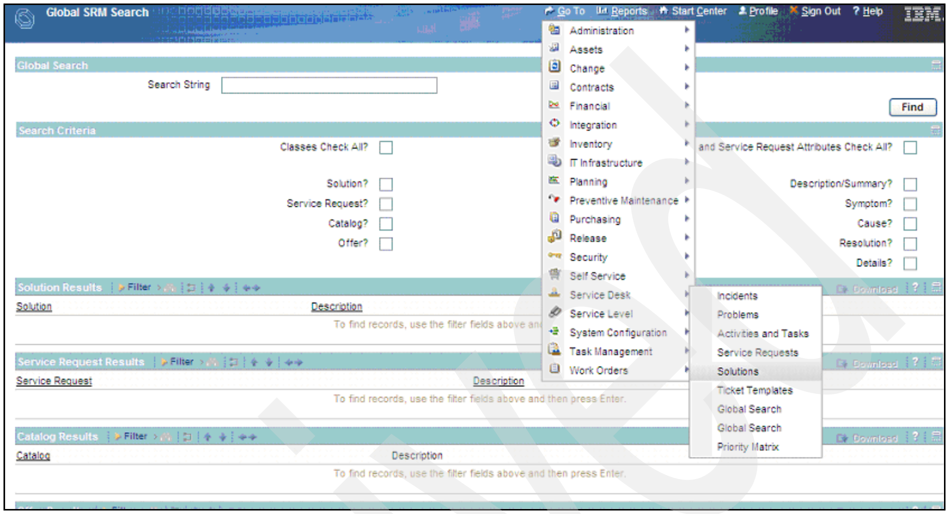


Figure 5-55 Go To Solutions

2. Search for a solution, for example, type 1005 on the Solution List tab and click the **1005** link to open the main application, as shown in Figure 5-56.

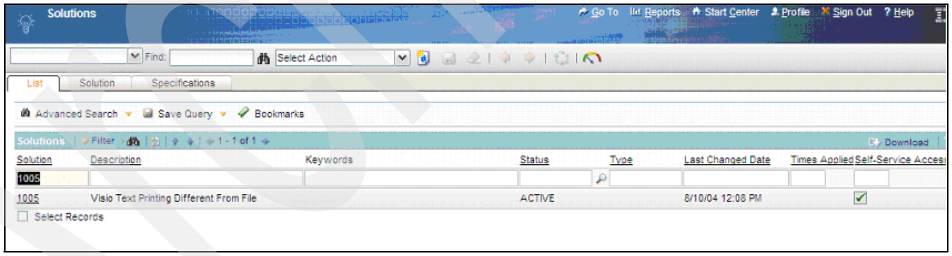


Figure 5-56 Solution List tab

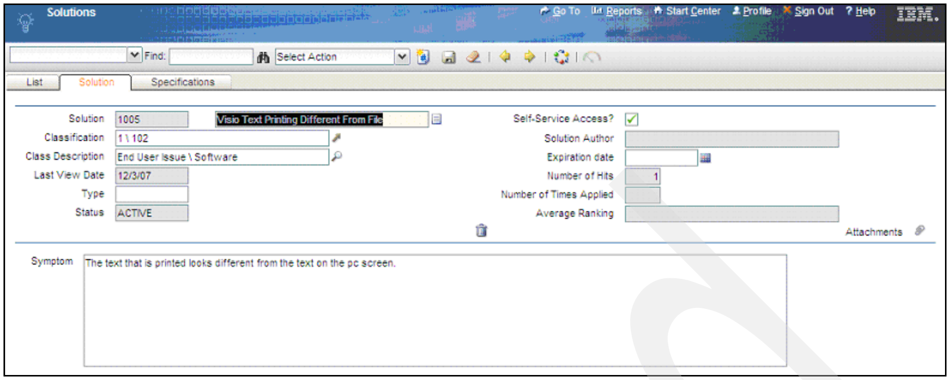


Figure 5-57 1005 link

3. Click the **Add New File** link, attach a file, and then add a new file to the solution by clicking the **Attachment Clip** icon, as shown in Figure 5-58.

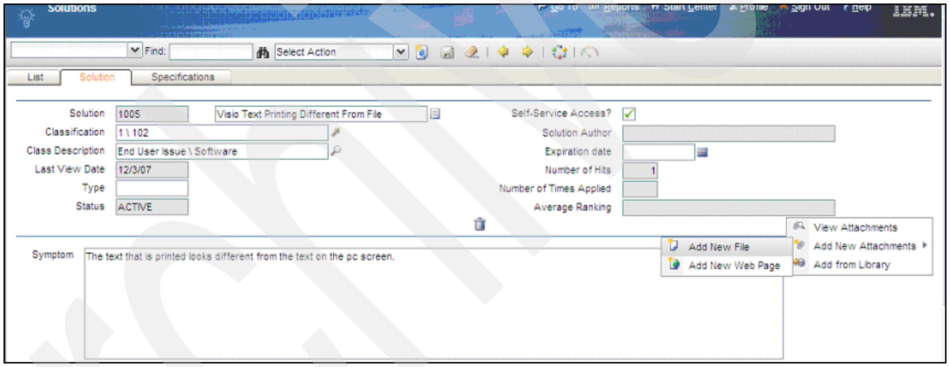


Figure 5-58 Solution tab

4. Click the **OK** button and the solution will now have the attachment, as shown in Figure 5-59 and Figure 5-60.

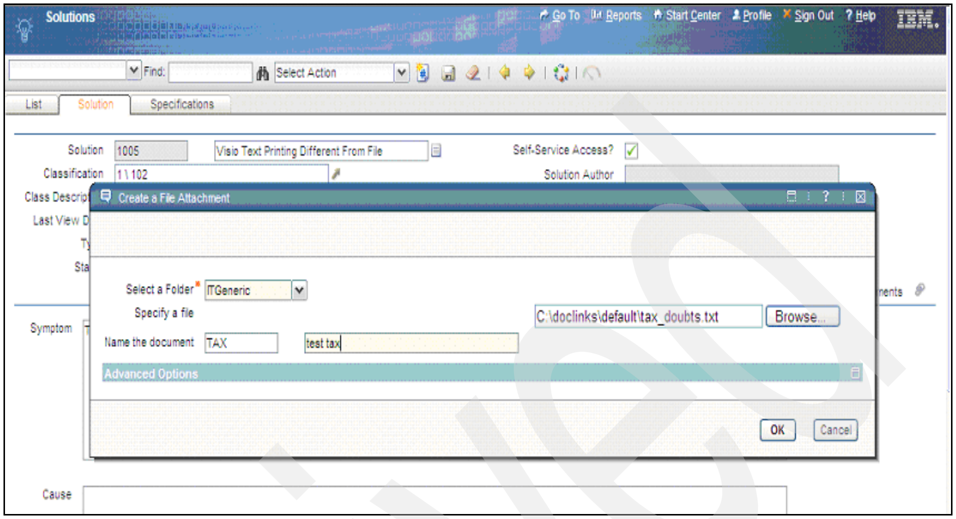


Figure 5-59 Attachment document to solution

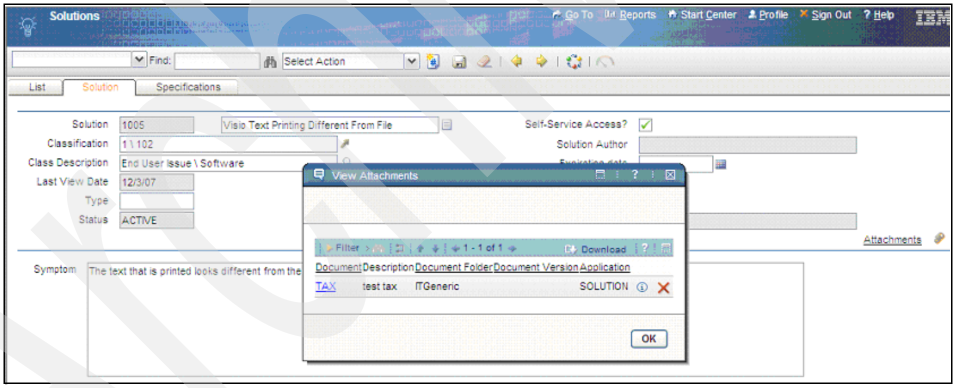


Figure 5-60 Attached document confirmation

5. To perform the search, go to the List tab of the Solutions application and select **Advanced Search** → **Attachment Search**, as shown in Figure 5-61 on page 337.

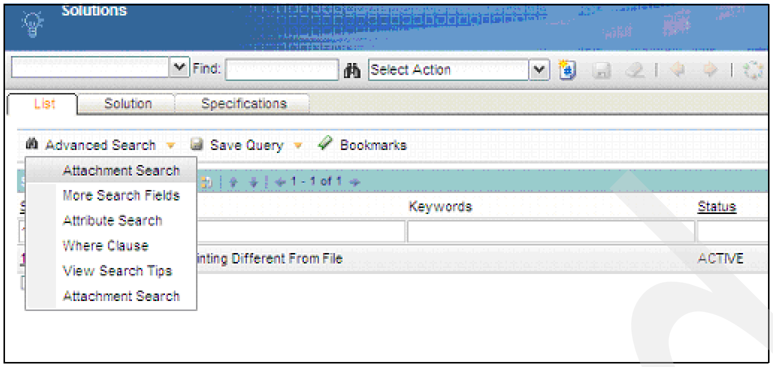


Figure 5-61 Solution Attachment Search feature

6. When you click **Find**, the result will contain all the solutions that have the term "deduction" inside their attached files, as shown in Figure 5-62.

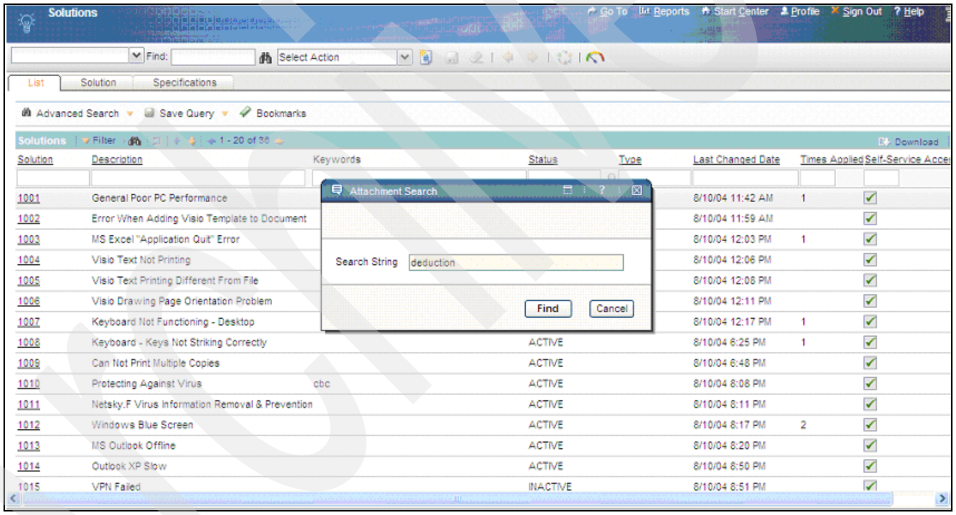


Figure 5-62 Attachment search string

7. This search result means that solution “1004” has an attachment containing the word “deduction”. In this case, the document under “Tax” contains the word, as shown in Figure 5-63.

