IBM

IBM OmniFind Enterprise Edition Version 8.4

Configuration and Implementation Scenarios

- IBM OmniFind Enterprise Edition V8.4 architecture
- Security considerations
- Business scenarios using single sign-on

Nagraj Alur
Mario Deschatelets
Ashish Jain
Sreeram Potukuchi
Wojciech Radzikowski
Filip Zawadiak

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM OmniFind Enterprise Edition Version 8.4 Configuration and Implementation Scenarios

May 2007

SG24-7394-00

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

First Edition (May 2007)

This edition applies to Version 8, Release 4, Modification 0 of IBM OmniFind Enterprise Edition (product number 5724-L31).

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

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

AIX 5L™
AIX®
Cloudscape™
Domino®
DB2 Universal Database™
DB2®
IBM®
Lotus Notes®
Lotus®
OmniFind™
OS/2®
QuickPlace®
Redbooks®
Redbooks (logo) ®
RDN™
System i™
Tivoli®
WebSphere®
Workplace™
Workplace Web Content Management™

The following terms are trademarks of other companies:

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

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

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

Microsoft, MS-DOS, SharePoint, Windows Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

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

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

Preface

This IBM® Redbooks® publication documents the procedures for implementing IBM OmniFind™ Enterprise Edition Version 8.4 technology in a single-server Windows® environment, two-server Linux® environment, and a four-server AIX® environment. Supported data sources include DB2®, Windows file system, DB2 Content Manager, Lotus® Domino®, Lotus Quickplace, WebSphere® Portal Server, Web Content Management (WCM), and Portal Document Manager (PDM). Tivoli® Directory Server (TDS) is the LDAP repository used in the scenarios.

It is aimed at IT architects and search administrators who are responsible for managing IBM OmniFind Enterprise Edition on Windows 2003, Red Hat Enterprise Linux, and AIX platforms.

The book offers a step-by-step approach to implementing a single-server, two-server, and four-server IBM OmniFind Enterprise Edition environment using typical customer scenarios.

This book is organized as follows:

- Chapter 1, “IBM OmniFind Enterprise Edition V8.4 architecture” on page 1 provides a detailed description of IBM OmniFind Enterprise Edition, its architecture and processing flow, security considerations, and key criteria in choosing between a single-server and multiple-server configuration for a given business requirement.
- Chapter 2, “Small business OmniFind scenario on a Windows 2003 Enterprise Edition platform” on page 53 describes a step-by-step approach to implementing a single-server IBM OmniFind Enterprise Edition configuration on a Windows 2003 platform within a typical small to medium (SMB) organization. It assumes an environment with a limited number of data sources to search, such as a Windows file system and Lotus Quickplace. Query time validation (impersonation) is implemented in this environment. Single sign-on is leveraged with Lotus Quickplace, and the identity management component (IMC) is leveraged for the Windows file system. Tivoli Directory Server is the user registry of choice in this environment. A single collection is defined. This chapter includes the configuration of two data sources, IMC and single sign-on, and the use of a customized sample search application.

- Chapter 3, “Medium-size organization OmniFind scenario on Red Hat Enterprise Linux platform” on page 141 describes a step-by-step approach to implementing a two-server IBM OmniFind Enterprise Edition configuration on a Red Hat Enterprise Linux platform within a typical medium organization. It assumes an environment with at three data sources to search, including WebSphere Portal Server, Web Content Management, and Portal Document Manager. Query time validation (impersonation) is implemented in this environment. Single sign-on is leveraged with WebSphere Portal Server, and IMC with Portal Document Manager. Tivoli Directory Server is the LDAP user registry of choice in this environment. Two collections are defined, one with security enabled and one without. This chapter includes the configuration of the three data sources, single sign-on, and the use of the sample search portlet that federates over both the collections (with and without security enabled).
- Chapter 4, “Large organization OmniFind scenario on an AIX platform” on page 299 describes a step-by-step approach to implementing a four-server IBM OmniFind Enterprise Edition configuration on an AIX 5L™ V5.3 platform within a typical large organization. It assumes an environment with three data sources to search, including Lotus Domino, IBM Content Manager, and DB2 UDB for LUW. Query time validation (impersonation) is implemented in this environment. Single sign-on is leveraged with Lotus Domino, and IMC with IBM Content Manager. Tivoli Directory Server is the LDAP user registry of choice in this environment. A single collection is defined that includes support for rule-based categorization, and a UIMA annotator. This chapter includes the configuration of the three data sources, single sign-on, IMC, and the use of a custom portlet.
- Chapter 5, “Merger of SMB and medium-size organizations” on page 409 describes the development of a federation portlet that searches collections spanning the Windows and Linux platforms as a consequence of the merger of the two insurance companies. This chapter includes the use of a custom search portlet that federates over the single server OmniFind Enterprise Edition implementation on Windows 2003 and the two server OmniFind Enterprise Edition on Red Hat Enterprise Linux.
- Appendix A, “Install Sample Search application portlet” on page 431 describes the installation of the sample search application portlet on WebSphere Portal Server.
- Appendix B, “Search Application Customizer” on page 459 describes the main features of the sample Search Application Customizer, and provides customization examples of the search application.
- Appendix C, “IBM OmniFind Enterprise Edition V8.4 control tables” on page 483 provides a brief description of the IBM OmniFind Enterprise Edition control tables defined in the Cloudscape™ database.

- Appendix D, “Configuring typical data sources” on page 503 describes the configuration of some of the more frequently used data sources supported by IBM OmniFind Enterprise Edition that is not included in the user scenarios.
- Appendix E, “Migration considerations” on page 535 describes the steps in migrating an OmniFind V8.3 system to OmniFind Enterprise Edition V8.4.
- Appendix F, “Troubleshooting aids” on page 561 describes the main troubleshooting tools available with IBM OmniFind Enterprise Edition and provides usage examples of them.

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, San Jose Center.

Nagraj Alur is a Project Leader with the IBM ITSO, San Jose Center. He holds a master’s degree in computer science from the Indian Institute of Technology (IIT), Mumbai, India. He has more than 30 years of experience in database management systems (DBMSs), and has been a programmer, systems analyst, project leader, independent consultant, and researcher. His areas of expertise include DBMSs, data warehousing, distributed systems management, database performance, information integration, and client/server and Internet computing. He has written extensively on these subjects and has taught classes and presented at conferences all around the world. Before joining the ITSO in November 2001, he was on a two-year assignment from the Software Group to the IBM Almaden Research Center, where he worked on Data Links solutions and an eSourcing prototype.

Mario Deschatelets is a Canada IBM GBS IT Consulting Architect for Portals and Enterprise Information Management. In this capacity, he has acquired experience in the design and architecture of large-scale and complex solutions for strategic technology-positioning, enterprise integrated solution, and standards definition. He has 15 years of experience in the design and implementation of Search Engines, Portals, Content Management solutions in various industry domains.

Ashish Jain is a member of the IBM India Software Labs (ISL). He is a Technical Manager for Content Management and is an IBM Certified Solutions Designer – DB2 Content Manager V8.3. His team of Information Management experts provide technical consultancy services to System Integrators in India and the Asia Pacific region. Ashish's expertise includes resolving system integrator technical issues, providing solution architectures, design reviews, pre-sales and post-sales support, and technical skills and services to accelerate the integration of IBM information management software with Business Partner applications. Ashish's first involvement with OmniFind was with the Government of India Portal in 2005, and he plans to continue to provide services and expertise to OmniFind customers in India.

Sreeram Potukuchi is a Manager/Database Architect with Werner Enterprises, Omaha, Nebraska. He holds a Masters Degree in Computer Information Systems from Boston University and has over 10 years of experience in the IT industry. His areas of expertise include Database Architecture, Design/Development, Administration, Performance Tuning, Distributed Systems Management, Data Warehousing and Business Intelligence. He is an IBM Certified Specialist - DB2 V7.1 User, IBM Certified Solutions Expert - DB2 UDB V7.1 Database Administration for UNIX®, Linux, Windows and OS/2®, IBM Certified Advanced Technical Expert - DB2 for Clusters, IBM Certified Database Administrator - DB2 UDB V8.1 for Linux, UNIX and Windows, and IBM Certified Advanced Database Administrator – DB2 Universal Database™ V8.1 for Linux, UNIX, and Windows.

Wojciech Radzikowski is an IT Specialist for IBM in Warsaw, Poland. He holds a Bachelors Degree in Computer Science from the Warsaw University of Technology. He works in the Software Services team that provides consultancy services to customers in Poland and across the CEMAAS region. His areas of expertise include WebSphere Portal, Web Content Management, and WebSphere II OmniFind (including the development of custom analysis engines using the UIMA SDK).

Filip Zawadiak is a Certified Senior IT Specialist in IBM Software Services for Lotus (ISSL) in Poland. He has 10 years experience in information technology. His areas of expertise include products like OmniFind, WebSphere Portal, Web Content Management, Lotus Notes/Domino, as well as deep knowledge of operating systems, networking protocols, and systems programming. His skills in debugging and reverse engineering help pinpoint and fix problems quickly. He has over 20 technical certifications in diverse IBM products. He holds a degree in Computer Engineering from Silesian Technical University. He is a member of Association for Computing Machinery (ACM).

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

David Been
Su Han Chan
Chandasekhar Iyer
Jordi Levant
Deborah Nakamura
Justo Perez
Vinod Dandala Reddy
Patricia Ropelatto
Maxime Tiran
Joel Waterman
Kathy Zeidenstein
IBM Silicon Valley Laboratory, San Jose, CA

Srinivas "Varma" Chitiveli
Kameron Cole
Dana Morris
Carolyn A Scott
IBM Software Group, USA

Zhao Xue Shan (Snow)
IBM China

Peter Kohlmann
Alexander Lang
Markus Lorch
Raiko Nitzsche
IBM Germany

Tim J Brown
IBM UK

Seiji Hamada
Koichi Hosono
Hiroaki Kikuchi
Hirofumi Nishikawa
IBM Japan Yamato Laboratory

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbooks publication dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll 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 Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review book 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

1

IBM OmniFind Enterprise Edition V8.4 architecture

In this chapter, we provide an overview of the IBM OmniFind Enterprise Edition V8.4 architecture and processing flow, a summary of what is new in IBM OmniFind Enterprise Edition V8.4, security considerations, and key criteria in choosing between single-server and multiple-server configurations for a given business requirement.

The topics covered are:

- IBM OmniFind Enterprise Edition overview
- IBM OmniFind Enterprise Edition architecture and processing flow
- What is new in IBM OmniFind Enterprise Edition V8.4
- Security considerations
- Choosing a particular topology

1.1 Introduction

The popularity and success of various commercial search engines that crawl Web sites on the internet has been generally tied to the quality of the search results. Search engines such as Google have used various algorithms to rank the search results in a way that tries to present the user with the most appropriate link at the very top. In addition to the proprietary algorithms used by the search engines, the quality of metadata retrieved from a Web site plays a significant role in the search quality.

Enterprises can benefit significantly by providing its employees search capabilities against the vast amount of data stored within the enterprise's various data sources. High quality enterprise search can increase productivity and help in the dissemination of information. However, searching within an intranet poses challenges that are significantly more complex than those faced by internet search engines. A detailed discussion of these challenges are beyond the scope of this book. However, enforcing native security of the various data sources when retrieving content, collecting useful metadata, and effective ranking algorithms for improving search quality are among the more challenging issues.

Important: Improving search quality in enterprise searches will require a combination of the invention of sophisticated document processing algorithms, flexible security infrastructures, and a concerted effort to educate enterprise content authors to develop metadata and content that can be easily searched. IBM OmniFind Enterprise Edition is an enterprise search engine architected to address these issues and deliver high quality enterprise search capabilities to an organization.

The following sections cover:

- IBM OmniFind Enterprise Edition overview
- IBM OmniFind Enterprise Edition architecture
- What is new in IBM OmniFind Enterprise Edition V8.4
- Security considerations
- Choosing a particular topology

1.2 IBM OmniFind Enterprise Edition overview

IBM OmniFind Enterprise Edition (previously known as WebSphere Information Integrator OmniFind Edition) delivers breakthrough full-scale secure enterprise search capabilities to help get the right information to the right people at the right time. Built on a flexible, open architecture, it offers a unique and powerful combination of performance, security, scalability, enterprise reach, and openness for applying advanced linguistic processing. By simply entering a keyword or phrase in a single query, users can quickly search across intranets, corporate public Web sites, relational database systems, file systems, and content repositories, and obtain a consolidated, ranked meaningful result. Users also have the ability to go beyond standard full-text search and perform parametric and semantic queries to dramatically improve the relevancy of search results.

IBM OmniFind Enterprise Edition is well adapted to work in the heterogeneous and multivendor world of the modern enterprise. Lotus Notes/Domino and WebSphere Portal customers, in particular, will benefit from the secure best-in-class integrations IBM OmniFind Enterprise Edition offers to them. IBM OmniFind Enterprise Edition effectively addresses the need for stringent security safeguards to protect content from unauthorized access using several techniques.

IBM OmniFind Enterprise Edition is the first commercially available UIMA[1] based platform for processing text-based information. UIMA enables seamless integration of text analytics components that analyze documents, extract knowledge, and identify higher-level concepts, such as people, places, organizations, products, problems and other “entities” that are buried within unstructured data. This knowledge can be used to create an enhanced index for searching, or routed to a traditional data mart or data warehouse for use in business intelligence and analysis applications.

Finally, IBM OmniFind Enterprise Edition is an archetype of the service-oriented architecture (SOA) paradigm. By searching information scattered across multiple sources throughout the enterprise, enhancing its value by enriching metadata through text analytics, and making it available as a service to people, processes, and applications that can take advantage of it, IBM OmniFind Enterprise Edition delivers a core function for any SOA environment.

[1] Unstructured Information Management Infrastructure (UIMA) is an open, extensible framework for processing unstructured information through text analytics for extracting insight from unstructured content to enable capabilities such as semantic queries, navigation of business intelligence reports, and custom analytics applications.

Note: An entry-level offering, IBM OmniFind Enterprise Starter Edition (previously known as WebSphere Information Integrator OmniFind Starter Edition), is also available providing the same functional capabilities as IBM OmniFind Enterprise Edition in a scaled-down implementation.

The main features of IBM OmniFind Enterprise Edition include the following:

- Multiple topology support

  IBM OmniFind Enterprise Edition supports three topology implementations: single-server, two-server, and four-server implementations.

- Wide content reach

  The range of data sources supported by IBM OmniFind Enterprise Edition include file systems, content repositories, databases, collaboration systems, intranets, extranets, and public-facing corporate Web sites. New sources are continually being added, and readers should refer to the http://www.ibm.com/software/data/enterprise-search/omnifind-enterprise/ Web site for an up-to-date list of supported data sources. Additional connectors can be built to other data sources through a Data Listener API.

- Security features

  Security is an integral element for enterprise search. Only authorized users may administer the system. In conjunction with the security mechanisms available in IBM WebSphere Application Server, you can configure administrative roles and authenticate all administrative users. By configuring administrative roles, you control users that can have access to various administrative functions.

  You may also specify options to associate security tokens with data when the data is being collected. By enabling security for the search applications, you can use these tokens to enforce access controls and ensure that only users with the proper credentials are able to query the data and view search results.

- Sample search application and SIAPI API

  Besides a free-text GUI search interface, IBM OmniFind Enterprise Edition provides an SIAPI API that can be incorporated easily into enterprise Java™ applications to develop custom search portals. IBM OmniFind Enterprise Edition includes a sample search application that can be used as a template for creating search applications that meet the special needs of ones organization.

  IBM OmniFind Enterprise Edition offers WebSphere Portal clients enhanced search capabilities with a broader content reach, and scalability to millions of documents. WebSphere Portal Search customers have seamless transition to IBM OmniFind Enterprise Edition, which imports and reuses existing portal

taxonomies for navigation and categorization, migrates rules for rule-based classification, and provides the same user experience as the WebSphere Portal's Search Center portlet.

- Heterogeneous platform support

  IBM OmniFind Enterprise Edition is supported on a number of operating system platforms, including Microsoft® Windows Server® 2003 Enterprise Edition Service Pack 1, IBM AIX 5L (32-bit and 64-bit support) V5.2 (maintenance level 7), IBM AIX 5L (32-bit and 64-bit support) V5.3 (maintenance level 3), Linux for Intel® (32-bit support) Red Hat Enterprise Linux AS 4 Update 2, Linux for Intel (32-bit support) SUSE Linux Enterprise Server V9.0 with Service Pack 2 (United Linux SP2), and Solaris™ (32-bit and 64-bit support) 9 with patch 118558-09. For specific details on versions supported, refer to *IBM OmniFind Enterprise Edition Installation Guide for Enterprise Search Version 8.4*, GC18-9282-03 and http://www.ibm.com/software/data/enterprise-search/omnifind-enterprise/.

1.3 IBM OmniFind Enterprise Edition architecture

Figure 1-1 provides an overview of the IBM OmniFind Enterprise Edition architecture and data flow.

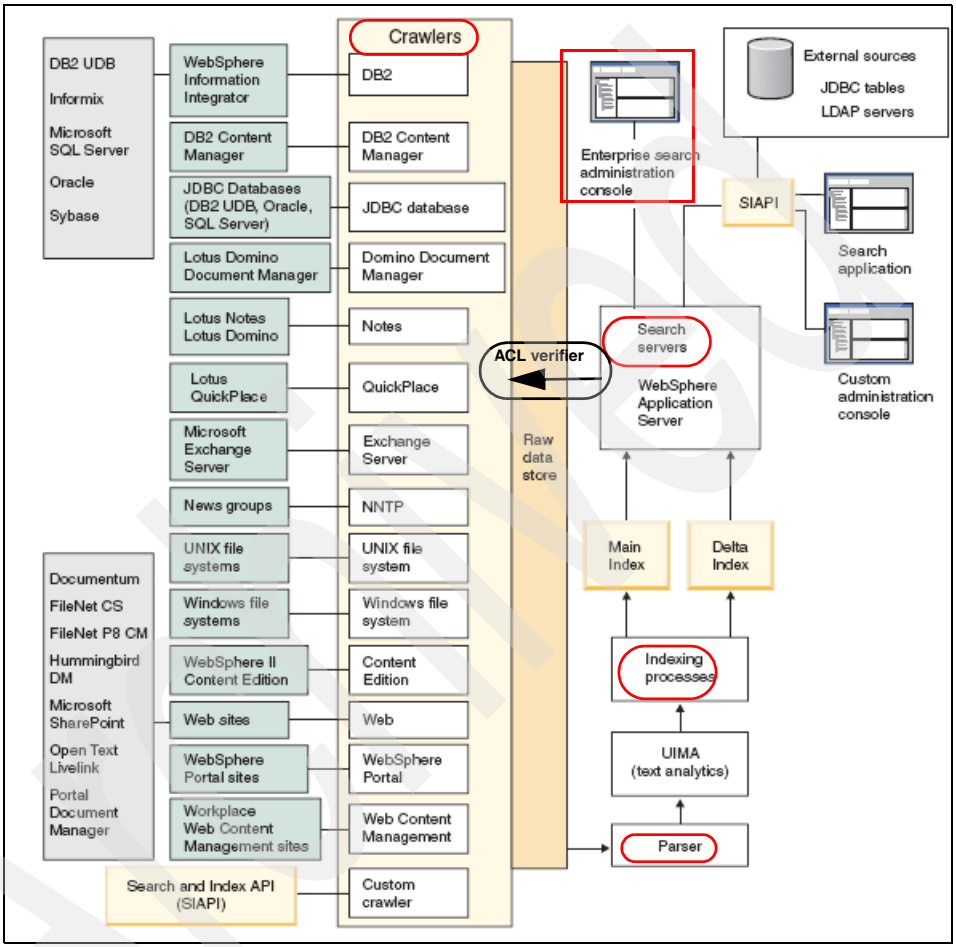

Figure 1-1 Main components and data flow

In this section, we briefly discuss the following topics:

- Main components
- Data flow
- Topologies supported
- Directory structure
- Collections, categories and scopes

1.3.1 Main components

The five main components of IBM OmniFind Enterprise Edition shown in Figure 1-2 (a simplified version of Figure 1-1 on page 6) are crawler, parser, indexer, search runtime, and admin console.

Note: The CCL server is the service that makes it possible for OmniFind to manage all of its processes (or components), and allows components to communicate with each other. The Controller/Monitor layer in OmniFind is responsible for coordinating and monitoring capabilities across all the components. DataListener is an API for pushing crawled data to the raw data store[a]. These will not be discussed further.

a. CCL is an internal communication method, but the DataListener APIs are described in the product documentation.

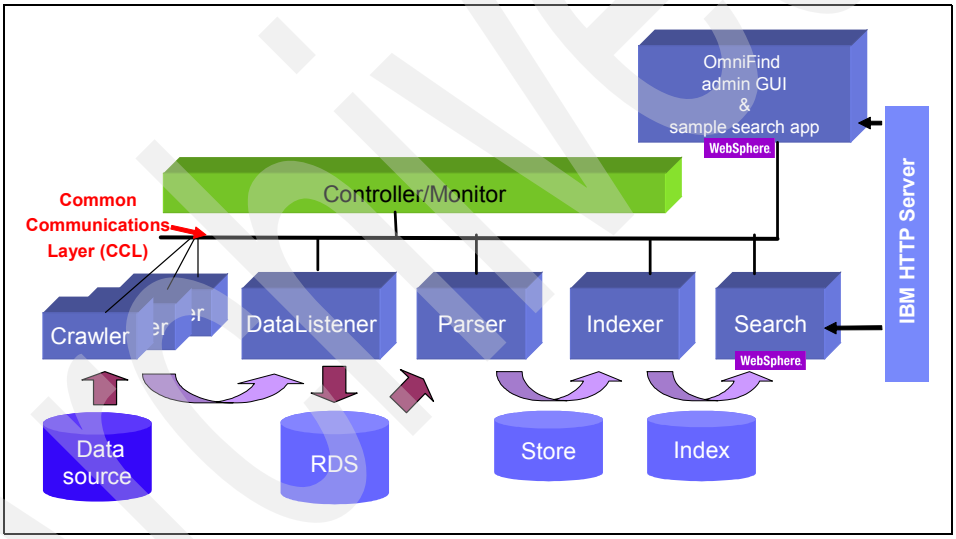

Figure 1-2 Main components and data flow

A brief description of the components shown in Figure 1-2 follows:

1. Crawler component

   This component performs the function of crawling the various data sources at intervals configured by the administrator, and populates a raw data store (file system based) with the content extracted from the data sources. Crawlers are available for a number of data sources, as shown in Figure 1-1 on page 6.

A crawler has the following general properties, though not all crawlers may have all of them:

a. May crawl multiple data sources and do so using multiple threads, which are configurable.

b. Some crawlers might also implement real-time verification from the search servers, as shown in Figure 1-2 on page 7. Crawlers may also implement additional services for discovery services, which are mostly used for the creation of administration screens, such as database lists, column lists, and user name checking.

c. Crawler properties are a set of rules that govern the behavior of a particular crawler when it crawls. For example, one can specify rules to control how the crawler uses system resources. The set of sources that is eligible to be crawled constitutes the crawl space of a crawler. After a crawler is created, one can edit the crawler properties at any time to alter how the crawler collects data. One can also edit the crawl space to change the crawler schedule, add new sources, or remove sources that are not to be searched any longer.

   One can start and stop crawlers manually, or set up crawling schedules. When scheduling a crawler, one specifies when it is to run initially and how often it needs to visit the data sources to crawl new and changed documents.

   The Web crawler runs continuously. After one specifies the uniform resource locators (URLs) to be crawled, the crawler returns periodically to check for data that is new and changed. The frequency of crawling the Web is determined automatically based on configuration guidelines that bound the lower and upper limit of crawl frequencies. For example, specifying a lower limit of weekly crawl frequency and an upper limit of daily crawl frequency will guide the determination of the actual crawl frequency in conjunction with statistics accumulated about the percentage of changes detected over past crawl actions.

   Unlike OmniFind V8.3, the NNTP crawler in OmniFind V8.4 does not run continuously. Its behavior is the same as the other data source crawlers such as the DB2 crawler or Notes crawler.

d. Each data source type is associated with a different crawler type, for example, all DB2 data sources have a DB2 crawler, while file system data sources have a file system crawler. However, multiple crawlers may be defined for different data sources of the same data source type. For example, one crawler may be defined for a set of DB2 tables in the human resources system, while another crawler may be defined for a set of DB2 tables in a data warehousing system.

e. All the crawler instances belonging to the same collection share the same raw data store. Each collection has its own raw data store.

f. One or more security tokens may be associated with the documents crawled. A security token plug-in may be written to generate the relevant security token(s) for a document using appropriate lookups of access control lists.

Attention: When running crawlers in OmniFind V8.4, we strongly recommend that you run the parser at the same time. This is because a file queue is used to store crawled data instead of the DB2 table used in OmniFind V8.3. The file queue can fill up if the parser is not used to parse and delete the documents in the file queue while the crawler is running.

2. Parser component

This component analyzes the documents that were collected by a crawler and prepares them for indexing. The parser component analyzes document content and document metadata. You can improve the quality and precision of search results by integrating custom text processing algorithms with enterprise search collections in the parser. These text processing algorithms are developed using the Unstructured Information Management Architecture (UIMA), which is a framework for creating, discovering, composing, and deploying text analysis functions.

The parser stores the results of the analysis in a file system data store for access by the indexing component.

In practice, the store comprises two distinct entities as follows:

– Main store

– Delta store, which corresponds to those documents that have not yet been merged into the (main) store.

Note: A shadow copy is also kept of the main store and the delta store to provide high availability of the data for search applications during parser processing. The parser updates the shadow copy (or passive copy) with documents, while the active copy is accessed by the search runtime. After a main index or delta index build operation (as discussed in 3 on page 11), the shadow and active copy's (of both the store and index) are toggled and the search runtime now has access to the latest updates. The shadow copy architecture also provides a backup capability in the event of corruption of the active copy.

A parser is associated with a single collection[2] and is multi-threaded. It can be started and stopped on demand, but is usually started once and runs continuously. The parser extracts the documents crawled for a given collection from the raw data store and after analysis, stores relevant portions of the document in the file system data store.