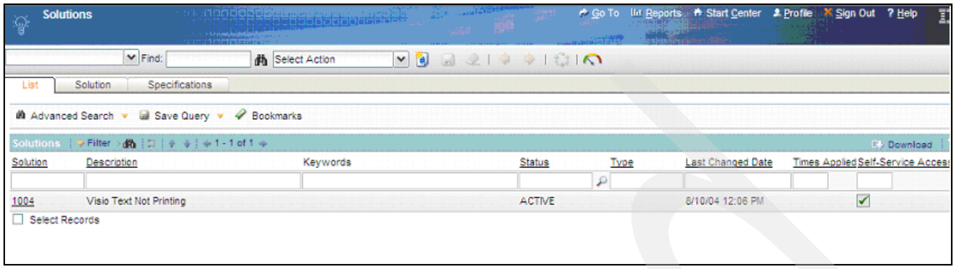


Figure 5-63 Attachment search result

8. When you point the mouse at the “TAX” link, you can see the actual file displayed on the bottom of the browser window. When you open the file, you can find the word, as shown in Figure 5-64.

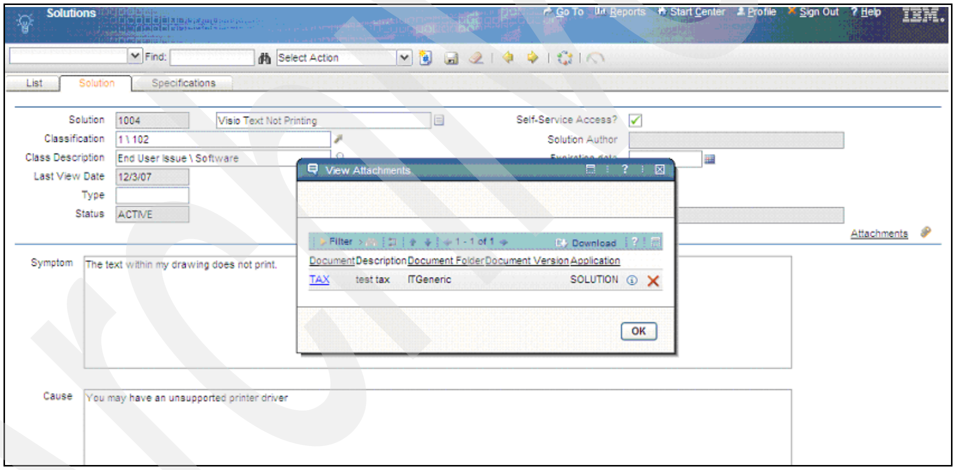


Figure 5-64 Solution detail from the attachment search result

9. In order to clear the results of the search, do the following:
   a. Go to the List tab and select **Advanced Search** → **Where Clause**, as shown in Figure 5-65.
   b. Clear the query and click **Find**. It will reset the search results, as shown in Figure 5-66.

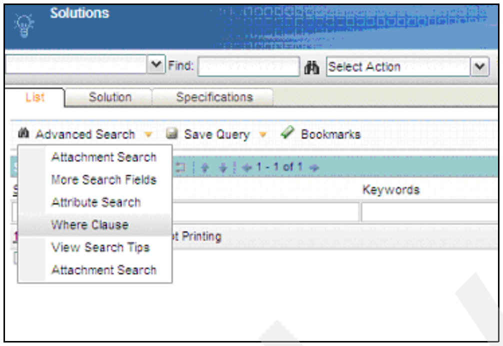


Figure 5-65 Solution Search where caluse

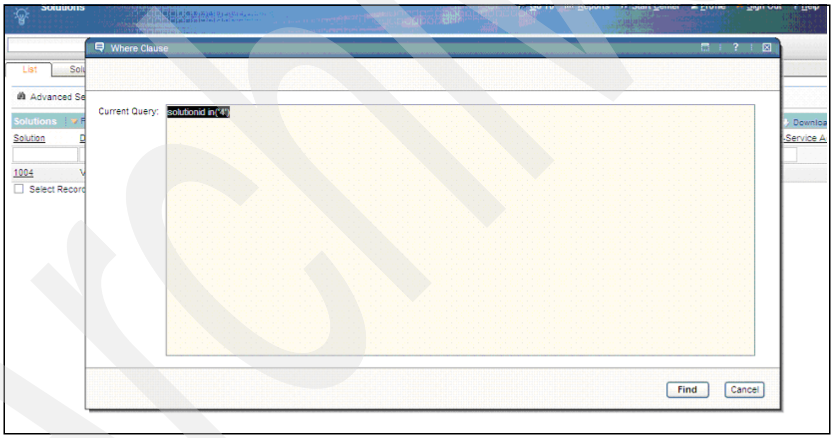


Figure 5-66 Clear search where clause

6

Best practices

This chapter will highlight best practices for high availability and system fine tuning for Tivoli Service Request Manager.

Availability and performance are the most important topics in today's IT environment. They are the basic requirements for any IT system in order for it support business in a smooth fashion. In order to meet the business requirements for availability and performance, IT administrators must have management processes to follow to ensure availability and performance. This requires the IT professional to adopt a strategy that is both *proactive* and *reactive* to conditions and events that would tend to adversely impact IT systems.

The proactive effort involves a number of tasks, including the following:

- Choosing the most effective IT architecture for the current and anticipated workload
- Adopting best practices in application design, development, and deployment
- Performing rigorous regression testing prior to deployment in a production environment
- Performing routine monitoring of key performance indicators to forestall potential performance problems, as well as gather information for capacity planning

The reactive effort involves having a well-defined methodology for identifying the root cause of a problem and resolving the problem by applying best practices.

ITIL provides the best practices for IT service management and the most relevant ITIL processes for availability and performance are Availability Management and Capacity Management.

In this chapter, we will focus on the following:

- "Availability Management" on page 343
- "Tivoli Service Request Manager availability considerations" on page 349
- "Fine tuning for Tivoli Service Request Manager components" on page 353

6.1 Availability Management

Before discussing Tivoli Service Request Manager availability considerations, we would like to introduce the ITIL Availability Management process. If you are familiar with ITIL processes, you can skip this section and proceed to 6.2, “Tivoli Service Request Manager availability considerations” on page 349.

Availability Management is a process inside ITIL service delivery processes. The goal of Availability Management is to ensure a cost effective operation for delivering a defined level of availability of IT services to meet the business requirements.

In today’s environment, business requires IT to provide the services for its operation in a timely way. Any service interruption causes loss of business and impacts customer satisfaction. Availability is normally the first priority consideration for IT services operation.

Availability Management understands the IT service requirements of the business. It plans, measures, monitors, and continuously strives to improve the availability of the IT infrastructure to ensure the agreed requirements are consistently met.

There are several indications that are important in Availability Management. These are the key items we will need to consider and manage in Availability Management processes. They are:

- Availability

  An IT service is able to perform the expected functionality over a specified time period.

- Reliability

  An IT service is able to perform the expected functionality, over a certain period of time, under prescribed circumstances.

- Maintainability

  An IT service can be easily maintained.

- Serviceability

  An IT service has all the relevant contract conditions of external suppliers to maintain it.

- Resilience

  The ability of an IT service to function correctly in spite of the incorrect operation of one or more subsystems.

6.1.1 Foundation of availability

The foundation of availability is to minimize downtime, maximize uptime, and generally provide a high level of service. It is different from recovery and is geared towards keeping systems running rather than recovering from them after an outage. The result of a unscheduled outage will directly impact your organization in terms:

- Enterprise demise
- Legal
- Business strategy
- Image
- Personnel morale
- Penalties

The following costs could be generate after a unscheduled outage:

- Lost revenue
- Lost user productivity
- Lost IT productivity
- Overtime payments
- Wasted goods
- Fines

6.1.2 Availability Management framework

When we understand what Availability Management is, it is time to define and establish the Availability Management framework.

This activity requires you to develop guidelines and the framework for Availability Management.

Establish the Availability Management framework

This activity develops guidelines and a framework for Availability Management. The following tasks are part of this activity:

1. Understanding the requirements and specifications for Availability Management.
2. Defining the strategy for Availability Management tools and capabilities, and how they should be sourced, should they be developed in-house or rely more on vendor capabilities.

3. Defining evaluation criteria for Availability Management solutions and services.
4. Establishing the framework for Availability Management by defining and implementing practices and systems that support process activities.
5. Based on these systems, determining skill requirements for the staff and assigning staff.
6. Finally, the structure and process of Availability Management, including escalation responsibilities, have to be communicated to the process users.
7. The establishment of the process framework also includes the continuous improvement of Availability Management, that is, the consideration of the Availability Management process evaluation and the implementation of recommended improvement actions.

These task are shown in Figure 6-1.

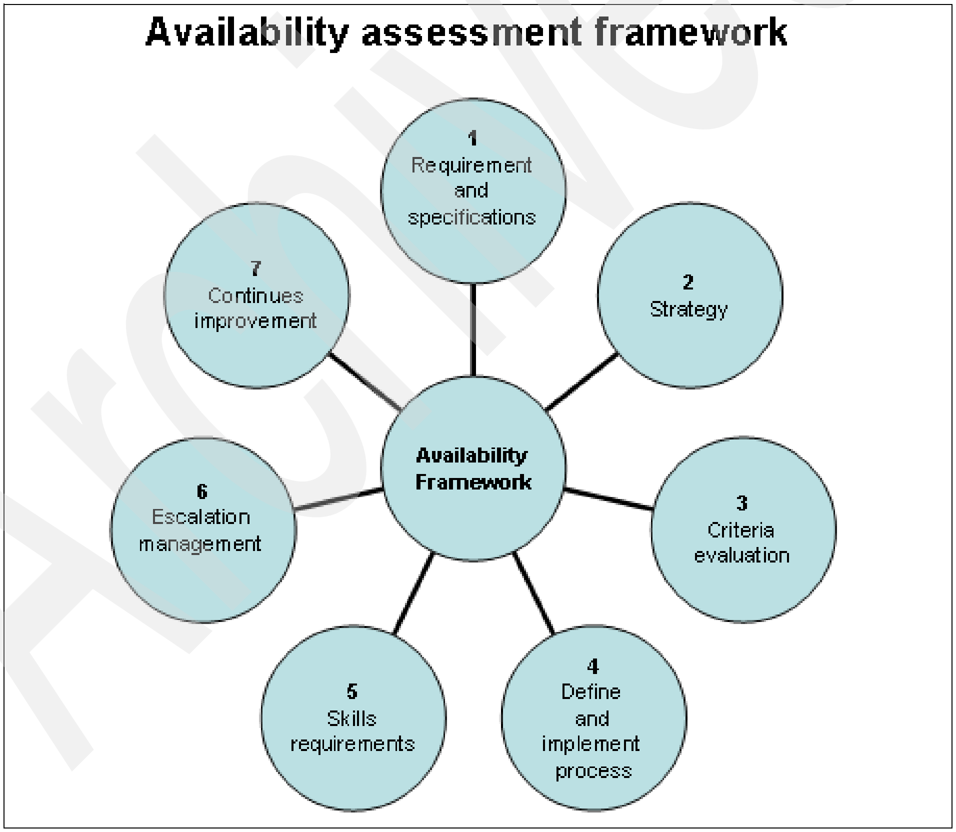


Figure 6-1 Availability assessment framework

Tools requirements for Availability Management

In order to effectively support the Availability Management process, a range of monitoring and management tools are required to support the activities in Availability Management. We need automation tools to help with activities like measurement, monitoring, analyzing, and reporting in Availability Management.

The required tools depend on the availability and automation requirements for daily Availability Management operations.

The tools should provide the following functions:

- Monitoring for the specified target of the IT resource or service
- Alerts for errors or threshold violations
- A reporting facility for generating the required management reports
- A centralized system management for reducing man made errors and enhance the quality of system management

6.1.3 Levels of availability

Different levels of availability have been defined within IBM to help customers address their business needs.

Based on the assessment for the Availability Management framework and your business objectives, you will have the ability to choose the level of availability defined in this section.

More than one consideration has to be taken before choosing the level of availability required to ensure that you will provide to your environment the added value expected for the level chosen.

> **Important:** Implementing a level of high availability that is not required by the business can require an investment that will not give you a *Return of Investment (ROI)*.

Each level of availability defines the percentage of uptime, what the service level should be, and the business hours for the support.

Economy

This level of availability is primarily for business hours:

- There is 98.5% uptime.
- There is low availability.
- The service level could be at 98.5%.

- There is no guaranteed minimum service level.
- This is an “office hours” systems with no support commitments.
- There is no guaranteed data recovery.

Critical

This level is the standard availability model:

- The typical service level is 99.0%.
- The minimum service level is about 98.5%.
- This is typically an “office hours” environment.
- There is a potential for data loss.

High available

This level is the high availability model:

- The typical service level is 99.5%.
- The minimum service level is about 99.0%.
- This can be a 24x7 environment, but is normally an extended “office hours” system.
- There is limited data loss.

Mission critical

This level is the very high availability model:

- The typical service level is 99.7%.
- The minimum service level is about 99.5%.
- This is typically a 24x7 environment.
- The recovery point objective is as close to zero as possible.

6.1.4 Key performance indicators

Once you have chosen your level of availability, you will have to make sure your infrastructure will be able to support and maintain it. The keys to successfully maintaining your availability level are the key performance indicators (KPIs).

Key performance indicators (KPIs) are the detailed specifications used to track business objectives. They are evaluated by management.

A KPI is associated with a specific process and is generally represented by a numeric value. A KPI may have a target and allowable margins, or lower and upper limits, forming a range of performance that the process should achieve. A KPI can be thought of as a metric with limits, and is itself made up of one or more metrics.

KPIs are created based on business objectives. A business objective is an executive statement of direction in support of a corporate strategy. The business objective is a high-level goal that is quantifiable, measurable, and results-oriented. For business measures modeling, the business objective is translated into a KPI that enables the organization to measure some aspect of the process against a target that they define.

The way to keep your management updated on the actual situation of your IT infrastructure are: report, on a daily or weekly basis, through a dashboard system or any other accessible tools.

What you will need to support the availability of your systems

The most important ITIL processes that will help your organization maintain your level of availability are listed and defined here:

- Service Management Processes
  - Incident management
  - Problem management
  - Change management

But, in order to effectively support the Availability Management process, a range of monitoring and management tools are needed to support the activities in Availability Management. We need automation tools to help with activities like measurement, monitoring, analyzing, and reporting in Availability Management. Your tools requirement depends on the availability and automation requirements for daily Availability Management operations.

The tools should provide the following functions:

- Monitoring for the specified target of the IT resource or service
- Alerts for errors or threshold violations
- A reporting facility for generating the required management reports
- A centralized system management for reducing man made errors and enhancing the quality of system management

As for the monitoring, we can consider several different views when setting a monitoring target. We could have three monitoring views: *individual IT component view*, *application view*, and *IT service view*. The reason to have the different views is because we have different measurement foci for the different aspects. From the IT services view, the concern is about service availability. From the application view, the concern is about application availability. And from the IT component view, the concern is about IT component availability. The three views have a layered relationship: the service view is on top, the application view is in the middle, and the IT component view is at the bottom. With the three different monitoring views, we have better control over service management for availability.

The focus for IT component view focuses on the resources monitoring for IT components. We want to know the status for each IT component, and if there is any existing errors for the IT component.

The focus for application view focuses on the composite application monitoring. We want to know the running status of the middleware and applications. The applications transaction may traverse multiple servers and its correctness depends on all the underlying IT components.

The focus for service view focuses on the IT services. We want to know the required IT services availability for a business operation.

6.2 Tivoli Service Request Manager availability considerations

Having discussed the Availability Management process defined by ITIL, we can move on to Tivoli Service Request Manager Availability Management considerations. You can install Tivoli Service Request Manager in two different ways:

- Basic configuration
- Advanced (or clustered) configuration

Each configuration is suitable for certain kinds of environments.

6.2.1 Basic Tivoli Service Request Manager configuration

A basic Tivoli Service Request Manager configuration consists of Tivoli Service Request Manager running on a single application server. The application server connects to a single instance of the Tivoli Service Request Manager database that is available on a database server. If the Maximo Enterprise Adapter is also configured for deployment, then additional messaging queues are set up. Maximo uses the additional queues to send data to external systems, and to receive data from external systems.

This configuration is suitable for up to 50 users. Even with fewer than 50 users, the basic Tivoli Service Request Manager configuration can become overloaded if a great deal of processing is being done. For example, running scheduled jobs (cron tasks) requires considerable memory and processing power.

When processing power and memory are overtaxed, the user who is using the application from a browser can experience unacceptable performance. If the basic configuration is performing poorly, you might need to deploy the advanced enterprise configuration.

Note: In the basic configuration, even if the application server process runs on a 64-bit computer, performance can be slow. Although a 64-bit computer allows greater memory utilization, the server process suffers from Java garbage collection pauses due to large heap space, and performance slows.

If your production system must support hundreds of users, the basic configuration will likely prove insufficient due to memory and processing limitations. A single instance of the application server to support the Maximo application cannot handle the load that is required in enterprise deployments.

Note: Because Tivoli Service Request Manager is an interactive application, the user who uses Tivoli Service Request Manager from a browser expects a response from the server to be immediate, or nearly so.

Other processes such as cron tasks and inbound messages from external systems do not require user interaction. The response time for these processes does not need to be immediate. These processes can be configured to run in separate clusters or on separate hardware.

6.2.2 Advanced Tivoli Service Request Manager configuration

You can use load distribution in a heavily used system that experiences unacceptable delays in Tivoli Service Request Manager and overall throughput. Load distribution separates the user-interface traffic from all other processing in Tivoli Service Request Manager.

Deploying Tivoli Service Request Manager on more than one server (or cluster) is an effective way to distribute user load and improve the user experience of system performance.

When you use clustering, implement the cluster configuration so that the system can scale well. You can set up multiple clusters, each consisting of multiple Java virtual machines (JVMs). The number of JVMs depends in part on the overall hardware and software limitations of the environment. You can tune the setup based on traffic and performance numbers.

A system of separate clusters with multiple JVMs as cluster members provides advantages in system administration. In such a setup, Tivoli Service Request Manager users are not affected by any problems in the queue cluster.

For your system to perform well, isolate the Tivoli Service Request Manager user-interactive applications to one cluster, and the asynchronous processes to a separate cluster. Such a configuration also helps the system to scale well.

The use of Tivoli Service Request Manager application reports can also put a large burden on the Tivoli Service Request Manager user interface performance. To remove the impact of reports on the user interface performance, run reports that require significant system resources as scheduled reports on separate clusters. To help the overall system performance, run only simple reports from the user interactive application.

You can further enhance each cluster simply by adding servers, depending on your needs. Figure 6-2 illustrates the high-level components of an advanced configuration

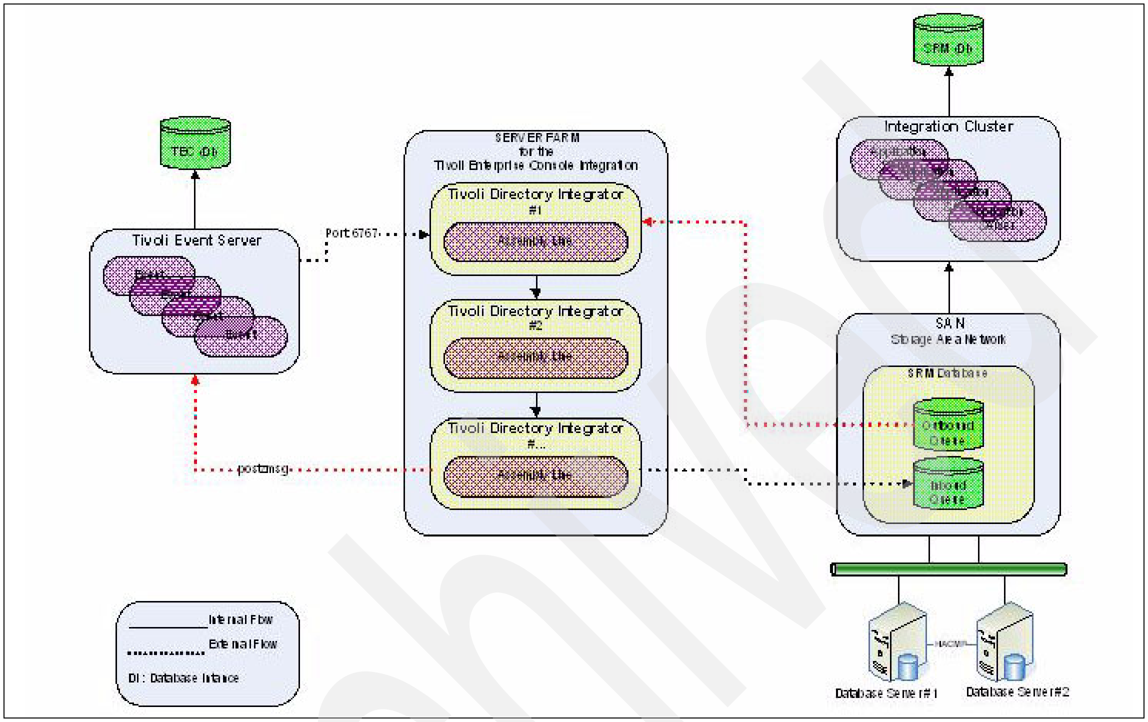


Figure 6-2 Advanced (clustered) Tivoli Service Request Manager configuration

Multiple cluster setup is also good idea when Tivoli Service Request Manager integration with external systems is implemented. The purpose of clustering is to separate the inbound queue by placing it in a separate cluster from the interactive users. Separating the queue this way prevents inbound traffic from affecting the user interface performance.

Every external system that Tivoli Service Request Manager needs to communicate with should be set up with its own sequential inbound and sequential outbound queues. If you also set up each sequential queue with its own cron task, administering the queues is easier. For example, you might set up a system with three clusters, and call them UI, Q, and Cron.

- UI cluster: Provides a connection for Tivoli Service Request Manager application users. Hosts the Web application and sequential outbound queues.
- Q cluster: Principally for hosting continuous and sequential inbound queues.

- Cron cluster: Where other Tivoli Service Request Manager cron tasks are configured.

The UI cluster is where users generate transactions when they save work in Tivoli Service Request Manager. The UI cluster should therefore host sequential outbound queues to send out user-generated Tivoli Service Request Manager transactions to external systems.

The Q cluster is intended for incoming data for Tivoli Service Request Manager.

The optional Cron cluster is set up for other Tivoli Service Request Manager cron tasks. Setting up a third cluster for cron tasks is helpful and should bring the system configuration close to its ideally segregated potential. If cron tasks might generate outbound Tivoli Service Request Manager transactions, then the Cron cluster needs its own set of sequential outbound queues. Outbound queue setup on the Cron cluster would be similar to outbound queue setup on the UI cluster and the Q cluster.