In a four server configuration, the parser is installed on the indexer server.

The parser performs the following tasks:

– Extracts text from whatever the format a document is in, for example, the parser extracts text from the tags in XML and HTML documents. The parser also extracts text from binary formats, such as Microsoft Word and Adobe® Acrobat® portable document format (PDF) documents.

– Detects the character set encoding of each document. Before doing any linguistic analysis, the parser uses this information to convert all text to Unicode.

– Detects the source language of each document.

– Applies parsing rules specified for the collection. The following can be configured for a parser:

  • Field mapping rules for XML documents

    This feature enables users to search structured and unstructured content in XML documents. If one maps XML elements to search fields, users can specify the field names in queries and search specific parts of XML documents.

    Note: Queries that search specific fields can provide more precise search results than free text queries that search all document content.

  • Categories

    This feature enables users to search documents by the categories to which the documents belong. Users can also select categories in the search results and browse only documents that belong to that particular category. “Categories” on page 25 describes the considerations involved with categories in greater detail.

[2] For all practical purposes, a collection corresponds to a single index that is created from the set of all the documents crawled. Collections are covered in detail in 1.3.5, “Collections, categorizations, and scopes” on page 24.

– Extracts text and adds tokens to enhance the retrievability of data. During this phase, the parser performs the following tasks:

  • Character normalization, such as normalizing capitalization and diacritical marks, such as the German umlaut.

  • Analyzing the structure of paragraphs, sentences, words, and white space. Through linguistic analysis, the parser decomposes compound words and assigns tokens that enable dictionary and synonym lookup.

  Figure 1-3 shows an example of tokenization by the parser.

– Interfaces with any text analysis UIMA based annotators that may have been configured for this collection for further text processing, as shown in Figure 1-1 on page 6.

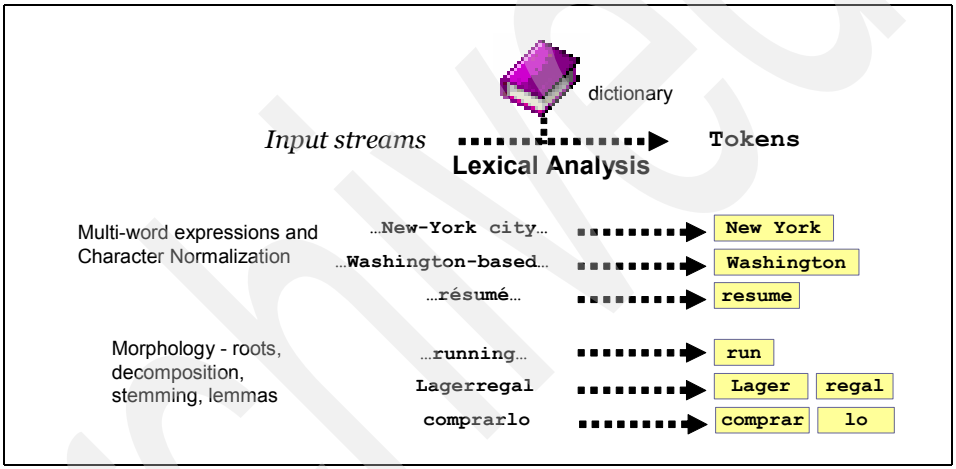

Figure 1-3 Example of tokenization

3. Indexer component

   This component may be scheduled at regular intervals to add information about new and changed documents to an index that is stored in a file system.

   In practice, the index comprises two distinct entities as follows:

   – Main index, which has the global ranking order of all the documents.

   – Delta index, which corresponds to those documents that have not yet been merged into the (main) index.

Note: A shadow copy is also kept of the main index and the delta index to provide high availability of the data for search applications during index build processing. The indexer updates the shadow copy (or passive copy) with documents, while the active copy is accessed by the search runtime. After a main index or delta index build operation, the shadow and active copy's (of both the store and index) are toggled and the search runtime now has access to the latest updates. The shadow copy architecture also provides a backup capability in the event of corruption of the active copy.

Note: Index building is multi-threaded across collections, but single threaded within a collection. Index build is not multi-threaded across collections. As with the parser, each collection has its own indexer process (component). That indexer instance runs independently of those of other collections. The indexer component itself is multi-threaded.

Building an index involves two stages:

– Main index build operation

When the main index operation is requested, the main index is rebuilt so that the structure is optimally organized. The indexing process reads *all* of the data that was collected by crawlers and analyzed by the parser. It merges the main store and delta store (if any) as well as the main index and delta index (if any).

Note: When the main index is built for the very first time by the reorganize index process, it is built from the main store only, since there are no delta stores or indexes.

As mentioned earlier, it is the shadow copy that is reorganized and switched to as the active copy after successful reorganization of the shadow copy.

– Delta index build operation

When the delta index operation is requested, the data that was crawled and parsed after the last time the delta index build operation cycle is added to the delta index. The delta store contains this information and the delta index is rebuilt. Again, as mentioned earlier, it is the shadow copy that is refreshed and switched to as the active copy after successful refresh of the shadow copy.

When configuring index options for a collection, one can specify schedules for reorganizing and refreshing the index. The frequency with which one reorganizes and refreshes the index depends on available system resources and whether the sources being indexed contain static or dynamic content. To ensure the availability of new information, one needs to schedule frequent delta index build operations.

During the indexing phase, algorithms are applied to identify duplicate documents, analyze the link structure of documents, and perform special processing on Web documents, such as document ranking.

After an index is built, one can configure scopes. A scope enables one to limit what users can see in the collection. Scopes are discussed in detail in “Scopes” on page 30.

4. Search runtime component

   This component works on behalf of a search application to process queries, search the index, and return search results to the search application. Figure 1-4 shows the workflow involved in the submission of a query and processing of the search results.

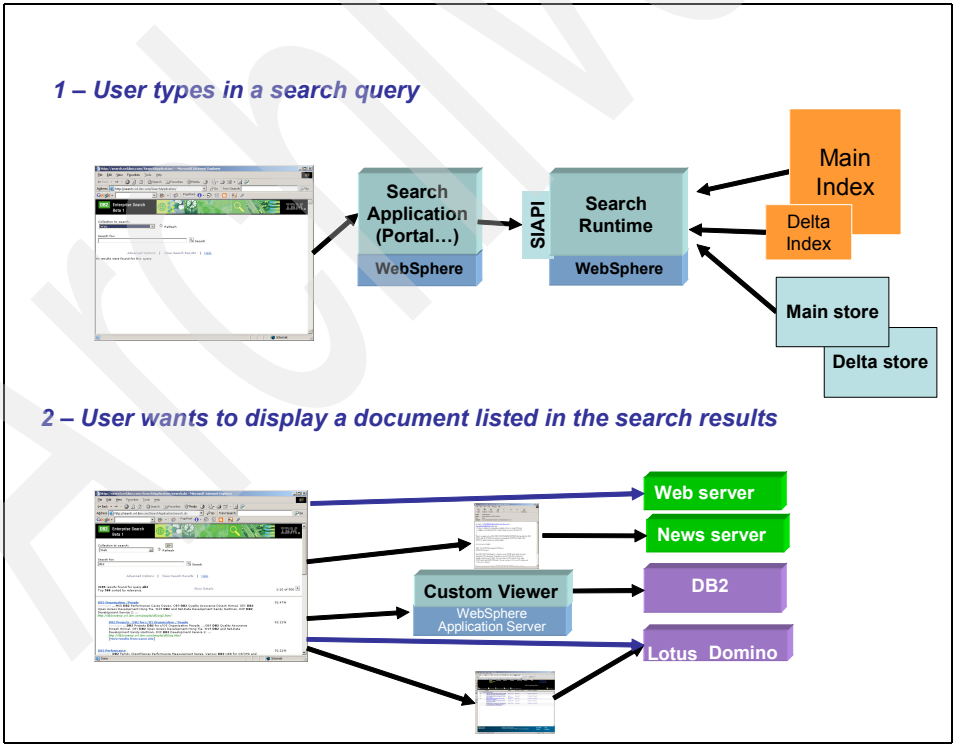

Figure 1-4 User search workflow

In Figure 1-4 on page 13, the search application receives input from a browser and invokes the search runtime through the search and index API (SIAPI). The search runtime executes the query against the index and returns a result to the search application that includes data collected from store, and “quick links”. The search application then presents the search results in an appropriate format.

Figure 1-5 on page 15 shows the results presented to the browser by the sample search application provided with WebSphere Information Integrator OmniFind Edition; the predefined links are explicitly identified with the title “Quick Links”.

The sample search application ESSearchApplication can be used as a template for developing custom search applications. The sample search application is a Struts framework application and is installed with IBM OmniFind Enterprise Edition; it is also made available as a portlet in WebSphere Portal Server. Some of its main features include:

- It demonstrates most of the search and retrieval functions that are available for enterprise search.
- It is a working example that enables one to search all active collections in one’s enterprise search system.
- It can be used to quickly test new collections before making them available to users.
- It includes options for specifying simple queries or queries with advanced options, such as options for searching categories or specifying the number of documents that can appear on a result page.
- If a collection includes documents in multiple languages, one can restrict the result set by specifying which languages one wants to search. One can also choose to see a summary of the results or details about each result document.

Note: A search application enables one to search collections in one’s enterprise search system. One can create any number of search applications, and a single search application can search any number of collections.

Figure 1-5 identifies certain other aspects of the results, such as lexical affinities, stemming, site collapse and stop-word elimination, a discussion of which is beyond the scope of this book. For a detailed understanding of these terms, the reader is urged to consult other documentation sources, such as *IBM OmniFind Enterprise Edition Version 8.4 Administering Enterprise Search,* SC18-9283.

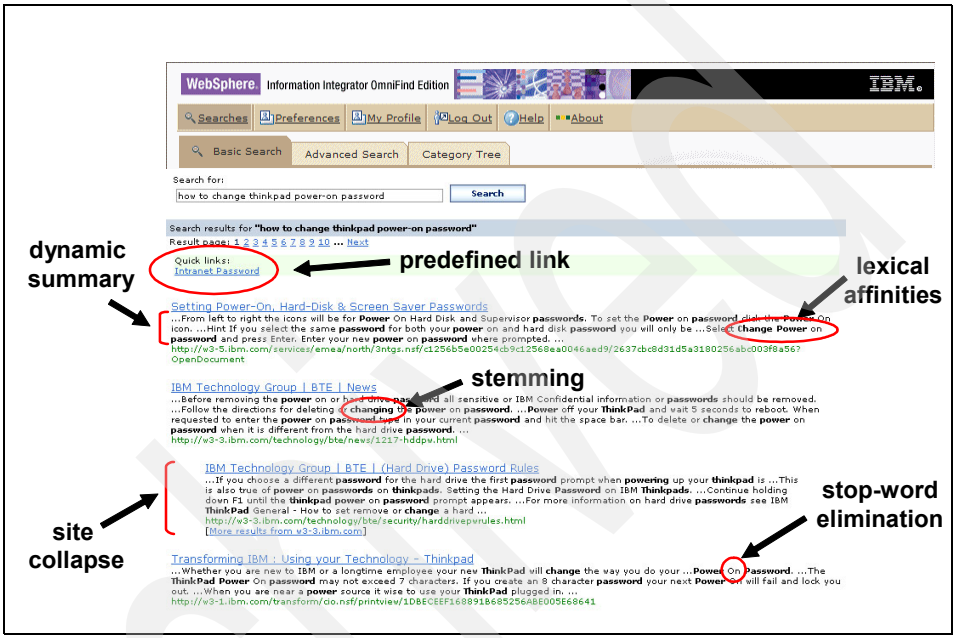

Figure 1-5 Search results

Attention: A custom search application may choose to present the search results returned from the search runtime in a different format. For example, the custom search application may choose *not to* explicitly identify the predefined links in the result, but instead only choose to place them at the top of the results set. The custom search application may be a portlet, servlet, or a Java application.

The user may scroll through the results presented by the search application and click a particular hyperlink to view the entire document of interest, as shown in Figure 1-4 on page 13. Depending upon the data source involved with the document requested, a custom viewer application may need to be written to access the data source and retrieve the document in question. For example, access to Web pages can be achieved without an intervening viewer application, while access to a database row will require a viewer

application to be written that connects to the target database and retrieves the row in question. This workflow implies the following requirements for a document to be retrieved for the user when a hyperlink is clicked:

- There is an ability to connect to the target data source where the document is stored from either the browser or the server that the hyperlink references.
- The connecting entity has the necessary security privileges to connect to the target data source and retrieve the document of interest.

Note: When configuring the search server(s)[a] for a collection, one can configure a search cache to hold frequently requested search results. A search cache can improve search and retrieval performance.

a. Multiple search servers are defined in two server and four server topologies

5. Admin console component

   This component is used to create and administer collections, start and stop other components (such as the crawler, parser, indexer, and search), monitor system activity and log files, configure administrative users, and associate search applications with collections.

   Note: Enterprise search administrators, collection administrators, operators, and monitors are all allowed to use the admin GUI; however, they might be restricted from certain views depending upon their specific role.

   Figure 1-6 on page 18 shows the admin console GUI. It shows a toolbar that provides various views, such as Collections, External sources, System, Security, and Search Customizer.

   - Collections view

     This view is used to create collections and administer the system

   - External sources view

     This view is used to specify options for making the data sources (without crawling or indexing them) searchable. You must specify information that enables your Java Database Connectivity (JDBC™) databases and Lightweight Directory Access Protocol servers to be accessed for enterprise search. After you associate the external sources with search applications, users can search these sources at the same time that they search collections with data that was crawled, parsed, and indexed.

– System view

  This view is used to configure alerts for system-level events, specify how many indexes can be built concurrently, and specify options for logging messages that are produced by system-level processes. Collection administrators, operators, and monitors cannot access this view.

– Security view

  This view is used to specify access controls for collections and the admin console. Collection administrators, operators, and monitors cannot access this view. Until you create your own search applications, you can use the sample search application to search all collections. After you create a search application, use the Security view to associate your application with the collections that it can search. If you enable security in IBM WebSphere Application Server, you can also use the Security view to configure administrative roles. By configuring administrative roles, you can allow more users to administer the system, yet restrict each user's access to specific functions and collections.

– Search Customizer view

  This view is used to invoke the Search Application Customizer.

– Monitor view

  This view is invoked by clicking the Monitor icon, and is used to monitor the system or collection components. Certain administrative roles can also start and stop component processes while monitoring them.

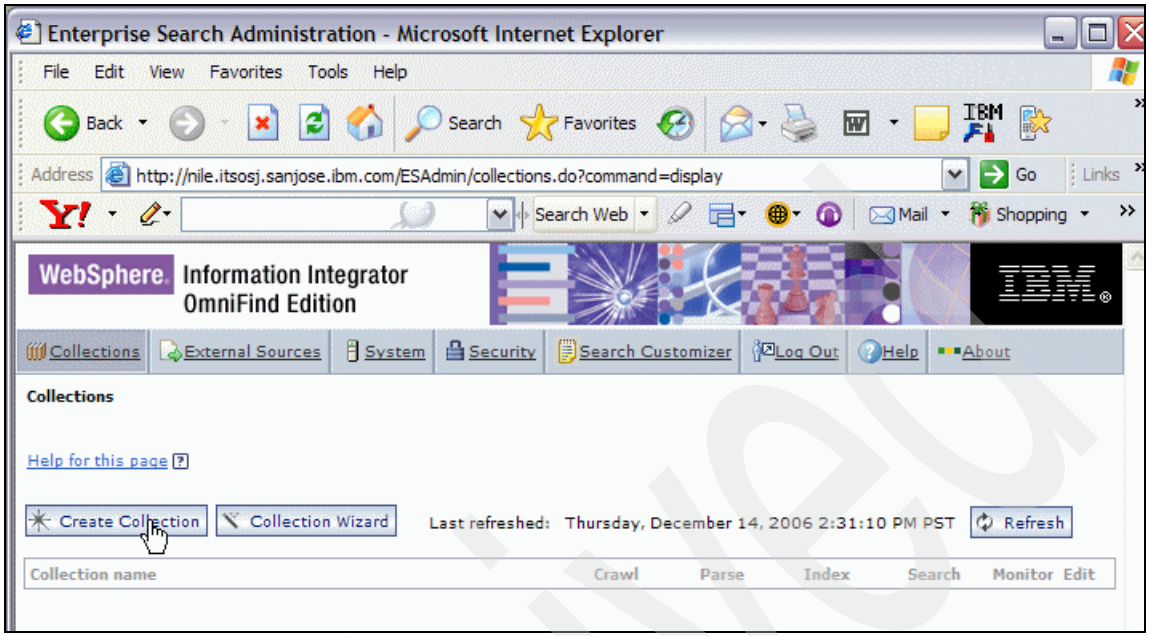

Figure 1-6 Admin console GUI

Note: The crawler, parser, indexer and search runtime components operate independently of one other. They can be started and stopped independently without affecting the others. For example, if a crawler or parser or indexer was to fail, the search runtime component could still be serving queries.

Figure 1-7 shows *some* of the key objects in IBM OmniFind Enterprise Edition. Some of the objects have been discussed already while others will be described in later sections.

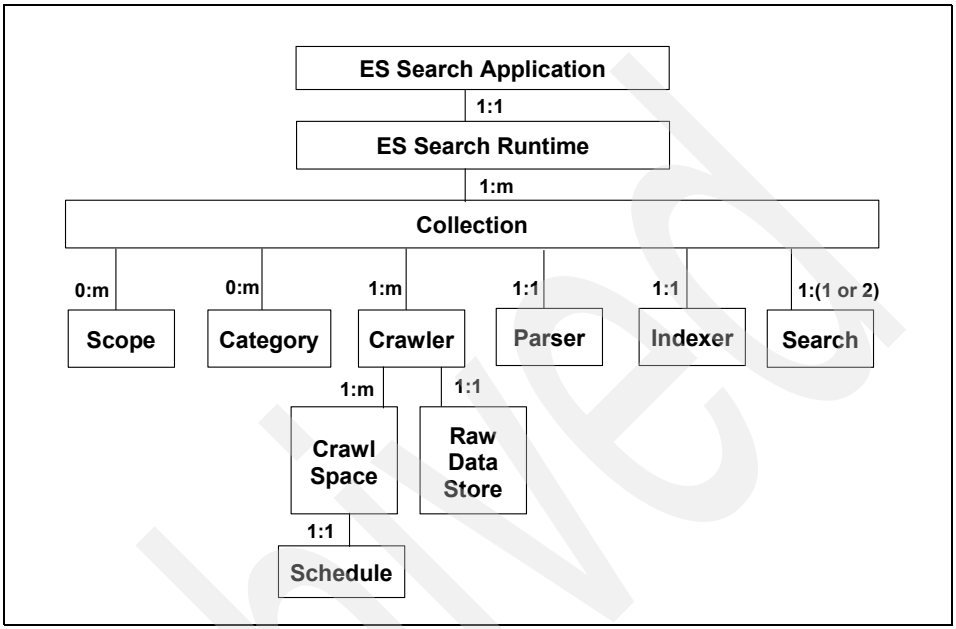

Figure 1-7 Some of the key objects in OmniFind Enterprise Edition and their object relationships

Figure 1-7 shows the following relationships between the objects:

- Each search application has one and only one search runtime associated with it.
- Each search runtime may have one or more collections associated with it.
- Each collection has the following relationships with other objects:
  - Each collection has zero or many scopes associated with it.
  - Each collection has zero or many categories associated with it.
  - Each collection has one or many crawler instances associated with it.
    - Each crawler instance has one or many crawl spaces associated with it. Each crawl space has one and only one schedule associated with it.
    - All crawler instances have the same collection raw data store associated with them.
  - Each collection has one and only one parser associated with it.
  - Each collection has one and only one indexer associated with it.

– Each collection may be associated with one or two search servers.

Figure 1-8 summarizes the key technologies deployed in the OmniFind environment.

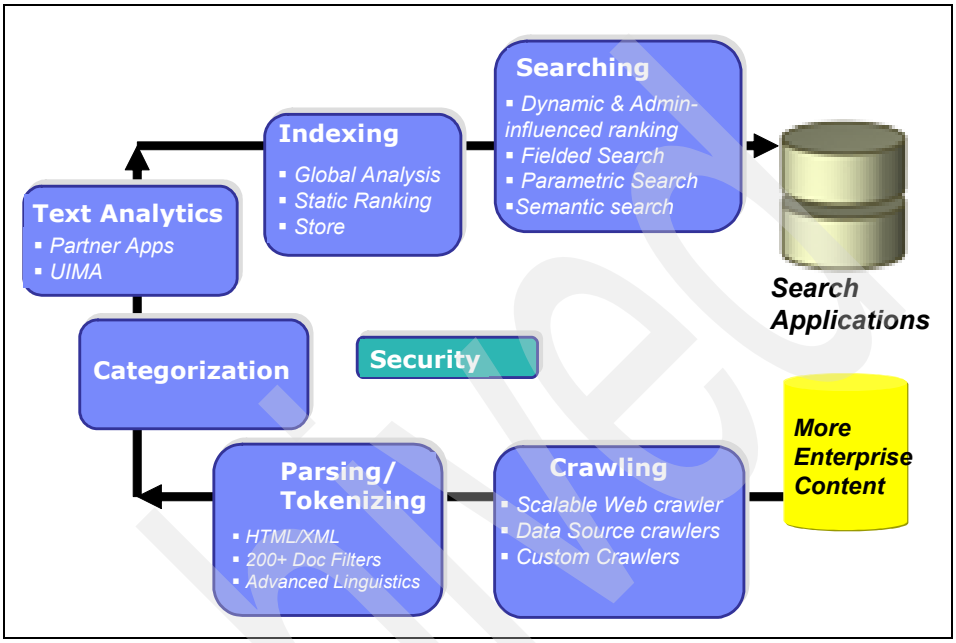

Figure 1-8 Key technologies

1.3.2 Data flow

Figure 1-2 on page 7 shows the data flow within OmniFind as follows:

1. Different crawlers write the crawled data to a single raw data store (RDS) which is file system based and located on the Indexer server. The DataListener API is used to perform this function, as described in 1.4.4, “Usability and performance enhancements” on page 34. However, the metadata associated with each crawler instance is written to a Cloudscape database (located on the Crawler server), with each crawler having its own set of database tables.
2. The parser component reads data from the collection’s RDS, parses it, and invokes any UIMA annotators that have been defined for complex text analytics. It then writes the output to the data store, which is file system based.
3. The Indexer reads the data from the store and creates an index.

For multi node installations, the index and data store is sent to the Search servers.

1.3.3 Topologies supported

IBM OmniFind Enterprise Edition currently supports three topologies:

1. A single server topology, where all the components are installed on a single server, as shown in Figure 1-9.

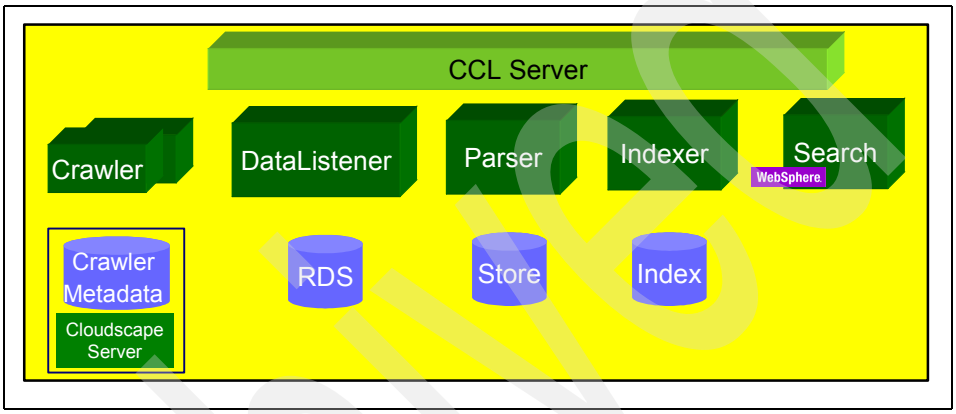

Figure 1-9 Single server topology

2. A two server topology, where the components are installed on two servers, as shown in Figure 1-10.

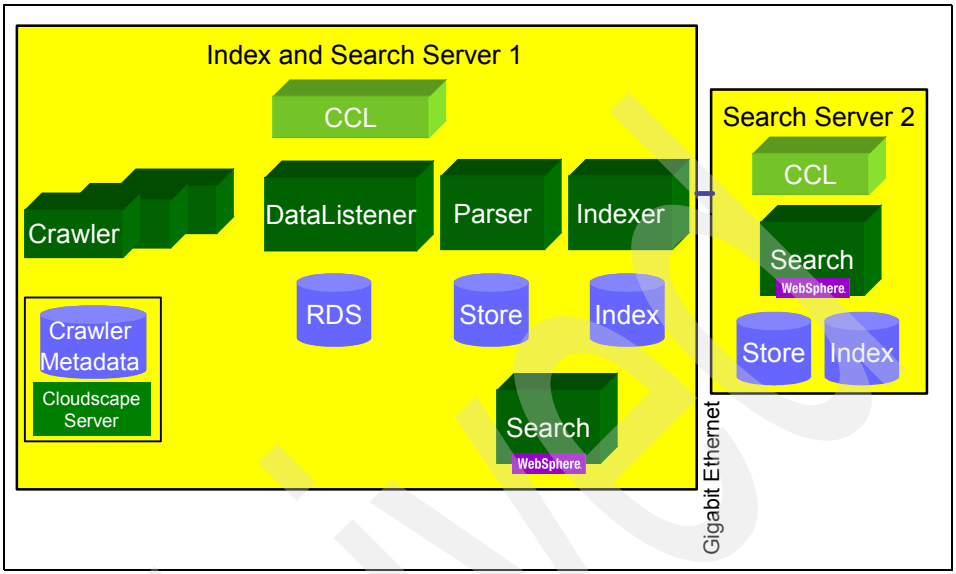

Figure 1-10 Two server topology

3. A four server topology, where the components are installed on four separate servers, with the search components installed on two servers, as shown in Figure 1-11.