6.3 Fine tuning for Tivoli Service Request Manager components

In this section, we will delve into the details of tuning Tivoli Service Request Manager for best performance. We have broken down this topic into several categories:

- Application Server performance testing and fine tuning
- Database Server and SQL fine tuning
- Network and bandwidth fine running
- Client workstation configuration
- Reporting configuration

6.3.1 Application server fine tuning

An application server is basically a software platform or framework to host and run applications, such as Tivoli Service Request Manager. IBM WebSphere Application Server and BEA WebLogic are the two standard application servers that Service Request Manager uses.

> **Note:** BEA WebLogic Server support for Tivoli Service Request Manager is planned to be available after the product is generally available.

The Service Request Manager Application Server maintains Tivoli Service Request Manager business application objects and configuration files. This section describes the benefits of tuning for optimal performance, highlights the tunable parameters of the major WebSphere Application Server components, and provides insight about how these parameters affect performance.

WebSphere Application Server provides tunable settings for its major components to enable you to make adjustments to better match the runtime environment to the characteristics of your application. Many applications can run successfully without any changes to the default values of these tuning parameters. Other applications might need changes, for example, a larger heap size, to achieve optimal performance.

Performance tuning can yield significant gains in performance even if an application is not optimized for performance. However, correcting shortcomings of an application typically results in larger performance gains than are possible with just altering tuning parameters. Many factors contribute to a high performing application.

Recommended hardware configuration for Tivoli Service Request Manager

In Tivoli Service Request Manager, every JVM must have a minimum of 1 CPU (one 800 Mhz or faster processor) and 1 GB of memory, but for better performance, each JVM should have a processor and 2 GB of memory dedicated to it. WebSphere itself should have one CPU and 1 GB of memory assigned to it. The OS should also have 1 CPU and 1 GB of memory assigned to it. That means to support 50 concurrent connections with no MEA installed you need a minimum of three CPUs and 4 GB of memory. For every 50 concurrent connections needed, another JVM and its requirements will be needed. If the MEA is involved with moderate to small activity, then one more JVM should be added and the MEA and cron task transactions should be separated from the user JVMs. If a high level of activity is expected, then the MEA should have its own JVM and cron tasks should have its own JVM.

Application performance testing

Application performance testing is an important component of the software deployment cycle. It becomes even more important with respect to Web applications, since a user's tolerance for slow applications is generally much lower than that of a captive internal audience. Performance testing has been a subject of many products and white papers recently, and has spawned a new IT industry dedicated to Web application testing.

In this section, we are going to talk about application server parameters that may affect system performances. When performance testing a Web application, several requirements must be determined either through interpretation of data from an existing application that performs similar work, or from best-guess estimates. Those requirements are:

- Average request rate

  What is the expected number of users who will access this application? This is generally expressed in hits per month, day, hour, or minute, depending on volumes. This should be re-evaluated regularly.

- Peak request rate

  How many pages will need to be served per second? This also should be re-evaluated regularly.

- Average concurrent users

  What is the average number of users accessing the application at the same time during regular usage hours? This should be planned for, expected, and re-evaluated on a regular basis.

- Peak concurrent users

  This is the maximum number of concurrent users that will visit your site during peak time.

- Regular usage hours

  This value defines your off-peak hours. This is required to simulate a realistic workload.

- Peak usage hours

  During this time, most of your traffic will happen and a performance degradation would impact most of your users.

- Site abandonment rate

  How long will a user stay on your page before he leaves the site or closes the browser?

The user base, especially for Web applications, is a difficult number to determine, especially if this is an application that is performing a new function on a Web site. Use existing Web server measurements to provide a "best-guess" number based on current traffic patterns for the site. If capturing these numbers is not possible, then the best option is to determine the breaking point for the application within the intended deployment infrastructure. This provides the ability to monitor once the application is live and to provide increased capacity prior to a negative user response due to load.

When dealing specifically with applications running in WebSphere Application Server, there are performance testing protocols that can be followed.

Refer to the following for more details on WebSphere Application Server Performance testing protocol tools:

http://www.software.ibm.com/wsdd/techjournal/0211_polozoff/polozoff.html

One important point to remember is that performance tuning is more of an "art" than a science. Do not worry if you get the impression that you will not be able to achieve your tuning goals if you believe you lack the "talent" for that art, because, on the other hand, performance tuning should also be seen as an art that strictly follows the recurring, monotonous trifold process of *testing*, *evaluating*, and *tuning*. This process requires that you know your system, your environment, and application very well, that you know what you want to test, what goal(s) you want to achieve, and also that you are familiar with the tools you are going to use for load testing, all of which implies more of a solid handicraft than an artist's creative work. But having a bit of intuition and developing a feeling for your work will most certainly help you in finding the best configuration. Finally, experience will be your most valuable friend, so get to know different load testing tools on various environments to gain comprehensive, in-depth knowledge.

It is important to keep in mind that it is impossible to make up for poor application design or application code by tuning WebSphere. It is also important to remember that performance tuning is an ongoing process. Continual reviews of performance should be done to ensure that changes in load, application behavior, and site behavior have not adversely impacted application performance. Also, there are no hard rules for performance tuning. What may be appropriate tuning for one application may not be appropriate for another. It is more important to understand the concepts associated with tuning, and make adjustments based on the understanding gained from those concepts.

Using the appropriate tools

Deploying applications that perform and scale in an acceptable manner is not an accidental occurrence. Producing high performance software requires that you include several rounds of stress testing during the development cycle. You can use one or more of the many open source or commercial stress testing tools to automate the execution of your stress tests.

The primary purpose of stress testing tools is to discover under what conditions your application's performance becomes unacceptable. You do this by changing the application inputs to place a heavier and heavier load on the application and measuring how performance changes with the variation in those inputs. This activity is also called load testing. However, load testing usually describes a very specific type of stress testing: increasing the number of users to stress test your application.

The simplest way to stress test an application is to manually vary the inputs (for example, the number of clients, size of requests, frequency of requests, and mix of requests) and then chart how the performance varies. If you have many inputs, or a large range of values over which to vary those inputs, you probably need an automated stress testing tool. Moreover, you will want test automation to repeat test runs following environmental or application-specific changes once you uncover an issue.

> **Note:** We recommend making only one change at a time in between test runs so the results are measurable.

If you are testing manually, it can be difficult to accurately reproduce an identical set of tests across multiple test executions. When it comes to having multiple users testing your application, it is almost impossible to run manual tests consistently and it can be very difficult to scale up the number of users testing the application.

Today, there is no generic, one-size-fits-all stress testing tool. Every application differs in what inputs it takes and how it executes them. Java and WebSphere-based Web applications generally receive requests from clients through the HTTP protocol. There are many stress testing tools that can simulate user activity over HTTP in a controlled and reproducible manner.

With so many stress testing tools available today, how can you choose the one that is most appropriate for your application? Some of the points to consider when evaluating stress testing tools are discussed in “Client interaction”.

Client interaction

The stress testing tool must be able to handle the features and protocols that your application uses.

Simulation of multiple clients

This is the most basic functionality of a stress testing tool, that is, scripted execution with the ability to edit scripts. If you cannot script the interaction between the client and the server, then you cannot handle anything except the most simple client requests. The ability to edit the scripts is essential; minor changes should not require you to go through the process of re-generating a script. This task includes:

- Session support

  If a stress testing tool does not support sessions or cookies, it is not very useful and may not be able to stress test Java and WebSphere applications.

- Configurable numbers of users

  The stress testing tool should let you specify how many simulated users are running each script or set of tasks, including allowing you to vary the number of simulated users over time. Many stress testing tools enable you to start with a small number of users and ramp up slowly to higher numbers of users.

- Reporting: success, errors, and failures

  The tool you choose must have a defined way to identify a successful interaction, as well as failure and error conditions. An error might be getting no Web page back at all, while a failure might be getting the wrong data back on the page.

- Page display and playback

  A useful feature in many stress testing tools is the capability to inspect some of the pages that are being sent to the simulated users or to replay entire test scripts. You can then be confident that the stress test is functioning as you expect.

- Exporting test results

  After running a stress test, you may want to be able to analyze the test results using various tools that are external to the stress testing tool, including spreadsheets and custom analysis scripts. Most stress testing tools include extensive built-in analysis functions, but being able to export the data gives you more flexibility to analyze and catalog the data in arbitrary ways.

- Think Time

  Real-world users do not request one Web page immediately after another. There are generally delays between viewing one Web page and the next. The term *think time* is the standard way of expressing the addition of a delay into a test script to more realistically simulate user behavior. Many stress testing tools support randomly generated think times based on a statistical distribution.

- Variable data

  Live users do not work with the same set of data on each interaction with your application. During a stress test, this should also be true of your simulated users. It is easier to make your simulated users appear to be working with varied data if the stress testing tool supports data input from lists, files, or a database.

- Script recording

  Rather than writing scripts, it is much easier to manually run through a session with your browser and have that session recorded for later editing. Most stress testing tools include provisions for capturing manual interaction with your application.

- Analysis tools

  Measuring performance is only half the story. The other, and perhaps more important, half of stress testing is analyzing the performance data. The type of analysis tools and degree of detailed analysis you can perform depends directly on what analysis tools are supported by the tool you select. Therefore, evaluate this support in the tools you are considering carefully.

- Load distribution

  Your deployed application may well need to support hundreds of concurrent users once in production. How can you simulate this level of traffic in a stress testing environment? A typical workstation running a stress testing tool will likely to begin to bottleneck, once approximately 200 virtual users are running. To simulate a greater number of users, you can distribute the stress testing load across multiple workstations. Many of the available stress testing tools support distribution of load and you will certainly want this feature for large scale stress testing.

- Measuring server-side statistics

  The basic stress testing tool measurement is client-based response times from client/server interactions. However, you may also want to gather other statistics, such as the CPU utilization or page fault rates. With this server-side data, you can then do useful things like view client response times in the context of server load and throughput statistics.

Application performance tuning guidelines

Tuning is about utilizing resources to their fullest potential, resulting in the fastest request processing possible. Many components in WebSphere Application Server have an impact on performance and tuning is highly application-dependent. This section discusses how to identify bottlenecks, gives a practical introduction to analyzing them, and finally gives recommendations on settings for major WebSphere environment properties.

It is important to stress once again that performance tuning is not an exact science. Factors that influence testing vary from application to application, and also from platform to platform. This section is designed to provide a primer for the reader in a way that describes areas that can be tuned to increase performance.

Adjusting WebSphere Application Server system queues

WebSphere Application Server establishes a queuing network, which is a group of interconnected queues that represent various components. There are queues established for the network, Web server, Web container, EJB™ container, Object Request Broker (ORB), data source, and possibly a connection manager to a custom back-end system. Each of these resources represents a queue of requests waiting to use that resource. Queues are load-dependent resources.

As such, the average response time of a request depends on the number of concurrent clients.

As an example, think of an application, consisting of servlets and EJBs, that accesses a back-end database. Each of these application elements reside in the appropriate WebSphere component (for example, servlets in the Web container) and each component can handle a certain number of requests in a given time frame.

Queuing before WebSphere

The first rule of tuning is to minimize the number of requests in WebSphere Application Server queues. In general, requests should wait in the network (in front of the Web server) rather than waiting in the WebSphere Application Server. This configuration allows only those requests that are ready to be processed to enter the queuing network. To accomplish this task, specify that the queues furthest upstream (closest to the client) are slightly larger, and that the queues further downstream (furthest from the client) are progressively smaller.

Note: The queuing network becomes progressively smaller as work flows downstream. When 200 client requests arrive at the Web server, 125 requests remain queued in the network because the Web server is set to handle 75 concurrent clients. As the 75 requests pass from the Web server to the Web container, 25 remain queued in the Web server and the remaining 50 are handled by the Web container. This process progresses through the data source until 25 user requests arrive at the final destination, the database server. Because there is work waiting to enter a component at each point upstream, no component in this system must wait for work to arrive. The bulk of the requests wait in the network, outside of WebSphere Application Server. This type of configuration adds stability, because no component is overloaded. The Edge Server Components can be used to direct waiting users to other servers in a WebSphere Application Server cluster.