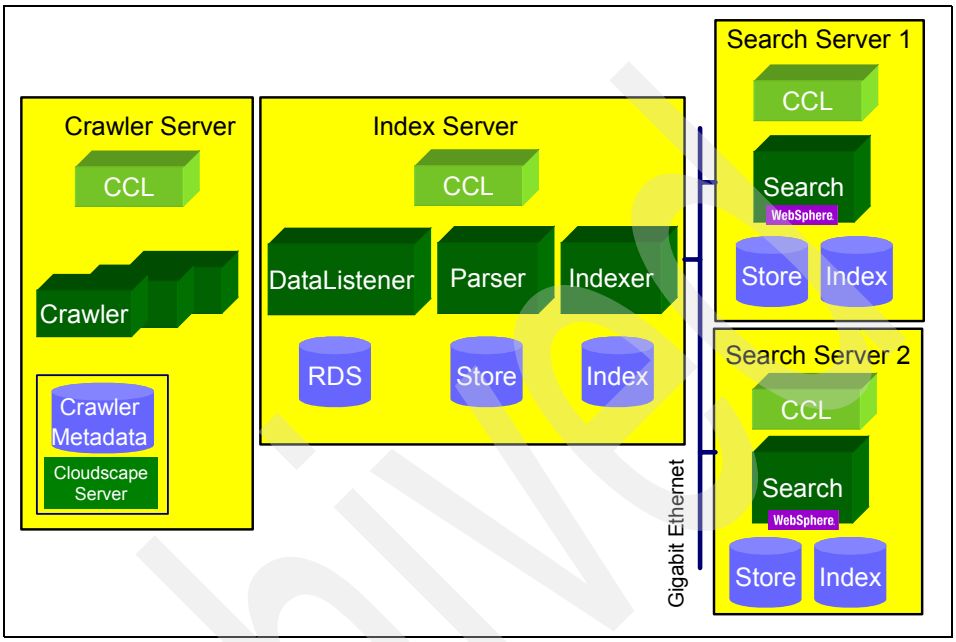

Figure 1-11 Four server topology

Considerations in choosing a particular topology are discussed in 1.6, “Choosing a particular topology” on page 51.

1.3.4 Directory structure

$ES_INSTALL_ROOT (%ES_INSTALL_ROOT% on Windows) and $ES_NODE_ROOT (%ES_NODE_ROOT% on Windows) are two environmental variables that are created by the OmniFind installer. They designate the two main directories in OmniFind as follows:

1. ES_INSTALL_ROOT points to the root installation directory, which contains all the libraries, binaries, dictionaries, documentation, and samples. The default value for this environment variable is:
   - /opt/IBM/es on Linux/AIX
   - c:\Program Files\IBM\es on Windows

   This directory can be modified during installation.

2. ES_NODE_ROOT points to the directory where config files, log files, and indexes are stored by default. The default value for this environment variable is:
   - /home/<omnifind user> on Linux/AIX
   - c:\Program Files\IBM\es\<omnifind user> on Windows

   The location of some of these files can be customized when creating a collection. Since a collection can grow rapidly, it is critical to plan for adequate space in the target directory.

Note: When a Fix Pack is applied, the files in ES_INSTALL_ROOT are most likely to be affected rather than those in ES_NODE_ROOT.

1.3.5 Collections, categorizations, and scopes

An user may query a collection, a category within a collection, or a scope within a collection.

In this section, we provide a brief overview of these objects and the relationship between them. As shown in Figure 1-7 on page 19, there can be zero to many scopes and categories associated with a given collection. The following subsections cover:

- Collections
- Categories
- Scopes
- Choosing between categories and scopes

Collections

An enterprise search collection contains the entire set of sources that users can search with a single query. It corresponds to an index that the search application queries. A collection has its own set of configuration files and processes, such as crawlers, parser, and indexer.

Note: To create a collection, one must be a member of the enterprise search administrator role. To add content to a collection or to specify options for how content in the collection can be parsed, indexed, or searched, one must be an enterprise search administrator or a collection administrator for the collection.

A number of configuration options are available when creating a collection, such as permitting static ranking, specifying the index location, and requesting categorization. A collection is empty until content is added to it through the definition and scheduling of one or more crawlers, and subsequent parsing and indexing of the crawled content. Additional configuration options are available when defining crawlers, and configuring the parser, indexer, and search components. For full details of these options, refer to *IBM DB2 OmniFind Enterprise Edition Version 8.4 Administering Enterprise Search,* SC18-9283.

A collection may be created using the Create Collection view in the GUI admin console, or by invoking the Collection Wizard, as shown in Figure 1-6 on page 18.

Categories

Categories enables one to group documents within a collection that share common characteristics, and users may then limit their search to a specific category by identifying them as the target of their search. Categories may be organized as a tree (taxonomy) to multiple levels of nesting, and a single document may belong to more than one category. A single collection may have zero or many categories associated with it.

Note: To configure categories, one must be a member of the enterprise search administrator role or be a collection administrator for the collection to which the categories belong.

OmniFind Enterprise Edition supports a rule-based approach for categorizing documents.

Categorization may be specified when creating the collection or when specifying parsing rules for the collection. The parser uses the rules that specified to associate documents with one or more categories:

- If a document passes at least one rule in a category, the parser associates the document with the category.
- If a document passes at least one rule in several categories, the parser associates the document with all of the categories.
- If a document does not pass any of the rules for a category, the parser does not associate the document with a category. Users can search for this document and retrieve it when they search the collection, but they will not be able to search a category and expect to retrieve the document.

When configuring a rule for categorizing documents, one may choose whether OmniFind Enterprise Edition is to use the URI of a document or the content in the document to determine whether the document belongs to the category as follows:

- URI pattern

  A URI rule applies to the document's URI. One may specify a partial URI (a pattern), and documents that have the specified pattern in their URIs pass the rule.

  For example, specifying the rule text as /hr/, then, in the following list, the first URI passes the rule, while the second URI does not:

  a. file:///corporate/hr/medicalform.doc http://company.com/human

  b. resources/medicalform.htm

Figure 1-12 shows an example of a rule-based category definition using a URI pattern.

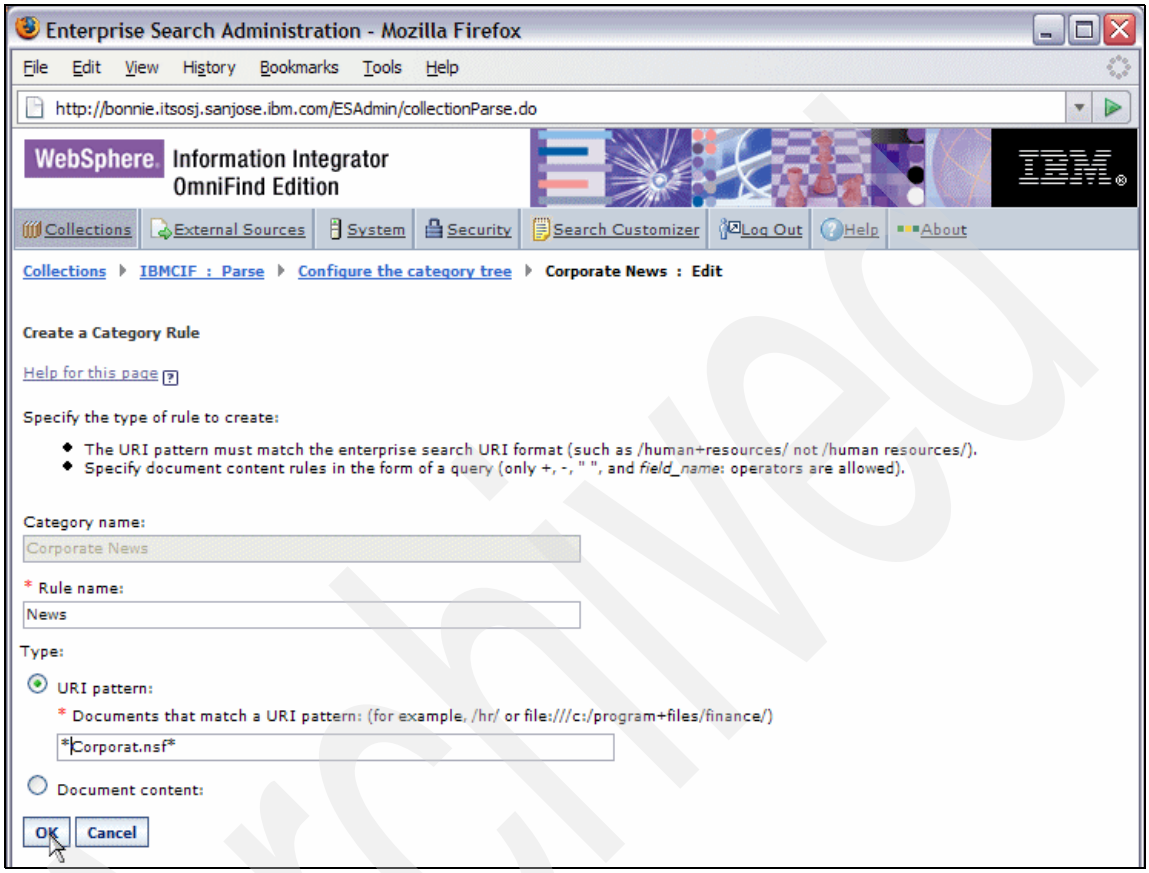

Figure 1-12 Configuring a rule-based category using a URI pattern

- Document content

  A content rule applies to the text of the document. One expresses the rule in the same format as a query. If the document is valid for the query, it passes the rule. When one configures the rule, one specifies the words and phrases that documents must contain or exclude, and one chooses a language for applying word stemming rules. For example:

  - The following rule specifies that if a document contains either the word *hr* or the phrase *human resources*, the document passes the rule:

    `hr "human resources"`

– The following rule specifies that if a document contains the word *hr* but not the word *benefits*, the document passes the rule:

```
+hr -benefits
```

Figure 1-13 shows an example of a rule-based category definition using document content.

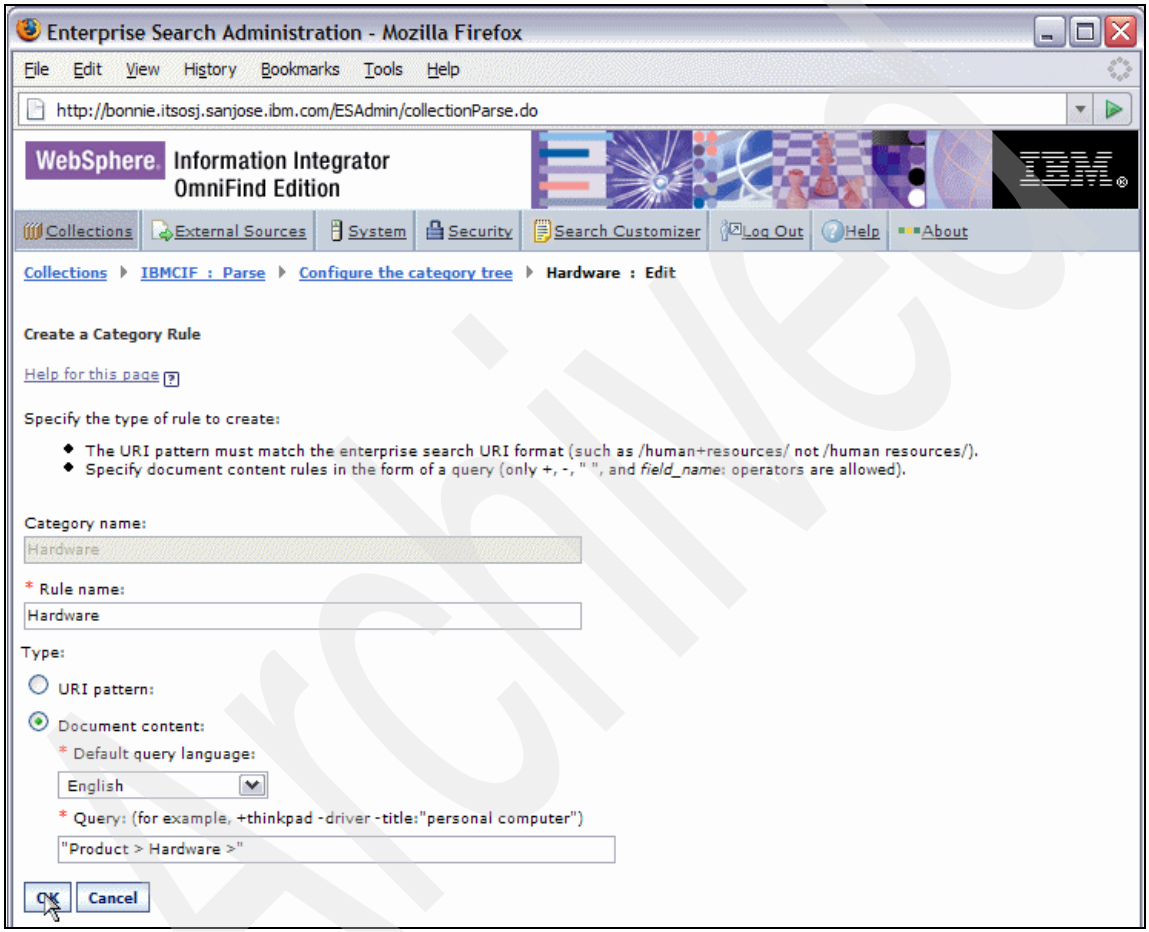

Figure 1-13 Configuring a rule-based category using document content

Attention: After the rule-based categories have been defined and the index created, they may be referenced in a query. Example 1-1 shows how to target a query to one or more rule-based categories, while Figure 1-14 shows how to determine the categoryid for the Products category (c3).

Example 1-1 Specifying a rule-based category in a query

```
(rulebased::<categoryid> OR rulebased::<categoryid2> OR ....) <search string>
      where
      <categoryid1>, <categoryid2> etc are the category ids in a collection
      <categoryid> may be determined from the admin console
      as shown in Figure 1-14, where the categoryid is "c3" for the Technical
      category

For example
(rulebased::c3 OR rulebased::c9) managers
```

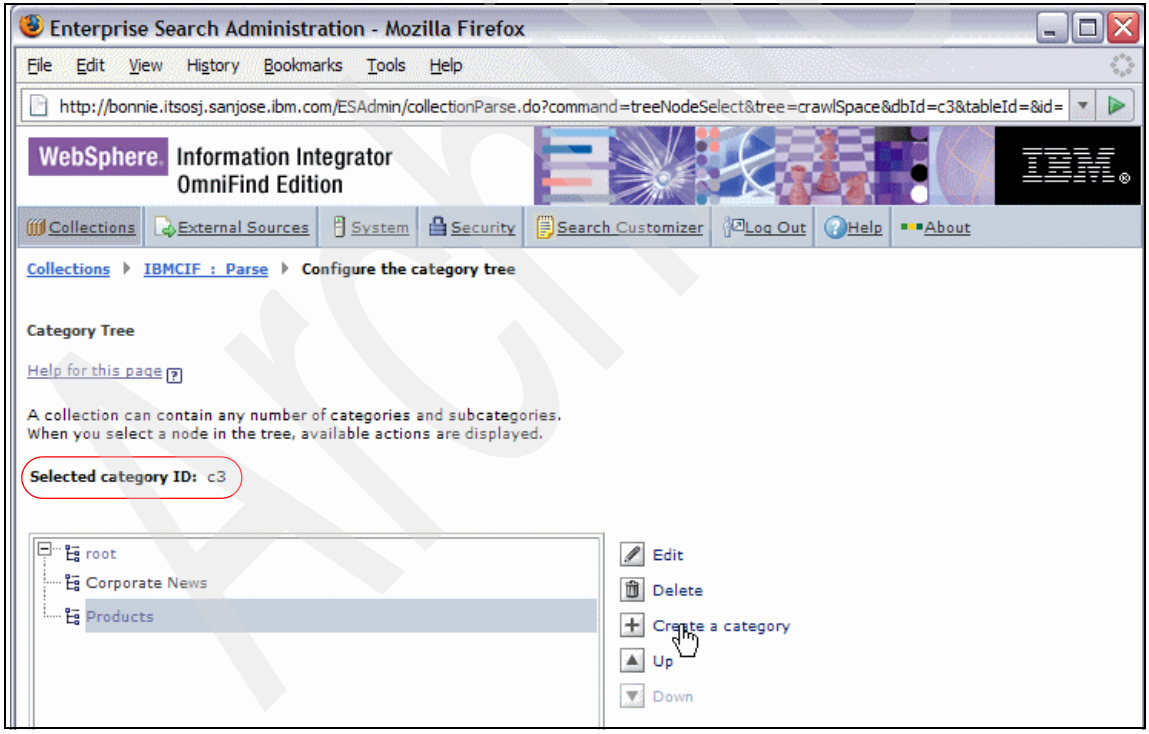

Figure 1-14 Determining the categoryid for Products category (c3)

Important: A category is *not* a security feature. It is *not* possible to restrict access to specific categories within a collection; if a user has access to a collection, then that person automatically has access to all the categories in the collection. A user can target one or more categories in a single query.

The OmniFind Enterprise Edition admin console is used to create and administer categories as follows:

- By selecting the categorization type when creating a collection, one can choose to use no categories, or use rule-based categories.
- When configuring parsing rules for the collection, one can change the categorization type, if necessary. If the categorization type change is made after documents are crawled and indexed, search quality will be degraded until the documents are re-crawled and the main index rebuilt.

Scopes

A scope is a group of related URLs in an index. By configuring a scope, the administrator can limit the documents that users can see in a collection by presenting them with a logical view of a collection. When users search the collection, they search only the documents in the scopes identified in their query.

Note: To configure scopes, one must be a member of the enterprise search administrator role or be a collection administrator for the collection that the scopes belong to.

The OmniFind Enterprise Edition admin console is used to configure scopes for a collection.

When creating scopes, one specifies a set of URIs that are in the index for a collection. Limiting the range of documents that users can search helps ensure that documents in the search results are specific to the information users seek. For example, one might create one scope that includes the URIs for the Technical Support department and another scope that includes the URIs for the Human Resources department.

To use this feature, support for searching scopes must be included in the search applications. Assuming that one's search application supports scopes, users in the Technical Support department will retrieve documents from the Technical Support scope, and users in the Human Resources department will retrieve documents from the Human Resources scope.

One can create as many scopes as one wants, although creating too many scopes can affect performance. In general, configure scopes so that most search requests need to filter only on one or two scopes. Because scopes can contain entire URIs or URI patterns, the same document can belong to more than one scope. When one creates, edits, or deletes a scope, the change becomes effective the next time the main index is rebuilt.

Attention: After the scopes have been defined and the index created, they may be referenced in a query. Example 1-2 shows how to target a query to one or more scopes.

Example 1-2 Specifying a scope in a query

```
(scopes::<scopename1> OR scopes::<scopename2> OR ....) “western region”

      where
      “western region” is the search string
      <scopename1>, <scopename2> etc are the scope names in a collection
```

Attention: Like categories, a scope is *not* a security feature. It is *not* possible to restrict access to specific scopes within a collection; if a user has access to a collection, then that person automatically has access to all the scopes in the collection. Unlike categories however, a user may target more than one scope within a single query.

1.4 What is new in V8.4

IBM OmniFind Enterprise Edition V8.4 and IBM OmniFind Starter Edition V8.4 offer a unique combination of content reach, security, scalability, enterprise integrations, and openness to deliver powerful enterprise search capabilities and a platform for processing unstructured data through text analytics. Both offerings build on the core capabilities of WebSphere Information Integrator OmniFind Edition V8.3, and feature new improvements in the areas of security, usability and performance, and WebSphere Portal integration.

Note: IBM OmniFind Enterprise Starter Edition V8.4 is not limited in functionality as compared to IBM OmniFind Enterprise Edition V8.4, except for its support of only the single-server and two-server configurations.

The new features are briefly described here and are categorized as content reach enhancements, security enhancements, scalability enhancements, usability and performance enhancements, taxonomy enhancements, enterprise integration enhancements, and miscellaneous changes.

1.4.1 Content reach enhancements

New content that can be crawled includes:

- Microsoft Windows SharePoint® Services Service Pack 2 for Windows 2003.
- IBM DB2 UDB for IBM System i™ V5.4
- WebSphere Portal V6.0 Web pages
- Native crawler for IBM Workplace™ Web Content Management (WCM) sites on WebSphere Portal V6 servers.
- JDBC connector to access JDBC-compliant relational data sources through Type 4 JDBC drivers, including IBM DB2 UDB V8.2, Microsoft SQL Server 2000, Microsoft SQL Server 2005, Oracle® 9i, and Oracle 10g.

For a complete and current list of data sources and required client software, visit http://www.ibm.com/software/data/enterprise-search/omnifind-enterprise/requirements4.html

1.4.2 Security enhancements

In IBM OmniFind Enterprise Edition V8.4, the ability to verify authorization with the owning repository at query time is supported for the following additional data sources:

- DB2 Content Manager Multiplatform
- FileNet P8 CM
- Hummingbird DM
- Windows SharePoint Services, Service Pack 2, for Windows 2003
- WebSphere Portal Web Content Management V6.0

Figure 1-1 on page 6 shows the current list of all data sources supporting verification of authorization with the owning repository at query time. For an up-to-date list, visit http://www.ibm.com/software/data/enterprise-search/omnifind-enterprise/

A new feature in IBM OmniFind Enterprise Edition V8.4 is support for single sign-on (SSO) authentication through WebSphere Application Server Lightweight Third-party Authentication (LTPA) for the following data source types:

- Domino Document Manager, Lotus Notes®, and Lotus QuickPlace® databases. The crawlers must use Domino Internet Inter-ORB Protocol (DIIOP) to connect to the Domino servers.
- Portal Document Management repositories accessible by the WebSphere Information Integrator Content Edition connector.
- When users use the IBM OmniFind Enterprise Edition search portlet to search collections from within WebSphere Portal, SSO security is also provided for documents crawled by the native Web Content Management (WCM) and WebSphere Portal crawlers.

For companies that have not yet implemented a full single SSO solution, IBM OmniFind Enterprise Edition continues to provide an optional built-in mechanism called Identity Management Component (IMC) for caching search users' various logins, to minimize the number of sign-in requests.

1.4.3 Scalability enhancements

IBM OmniFind Enterprise Edition V8.4 has the following scalability enhancements:

- IBM OmniFind Enterprise Starter Edition V8.4 is limited to two-processor implementations. If you wish to start with a small implementation, you can seamlessly move the application to OmniFind Enterprise Edition with the purchase of a trade-up license.
- Support for simultaneously performing main index build and delta index build on the same collection.[3] This parallelism capability improves index content latency by incorporating changes occurring in the delta store into the delta index and copying it over to the search server(s), even as the main index is being built.

[3] In V8.3, it was not possible to perform a delta index build (called refresh index in V8.3) while the main index was being built.

1.4.4 Usability and performance enhancements

IBM OmniFind Enterprise Edition V8.4 includes several new features and enhancements that make the product easier to install, use, and service for a faster up-and-running experience as follows:

- It replaces the V8.3 IBM DB2 UDB database with embedded and server open source (Cloudscape) databases for storing crawler metadata and security credentials. The crawled data (raw data store) is no longer stored in the IBM DB2 UDB database on the crawler server, but in file systems in the parser/indexer server. This is both a performance feature as well as a usability feature, because it eliminates the requirement for IBM DB2 UDB database skills to configure and administer the system.

  Figure 1-15 on page 35 shows the V8.4 RDS architecture. DB2 tables have been replaced with a simple file queue solution. The DataListener component that accepts documents from the crawlers and writes them to the collection RDS has been moved from the crawler server to the indexer server. The RDS queue size has a 10 GB default size limit per collection. The RDS queue (actually a file) resides on the indexer server and partitioned. It is made up of 64 MB (default) chunks or bundles. By maintaining a small bundle size, we try to avoid any conflicts that could arise with the file size ulimit being set on UNIX systems. Every time a 64 MB chunk gets filled or the parser driver asks the DataListener for more data, a new 64 MB file is created. Unlike OmniFind V8.3, there is one RDS queue per collection and each RDS queue contains crawled data that belongs to all the crawlers belonging to a specific collection. The RDS queue is removed when the collection (not the crawler) is removed. The RDS queues are platform independent. They can be copied from a Windows platform and read on any UNIX system, and vice versa.

  The RDS file queues for a collection reside in ES_NODE_ROOT/data/<collectionid>/rds/rds_a* on the indexer node. RDS file queues (rds_a.<time stamp>) are deleted by the parser driver when all the documents are parsed.

  Figure 1-15 on page 35 shows the following:

  - Collection col1 crawlers crawler1 and crawler2 extract data and send it to the DataListener component, which then writes it to the collection's RDS input queue col1/rds_a (collection input queue). When this queue col1/rds_a reaches the 64 MB limit or the parser driver asks the DataListener for more data, it is flushed as a 64 MB chunk (col1/rds_a.<time stamp> which are the collection output queues) and a new col1/rds_a RDS queue is created.

    The flushed RDS queue chunks (col1/rds_a.<time stamp>) are read by the parser, which is independent of crawlers being added or dropped.

– The same process occurs for collection col2 and its crawlers. But this collection has its own set of RDS queues.

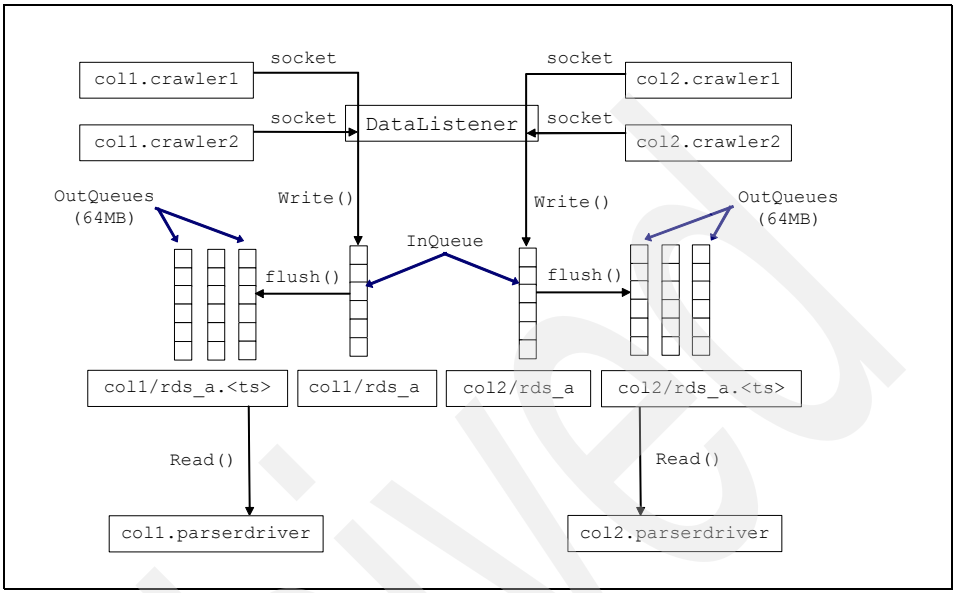

Figure 1-15 Raw data store (RDS) architecture

- Support for a two server configuration where the search component is duplicated on a second server. With such a configuration, one server comprises the crawler, parser, indexer and search components, while the second server only has the search component, as shown in Figure 1-16. This provides an intermediate configuration between the single server and four server configurations (that were supported in V8.3) that scaled the search component without having to provide for a separate server for the crawler component. By making the search component available on two different servers, this option provides high search availability and server redundancy without the requirement to go to a four-server configuration.

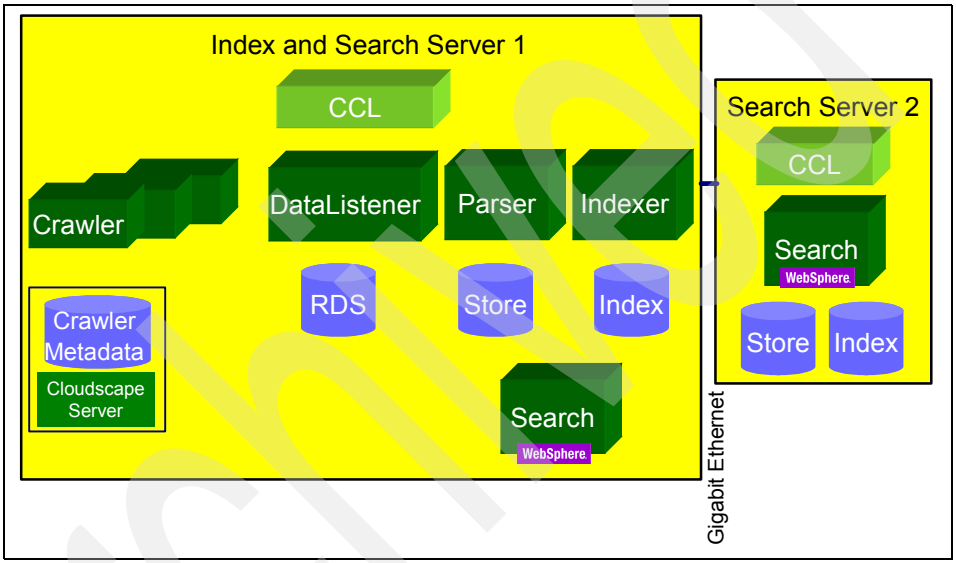

Figure 1-16 Two server configuration

- IBM OmniFind Enterprise Edition V8.4 also includes a number of enhancements to the sample search application as follows:
  - A new tool called the Search Application Customizer is provided to enable customization of the search application through a graphical user interface (GUI), making it easier to tailor the search application for the planned usage.

    Additionally, the sample search application supports the creation of multiple configuration files that may be specified as a parameter during search application invocation. This enables you to customize the sample search application for different users to not only provide a different look and feel but also target different collections for searching, as described in Appendix B, “Search Application Customizer” on page 459.

- Users can now sort search results by selecting the field to sort by, such as title, in addition to sorting by relevance or date. This functionality is implemented through SIAPI, which the search application invokes.
- Another valuable feature taking advantage of the UIMA infrastructure is the capability to use semantic search through synonym expansion, illustrated with sample annotators for phone numbers, Web Addresses, and e-mail addresses, as described in "LSTEP4h: Configure UIMA annotator for CUSTINFO collection" on page 233, "LSTEP4i: Configure synonym dictionary for CUSTINFO collection" on page 247, and 3.3.4, "LSTEP4: Create CUSTINFO and GENINSINFO collections" on page 147.

- Sample Search portlet installation script, as described in Appendix A, "Install Sample Search application portlet" on page 431.
- esadmin startall and stopall scripts that allow you to stop and start all OmniFind components with one single command. These include the CCL server on the local node, WebSphere HTTP server on the search nodes, Information Center on the search nodes, and all sessions that are normally started by the **`esadmin start`** command.

1.4.5 Taxonomy changes

IBM OmniFind Enterprise Edition does not require a predefined taxonomy in order to deliver highly relevant search results. However, it can take advantage of taxonomy tags to influence both the results and interface of a search application. To simplify enterprise search deployments, IBM OmniFind Enterprise Edition V8.4 has the ability to configure rules to control which documents are associated with categories in a collection. It also provides an ability to integrate the IBM Classification Module component of IBM OmniFind Discovery Edition (previously known as WebSphere Content Discovery Server) into IBM OmniFind Enterprise Edition as a UIMA annotator.

Model-based categories are no longer supported in IBM OmniFind Enterprise Edition V8.4.

1.4.6 Enterprise integration enhancements

In addition to the broad set of enterprise connectors, IBM OmniFind Enterprise Edition also offers a full search and indexing API, and supports UIMA. IBM OmniFind Enterprise Edition provides an open platform for integrations on the front end, to back-end data sources and systems, and to information extractors during the indexing pipeline.[4] OmniFind is unique in that it exposes the heart of its indexing pipeline through an open source API. UIMA allows pluggable annotators to be inserted into the standard document indexing flow in order to infer and extract searchable structured metadata from content. In addition to the search capability, this extracted knowledge can also be used to provide stand-alone reports or to drive analytics applications.

IBM OmniFind Enterprise Edition V8.4 enhancements include:

- In addition to a stand-alone search application, IBM OmniFind Enterprise Edition includes WebSphere Portal integrations that can extend the standard search capability of WebSphere Portal. This allows for a single search interface within WebSphere Portal to cover a full set of enterprise content repositories in a secure manner.

  Integrations to desktop search tools, such as Google Desktop and X1, were available in V8.3.

- IBM OmniFind Enterprise Edition V8.4 provides enhanced integration to WebSphere Portal V6, including:
  - A native Web Content Management crawler for crawling IBM Workplace Web Content Management™ (WCM) sites using native security.
  - Support for the new WebSphere Portal V6 Search Center, making IBM OmniFind Enterprise Edition collections searchable through the Search Center portlet.
  - Offers the ability to redirect queries from the WebSphere Portal Search bar or Search Center to the IBM OmniFind Enterprise Edition search portlet.
  - As described earlier, an automated command-line script with which the IBM OmniFind Enterprise Edition search portlet is easily and quickly deployed.

[4] UIMA is the foundation of the parser processing pipeline. Every parser processing action, like language detection, tokenization, lemmatization, and categorization, are in fact UIMA modules. You can add to this indexing pipeline with your own custom or off-the-shelf annotators.

1.4.7 Miscellaneous changes

There are some miscellaneous changes in terminology as well. The Reorganize Index process in the administration GUI is now renamed to Main Index, while Refresh Index is now called Delta Index.

1.5 Security considerations

In enterprise search products, such as IBM OmniFind Enterprise Edition, where information about documents throughout an enterprise are made accessible to users, it is critical that stringent security safeguards be implemented to ensure that only authorized persons have access to sensitive information. IBM OmniFind Enterprise Edition addresses this requirement in several ways, as discussed in this section.

To ensure that only users who are authorized[5] to access content do so, and to ensure that only authorized users are able to access the administration console, IBM OmniFind Enterprise Edition coordinates and enforces security at several levels, such as Web server, collection level security, document level security, and encryption as follows:

- Web server

  This is the first level of security. If you enable global security in WebSphere Application Server, you can assign users to administrative roles[6] and authenticate[7] users who administer the system. When a user logs in to the administration console, only the functions and collections that the user is authorized to administer are available to that user.

[5] Authorization is the mechanism by which a system grants or revokes the right to access some data, or perform some action. Typically, the steps involved are that a user first logs in to a system with a user ID using some authentication system. Then the authorization mechanism controls what operations the user may or may not perform by comparing the user ID to an access control list. Access control refers to limiting what users can do after they identify themselves and are authenticated. An access control list (ACL) is the most common way in which access to resources is limited. An ACL is a list of user identifications (user names, group names, user roles, and so on). Each user identification is associated with a set of permissions that define the user's rights and privileges.

[6] Roles are an abstract logical grouping of users that are predefined by the IBM OmniFind Enterprise Edition to have certain privileges.

[7] Authentication is the process by which a system verifies that users are who or what they declare themselves to be. Because access is typically based on the identity of the user who requests the resource, authentication is essential to effective security. Enterprises generally perform user authentication during the login process. Typically, the user identifies himself or herself through a user ID and associated password. The authentication process is a part of the IT infrastructure and is typically performed by the Web server or application server in conjunction with a user registry, such as an LDAP server.

Search applications can also use WebSphere Application Server security mechanisms to authenticate users who search collections.

- Collection-level security

  You can enable security at the collection level when you create a collection; this setting cannot be changed after the collection is created. If you do not enable collection-level security, you cannot later specify document-level security controls.

  When collection-level security is enabled:

  – You can configure options to enforce document-level security, such as associating security tokens with documents as they are crawled, requiring current credentials to be validated during query processing, and specifying whether anchor text in Web documents is to be indexed.

  – You can enforce security by mapping search applications (not individual users) to the collections and external sources that they can search. You then use standard access control mechanisms to permit or deny users access to search applications.

- Document-level security

  You can enable document-level security when you configure crawlers for a collection, You can specify options to associate security tokens with data as the data is collected by crawlers. These tokens are stored with the documents in the index. Your search applications must provide at least one of these tokens, in order to be able to access these documents. This ensures that only users with the proper credentials are able to query the data and view search results.

  For certain types of data sources, you can additionally choose to validate a user's login credentials with current access controls at the back-end data sources during query processing; this option is sometimes called impersonation. This extra layer of security ensures that a user's privileges are validated in real time with the native data source. This capability can protect against instances in which a user's credentials change after a document and its security tokens are indexed.

- Encryption

  To protect sensitive data, encryption is used to encode the authentication data portion of all messages that are transmitted through the enterprise search system. This is to prevent a malicious and unauthorized user from gaining access to data while it is in transit. For example, you can configure the search servers use protocols such as the Secure Sockets Layer (SSL) for communications between the search application and search runtime.

All passwords that are stored by the system (in configuration files such as the password for the enterprise search administrator, which is specified when the product is installed, and the credentials database when identity management is enabled) are also encrypted. The data in the index is not encrypted; however, it is not in clear text, but compressed binary format.

Attention: For increased security, you need to ensure that the server hardware is appropriately isolated and secure from unauthorized intrusion. By installing a firewall, you can protect the enterprise search servers from intrusion through another part of your network. Also ensure that there are no open ports on the enterprise search servers. Configure the system so that it listens for requests only on ports that are explicitly assigned to enterprise search activities and applications.

Figure 1-17 on page 42 provides a high-level overview of IBM OmniFind Enterprise Edition's security features. Each of the features shown in Figure 1-17 on page 42 are briefly described here. The security controls involved depend upon whether WebSphere Application Server global security is enabled or not.

The main topics discussed here are as follows:

- WebSphere Application Server global security is not enabled.
- WebSphere Application Server global security is enabled.
- Encryption security.
- Processing flow involving pre-filtering and post-filtering.

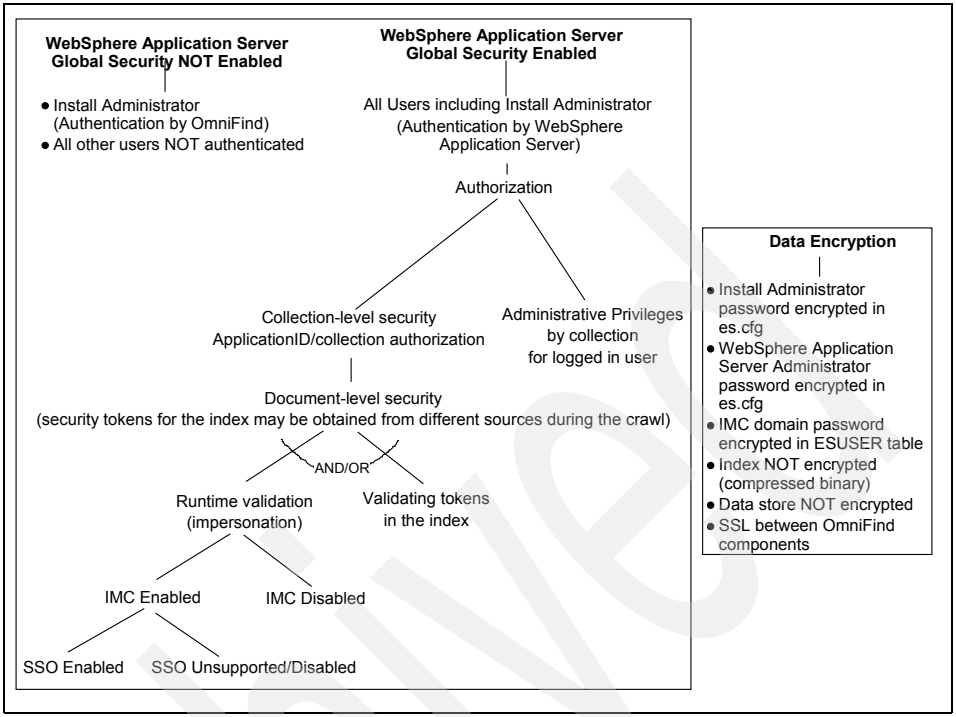

Figure 1-17 Security overview

1.5.1 WebSphere Application Server global security is not enabled

When WebSphere Application Server global security is not enabled, the only security provided is the authentication of the install enterprise search administrator by IBM OmniFind Enterprise Edition, and the encryption of the corresponding password in the es.cfg configuration file.

The messages that are transmitted through the enterprise search system are not encrypted either.

As shown in Figure 1-17, collection-level security or document-level security do not apply when WebSphere Application Server global security is not enabled.

1.5.2 WebSphere Application Server global security is enabled

When WebSphere Application Server global security is enabled, WebSphere Application Server is responsible for the authentication of all users accessing OmniFind, including the install enterprise search administrator. The messages that are transmitted through the enterprise search system are also encrypted, as are the credentials stored in the profile database where the identity management component stores user credentials.

As shown in Figure 1-17 on page 42, when WebSphere Application Server global security is enabled, the following broad tasks can be performed:

- Authenticated users can perform administrative functions
- Collection-level security can be implemented

Authenticated users can perform administrative functions

Authenticated users can perform administrative functions at a collection-level[8] if they have administrative roles assigned to them. Roles are an abstract logical grouping of users that are predefined by IBM OmniFind Enterprise Edition. By assigning users to roles, one can restrict access to specific collections and control the functions that each administrative user can perform. The user IDs that one assigns to administrative roles in IBM OmniFind Enterprise Edition must exist in a WebSphere Application Server active user registry.

Administrative roles are assigned through the administration console by a user that has the privilege to do so. The administrative roles that can be assigned to a user are as follows:

- Enterprise Search administrator

  This role has super user access to all administrative functions of the IBM OmniFind Enterprise Edition system. These users create collections and have the authority to administer all aspects of the IBM OmniFind Enterprise Edition search system.

  As mentioned earlier, when IBM OmniFind Enterprise Edition is installed, the installer specifies the user ID and password for the first enterprise search administrative user. This user can assign other users to the enterprise search administrator role.

- Collection administrator

  This role can edit, monitor, and control the operation of specific collections or all collections. These users cannot create collections or administer components that span collections.

[8] The enterprise search administrator role has privileges over all the collections, unlike the collection administrator, operator administrator, and monitor administrator roles that may be granted privileges over specific collections.

- Operator

  This role can monitor system activity and control the operation of specific collections or all collections. For example, these users can start and stop collection activity, but they cannot create collections, edit collections, or administer components that span collections.

- Monitor

  This role can monitor system activity for specific collections or all collections. They cannot control operations (such as the starting and stopping component), create collections, edit collections, or administer components that span collections.

Collection-level security can be implemented

Besides the administrative roles that can be restricted to operate on specific collections, IBM OmniFind Enterprise Edition collections can also be restricted to specific search applications by the Enterprise Search administrator or Collection administrator. Search applications are client programs that issue search requests using the IBM Search API (SIAPI). These programs may be IBM OmniFind Enterprise Edition supplied or custom developed.

The ability to search different collections is controlled by mapping search applications to the collections and external sources that they can search. An application named Default enables the sample search application to be used as provided, to search all collections and external sources. All search applications are required to pass a valid application name (APPID) to the enterprise search application programming interface (SIAPI). Only the collections and external sources associated with this APPID can be searched by the search application.

Note: Before a search application can access a collection or external source, an enterprise search administrator must associate the search application (APPID) with the specific collections and sources that it can search.

A search application can search all of the collections and external sources in an enterprise search system, or search only the collections and external sources that you specify.

The sample search application (ESSearchApplication) has a properties file that specifies the application name to use. The default location for this properties file is ES_INSTALL_ROOT\installedApps\ESSearchApplication.ear\ESSearchApplication.war\WEB-INF\config.properties.

The initial value for the application name is Default. If you change this value, you change the list of collections and external sources that the ESSearchApplication application can search.

Note: When collection-level security is enabled, global analysis processes do not identify duplicate documents in the collection.

As shown in Figure 1-17 on page 42, when collection-level security is implemented, document-level security controls can also be implemented, as described in “Document-level security controls” on page 45.

Document-level security controls

Document-level security control is enabled by selecting the “Enable security for the collection” drop-down option during the creation of a collection, as shown in Figure 2-24 on page 81.

Document-level security control ensures that users searching for documents can only access those documents that they are authorized to see.

OmniFind supports many approaches for configuring document-level security controls as follows:

- Documents can be pre-filtered and associated with security tokens at crawl time before they are added to the index. The security token (with the exception of the default token) may represent a valid operating system group ID, user ID, or any other value as determined by the Enterprise Search administrator.

  Note: By default, each document without a token is considered to be a public document and therefore available to everyone.

  IBM OmniFind Enterprise Edition provides selected combinations of four methods to replace the public security tokens with a different value:

  a. The value that can be extracted from the access control lists for certain data sources.
  b. The value that can be specified by the Enterprise Search administrator using the administration console.
  c. The value can be extracted from an Enterprise Search administrator-designated field in the crawled document.

d. For most crawler types, a custom Java class (plug-in) can be used to associate security tokens with documents in the index. This API provides an entry-point to facilitate deployment and integration of IBM OmniFind Enterprise Edition in the existing security infrastructure. The goal is permit the application of business and security rules to achieve document-level security. The security plug-in is supported for all the crawlers except the Web crawler (HTTP/HTTPS) and the newsgroup (NTTP) crawler.

   The intent of this API allows customers to write a Java routine that implements logic to gather Access Control Lists (from data sources that OmniFind does not currently automatically support ACL extraction) when crawling documents. For each document fetched from the data source, the IBM OmniFind Enterprise Edition crawlers will call the Security token plug-in, which will return the security tokens to be associated with the document in the IBM OmniFind Enterprise Edition index. The implementation of the security token plug-in is open and can be used to access the data sources to gather the ACLs or to access a central repository managing Access controls, such as Tivoli Access Manager.

Note: For documents crawled by a Web crawler, the anchor text in documents that contain links to forbidden documents can be excluded from the index.

- For some data types, search results can be post-filtered to validate the user's login credentials against current access control data. The enterprise search identity management component (IMC) can encrypt the various credentials that users need to access different repositories, and store the encrypted credentials in profiles. If the sources to be searched are protected by a product that provides single sign-on (SSO) security, the IMC can control access to documents without requiring users to create profiles. This is described in more detail later in 1.5.4, "Processing flow involving pre-filtering and post-filtering" on page 47.

1.5.3 Encryption security

During the administration of a collection, there are instances where the administrator is asked to provide sensitive information, such as passwords used by the crawlers to access the back-end data sources.

All supplied passwords are masked when they are entered or displayed in the administration console. In addition, the passwords are further encrypted using industry proven techniques before being stored in the configuration database. Encryption provides additional protection of the passwords even if access is granted to the IBM OmniFind Enterprise Edition machines.

When IMC is enabled, the passwords of the credentials stored in the IMC Cloudscape database are encrypted.

Note: The password for the enterprise search administrator user ID specified during IBM OmniFind Enterprise Edition installation is also stored in encrypted format.

1.5.4 Processing flow involving pre-filtering and post-filtering

As mentioned earlier, there are two distinct approaches to filtering documents to ensure that search results contain only the documents that the user who submitted the search request is authorized to view:

- The first approach is to replicate the document's native access control lists (ACLs) at crawl time into the index and to rely on the search engine to compare user credentials to the replicated document ACLs. Pre-filtering the documents, and controlling which documents are added to the index, results in the best performance. However, it is difficult to model all of the security policies of the various back-end sources in the index and implement comparison logic in a uniform way. This approach is also not as responsive to any changes that might occur in the source ACLs.
- The second approach is to post-filter documents in the result set by consulting the back-end sources for current security data. This approach allows the contributing back-end sources to be the final arbiters of the documents returned to the user, and ensures that the result set reflects current access controls. However, this approach results in degraded search performance because it requires that connections exist with all of the back-end sources. If a source is not accessible, then links to documents must be filtered out of the result set along with documents that the user is not authorized to view.

Important: In a multiple server configuration, post-filtering is done at the crawler server for some source types. If the crawler server is brought down for maintenance, users experience no results when they query enterprise search collections. In addition, no results are returned if the back-end servers that are required to control access are not accessible.

For OmniFind, support for enforcing access controls relies on a combination of these two approaches. The design provides optimum performance while maintaining the precise security policies of the originating document repositories. By storing high-level access control data in the index, the system can provide an interim (potentially smaller) result set that can then be post-filtered to verify current access controls. The assumption is that if the user has access to the repository that owns the document, then chances are that the user also has access to the document.

Note: The access control data that is stored in the index varies with the crawler type. For example, the Notes crawler can store database-level and server-level access controls, and the QuickPlace crawler can store access controls for servers, places, and rooms.

All data source types in an OmniFind system support the ability to index native access control lists during crawl time. As mentioned earlier, some data source types also support the ability to post-filter the result set and verify the user's current credentials.

The two-pronged security design encompasses the following tasks, as shown in Figure 1-18.

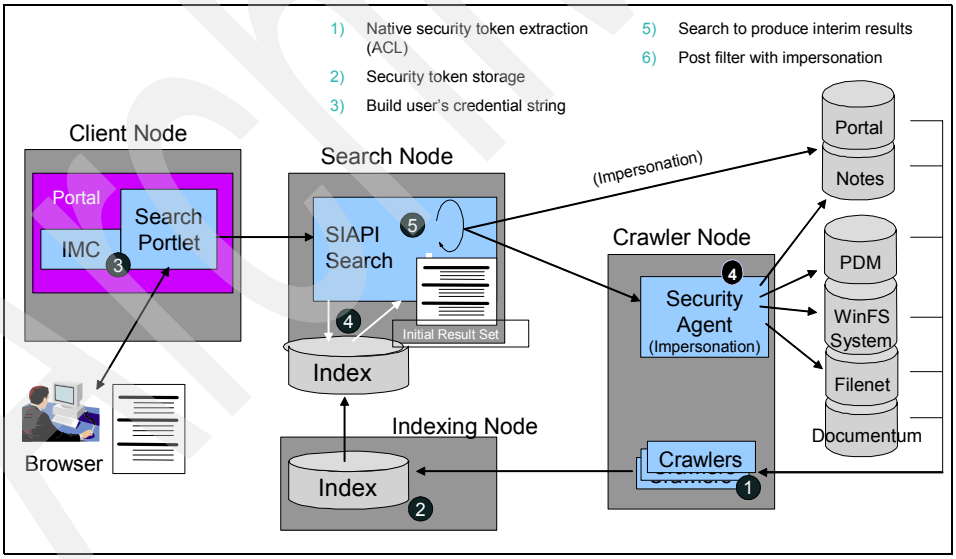

Figure 1-18 Pre-filtering and post-filtering processing flow

Each of the steps in the pre-filtering and post-filtering process shown in Figure 1-18 on page 48 is briefly described here:

1. Extracting native ACL information during crawl time.
2. Storing server and database ACL information in the index.
3. Creating the user's security context when the user logs in or when the session is initialized. This task must account for the different identifiers that a single user must use to access the various back-end sources.
   - If you disable IMC, then it is the responsibility of your search application to generate the user security context (USC) string. After it is generated, the USC string is used to set the ACL constraints value on each query.
   - If IMC is enabled (this is the default), the search servers can use the following approaches to validate a user's current credentials during query processing.
     - The search application can prompt the user to register the credentials that they need to access various domains in a user profile. The profile, which is encrypted and stored in a secure data store, enables the user to search the secure domains. If credentials are not specified for a domain that requires current credentials to be validated, documents from that domain are excluded from the search results.
     - Additionally, if documents in a collection were crawled by a crawler that provides support for single sign-on (SSO[9]) security, and you specify that you want to use SSO security to control access to documents, the system will use SSO security methods to authenticate users for the duration of a search session. The user does not need to create a profile that specifies credentials or provide a user ID and password when searching secure domains.

[9] SSO authentication enables a user to be authenticated one time and gain access to many resources without being prompted to present credentials again. In an enterprise search system, SSO authentication eases the burden of managing the many user names and passwords that users must specify to access documents in secure collections. IBM WebSphere Application Server and Lotus Domino support a form of SSO that is known as Lightweight Third-Party Authentication (LTPA). When a user attempts to access either product, the user is asked to authenticate with a user name and password. This user name and password are verified against an LDAP repository that both products share. After the user is authenticated, a session cookie is created to contain the LTPA token. The user can then access other resources on any server that has the same authentication configuration without being prompted to specify credentials again. This token persists as long as the browser session is valid.