If resources are more readily available on the application server or database server, it may be appropriate to tune so that each request from the Web server has an available application server thread, and every application server thread has an available database connection. The need for this type of configuration depends on the application and overall site design.

Determining optimum queue sizes

A simple way to determine the right queue size for any component is to perform a number of load runs against the application server environment at a time when the queues are very large, ensuring maximum concurrency through the system.

One approach would be as follows:

- Set the queue sizes for the Web server, Web container, and data source to an initial value, for example, 100.
- Simulate a large number of typical user interactions entered by concurrent users in an attempt to fully load the WebSphere environment. In this context, "concurrent users" means simultaneously active users that send a request, wait for the response, and immediately resend a new request upon response reception without think time.
- Measure overall throughput and determine at what point the system capabilities are fully stressed (the saturation point).
- Repeat the process, each time increasing the user load. After each run, record the throughput (requests per second) and response times (seconds per request) and plot the throughput curve.

The throughput of WebSphere Application Server is a function of the number of concurrent requests present in the total system. At some load point, congestion will start to develop due to a bottleneck and throughput will increase at a much lower rate until reaching a saturation point (maximum throughput value). The throughput curve should help you identify this load point.

It is desirable to reach the saturation point by driving CPU utilization close to 100%, since this gives an indication that a bottleneck is not caused by something in the application. If the saturation point occurs before system utilization reaches 100%, there is likely another bottleneck that is being aggravated by the application. For example, the application might be creating Java objects causing excessive garbage collection mark phase bottlenecks in Java that you may notice as only one processor being utilized at a time on multi-processor systems. On uniprocessor systems, you will not be able to see the symptom, but only the problems it is causing.

The most manageable type of bottleneck occurs when the CPUs of the servers become fully utilized. This type of bottleneck can be fixed by adding additional or more powerful CPUs.

Determining the maximum concurrency point

The number of concurrent users at the throughput saturation point represents the maximum concurrency of the application. For example, if the application saturated the application server at 50 users, 48 users might give the best combination of throughput and response time. This value is called the Max Application Concurrency value. Max Application Concurrency becomes the preferred value for adjusting the WebSphere Application Server system queues. Remember, it is desirable for most users to wait in the network; therefore, queue sizes should decrease when moving downstream farther from the client.

For example, given a Max Application Concurrency value of 48, start with system queues at the following values: Web server 75, Web container 50, and data source 45. Perform a set of additional experiments by adjusting these values slightly higher and lower to find the best settings

Adjusting queue settings for access patterns

In many cases, only a fraction of the requests passing through one queue enters the next queue downstream. In a site with many static pages, many requests are fulfilled at the Web server and are not passed to the Web container. In this circumstance, the Web server queue can be significantly larger than the Web container queue. In the previous section, the Web server queue was set to 75 rather than closer to the value of Max Application Concurrency. Similar adjustments need to be made when different components have different execution times. As the percentage of static content decreases, however, a significant gap in the Web server queue and the application server queue can create poorly performing sites overall. Remember that tuning is an art, not a science, and different Web sites have different requirements.

For example, in an application that spends 90% of its time in a complex servlet and only 10% making a short JDBC™ query, on average 10% of the servlets are using database connections at any time, so the database connection queue can be significantly smaller than the Web container queue. Conversely, if much of a servlet execution time is spent making a complex query to a database, consider increasing the queue values at both the Web container and the data source. Always monitor the CPU and memory utilization for both the WebSphere Application Server and the database servers to ensure the CPU or memory are not being overutilized.

Configuring the queues

Within WebSphere Application Server, the queues are represented as pooled resources, for example, thread pools or database connection pools. Pool settings determine the maximum concurrency level of a resource. This section describes how the different queues are represented in WebSphere and their settings.

Data source

There are two settings to be concerned with for determining data source queues:

- Connection pool size
- Prepared statement cache size

Connection pool size

When accessing any database, the initial database connection is an expensive operation. WebSphere Application Server supports JDBC 2.0 Standard Extension APIs to provide support for connection pooling and connection reuse.

The connection pool is used for direct JDBC calls within the application, as well as for enterprise beans using the database.

Tivoli Performance Viewer can help find the optimal size for the connection pool. Use a standard workload that represents a typical number of incoming client requests, use a fixed number of iterations, and use a standard set of configuration settings. Watch the Pool Size, Percent Used, and Concurrent Waiters counters of the data source entry under the JDBC Connection Pools module. The optimal value for the pool size is that which reduces the values for these monitored counters. If Percent Used is consistently low, consider decreasing the number of connections in the pool.

Better performance is generally achieved if the value for the connection pool size is set lower than the value for the Max Connections in the Web container. Lower settings for the connection pool size (10-30 connections) typically perform better than higher (more than 100) settings. On UNIX® platforms, a separate DB2 process is created for each connection. These processes quickly affect performance on systems with low memory, causing errors.

Each entity bean transaction requires an additional connection to the database specifically to handle the transaction. Be sure to take this into account when calculating the number of data source connections.

The connection pool size is set from the Administrative Console using these steps:

1. Select **Resources** → **JDBC Providers** in the console navigation tree.
2. Select the appropriate scope (cell, node or server), depending on your configuration.
3. Open the JDBC provider configuration by clicking the name of the provider.
4. Select the **Data Sources** entry under Additional Properties.
5. Open the data source configuration by clicking the data source name.
6. Select **Connection pool properties**.
7. Use the Minimum connections and Maximum connections fields to configure the pool size. The default values are 1 for min connections and 10 for max connections.
8. Save the configuration and restart the affected application servers for the changes to take effect.

Deadlock can occur if the application requires more than one concurrent connection per thread, and the database connection pool is not large enough for the number of threads. Suppose each of the application threads requires two concurrent database connections and the number of threads is equal to the maximum connection pool size. Deadlock can occur when both of the following are true:

- Each thread has its first database connection, and all are in use.
- Each thread is waiting for a second database connection, and none would become available, since all threads are blocked.

To prevent deadlock in this case, the value set for the database connection pool must be at least one higher than the number of waiting threads in order to have at least one thread complete its second database connection.

To avoid deadlock, code the application to use, at most, one connection per thread. If the application is coded to require C concurrent database connections per thread, the connection pool must support at least the following number of connections, where T is the maximum number of threads:

T * (C - 1) + 1

The connection pool settings are directly related to the number of connections that the database server is configured to support. If the maximum number of connections in the pool is raised, and the corresponding settings in the database are not raised, the application fails and SQL exception errors are displayed in the SystemErr.log file.

Prepared statement cache size

The data source optimizes the processing of prepared statements to help make SQL statements process faster. It is important to configure the cache size of the data source to gain optimal statement execution efficiency. A prepared statement is a precompiled SQL statement that is stored in a prepared statement object. This object is used to efficiently execute the given SQL statement multiple times. If the JDBC driver specified in the data source supports precompilation, the creation of the prepared statement will send the statement to the database for precompilation. Some drivers might not support precompilation and the prepared statement might not be sent until the prepared statement is executed.

If the cache is not large enough, useful entries will be discarded to make room for new entries. In general, the more prepared statements your application has, the larger the cache should be. For example, if the application has five SQL statements, set the prepared statement cache size to 5, so that each connection has five statements.

Tivoli Performance Viewer can help tune this setting to minimize cache discards. Use a standard workload that represents a typical number of incoming client requests, use a fixed number of iterations, and use a standard set of configuration settings. Watch the PrepStmtCacheDiscardCount counter of the JDBC Connection Pools module. The optimal value for the statement cache size is the setting used to get either a value of zero or the lowest value for PrepStmtCacheDiscardCount.

As with the connection pool size, the statement cache size setting requires resources at the database server. Specifying too large a cache could have an impact on database server performance. We highly recommend that you consult your database administrator to determine the best setting for the prepared statement cache size.

Note: The statement cache size setting defines the maximum number of prepared statements cached per connection.

The cache size is set from the Administrative Console using these steps:

1. Select **Resources** → **JDBC Provider** in the console navigation tree.
2. Select the appropriate scope (cell, node, or server), depending on your configuration.
3. Open the JDBC provider configuration by clicking the name of the provider.
4. Select the **Data Sources** entry in the Additional Properties window.
5. Open the data source configuration by clicking the data source name.
6. Select **WebSphere Application Server data source properties**.
7. Use the Statement cache size field to configure the total cache size.
8. Save the configuration and restart the affected application servers for the change to take effect.

EJB container

You can use the following parameters to make adjustments that improve performance for the EJB container.

Cache settings (cache size and cleanup interval)

To determine the cache absolute limit, multiply the number of enterprise beans active in any given transaction by the total number of concurrent transactions expected. Then, add the number of active session bean instances. Use the Tivoli Performance Viewer to view bean performance information. The cache settings consist of two parameters: the cache size and the cleanup interval.

The cleanup interval specifies the interval at which the container attempts to remove unused items from the cache in order to reduce the total number of items to the value of the cache size.

The cache size specifies the number of buckets in the active instance list within the EJB container.

To change these settings, click **Servers** → **Application servers** → **<AppServer_Name>** → **EJB Container Settings** → **EJB container** → **EJB cache settings**.

The default values are Cache size=2053 buckets and Cache cleanup interval=3000 milliseconds.

ORB thread pool size

Method invocations to enterprise beans are only queued for requests coming from remote clients going through the RMI activity service. An example of such a client is an EJB client running in a separate Java Virtual Machine (another address space) from the enterprise bean. In contrast, no queuing occurs if the EJB client (either a servlet or another enterprise bean) is installed in the same JVM that the EJB method runs on and the same thread of execution as the EJB client.

Remote enterprise beans communicate by using the RMI/IIOP protocol. Method invocations initiated over RMI/IIOP are processed by a server-side ORB. The thread pool acts as a queue for incoming requests. However, if a remote method request is issued and there are no more available threads in the thread pool, a new thread is created. After the method request completes, the thread is destroyed. Therefore, when the ORB is used to process remote method requests, the EJB container is an open queue, due to the use of unbounded threads.

The degree to which the ORB thread pool value needs to be increased is a function of the number of simultaneous servlets (that is, clients) calling enterprise beans and the duration of each method call. If the method calls are longer or the applications spend a lot of time in the ORB, consider making the ORB thread pool size equal to the Web container size. If the servlet makes only short-lived or quick calls to the ORB, servlets can potentially reuse the same ORB thread. In this case, the ORB thread pool can be small, perhaps even one-half of the thread pool size setting of the Web container.

The ORB thread pool size is configured from the Administrative Console using these steps:

1. To change these settings, click **EJB cache settings**.

2. Select **Servers** → **Application servers** → **<AppServer_Name>** → **Container Services**.
3. Select **ORB Service** → **Thread Pool**.
4. Use the Maximum Size field to configure the maximum pool size. Note that this only affects the number of threads held in the pool (the actual number of ORB threads can be higher).
5. Save the configuration and restart the affected application server for the change to take effect.

Web container

To route servlet requests from the Web server to the Web containers, a transport connection between the Web server plug-in and each Web container is established. The Web container manages these connections through transport channels and assigns each request to a thread from the Web container thread pool.

Thread pool

The Web container maintains a thread pool to process inbound requests for resources in the container (that is, servlets and JSPs).