To validate a user's credentials, the IMC must obtain the user's group information for each of the user's identities and add this information to a user security context (USC) string. This group information[10] is used to filter results in accordance with access control data that is stored in the enterprise search index or in accordance with SSO authentication data. The IMC does this by using SSO tokens or by using the user's credentials to connect to the back-end system and request the groups for which the user is a member.

When users search collections that require current credentials to be validated, the system can use SSO security methods to deny or permit access to documents. Users are not prompted for credentials when they search sources that support SSO authentication. The IMC is used if all of the following conditions are true:

- SSO is properly enabled in WebSphere Application Server and the target domains.
- Security is enabled in at least one of the collections that the search application can search.
- The options to use the IMC and SSO security are enabled in the enterprise search administration console.
- The option to use SSO security and options to enforce document-level security (such as indexing access controls or validating current credentials during query processing, as shown in Figure 2-37 on page 95) were selected when the certain crawler types (Content Edition[11], Domino Document Manager[12], Notes[13], and QuickPlace[14]) were configured.

Note: When users use the Search portlet for enterprise search to search collections from within WebSphere Portal, SSO security is also provided for documents crawled by the Web Content Management (WCM) and WebSphere Portal crawlers.

[10] When you configure IMC options in the administration console, you can specify how often this group information is to be refreshed. You can extract new group data each time that the user logs in to the search application, or you can extract the group data on a regular basis, such as every three days.

[11] Available for Portal Document Manager repositories only
[12] Available for crawlers that use the DIIOP protocol only
[13] Available for crawlers that use the DIIOP protocol only
[14] Available for crawlers that use the DIIOP protocol only

4. Processing the search with the user's security context and producing an interim result set that contains only those documents that the user has access to at the repository level. If no security token is supplied, then the default public token is automatically applied during search. The security token(s) stored with the documents are compared to the security token(s) sent in from the search application. Only those documents that match the security token specification are returned. The search application has the flexibility to specify inclusion and exclusion in the list of security tokens.

   Note: If a document in the index has the default token and contains matching search terms, then that document is returned in the search result bypassing the validation of credentials during query processing for this document.

5. Post-filtering the interim result set by consulting the back-end sources that contributed documents to the result set for current native ACL information. As mentioned earlier, in a multi-server configuration, post-filtering is done at the crawler server for some data sources, as shown in Figure 1-18 on page 48.

1.6 Choosing a particular topology

IBM OmniFind Enterprise Edition provides a single-server, two-server, and a four-server configuration on a variety of platforms, including IBM AIX, Microsoft Windows, Sun Solaris, and Linux.

The main reason for the multi-server installation is to provide high availability. But it also allows for balancing the workload over multiple servers. For example, if more than one base component is busy all the time during production, and the hardware configuration used does not have enough capacity to serve all components at the same time. In this case, a multi-server installation can separate the active components to different physical machines.

Note: This also applies to a two-server configuration, where one machine still runs all components.If high search performance is required, but the crawl, parse, and index workload is too much to handle on a single machine, the search workload could instead be offloaded to the additional search server while the other workload is handled by the main server. In this scenario, if the second search server goes down, the main server can still serve as a search backup with reduced search performance.

Examples of multi-server scenarios include:

- High availability requirement for search servers.
- Maintaining search performance when creating new indexes takes a long time and uses most of the available resources of the indexer server.
- Increase the possible query load.
- When lots of collections have to be active concurrently and the used hardware configuration (memory and CPU) cannot scale adequately.
- When lots of or very large data sources have to be crawled and the document update rate is quite high, the crawler server is busy all the time.

IBM OmniFind Enterprise Edition V8.4 has significant functionality, scalability, and performance enhancements that impact the selection of a particular configuration for a given business requirement. We are currently in the process of developing capacity planning guidelines for IBM OmniFind Enterprise Edition V8.4, and will make this available on the following Web site in the near future

http://www-306.ibm.com/software/data/enterprise-search/omnifind-enterprise/support.html

Important: For users interested in more accurate projections, we recommend that you set up a representative hardware and workload environment and measure and extrapolate the monitored response times, throughput and memory, CPU, and disk space utilization.

2

Small business OmniFind scenario on a Windows 2003 Enterprise Edition platform

In this chapter, we describe a step-by-step approach to implementing IBM OmniFind Enterprise Edition in a single server Microsoft Windows 2003 Enterprise Edition platform for a hypothetical insurance company.

The topics are:

- Business requirement
- Environment configuration
- Configure the environment

2.1 Business requirement

Our fictitious company Northwest Insurance is a fast growing insurance company that has plans in the future to expand through acquisitions of other insurance companies in different geographic markets and domains, such as auto insurance and home insurance.

Northwest Insurance is a general insurance company offering life and home insurance products in the Pacific Northwest region of the United States, with an employee force of less than a 1000 supporting 20,000 customers. In a competitive marketplace, Northwest Insurance needed to provide superior customer service by improving the productivity of its 300 strong sales, marketing and technical personnel by making customer information spread across multiple data sources available on demand in response to customer and prospect inquiries. Such a system would also promote cross selling and up selling among its customers.

One of the solutions aimed at achieving this objective was to implement an enterprise search system for the 20,000 customer-related documents located in its Windows file system and Lotus QuickPlace systems. Given the sensitive nature of customer information, Northwest Insurance requires the enterprise search solution to support and leverage the native security capabilities of the underlying data sources; unauthorized access to secure documents had to be prevented. The solution also needs to provide a simple GUI interface available as a portlet in WebSphere Portal Server.

From an IBM OmniFind Enterprise Edition implementation perspective, this translates to having:

- Enabled WebSphere global security with an LDAP repository.
- A single collection with collection security enabled in order to enforce document-level security and single sign-on.
  - The Lotus QuickPlace system contains information about product offerings, corporate news, and discussion rooms.
  - The Windows file system contain information about policy and claim details, application forms, customer correspondence, and internal presentations.
- Enable Identity Management Component (IMC) and single sign-on.
- A Windows file system crawler and Lotus QuickPlace crawler.
- The sample search portlet installed on WebSphere Portal Server.

2.2 Environment configuration

Northwest Insurance's workload demands (small employee community, number of documents to be indexed, and the absence of a need for a high availability environment) permit the adoption of a sufficiently configured single server IBM OmniFind Enterprise Edition environment. The Windows 2003 Enterprise Edition platform is considered sufficient to address Northwest Insurance's enterprise search solution needs.

Figure 2-1 on page 56 shows the configuration used in the Northwest Insurance single server Windows 2003 configuration, including:

- A Windows 2003 server (kazan.itsosj.sanjose.ibm.com) provides Northwest Insurance's enterprise portal through which authorized users will access the enterprise search solution. This server uses LDAP (Tivoli Directory Server) to address the security requirements of the enterprise search solution.
- Tivoli Directory Server (boron.itsosj.sanjose.ibm.com) is installed on a separate server that is physically well secured, given the sensitive nature of the information it contains.
- Single Windows 2003 server (nile.itsosj.sanjose.ibm.com) for the IBM OmniFind Enterprise Edition search, indexer, parser, and crawler components.
- The Windows file system data source is located on the same server as IBM OmniFind Enterprise Edition, while the Lotus QuickPlace data source is installed on another server (kazan.itsosj.sanjose.ibm.com).
- The various crawler types defined in this configuration.

> **Note:** The Windows 2003 server has direct connectivity and authorization to the data sources in order to render the document that a user selects in the search result.

- IBM OmniFind Enterprise Edition administrators administer and manage the environment through the administration console GUI.

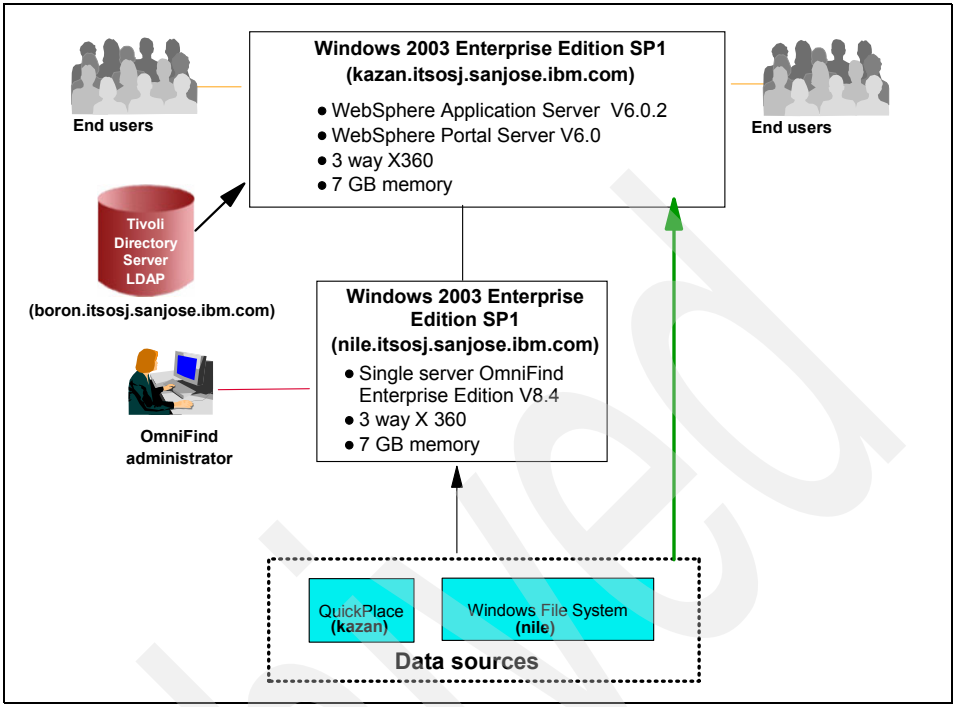

Figure 2-1 Northwest Insurance's single server Windows 2003 enterprise search solution

2.3 Configure the environment

In this section, we document the step-by-step configuration of the single server Windows 2003 configuration in our fictitious company Northwest Insurance. Figure 2-2 on page 57 lists the main steps involved in configuring this environment. First, the administrator and users that need to access the IBM OmniFind Enterprise Edition environment must be defined in the Tivoli Directory Server LDAP repository. Next, global security must be enabled in the WebSphere Application Server of the IBM OmniFind Enterprise Edition search server. Once global security is enabled, the es.cfg configuration file must be updated with the WebSphere Application Server user ID and password. Northwest Insurance's collection can now be created (with collection security enabled to enforce document-level security), populated, and queried using the sample search Web application, and sample search portlet.

Each of these steps is described in detail in the following subsections.

Note: We assume that you have verified that the IBM OmniFind Enterprise Edition has had the proper prerequisites correctly installed in the Windows 2003 server.

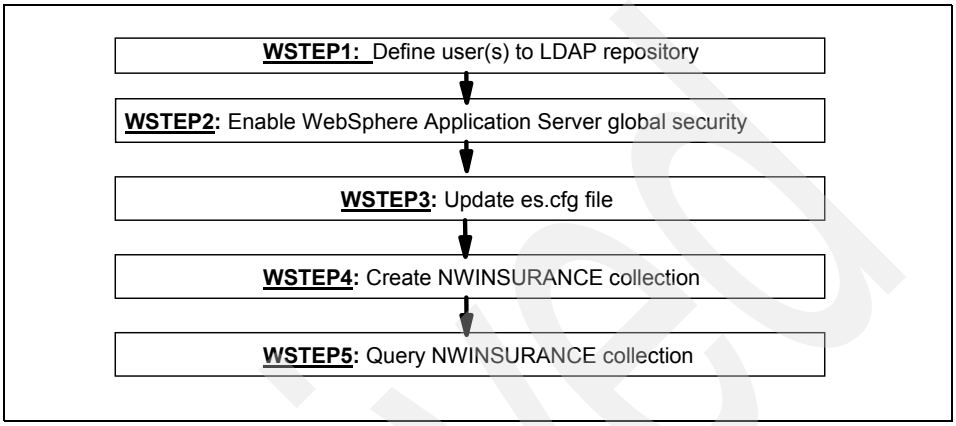

Figure 2-2 Steps to configure Northwest Insurance's single server configuration

Attention: In all the following sections, for the purposes of avoiding screen capture overload, we have *not* included all the windows that you would typically navigate through in order to perform the desired function. Instead, we have focused on including select screen captures (and in some cases, portions of selected windows) that highlight the key items of interest, thereby skipping both initial as well as intervening windows in the process.

2.3.1 WSTEP1: Define users in LDAP repository

In this step, we add the user esadmin and its associated password to the Tivoli Directory Server (TDS) LDAP repository using the Tivoli Directory Server Web Administration Tool.

Figure 2-3 on page 59 shows the login window to the Tivoli Directory Server Web Administration Tool with the appropriate user ID and password.

Attention: We assume that the LDAP tree root and user and group object templates have been appropriately configured.

Note: All LDAP related entries depend on an exact LDAP server configuration. In our case, the LDAP tree starts with ou=itso,o=ibm object. Two objects (cn=users for users and cn=groups for groups) of class container were created below this object. Next, a realm (cn=itso_realm) was configured, which specifies default locations for the newly created user and group objects and their templates, including the choice of RDN™[a] (prefixes) and object classes. We chose as a template object class "inetOrgPerson", and RDN attribute uid. This results in the full distinguished names (DNs) of newly created user objects looking like uid=username,cn=users,ou=itso,o=ibm. For groups, we chose the object class "group" and RDN attribute cn. Full DNs of newly created groups will look like cn=groupname,cn=groups,ou=itso,o=ibm. Based on these configuration, the following LDAP configurations are derived:

- User prefix: uid
- Group prefix: cn
- Base DN: ou=itso,o=ibm
- User filter: (&(uid=%V)(objectClass=inetOrgPerson))
- Group filter: (&(cn=%V)(objectClass=group))

User and group filters are configured in the "Advanced LDAP user registry setting" window. For further details, refer to the IBM Redbooks publication *Understanding LDAP - Design and Implementation*, SG24-4986.

a. Relative distinguished name

After a successful login, navigate to the Add user window by expanding the Users and groups folder in the navigation pane, as shown in Figure 2-4 on page 60. Add the user ID of esadmin and other required information for this window, and click **Finish** to complete the addition of this user to the LDAP repository.

After the esadmin user ID has been added, you need to specify the password for the esadmin user, as shown in Figure 2-5 on page 61 through Figure 2-9 on page 65. This involves a series of steps as follows:

1. Navigate to the Manage entries item in the Directory management folder in the navigation pane, select the ou=itso,o=ibm entry in the relative distinguished name (RDN) column, and click **Expand**, as shown in Figure 2-5 on page 61.
2. Select the cn=users entry in the RDN column, and click **Expand**, as shown in Figure 2-6 on page 62,
3. Select the uid=esadmin entry in the RDN column, and click **Edit attributes...**, as shown in Figure 2-7 on page 63.

4. Click the **Optional attributes** link, as shown in Figure 2-8 on page 64.
5. Provide the password in the userPassword field, and click **OK** to complete the addition of a user ID to the LDAP repository.

Note: This process will have to be repeated for every user ID that needs to access IBM OmniFind Enterprise Edition. You will also need to add users for the administration of WebSphere Application Server, namely, wasadmin (WebSphere Application Server console administrator ID) and ldapbind (WebSphere Application Server bind ID).