Tivoli Performance Viewer can help tune the Web container thread pool size settings. Use a standard workload that represents a typical number of incoming client requests, use a fixed number of iterations, and use a standard set of configuration settings. Watch the PercentMaxed and ActiveCount counters of the Thread Pools module. If the value of the PercentMaxed counter is consistently in the double digits, then the Web container could be a bottleneck and the number of threads should be increased. On the other hand, if the number of active threads are significantly lower than the number of threads in the pool, consider lowering the thread pool size for a performance gain.

The Web container thread pool size is configured from the Administrative Console using these steps:

1. Select **Servers** → **Application servers** → **<AppServer_Name**>.
2. Select the **Thread Pools** entry under Additional Properties.
3. Select the **WebContainer** entry in the thread pools list of the workspace.
4. Use the Maximum Size field to configure the maximum pool size. Note that in contrast to the ORB, the Web container only uses threads from the pool, hence a closed queue. The default value is 50.
5. Save the configuration and restart the affected application server for the change to take effect.

Important: Checking the **Allow thread allocation beyond maximum thread size** box on the Thread Pool Configuration page allows for an automatic increase of the number of threads beyond the maximum size configured for the thread pool. As a result of this, the system can become overloaded because too many threads are allocated.

HTTP transport channel maximum persistent requests

The maximum persistent requests is the maximum number of requests allowed on a single keep-alive connection. This parameter can help prevent denial of service attacks when a client tries to hold on to a keep-alive connection. The Web server plug-in keeps connections open to the application server as long as it can, providing optimum performance.

A good starting value for the maximum number of requests allowed is 100 (which is the default value). If the application server requests are received from the Web server plug-in only, increase this parameter's value.

The maximum number of requests allowed is configured from the Administrative Console using these steps:

1. Select **Servers** → **Application servers** → **<AppServer_Name>**.
2. Select **Web Container Settings** → **Web container transport chains** under Container Settings.
3. Select the transport chain you want to modify, for example, WCInboundDefault.
4. Select **HTTP Inbound Channel (HTTP #)** in the Transport Channels window.
5. Enter a value in the Maximum persistent requests field. Click **OK**.
6. Save the configuration and restart the affected application server for the change to take effect.

HTTP transport channel read timeout

Specifies the amount of time, in seconds, the HTTP transport channel waits for a read request to complete on a socket after the first read request occurs. The read being waited for could be an HTTP body (such as a POST) or part of the headers if they were not all read as part of the first read request on the socket.

The read timeout is configured from the Administrative Console using these steps:

1. Select **Servers** → **Application servers** → **<AppServer_Name>**.
2. Select **Web Container Settings** → **Web container transport chains** under Container Settings.

3. Select the transport chain you want to modify.
4. Select **HTTP Inbound Channel (HTTP #)** in the Transport Channels pane.
5. Enter a value in the Read timeout field (the default starting value is 60). Click **OK**.
6. Save the configuration and restart the affected application server for the change to take effect.

HTTP request threads

Application servers use threads to listen for and process requests from the browser. A thread represents a work item or task. Application servers (WebSphere Application Server and BEA WebLogic) have default settings for number of threads.

WebSphere Application Server has a default minimum thread size of 5. WebLogic has a default thread count of 25 (when started in production mode).

You can, and often should, revise these values to improve system performance. To optimize performance, you want to try to determine the right number of threads.

The correct number of threads

If the thread count is too high, you can have too many simultaneous requests to the CPU. Servicing the requests becomes inefficient If the thread count is too low, because the processor is not fully utilized.

The queue grows and work backs up. The correct number of threads varies from site to site. The number depends on the volume of front-end transactions (user load) and back-end transactions (nonuser load).

Use the WebSphere and WebLogic application server monitoring features to set and monitor thread usage and queue times for your application servers. An optimal number of threads can often be found in the 40 to 55 range.

Managing integration loads

If you are using the Service Request Manager Enterprise Adapter, your system has front-end users plus potentially heavy resource use from integration.

Integration volume can affect the user experience of system performance. One way to address potential performance degradation is to control resources used by integration. To do this, you can limit the number of message-driven beans (MDBs) on the continuous queue of the integration framework. (Another way to address performance is to separate inbound integration traffic to another server or cluster.)

A default installation of Service Request Manager does not have a specific number of MDBs in the deployment descriptor file. The number of MDBs that gets created is managed by the container. Depending on the load on your system, you might need to reset this number. It is generally best to start off with a small number, such as two MDBs. For a small or medium-sized implementation, two or three MDBs usually yields acceptable performance. As the system load increases, you might need to gradually increase the number of MDBs. You generally should not exceed ten MDBs. More than ten MDBs does not necessarily yield higher throughput and it can significantly increase resource usage (CPU cycles in particular).

Test different numbers of MDBs in a development setting to determine an appropriate number before you establish the number of MDBs for your production system.

Load balancing

Load balancing is the distribution of the task load across multiple instances of an application. User load comes from logged-in Service Request Manager users. Nonuser load comes from such things as scheduled jobs (cron tasks) and Service Request Manager Enterprise Adapter incoming transactions. Distribute user load and nonuser load to different application servers or clusters. For user load, software load balancers and hardware load balancers are available. A software load balancer is included with the Tivoli Service Request Manager system.You can use a commercially available hardware device for hardware load balancing.

For more details and information about WebSphere Application Performance fine tuning, please refer to the following Web site:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/welc6toptuning.html`

6.3.2 Database Server and SQL fine tuning

Universal Database environments range from stand-alone systems to complex combinations of database servers and clients running on multiple platforms. Critical to all these environments is the achievement of adequate performance to meet business requirements. Performance is typically measured in terms of response time, throughput, and availability.

The performance of any system is dependent upon many factors, including system hardware and software configuration, number of concurrent users, types of users, and the application workload. Performance management is a complex issue, and can be defined as modifying the system and application environment in order to satisfy previously defined performance objectives. These objectives must be quantitative, measurable, and realistic. Units of measurement include response time for a given workload, transactions per second, I/O operations, CPU use, or a combination of the above. Without well defined performance objectives, performance is a hit or miss exercise, with no way of delivering on any service level agreements that may be negotiated with users.

Performance management is an iterative process, that involves constant monitoring to determine whether performance objectives are being met even as environments and workloads change over time. When performance objectives are not being met, then appropriate changes must be made to the hardware or software environments, as well as the performance objectives themselves, in order to ensure that they will be met.

From a database perspective, performance problems can arise out of a combination of poor application and system design, inadequate CPU, memory disk, and network resources, and suboptimal tuning of these resources. In this chapter, we will talk about information for improving system performance by focusing on two major areas:

- Database tuning
- User queries and SQL tuning

Database performance tuning can dramatically affect the throughput of your application. For example, if your application requires high concurrency (multiple, simultaneous interactions with back-end data), an improperly tuned database can result in a bottleneck. Database access threads accumulate in a backlog when the database is not configured to accept a sufficient number of incoming requests.

Tivoli Service Request Manager database tuning

The database is central to Service Request Manager functionality. The database stores all data that is collected and calculated by the Service Request Manager application. The database also stores metadata for configuring and maintaining the environment.

The database server processes all transactions from the Service Request Manager application. The integrated reporting function accesses the data in the database to generate documents, such as work orders and purchase orders. Reporting also generates resource-intensive management reports.

Because all functionality is based on database performance, the database should be a key focus for performance tuning.

You can apply standard database-tuning techniques to Service Request Manager. Periodically monitor a production Service Request Manager database during peak load. You can use any appropriate monitoring tools or techniques. If necessary, adjust parameters to resolve bottlenecks that are reported by the monitoring tools.

Here we will talk about the default database for IBM Service Request Manager, which is DB2. DB2 has many parameters that can be configured to optimize database performance.

The rest of this section shows the various DB2 fine tuning configuration that would improve the overall performance of the Tivoli Service Request Manager system.

DB2 logging

DB2 has corresponding log files for each database. Performance improvements can be gained by setting the log files on a different hard drive from the database files.

DB2 Configuration Advisor

Located in the DB2 Control Center, this advisor calculates and displays recommended values for the DB2 buffer pool size, the database, and database manager configuration parameters, with the option of applying these values. Refer to the online help facility within the Control Center for more information about the advisor.

Use TCP sockets for DB2 on Linux

On Linux platforms, whether the DB2 server resides on a local machine with WebSphere Application Server or on a remote machine, configure the DB2 application databases to use TCP sockets for communication with the database.

The directions for configuring DB2 on Linux can be found in the WebSphere Application Server installation documentation for the various operating systems. This document specifies setting DB2COMM for TCP/IP and the corresponding changes required in the /etc/services file.

The default is to use shared memory for local databases, but we recommend that you change the specification for the DB2 application databases and for any session databases from shared memory to TCP sockets.

DB2 MaxAppls and DB2 MaxAgents

When configuring the data source settings for the databases, confirm the DB2 MaxAppls setting is greater than the maximum number of connections for the data source. If you are planning to use multiple cluster members, set the

MaxAppls value as the maximum number of connections multiplied by the number of cluster members.

The same relationship applies to the session manager number of connections. The MaxAppls setting must be equal to or greater than the number of connections. If you are using the same database for session and data sources, set the MaxAppls value as the sum of the number of connection settings for the session manager and the data sources.

DB2 buffpage

Buffpage is a database configuration parameter. A buffer pool is a memory storage area where database pages containing table rows or index entries are temporarily read and changed. The purpose of the buffer pool is to improve database system performance. Data can be accessed much faster from memory than from disk.

The goal of buffer pool tuning is to help DB2 make the best possible use of the memory available for buffers. The overall buffer size has a great effect on DB2 performance, since a large number of pages can significantly reduce I/O, which is the most time consuming operation. However, if the total buffer size is too large, and there is not enough storage to allocate them, then a minimum buffer pool for each page size will be allocated, and performance will be sharply reduced. To calculate the maximum buffer size, all other storage utilization must be considered by DB2 as well as the operating system and any other applications. Once the total available size is determined, this area can be divided into different buffer pools to improve utilization. If there are table spaces with different page sizes, then there must be at least one buffer pool per page size.

DB2 query optimization level

When a database query is executed in DB2, various methods are used to calculate the most efficient access plan. The query optimization level parameter sets the amount of work and resources that DB2 puts into optimizing the access plan. The range is from zero to 9. An optimization level of 9 causes DB2 to devote a lot of time and all of its available statistics to optimizing the access plan.

The optimization level is set on individual databases and can be set with either the command line or with the DB2 Control Center. Static SQL statements use the optimization level specified by the **`prep`** and **`bind`** commands. If the optimization level is not specified, DB2 uses the default optimization, as specified by the dft_queryopt parameter. Dynamic SQL statements use the optimization class specified by the current query optimization special register, which is set using the SQL Set statement. For example, the following statement sets the optimization class to 1:

```
Set current query optimization = 1
```

If the current query optimization register has not been set, dynamic statements will be bound using the default query optimization class.

The default value is 5. We recommend that you set the optimization level for the needs of the application. High levels should only be used when there are very complicated queries.

DB2 reorgchk

The performance of the SQL statements can be impaired after many updates, deletes, or inserts have been made. Performance can be improved by obtaining the current statistics for the data and rebinding.

Use the following DB2 command to issue runstats on all user and system tables for the database you are currently connected to:

```
reorgchk update statistics on table all
```

You should then rebind packages using the **bind** command.

In order to see if runstats has been done, issue the following command on DB2 CLP:

```
db2 -v "select tbname, nleaf, nlevels, stats_time from
sysibm.sysindexes"
```

If no runstats has been done, nleaf and nlevels will be filled with -1 and stats_time will have an empty entry “-”. If runstats was done already, the real-time stamp when the runstats was completed will also be displayed under stats_time. If you think the time shown for the previous runstats is too old, execute runstats again.