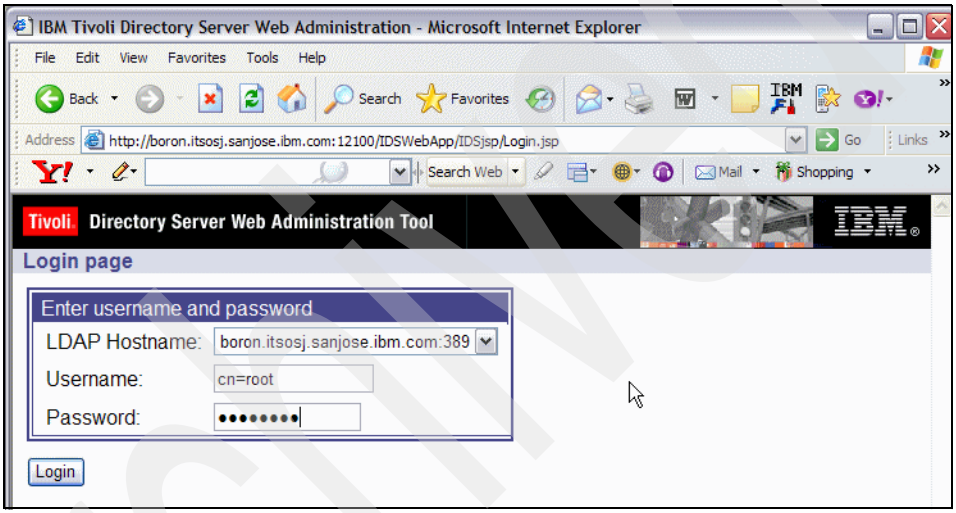

Figure 2-3 Log in to the Tivoli Directory Server Web Administration Tool

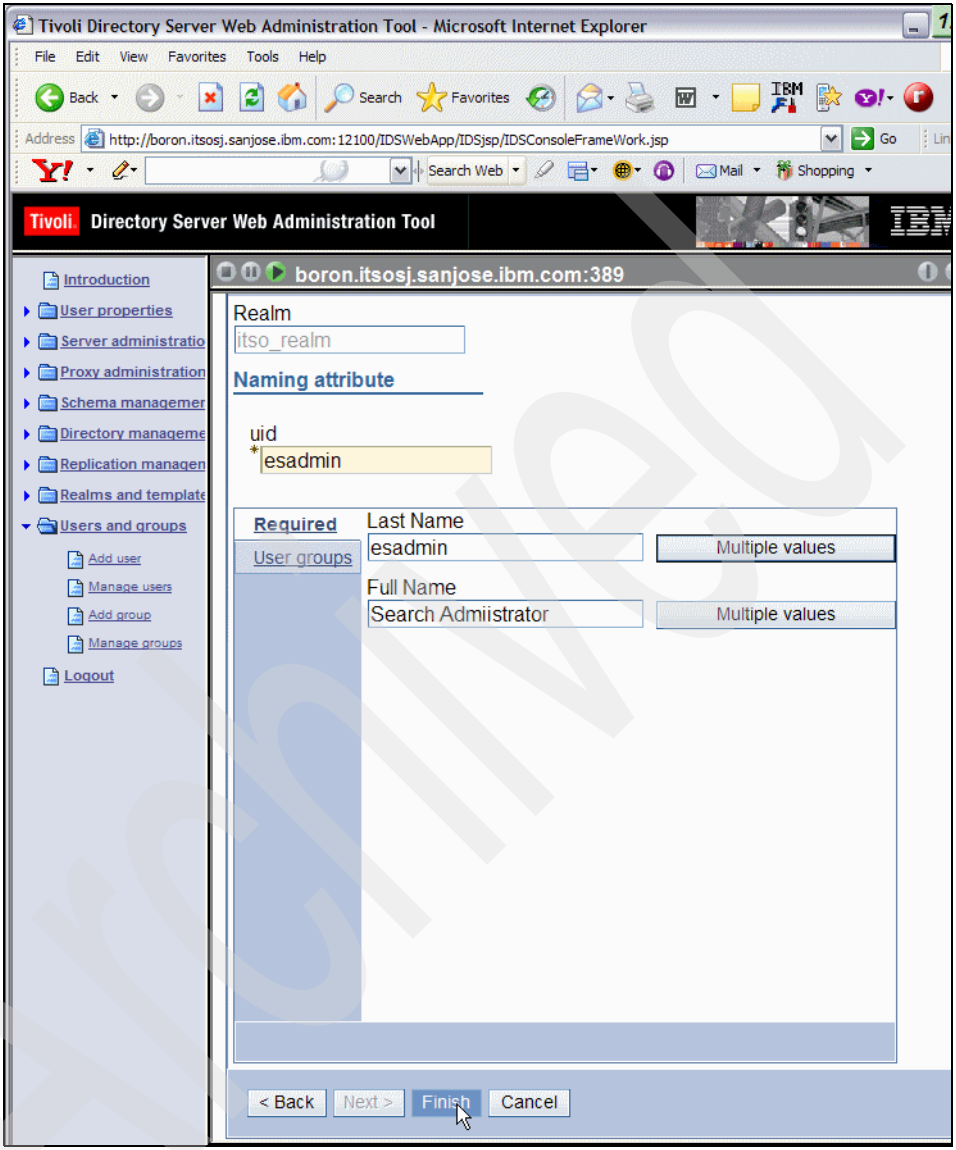

Figure 2-4 Add the esadmin user ID to the itso realm

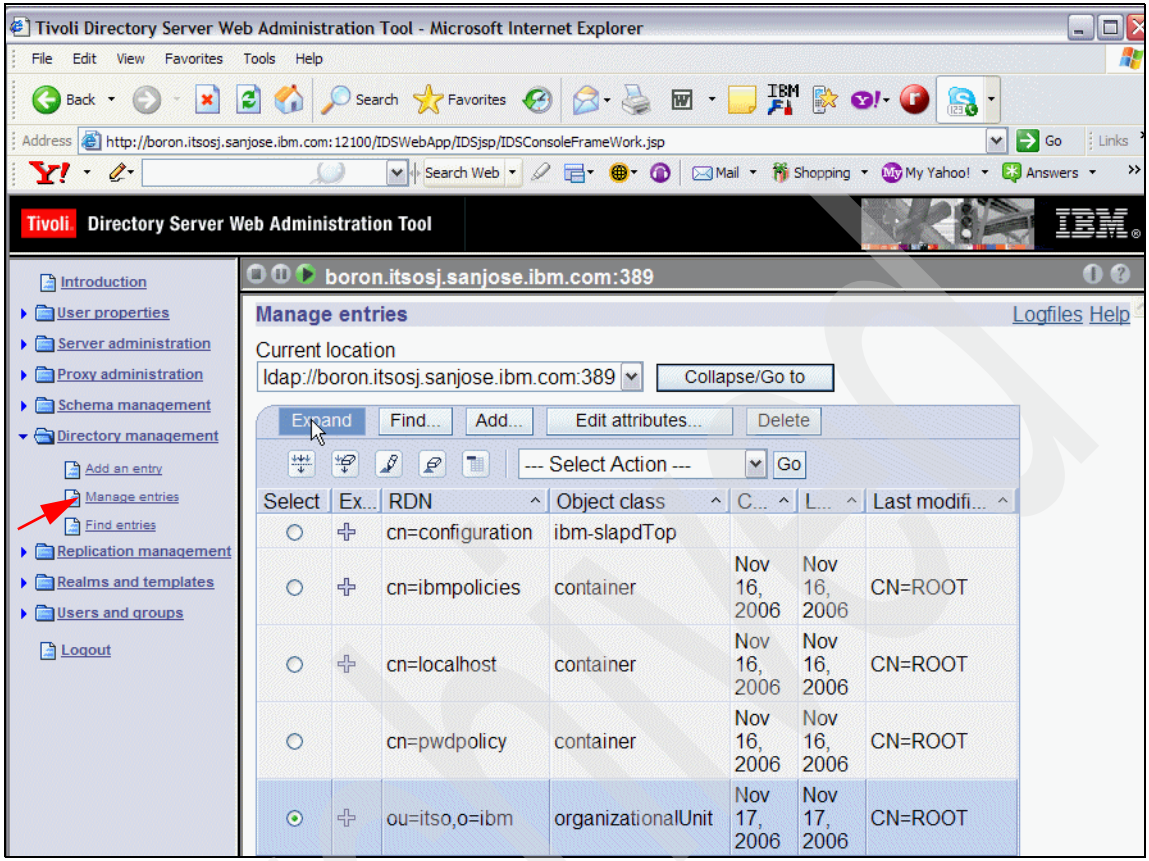

Figure 2-5 Specify the password for the newly added user esadmin 1/5

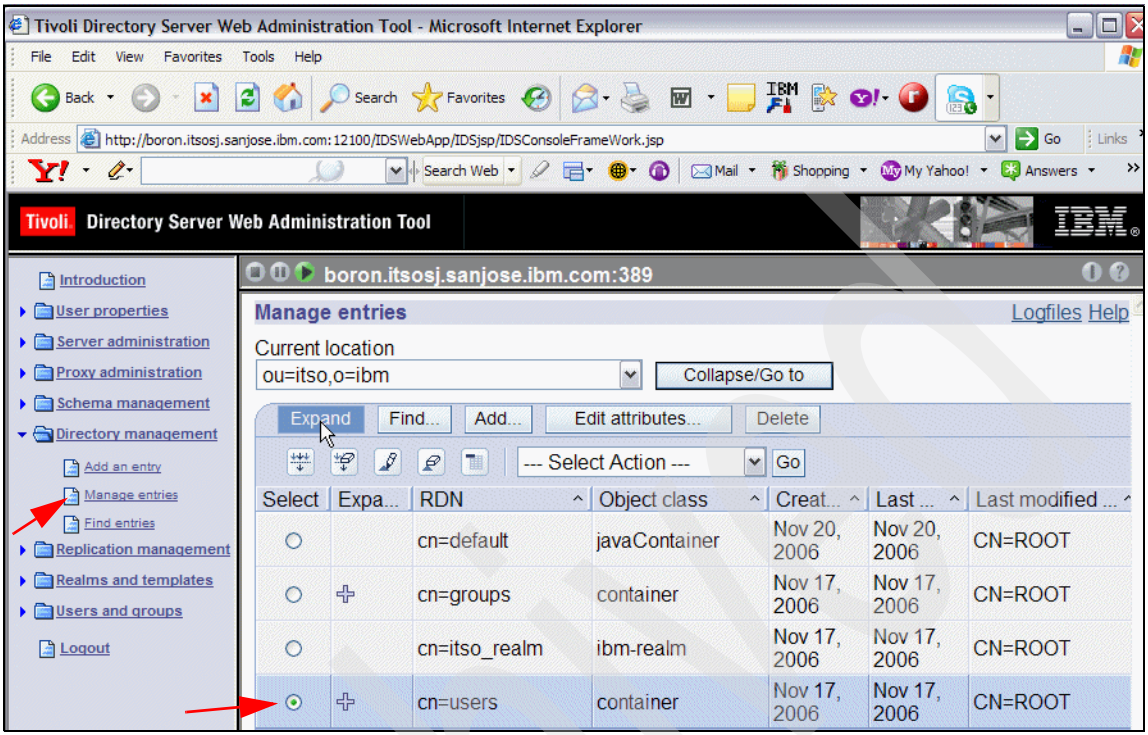

Figure 2-6 Specify the password for the newly added user esadmin 2/5

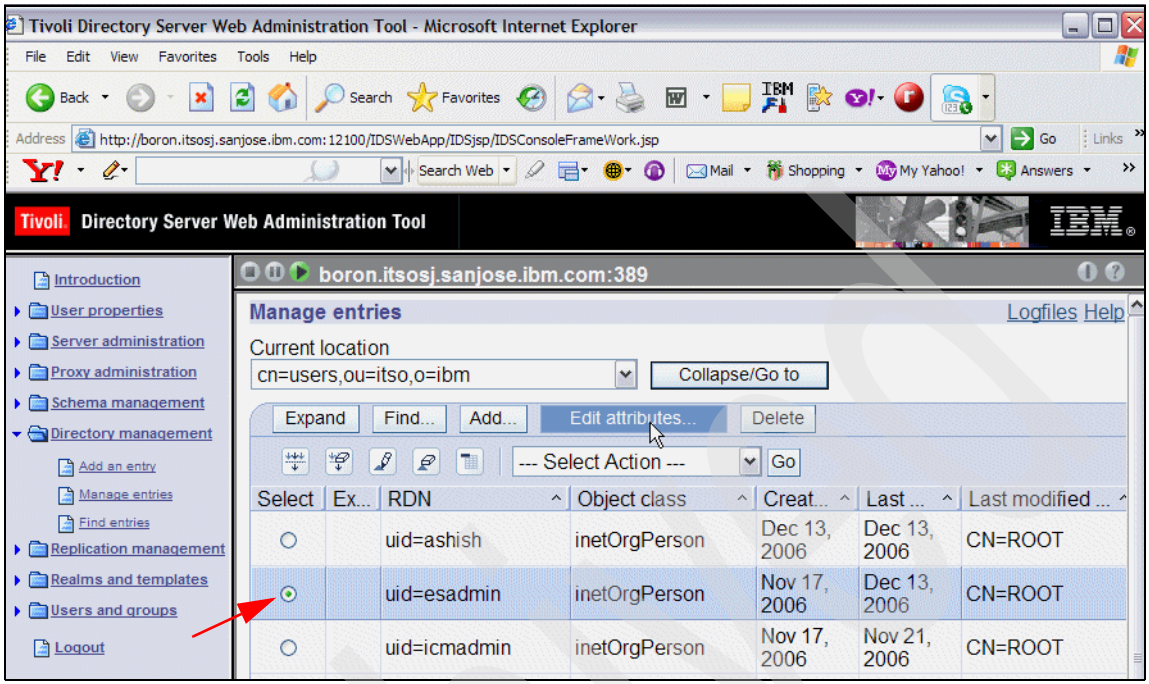

Figure 2-7 Specify the password for the newly added user esadmin 3/5

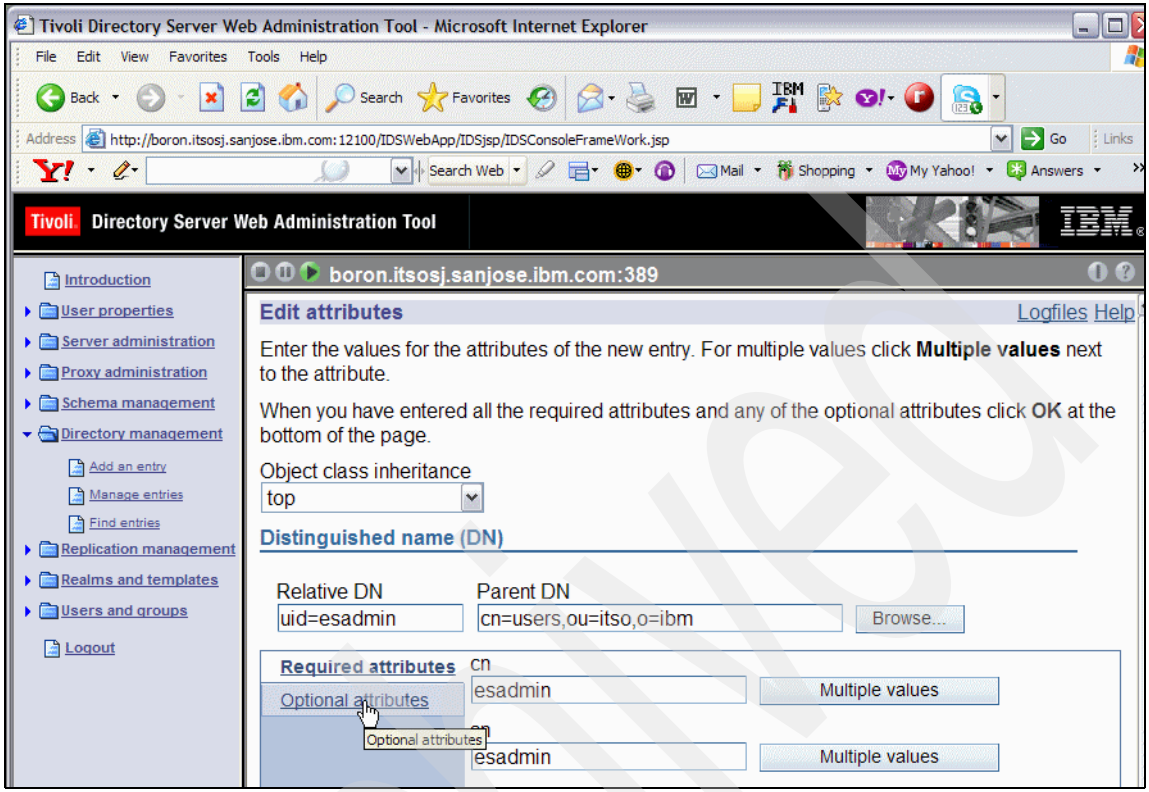

Figure 2-8 Specify the password for the newly added user esadmin 4/5

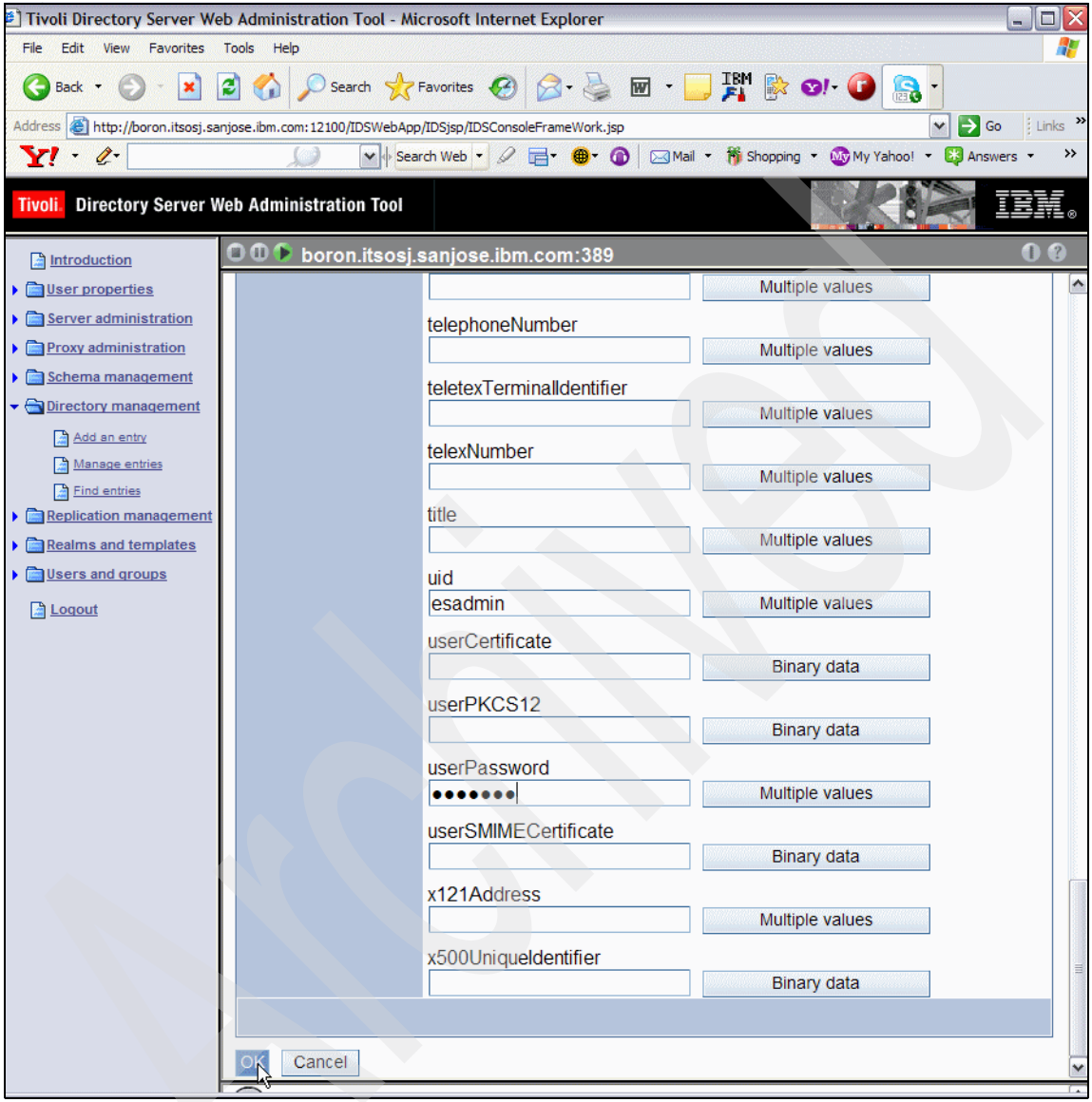

Figure 2-9 Specify the password for the newly added user esadmin 5/5

2.3.2 WSTEP2: Enable WebSphere Application Server global security

In this step, we enable global security in the WebSphere Application Server with the Search Runtime, and specify that it uses the Tivoli Directory Server as the LDAP repository as its user registry. LTPA keys are generated and then exported to other servers participating in the single sign-on domain.

Figure 2-10 on page 68 through Figure 2-16 on page 74 show some of the main windows for enabling global security, and generating and exporting the LTPA key. Figure 2-17 on page 75 through Figure 2-20 on page 77 show the import of the previously exported LTPA key into kazan.itsosj.sanjose.ibm.com (where Lotus QuickPlace and WebSphere Portal Server are installed) and the verification that single sign-on is working correctly.

After logging in to the WebSphere Administrative Console, navigate to Global security under Security in the navigation pane, as shown in Figure 2-10 on page 68. Click the **LDAP** link to provide details of the LDAP repository to be used (Figure 2-11 on page 69).

In Figure 2-11 on page 69, provide the LDAP host name (boron.itsosj.sanjose.ibm.com), the user ID (uid=wasadmin,cn=users,ou=itso,o=ibm), and password to log in to the LDAP repository, Base distinguished name (ou=itso,o=ibm), Bind distinguished name (uid=ldapbind,cn=users,ou=itso,o=ibm), Bind password, and other details as shown, and click **OK**.

The LTPA settings should be configured by clicking the **LTPA** link in Figure 2-12 on page 70. The password corresponds to that for logging in to the WebSphere Administrative Console. We explicitly generate the LTPA key by first specifying the name of the key and its target location (c:\ltpakey.key) in the Key file name box, and then clicking **Generate keys**, as shown in Figure 2-13 on page 71. After the keys are generated, the LTPA key can be exported as a particular file name in a target directory (c:\ltpa.key) in the Key file name box, as shown in Figure 2-14 on page 72. Single sign-on must be enabled as well by clicking the **Single sign-on (SSO)** link, as shown in Figure 2-14 on page 72. Check the **Enabled** box in Figure 2-15 on page 73. On the Global Security panel, select the **Enable global security** check box. The Enforce Java 2 security check box will be automatically selected. Verify that the Active user registry drop-down list has Lightweight Directory Access Protocol (LDAP) user registry selected, and that the Active authentication mechanism drop-down list has Lightweight Third Party Authentication (LTPA) selected, as shown in Figure 2-16 on page 74. Click **OK**, and then follow the appropriate windows to save all the configuration changes made, and log out. WebSphere Application Server global security is now enabled. Restart WebSphere Application Server and verify that global security is

enabled by logging in to the WebSphere Administrative Console; you will now be prompted for a password. This is not shown here.

The exported key (c:\ltpa.key) must then be copied over to a directory on the appropriate servers (e:\ltpa.key in our case on kazan.itsosj.sanjose.ibm.com, where WebSphere Portal Server and Lotus QuickPlace are installed) that participate in single sign-on. Figure 2-17 on page 75 through Figure 2-20 on page 77 show the main steps involved in importing the LTPA key and verifying the success of the procedure.

After logging in to the WebSphere Application Server on kazan.itsosj.sanjose.ibm.com, navigate to the LTPA settings window, as shown in Figure 2-17 on page 75. After specifying the directory (e:\ltpa.key) where the copied LTPA key resides in the Key file name box, click **Import keys**. Single sign-on must be enabled by clicking the **Single sign-on (SSO)** link, as shown in Figure 2-17 on page 75. Check the **Enabled** box in Figure 2-18 on page 76 and save all the configuration changes by following all the prompts.

To verify that single sign-on is working, you must first log in to the WebSphere Application Server associated with the IBM OmniFind Enterprise Edition Search Runtime (nile.itsosj.sanjose.ibm.com). Then, in the *same* browser session, type the Lotus QuickPlace and WebSphere Portal Servers URLs on kazan.itsosj.sanjose.ibm.com, as shown in Figure 2-19 on page 76 and Figure 2-20 on page 77 respectively. When the appropriate Web page is displayed *without* a prompt for credentials, it indicates that single sign-on is working correctly.