DB2 MinCommit

This parameter allows delayed writing of log records to a disk until a minimum number of commits have been performed, reducing the database manager impact associated with writing log records. For example, if MinCommit is set to 2, a second commit would cause output to the transaction log for the first and second commits. The exception occurs when a one-second timeout forces the first commit to be output if a second commit does not come along within one second. In test applications, up to 90% of the disk input and output was related to the DB2 transaction log. Changing MinCommit from 1 to 2 reduced the results to 45%.

Try to adjust this parameter if the disk input/output wait is more than 5% and there is DB2 transaction log activity from multiple sources. When a lot of activity occurs from multiple sources, it is less likely that a single commit will have to wait for another commit (or the one-second timeout).

Do not adjust this parameter if you have an application with a single thread performing a series of commits (each commit could hit the one-second delay).

To view the current value for a particular database, follow these steps:

1. Issue the DB2 command **`get db cfg for <dbname>`** (where <dbname> is the name of the application database) to list the database configuration parameters.
2. Look for “Group commit count (MINCOMMIT)”.
3. Set a new value by issuing the DB2 command **`update db cfg for <dbname> using mincommit n`** (where n is a value between 1 and 25 inclusive). The new setting takes effect immediately.

The following are several metrics that are related to DB2 MinCommit:

- The disk input/output wait can be observed on AIX or any other UNIX based operating system with the command **`vmstat 5`**.

  This shows statistics every 5 seconds. Look for the wa column under the CPU area.

- The percentage of time a disk is active can be observed on AIX with the command **`iostat 5`**.

  This shows statistics every 5 seconds. Look for the %tm_act column.

- The DB2 command **`get snapshot for db on <dbname>`** (where <dbname> is the name of the application database) shows counters for log pages read and log pages written. The default value is 1. We recommend that you set MinCommit to 1 or 2 (if the circumstance permits).

Indexing

Indexing a database requires a good understanding of the data, user functions, and how databases use indexes. Indexes use key parts of data from a table in a binary structure to enhance searching capability. Each record of data in the table must have associated data in the index.

Indexing can greatly increase search speeds. However, a drawback of indexes is that for each insert, update, or delete, the index must also be updated. Database administrators often apply many indexes to a table to enhance searching, and then find that other activities have slowed. You should review indexes to ensure that you have the right balance for searching and updating tables.

Some special index types are available on each database platform that are not available in the Service Request Manager Database Configuration application. These index types can be created and maintained from the back end, and they can improve performance in specific cases. For example, on Oracle you might create a bitmap index or a function-based index if you determine that these indexes would improve certain queries.

If you use special index types, the Service Request Manager system administrator must remember to remove any special indexes before configuring database changes. After the database is configured, the special indexes must be replaced.

Statistics

Keep index statistics up-to-date. In the Service Request Manager Database Configure application, you can use a Select Action menu action that does this. You can also update index statistics from the back end.

With Service Request Manager Release 6, the Update Statistics action is enhanced to function on Oracle by running the Service Request Manager_GATHER_TABLE_STATS procedure.

Customization

Customizing Service Request Manager can change the way information is selected from the database. Some customizations include additional tables and columns. If you have customized Service Request Manager, you should carefully compare indexes to the user functions that use them. Ensure that you implemented the right balance of indexes.

Load testing

If possible, do load testing during the implementation phase to expose performance problems before you put Service Request Manager into production. If you have the equipment to perform load testing, you can use load testing after Service Request Manager is in production to determine if there is any performance impact from patches or from data growth over time.

For more details on performance fine tuning with the DB2 UDB database, please refer to *DB2 UDB/WebSphere Performance Tuning Guide*, SG24-6417, which can be found at `http://www.redbooks.ibm.com/redbooks/pdfs/sg246417.pdf`.

Oracle database note:

On Oracle, you should specify the following two initialization parameters:

- Set the Cursor_sharing parameter to SIMILAR or FORCE so that the user entered literal values are converted to bind variables.
- Set the Nls_length_semantics parameter to CHAR when the database character set is a double-byte or unicode character set.

SQL server database note:

Microsoft SQL Server poses some challenges in environments like Service Request Manager that have multiple concurrent-user transactions. When records are selected, SQL Server escalates locks from “record” to “page” to “table,” based on an internal algorithm. If user queries are not efficient, database locks can impede other users who are accessing the system.

Some customers running Service Request Manager on SQL Server have improved database performance by turning on the SQL Server row-level locking that is supported by Microsoft. Turning on row-level locking reduces the locked areas of the database. But note that row-level locking also can result in a higher number of locks, because locks are based on record instead of memory page.

User queries and SQL tuning

SQL is a high-level language that provides considerable flexibility in writing queries to deliver the same answer set. However, not all forms of the SQL statement deliver the same performance for a given query. It is therefore vital to ensure that the SQL statement is written in a fashion to provide optimal performance.

DB2 UDB, Oracle, and Microsoft SQL provides the SQL compiler that creates the compiled form of SQL statements. When the SQL compiler compiles SQL statements, it rewrites them into a form that can be optimized more easily. This is known as *query rewrite*. The SQL compiler then generates many alternative execution plans for satisfying the user’s request. It estimates the execution cost of each alternative plan using the statistics for tables, indexes, columns, and functions, and chooses the plan with the smallest execution cost. This is known as *query optimization*.

It is important to note that the SQL compiler must choose an access plan that will produce the result set for a given query. We recommend the following guidelines to ensure that the SQL compiler chooses the optimal access plan for a given query:

Most of the WHERE clauses in Service Request Manager queries are generated by individual users on the List tabs of Service Request Manager applications. This powerful feature of Service Request Manager can produce a lot of inefficient SQL. You can improve the ease of use and convenience for users by setting the appropriate search types for database columns.

Using appropriate search types also reduces the load on the database Appropriate Search Type options. You can change the default search type of WILDCARD to TEXT or EXACT. TEXT and EXACT searches can use indexes. You specify the search type for a database column in the Database Configuration application. You also can change the search type for groups of columns.

EXACT search types

When a user does a search, the default method of searching on many Service Request Manager database fields is to use wildcards (SEARCHTYPE = WILDCARD). The WILDCARD search type causes Service Request Manager to construct a condition of the form column, such as '%value%', when the user enters “value” in a field on the List tab. In wildcard searching, the database engine cannot use indexes. Searching without indexes can result in slower search times, especially on tables with a large number of rows. You can specify a search type of EXACT when word searching is not needed.

For example, key fields (such as Work Order and Purchase Order) and value list fields (such as Work Order Status) can benefit from the indexing that is used in EXACT searches. EXACT searches use wildcarding only if a user explicitly enters wildcard characters on the List tab or in the WHERE clause.

TEXT search types

Most Service Request Manager tables have one or more longer character columns for descriptions, memos, or remarks. You can provide a search type of TEXT, and a corresponding Oracle Text index or SQL Server full text catalog, for columns that have a lot of text. (Full text indexing is not available with Service Request Manager on IBM DB2.) Text indexing puts some load on the database because of the constant need for background processing to keep the indexes synchronized. However, text indexing produces efficient word searching of description fields.

Specify a search type of TEXT on description fields that word searching is required for. Use TEXT on description fields of tables with large numbers of rows (tens of thousands, for example).

The text search engine takes some time to refresh the indexes, so new records might not be found until the text search index refreshes itself. On Oracle, you can modify the procedure Service Request Manager_ts_job_call to change the schedule of the synchronization process to any interval that you want.

On SQL Server, you can set and modify the population schedule for the Full Text Catalog. Use SQL Server 2000 Enterprise Manager or SQL Server 2005 Management Studio.

Wildcard search type

Tables with fewer than 2000 of an Oracle Text index or SQL Server full text catalog, or for columns that have a lot of text (full text indexing is not available with Service Request Manager on IBM DB2) puts load on the database because of the constant need for background processing to keep the indexes synchronized. However, text indexing produces efficient word searching of description fields.

Specify a search type of TEXT on description fields that word searching is required for. Use TEXT on description fields of tables with large numbers of rows (tens of thousands, for example).

The text search engine takes some time to refresh the indexes, so new records might not be found until the text search index refreshes itself.

On Oracle, you can modify the procedure Service Request Manager_ts_job_call to change the schedule of the synchronization process to any interval that you want.

On SQL Server, you can set and modify the population schedule for the Full Text Catalog. Use SQL Server 2000 Enterprise Manager or SQL Server 2005 Management Studio.

You can use the default search type wildcard on description fields of tables that have relatively small number of rows. You can use Oracle Text index or SQL Server full text catalog for columns that have a lot of text. (Full text indexing is not available with Service Request Manager on IBM DB2.) Text indexing puts some load on the database because of the constant need for background processing to keep the indexes synchronized. However, text indexing produces efficient word searching of description fields.

Specify a search type of TEXT on description fields that word searching is required for. Use TEXT on description fields of tables with large numbers of rows (tens of thousands, for example).

The text search engine takes some time to refresh the indexes, so new records might not be found until the text search index refreshes itself.

On Oracle, you can modify the procedure Service Request Manager_ts_job_call to change the schedule of the synchronization process to any interval that you want. On SQL Server, you can set and modify the population schedule for the Full Text Catalog. Use SQL Server 2000 Enterprise Manager or SQL Server 2005 Management Studio.

NONE search type

The NONE search type prevents a column from being used in a WHERE clause, and still allows the display of the returned values on the List tab under “Finding what users would query”.

Service Request Manager users usually have a well-defined set of columns that they want to query in each application. You can identify these columns in user team interviews during implementation, or later, by examining reports of slow-running SELECT statements. You can then index these columns to improve system performance.

Users can create and save their own queries, and can also share queries with other users. Saved queries are stored in a table named QUERY. You should periodically review these saved queries for inefficient conditions and use of unindexed columns.

Someone with SQL expertise can help create special-purpose queries (for example, return all PM work orders created since Monday of this week). Doing so can save users the effort of querying larger sets and sorting and scrolling through them. You also can set up users with an efficient default query for their most-used applications so that they see their preferred record set when they enter the application. For example, in Work Order Tracking, you might specify a default query for supervisor Mike so that he would initially see only work orders with Mike in the Supervisor field.

Restrict querying in application

You can control or restrict user access to query features and user ability to query on specific columns by using a combination of methods:

- Application Designer
- Application cloning
- Security groups

Application Designer

You can use the Application Designer to customize an application by adding or removing columns from the List tab. You can then ensure that the columns to be queried are all indexed.

Application cloning

You can clone an application and then use the Application Designer to create an alternate version of an application that has a restricted number of columns that can be queried.

Security groups

After you clone applications, you can use security groups to assign users to specific application clones. You can also use security groups to prohibit access to the More Search Fields and Where Clause advanced query options. By prohibiting access to those options, you limit users to querying on the List tab of the application.

Key performance indicators

Key performance indicators (KPIs) display the state of systems and processes in Service Request Manager. Because KPIs can be user-defined, you should monitor them for efficiency.

Live KPI on start centers

When a KPI is defined on a Start Center, you have a choice of how to retrieve information. You can run an immediate query to get the information, or you can retrieve the information from a table that is updated by a cron task.

The immediate query runs every time the Start Center opens. This approach can cause a long delay in opening the Start Center, and it puts a load on the database.

KPI query

You should periodically review queries that are written for KPIs. Check for SQL efficiency and index usage.

KPI cron task frequency

How up-to-date does your KPI information need to be? Five minute intervals might sound like a good idea initially, but 60 minute or 120 minute intervals might be as useful to the people who want to see the information. Longer KPI cron task intervals reduce the load on the database from KPIs and can improve system performance. In general, use longer rather than shorter KPI cron task frequency intervals when the value of the data to the user is essentially the same at longer intervals.

KPI best practices

Set all KPIs to retrieve their data from the KPI table. Use cron tasks to run KPI queries at reasonable intervals.

Escalations

Escalations are, in effect, batch users that do not have the impact of a user interface. But escalations are very resource intensive whenever they do something.

Efficiency and frequency

An escalation selects a set of records and a list of one or more actions in the result set. The column in the WHERE clause of the selection should be efficiently indexed. Set the frequency or schedule of escalation accordance to its importance:

- An escalation that dispatches emergency work orders for critical safety issues might warrant a frequency of five minutes.
- An escalation that ensures that service requests from executives that must be dealt with promptly might need to run every fifteen minutes.
- An escalation that closes work orders that were completed 90 or more days ago might need to run only once a week at an off-peak time.

Report

Review custom reports for efficient SQL and use of indexes. Most reports receive a WHERE clause from Service Request Manager. For all these cases, improving the efficiency of user queries also improves the efficiency of reports.

6.3.3 Network and bandwidth fine tuning

In this section, we discuss the approaches to improve system performance by focusing on various network and bandwidth issues.

Service Request Manager in a network

Clients connect to the Service Request Manager application over the network. The application also communicates with its various parts (application server, database, and report server) over the network. If any segment of the network performs poorly, the user experiences a system that is slow and hard to navigate.

Service Request Manager is a Web-based product that operates on a request and response basis. If the requests and responses are delivered slowly, Service Request Manager itself has no control over response time.

Optimum network configurations for Service Request Manager should include the ability to produce 50 ms or faster round-trip packet response between the client and the server. The system needs enough bandwidth to support 6 KBps per user. Users might begin to experience performance degradation if the network does not operate within these parameters.

Windows terminal server or Citrix

You can use network caching, acceleration, and compression utilities to improve network performance.

Other options for resolving low bandwidth or high-latency network performance issues include services such as Windows Terminal Server and Citrix. These services can help provide maximum performance between the Windows Terminal Server or Citrix client and the Tivoli Service Request Manager server with a minimum of traffic between the Windows Terminal Server or Citrix server and the user.

A benefit of running Tivoli Service Request Manager through Citrix is that Citrix traffic is treated as business traffic. In some customer environments, business traffic can take priority over non-business traffic.

Some Tivoli Service Request Manager customers have found considerable improvements in network performance when they use Citrix or Windows Terminal Server.

Image and JavaScript browser caching

You can enable a browser file caching filter on the application server. This lets the browser store images, CSS files, and JavaScript™ files locally. Storing images and files locally benefits performance in two ways:

- Images and files do not need to be constantly requested from the server.
- Because images are not downloaded as often, less bandwidth is required.

Browser file caching is not enabled by default. *We recommend that you enable browser file caching.* For information about how to enable this feature, see the following Tivoli Service Request Manager Support Web page:

`http://www-1.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&q1=cache&uid=swg21262009&loc=en_US&cs=utf-8&lang=en`

For Microsoft Internet Explorer 7, set the number of days until refresh is enabled to 32 days or longer.

Use QoS guarantees for Tivoli Service Request Manager traffic

You can prioritize Tivoli Service Request Manager traffic over standard Web traffic. Contact your network administrator for details about prioritizing Tivoli Service Request Manager traffic.

Network appliances

Many customers use network appliances, such as those provided by Juniper® Networks, RiverBed, and Expand Networks, among other vendors.

Network appliances can help compress data and optimize bandwidth. Customers report that network appliances can prove very beneficial to system performance, especially in a high-latency environment.

6.3.4 Client workstation configuration

Some Tivoli Service Request Manager customers report that client workstation configuration is initially the most important area to focus on when users report performance issues.

If you provide users with adequate workstations that have sufficient memory, you will notice performance improvement.

Configuration recommendations

Consider using the following configuration for client workstations:

- Processor: Intel Pentium® or Xeon processor.
- Memory: At least 1 to 2 GB of RAM.

  Although 512 MB of RAM is a published recommendation, many customer sites find that this is insufficient for robust performance in an enterprise setting. Some customers report that users who had performance issues with 512 MB or 1 GB of RAM have fewer or no issues after increasing RAM.
- Monitor screen resolution for mobile computer: 1024 x 768.
- Desktop screen resolution: 1280 x 1024.
- Operating system: Microsoft Windows XP.
- Browser: Microsoft Internet Explorer 6 or 7. Use Internet Explorer 7 if the Tivoli Service Request Manager release supports Internet Explorer 7. Some customers report noticeable performance improvements with Internet Explorer 7.
- File reader: Adobe® Reader 6 or higher to read the PDF files.

Workstation configuration

Install antivirus software on all client workstations. Scanning for viruses can affect system performance. If possible, schedule virus scans during low usage periods using scheduled cron tasks or jobs.

A cron job is a task scheduled to run at a specified frequency, for example, every two minutes, every five hours, or every 48 hours. The Reorder routine and PM generation are examples of Tivoli Service Request Manager cron tasks. Cron tasks usually run as background tasks and no user response or action would be required for the completion of the task or job.

When there are large numbers of cron tasks running concurrently, there are possible chances for an user load, and in turn the user response time might decline. To improve response time for a user, you can set up cron tasks to run outside the user environment.

Cron jobs or tasks can also take advantage of clustering. You can run cron tasks on multiple separate computers to improve throughput.

6.3.5 Reporting configuration

In this section, we will talk about the best practices in reporting. Running reports is a very resource-intensive operation and has the potential to be one of the biggest factors in poor system performance.

To help keep reports from affecting your system performance, you should isolate the reporting function as much as possible.

- Isolate reporting in time.
- Isolate reporting by user.
- Isolate reporting by cluster and application server.
- Isolate reporting by database.
- Manage the report database.
- Configure the report server.
- Intensive reports in off-peak hours.

Many reports that consume significant resources are not needed immediately, and do not necessarily need to be run on up-to-the-minute data. You should run such reports in off-peak hours, such as overnight or on the weekend.

Time-based reports such as end-of-month and end-of-quarter reports can be run in off-peak hours, on a copied database from a specific date and time. Because you do not need to run these reports on the current database, you can protect the production database from being slowed by these reports.

Efficient use of reports

The more users that run reports, and the more reports they run (especially database-intensive reports), the greater the potential effect on system performance. You should establish business practices to help manage the amount of report use, especially during peak system-use hours. Limit the number of users who can run reports. Limit the number of reports that users can run.

During peak business hours, try to limit report use to reports that users need for their daily work, such as Print Work Orders, Print POs, and so on.

Running reports on a dedicated cluster server

If your users do extensive reporting, a good practice is to establish one or more application servers that are dedicated to running reports. You can size the clustering of report application servers based on demand. Establish a separate cluster for running scheduled reports (cron jobs).

Dedicated database for reporting

Some customers report that providing a separate database to run reports on is *the* single practice that gives the greatest boost to system performance.

Configure a independent or dedicated Tivoli Service Request Manager database that has a copy of the production data, and use that as an off-line database for reporting. Mirror the Tivoli Service Request Manager production database on a separate database server, and run resource intensive reports on the mirror database. Create a separate Tivoli Service Request Manager application that connects to the reporting database and synchronize the production and reporting databases periodically.

For example, you might synchronize the databases at the end of every day or once a week, depending on your needs. With this setup, reports that require more system resources can be run by just a few administration users. Because they are run on a separate mirror database, these reports do not affect the performance of the production system.

Periodic maintenance of database for reports

By default, Tivoli Service Request Manager saves reports to the database. Over time, the volume of saved reports can affect database performance. It is a good practice to periodically delete unneeded reports from the database. Consider using a cron task to delete reports at a given interval after they run.

Abbreviations and acronyms

BIRT	Business Intelligence and Reporting Tools
BPEL	Business Process Execution Language
BPM	Business Process Management
CCMDB	Change and Configuration Management Database
CDM	Common Data Model
CIO	Chief Information Officer
EAR	Enterprise Archive
FIERA	Flexible Internet Evaluation Report Architecture
HIPAA	Health Insurance Portability and Accountability Act
HTML	Hypertext Markup Language
IBM	International Business Machines Corporation
ISC	IBM Solution Console
IT	Information Technology
ITIL	IT infrastructure Library
ITSO	International Technical Support Organization
ITUP	IBM Tivoli Unified Process
JVMs	Java Virtual Machines
KPI	Key Performance Indicator
LDAP	Lightweight Directory Access Protocol
LTPA	Lightweight Third-Party Authentication
MBO	Maximo Business Objects
MDBs	Message-driven Beans
MEA	Maximo Enterprise Adaptor
OMP	Operational Management Products
OPAL	Open Process Automation Library
ORB	Object Request Broker
PMP	Process Management Products
RAC	Rational Agent Controller
RMC	Rational Method Composer
SOX	Sarbanes-Oxley Act
SSO	Single Sign-On
STS	Secure Token Service
TEP	Tivoli Enterprise Portal
URL	Uniform Resource Locator
eTOM	Enhanced Telecom Operations Map

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 390. Note that some of the documents referenced here may be available in softcopy only.

- *DB2 UDB/WebSphere Performance Tuning Guide*, SG24-6417
- *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565
- *IBM Tivoli Common Data Model: Guide to Best Practices*, REDP-4389
- *Implementing IBM Tivoli Service Request Manager Service Catalog*, SG24-7613
- *Integration Guide for IBM Tivoli Service Request Manager V7.1*, SG24-7580

Online resources

These Web sites are also relevant as further information sources:

- BIRT information

  `http://www.eclipse.org/birt/phoenix/`
- ITUP information

  `http://www.ibm.com/software/tivoli/features/it-serv-mgmt/itup/index.html`
- WebSphere Application Server Performance testing protocol tools

  `http://www.software.ibm.com/wsdd/techjournal/0211_polozoff/polozoff.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

U

V

W

X

IBM

Redbooks

Implementing IBM Tivoli Service Request Manager V7.1 Service Desk

(0.5" spine)
0.475"<->0.875"
250 <-> 459 pages

Implementing IBM Tivoli Service Request Manager V7.1 Service Desk

Learn how to deploy SRM in your environment

Experiment with Service Desk functions

Improve the performance of your SRM installation

IBM Tivoli Service Request Manager V7.1 provides a unified and integrated approach in dealing with all aspects of service requests to enable a "one-touch" IT service experience, backed up by an optimized delivery and support process. It is a powerful solution that closely aligns IT operations and the business, improving IT service support and delivery performance.

This book provides information that can be used by clients, Business Partners, or IBM field personnel who are looking to engage in an effort to implement ITIL based Service Desk processes in an enterprise environment utilizing the IBM Tivoli Service Request Manager V7.1 product. This book is divided into three parts:

- Concepts and components: Provides an overview of the IBM Tivoli Service Request Manager product Service Desk functions and some of the standards that drive its development. Also provides the reader with a better understanding of the various components, logical and physical, that make up the product and the functions that they provide.
- Planning and Installation: Provides information related to the actual installation of the IBM Tivoli Service Request Manager product components, including information related to hardware and software requirements.
- Demonstration scenarios and best practices: Focuses on demonstration scenarios using some of the new features of Tivoli Service Request Manager, such as reporting, survey, and search functions. We also provide best practices for fine tuning and high availability of Tivoli Service Request Manager components.

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

SG24-7579-00 ISBN 0738431192