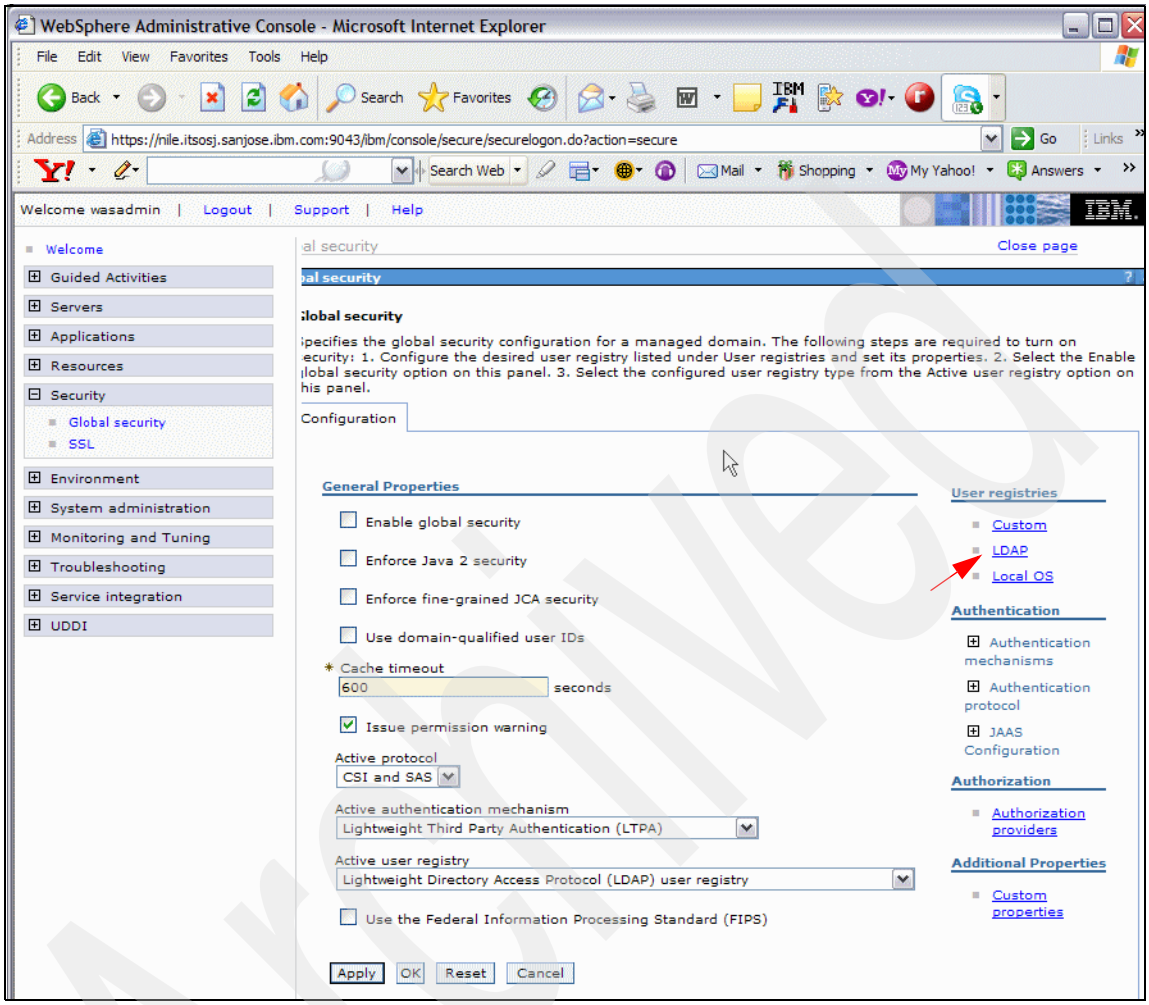

Figure 2-10 WebSphere Application Server global security enablement

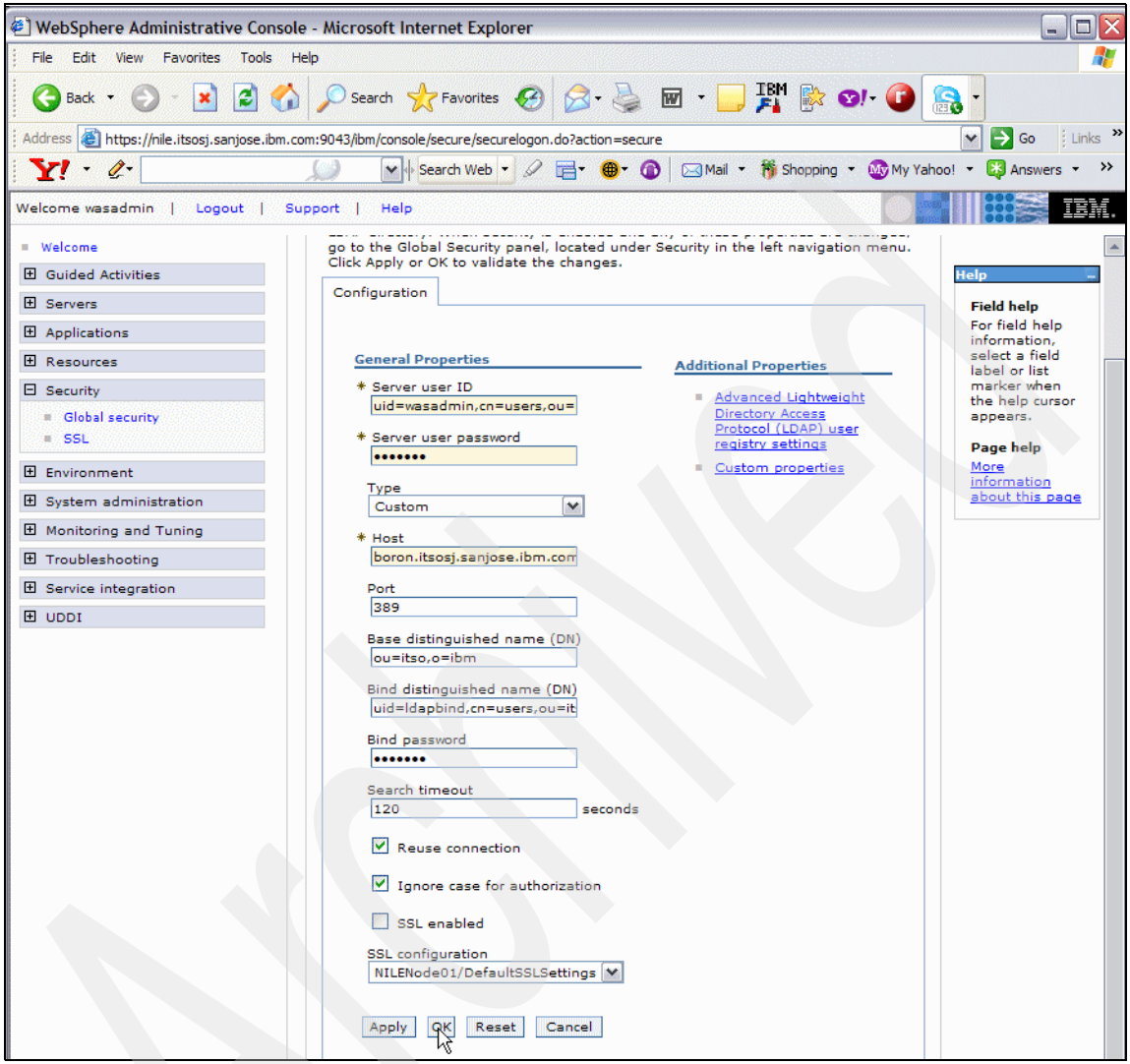

Figure 2-11 Provide details of the LDAP repository to be used

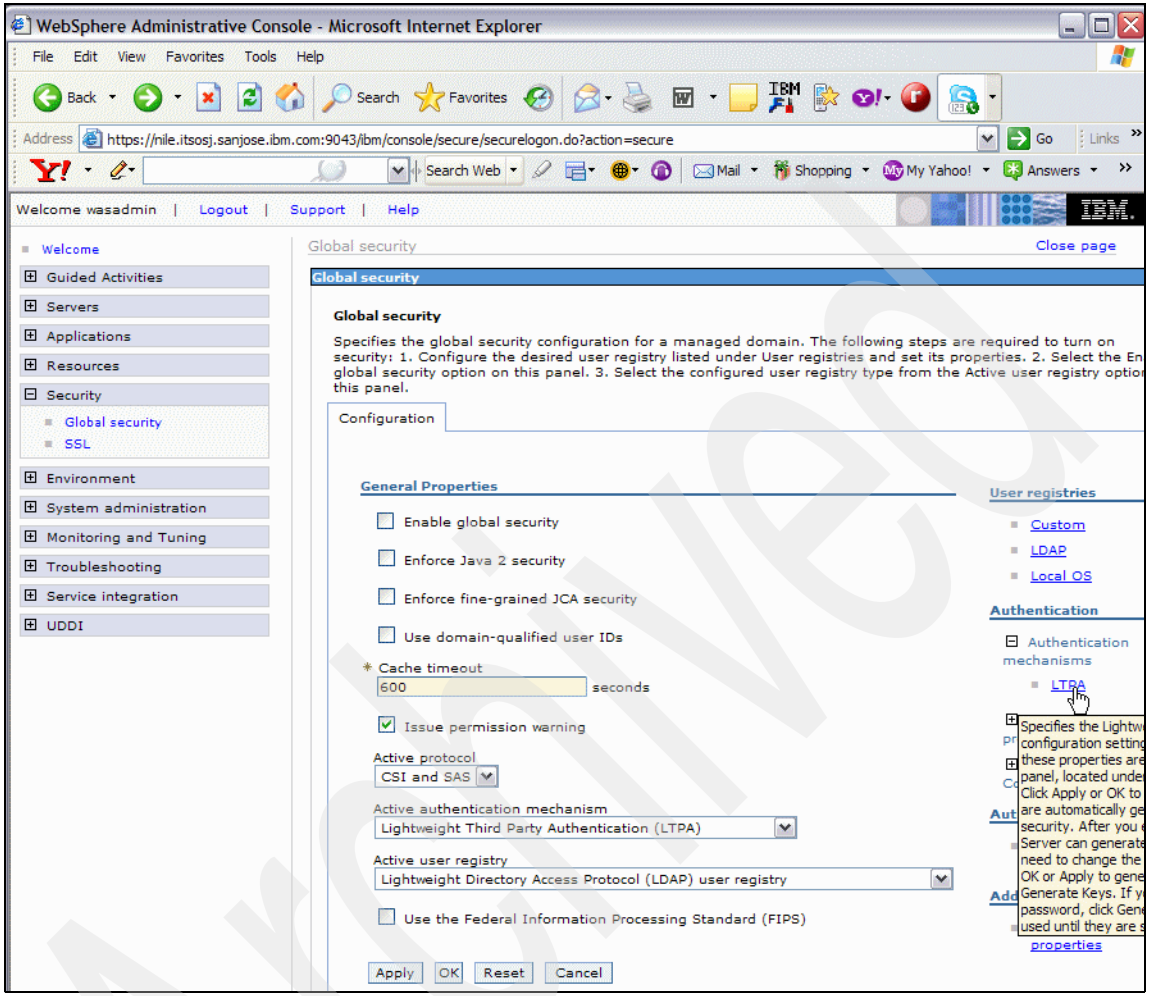

Figure 2-12 LTPA settings 1/2

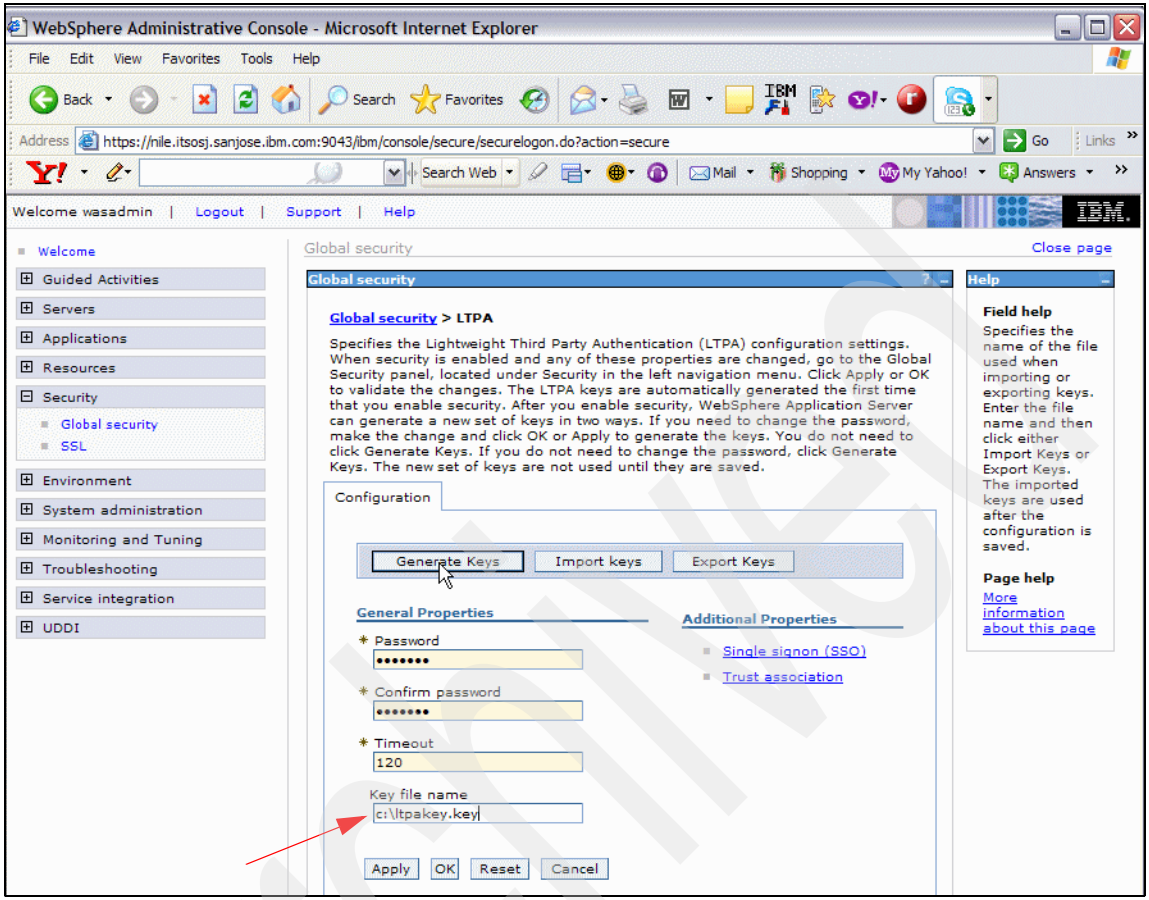

Figure 2-13 LTPA settings 2/2

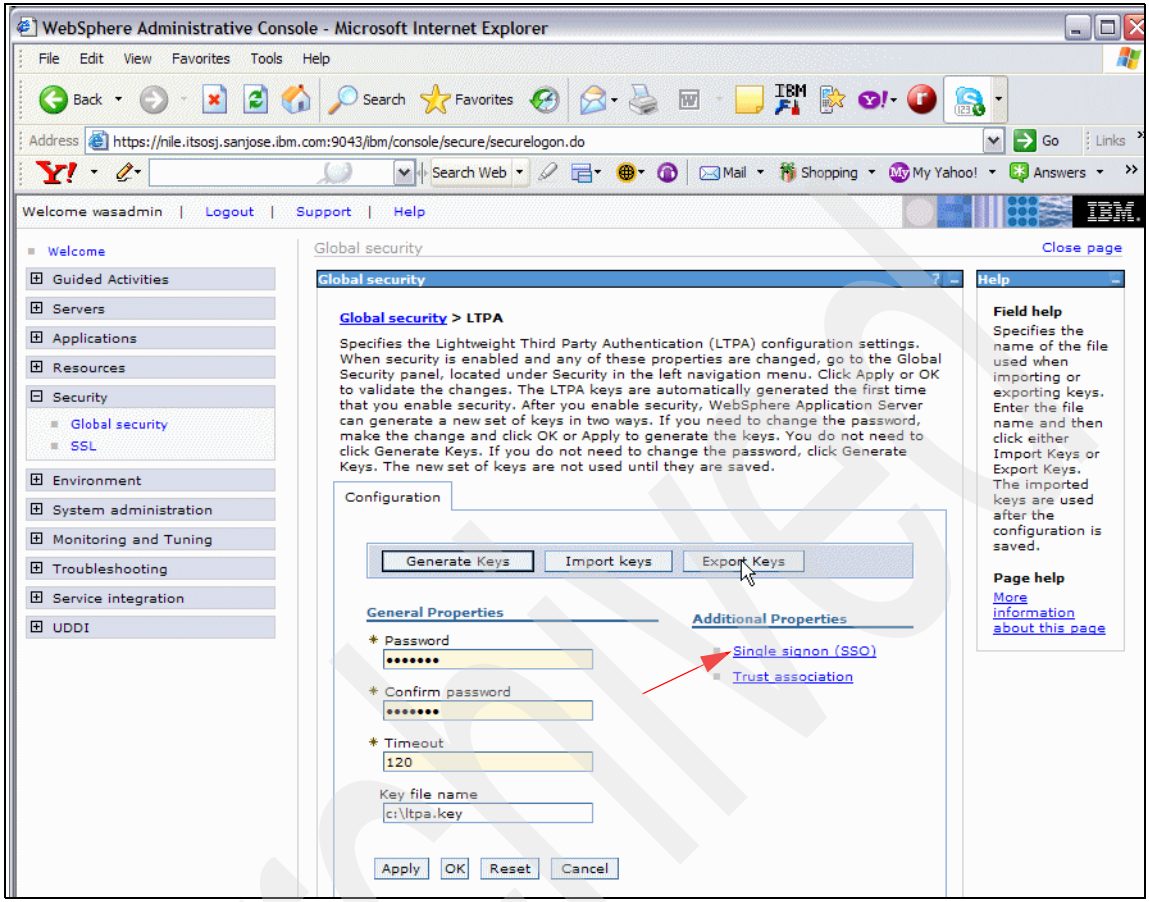

Figure 2-14 Export LTPA keys from WebSphere Application Server

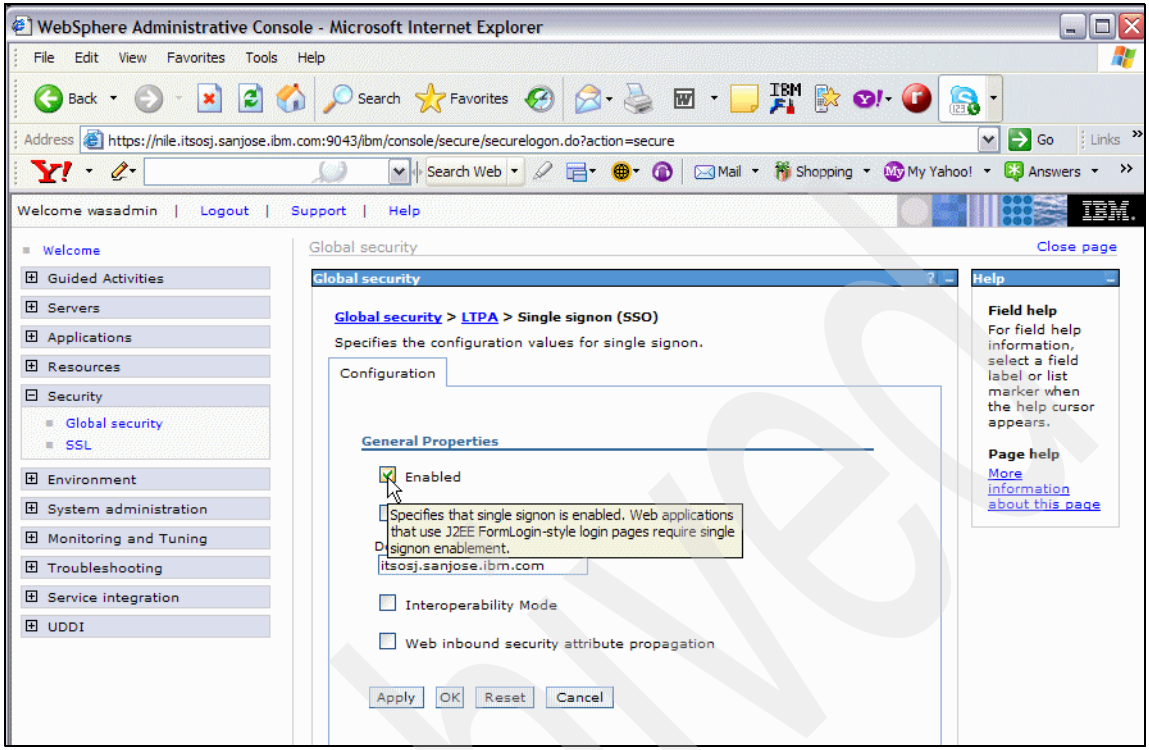

Figure 2-15 Enable Single sign-on (SSO)

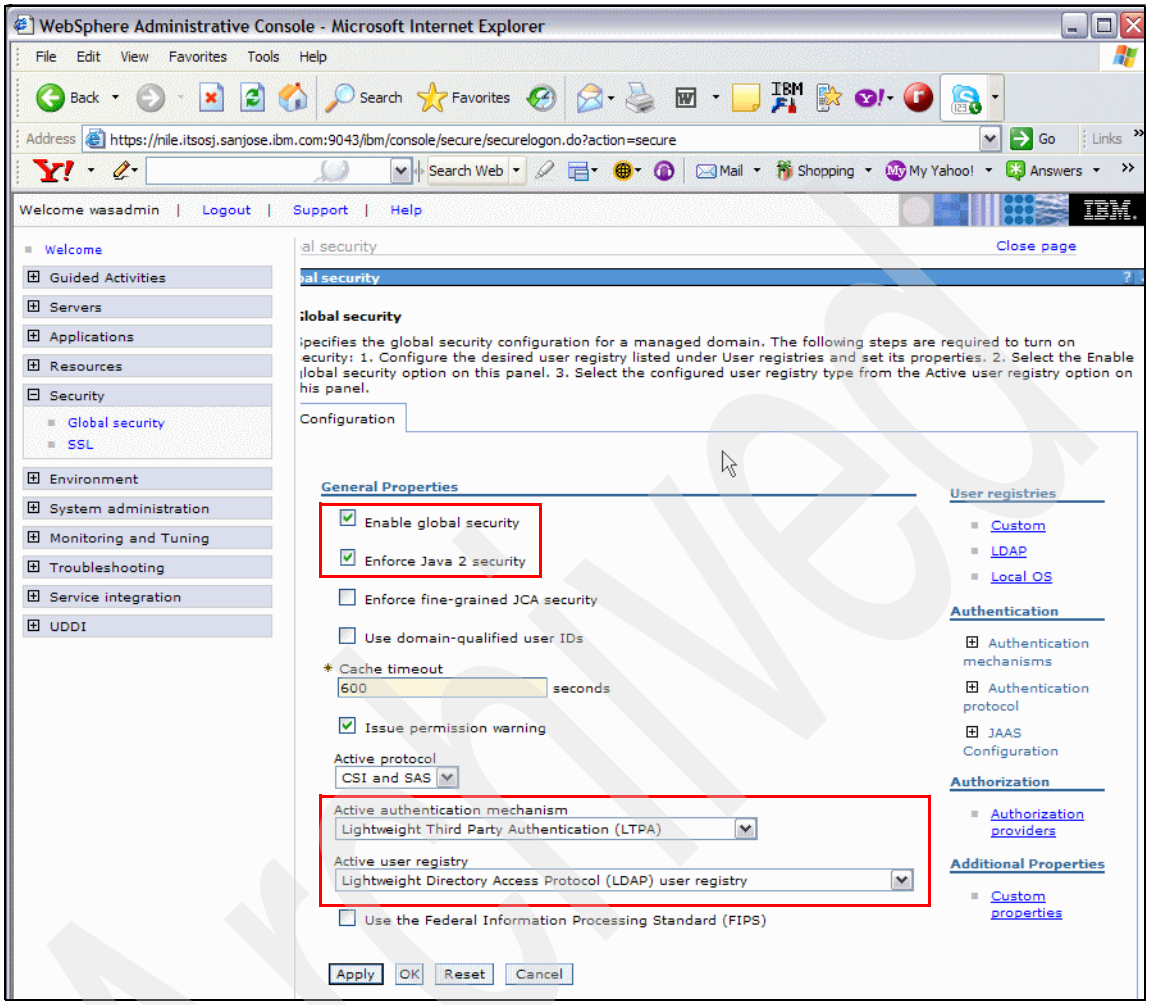

Figure 2-16 WebSphere Application Server global security enablement

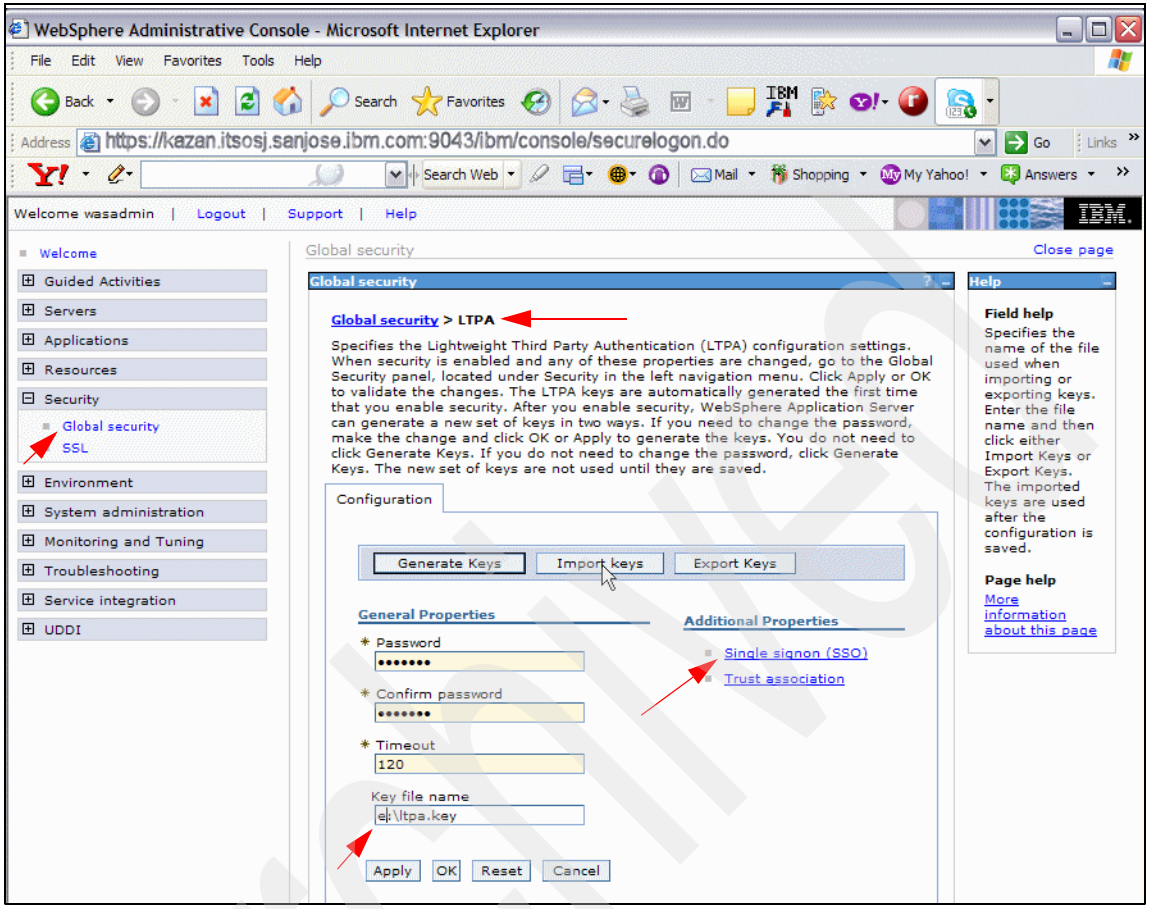

Figure 2-17 Import LTPA keys

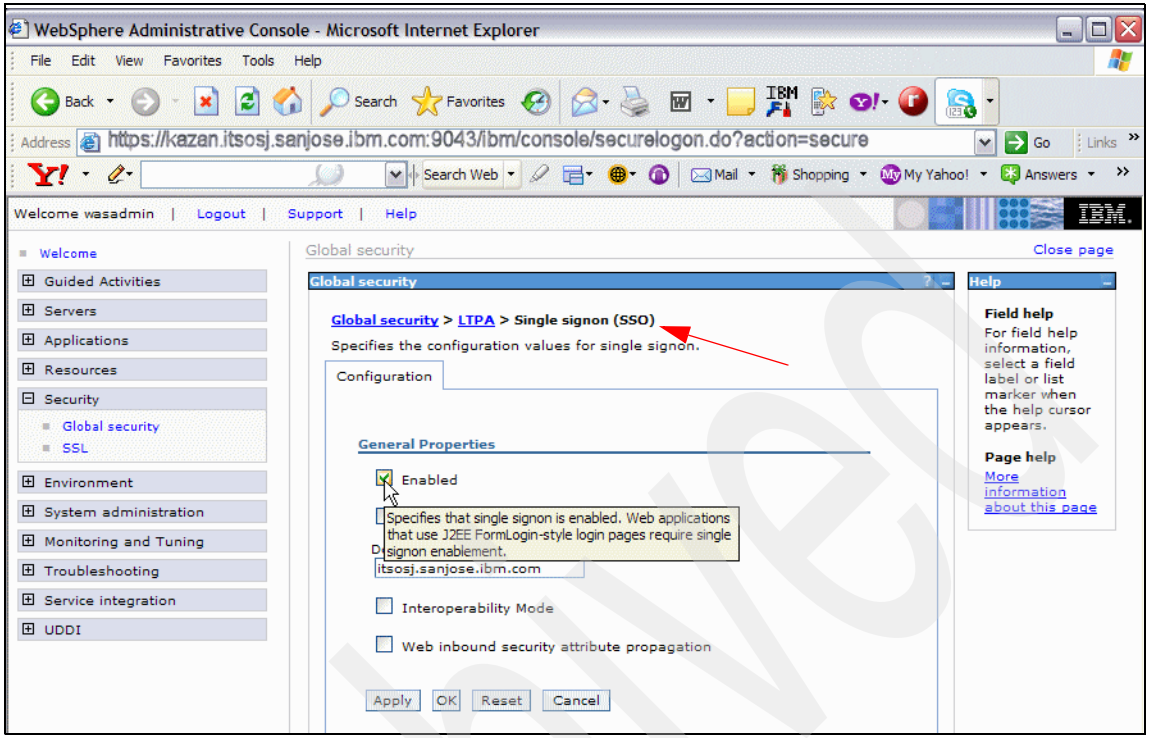

Figure 2-18 Enable single sign-on

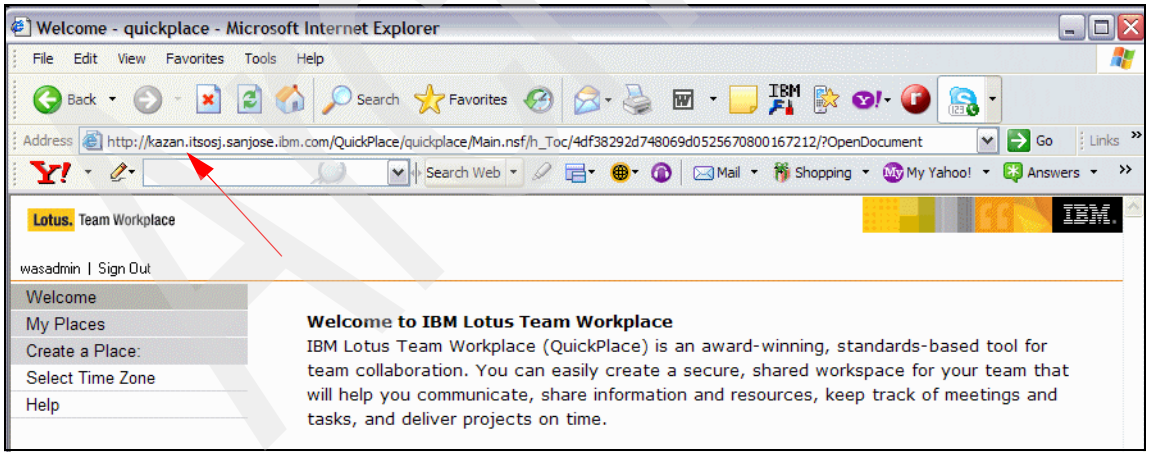

Figure 2-19 Verify that the LTPA token is accepted by Lotus QuickPlace

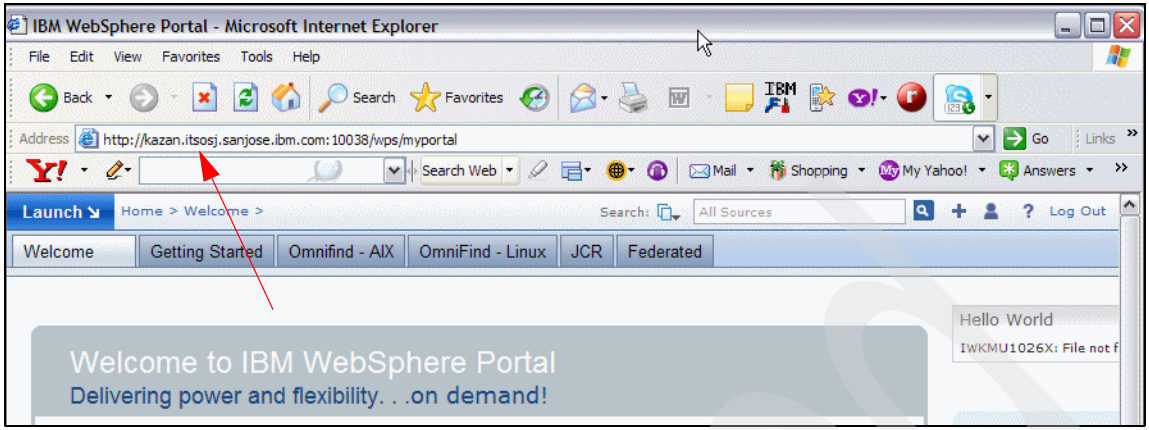

Figure 2-20 Verify LTPA token is accepted by WebSphere Portal

2.3.3 WSTEP3: Update es.cfg file

Once WebSphere global security is enabled in the IBM OmniFind Enterprise Edition Search Runtime server, you must update the es.cfg file with the WebSphere Application Server user ID and password.

- The WebSphere Application Server user ID must be supplied in the WebSphere Application ServerUser property using an editor, such as Notepad. Example 2-1 on page 78 shows this property to have a value of wasadmin.
- The WebSphere Application Server password needs to be stored in encrypted form in the es.cfg file in the WASPassword property. For this to occur, you must execute the **eschangewaspw** command as follows (shown for Windows; on UNIX, it would be the **eschangewaspw.sh** command):

  `eschangewaspw` *`wasadmin`*

  where wasadmin is your WebSphere Application Server Admin password

Example 2-1 shows the partial contents of the es.cfg file with the encrypted password.

Note: When WebSphere global security is enabled, the Common Communications Layer (CCL) component of IBM OmniFind Enterprise Edition needs to authenticate with WebSphere Application Server in order to start the search runtime. It obtains the required user ID/password from the es.cfg file; it has the key to decrypt the WASPassword.

Example 2-1 es.cfg file contents

```
.....
LocalHostName=NILE.itsosj.sanjose.ibm.com
DBName=fountain
.....
WASUser=wasadmin
.....
WASPassword=cRJygNQdU7gx4k7JU7PKtOt5EFa3DylJ
```

2.3.4 WSTEP4: Create NWINSURANCE collection

In this step, we create the NWINSURANCE collection with the appropriate crawlers, then parse and index the crawled data. The individual steps involved are shown in Figure 2-21 and described in more detail in the following subsections.

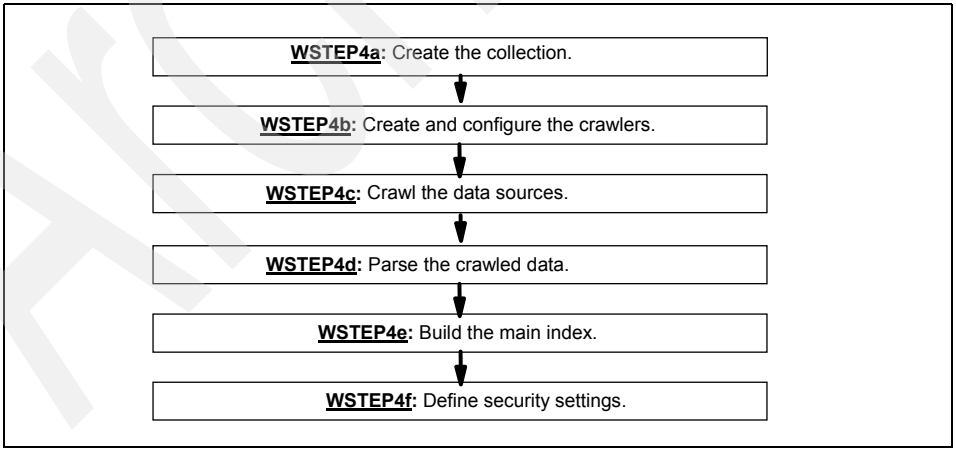

Figure 2-21 Steps to create and configure the IBMCIF collection

Note: A number of parameters can be specified during the creation and configuration of a collection and the associated crawlers. A description of these parameters is beyond the scope of this book. You are encouraged to read the product documentation or invoke **Help** on the GUI for detailed information about these parameters.

WSTEP4a: Create the collection

After logging in to the GUI administration console as the enterprise search administrator (Figure 2-22 on page 80), click the **Collections** view and click **Create Collection**, as shown in Figure 2-23 on page 80. Provide details in Figure 2-24 on page 81 about the collection, such as the Collection name (NWINSURANCE), Collection security[1] (Enable security for the collection), Document importance (Do not apply any static ranking), Categorization type (None) during parsing, and Language to use (English) during search. Click **OK** to complete the creation of the collection and proceed to Figure 2-25 on page 83 to define and configure the crawlers in the created collection.

Note: The key point here is to enable security for the collection so that document-level security can be enforced, since this option cannot be changed once the collection is created.

Once the collection is created, we can proceed to the creation of the Lotus QuickPlace and Windows file system crawlers, as described in “WSTEP4b: Create and configure the crawlers” on page 82.

[1] Required for enforcing document-level security

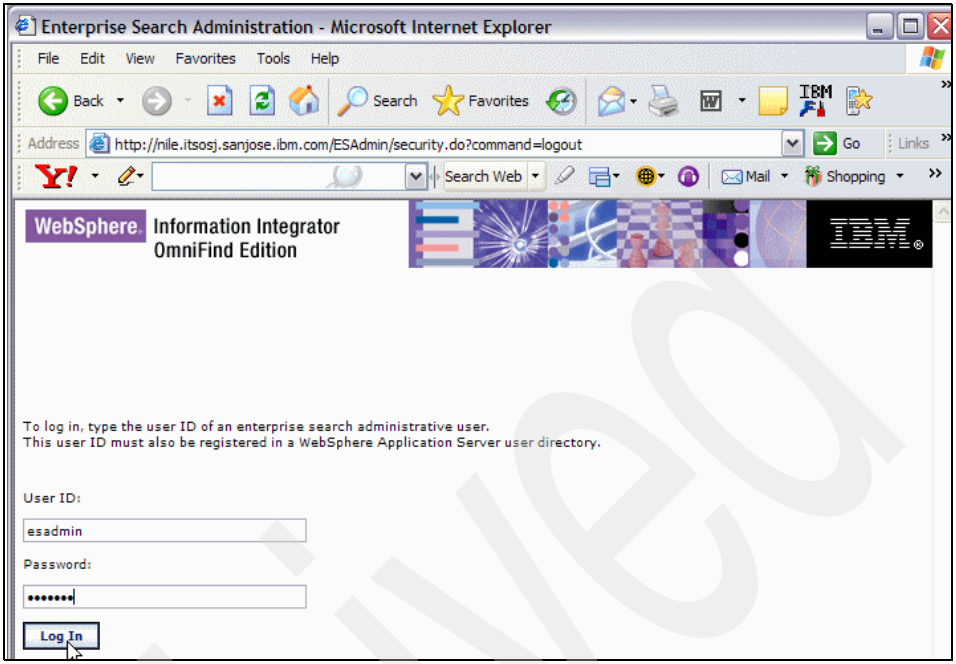

Figure 2-22 Login to the GUI administration console

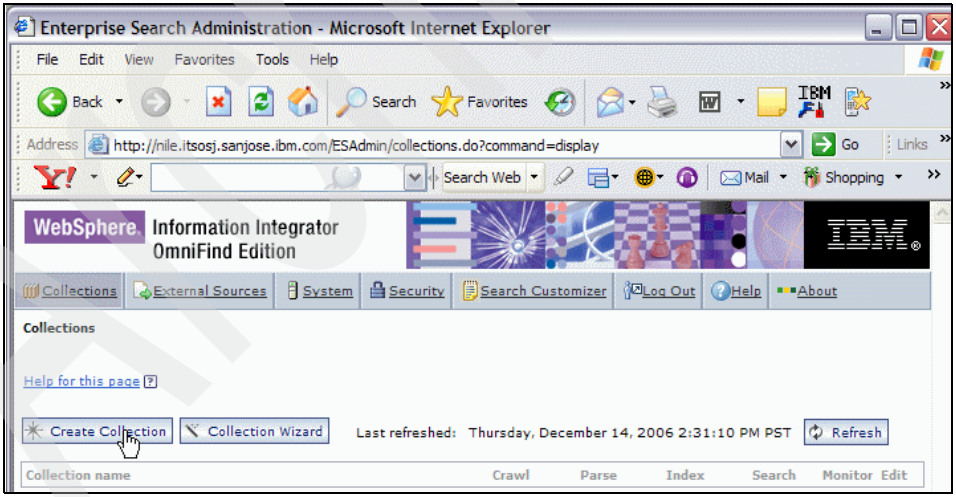

Figure 2-23 Create Collection

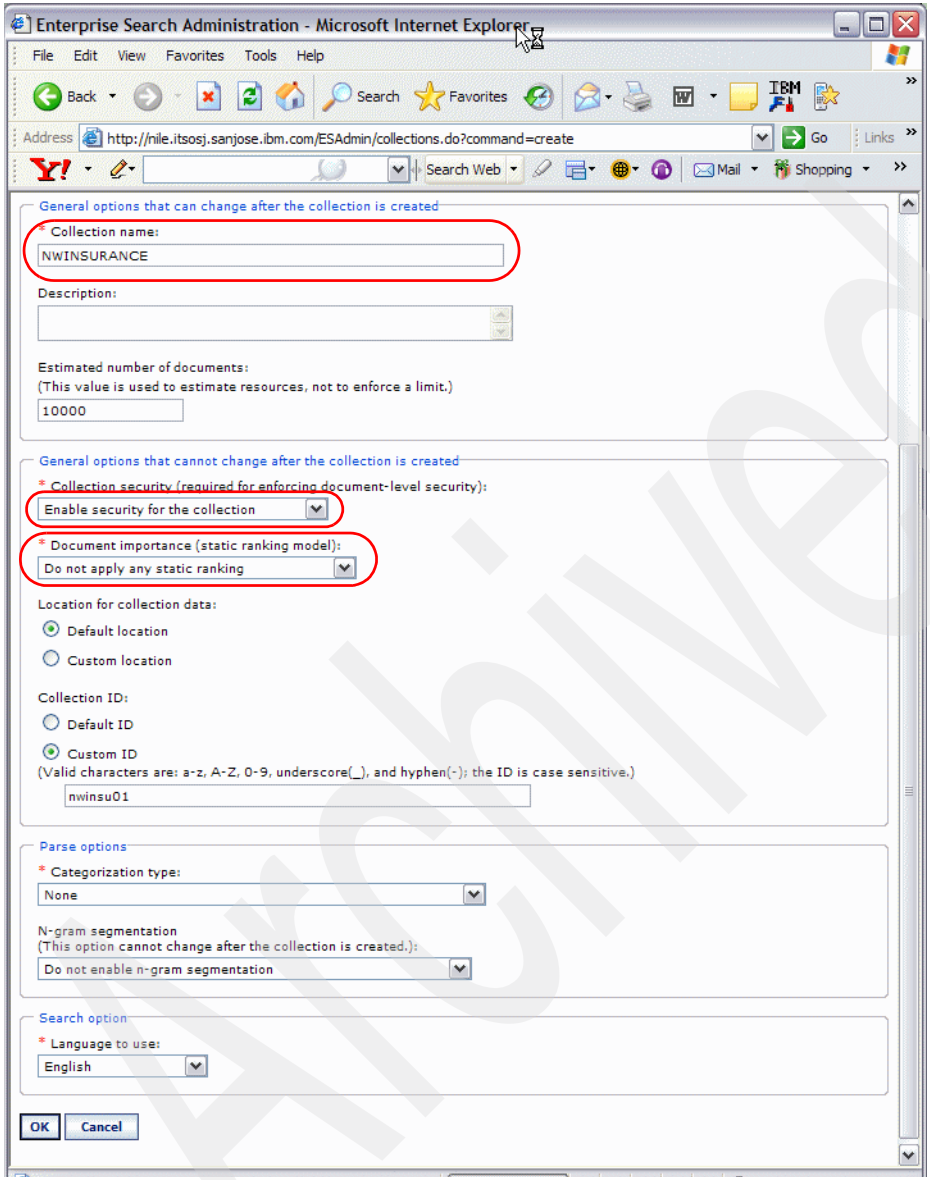

Figure 2-24 NWINSURANCE collection details

WSTEP4b: Create and configure the crawlers

In this step, the Lotus QuickPlace and Windows file system crawlers are defined with the appropriate security configurations, followed by a full crawl of the data sources involved.

- Lotus QuickPlace crawler

  Figure 2-25 on page 83 through Figure 2-52 on page 110 describe the creation and configuration of the Lotus QuickPlace crawler, followed by a full crawl of the crawls spaces defined.

  After logging in to the administration console, select the **Collections** view and click the **Crawl** icon, as shown in Figure 2-25 on page 83. In the following window (Figure 2-26 on page 84), switch to Edit mode by clicking the **Edit** icon. From the **Crawl** tab in Figure 2-27 on page 85, click **Create Crawler**. Select QuickPlace from the drop-down list for Crawler type and click **Next** in Figure 2-28 on page 86.

  Provide details of the QuickPlace crawler in Figure 2-29 on page 87, such as the Crawler name (NW_INSU_QP) and Maximum number of documents to crawl (20000). Click **Next** to provide further details in Figure 2-30 on page 88, such as the Lotus QuickPlace server name (kazan.itsosj.sanjose.ibm.com), the Protocol to use (DIIOP), the user ID (esadmin) and password to access the server using this protocol, and whether single sign-on security (SSO) should be enabled. Other information to be provided includes the LDAP server information, such as the server name (boron.itsosj.sanjose.ibm.com), port number (389), the Base DN (ou=itso,o=ibm), whether or not to use SSL to connect to the LDAP server (Do not use SSL to connect to the LDAP server), and the credentials to access the LDAP server (user name uid=ldapbind,cn=users,ou=itso,o=ibm). Click **Next** in Figure 2-30 on page 88 to select the QuickPlace spaces in crawl (Figure 2-31 on page 89).

  Figure 2-31 on page 89 shows the selected QuickPlace places to crawl (Industry Learning | Insurance..) obtained by first discovering available places ('*' in the Place name or pattern followed by a click of **Search for places**, which lists all those found with the matching criteria in the Available places box and then copying those of interest to the Places to crawl box). Click **Next** to specify the crawl schedule (Figure 2-32 on page 90). Since we chose to schedule the crawls manually, click **Next** in Figure 2-32 on page 90. The next step is to identify all the documents to be crawled. As shown in Figure 2-33 on page 91, we chose to crawl all the rooms, and clicked **Next**. The next step is to specify options for the entire crawl space by clicking **Edit Crawl Space Options**, as shown in Figure 2-34 on page 92.

Figure 2-35 on page 93 through Figure 2-38 on page 96 describe the default options for searching all the documents in the crawl space. You can select the fields to participate in free text search, fielded search, parametric search, and complete match support. You can also specify the fields that can be explicitly used to sort search results. Additionally, you can specify whether attachments should be crawled, and types of attachments to exclude, Since collection security is enabled for the NWINSURANCE collection, you can specify whether or not credentials should be validated during query processing, and the options for indexing access controls. Click **OK** when all the options have been selected, as shown in Figure 2-37 on page 95. Click **Finish** in Figure 2-38 on page 96 to complete the creation and configuration of the Lotus QuickPlace crawler.

We can now proceed to create and configure the Windows file system crawler on “Windows file system crawler” on page 96.

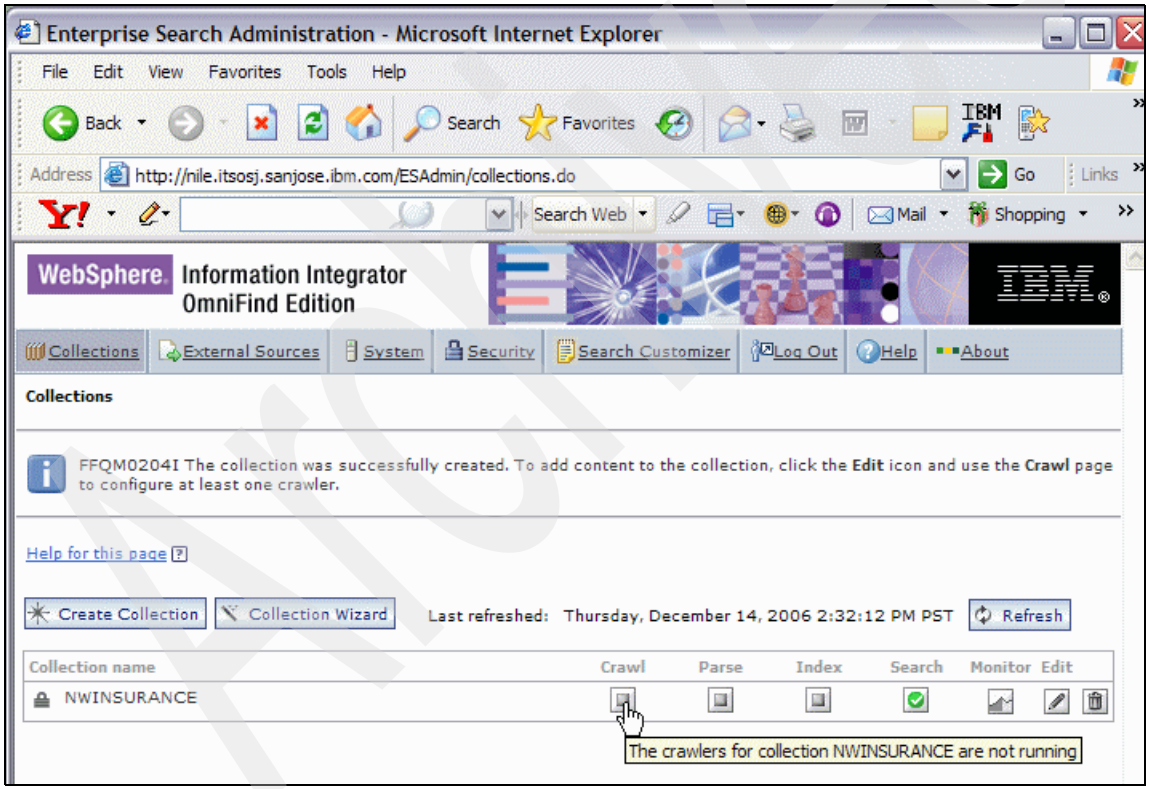

Figure 2-25 Click Crawl icon

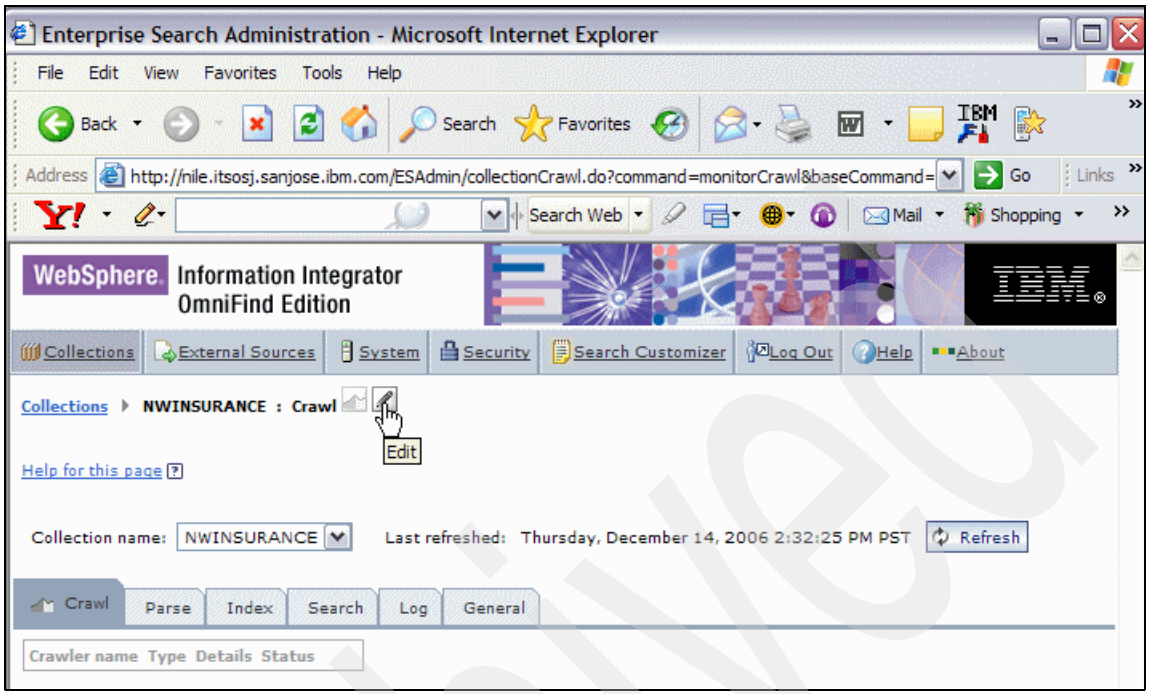

Figure 2-26 Click Edit icon

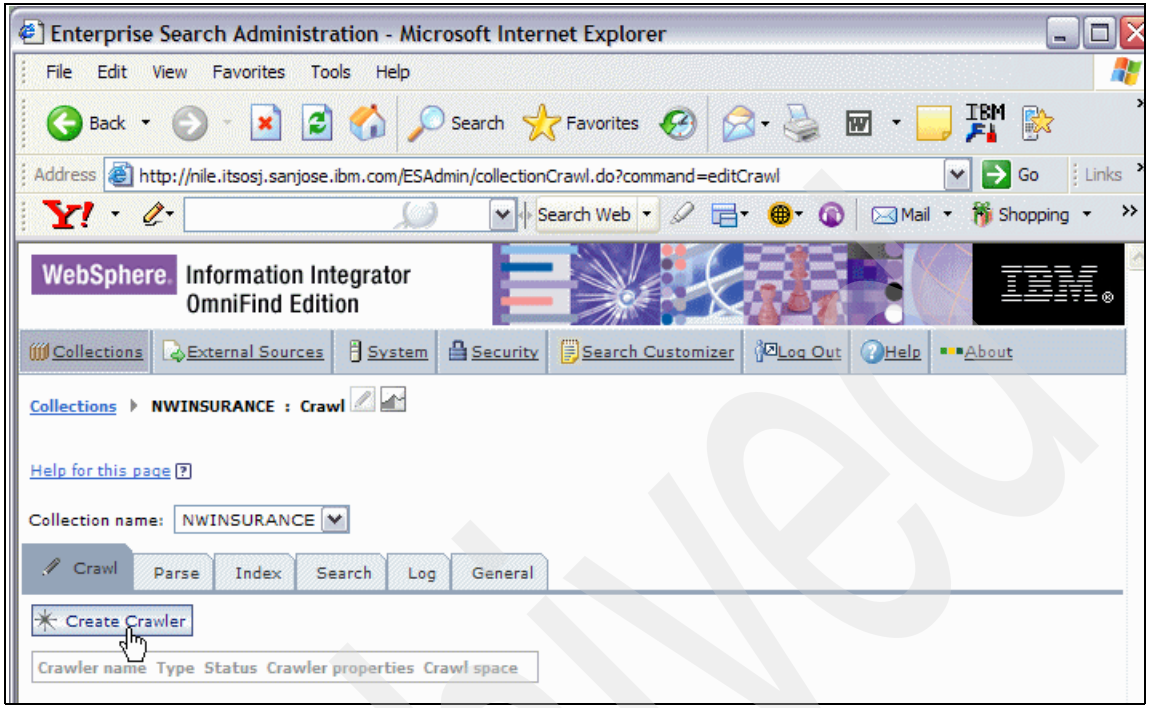

Figure 2-27 Create Crawler

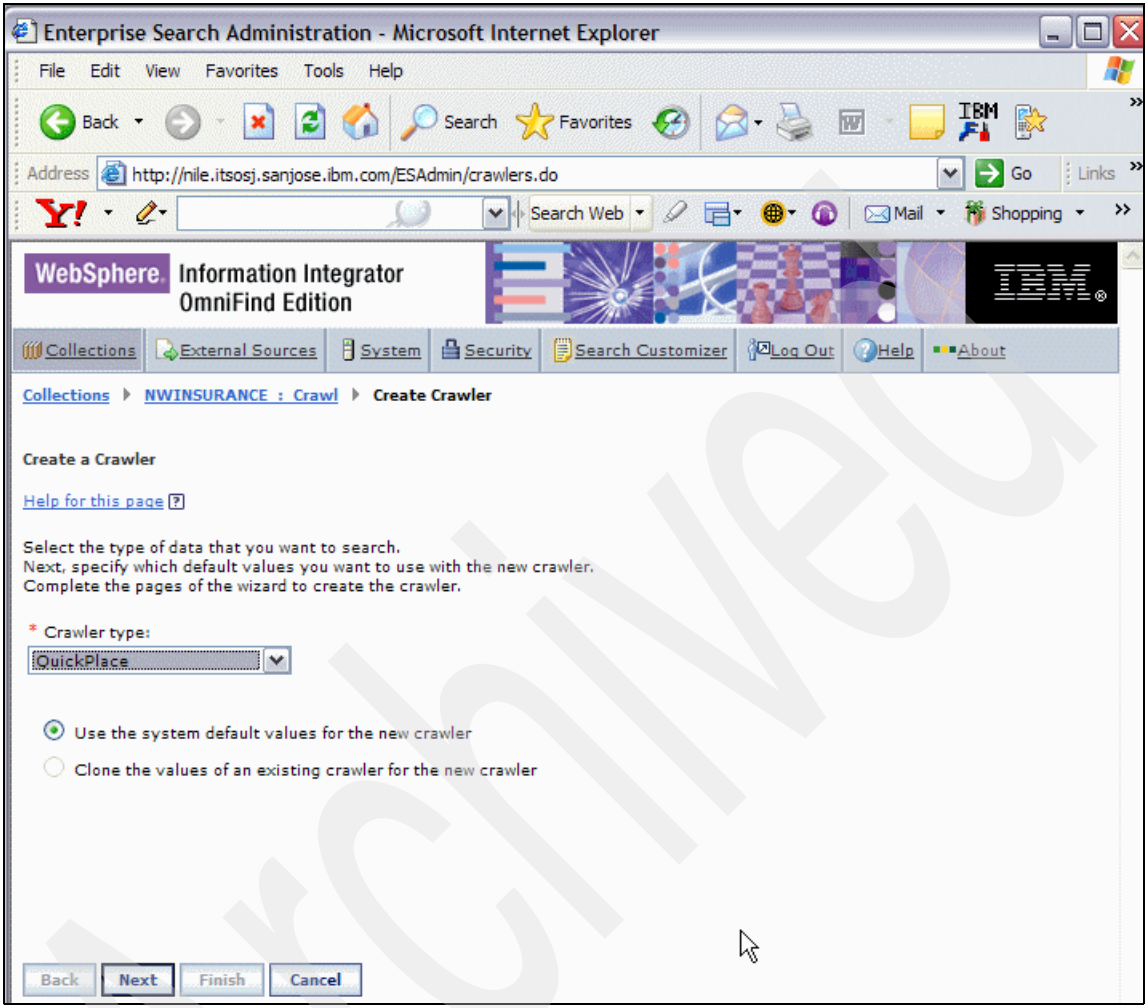

Figure 2-28 QuickPlace crawler type

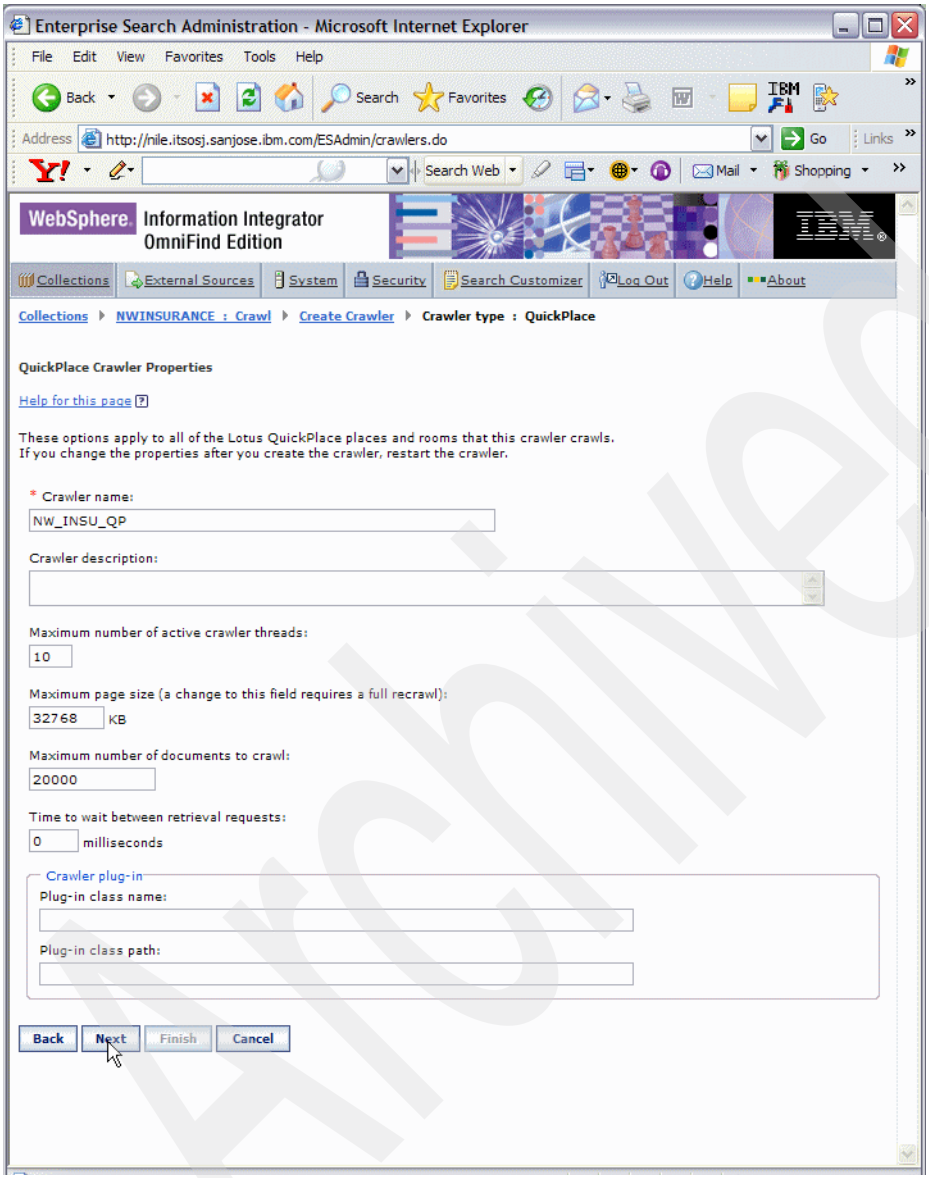

Figure 2-29 Crawler details 1/2

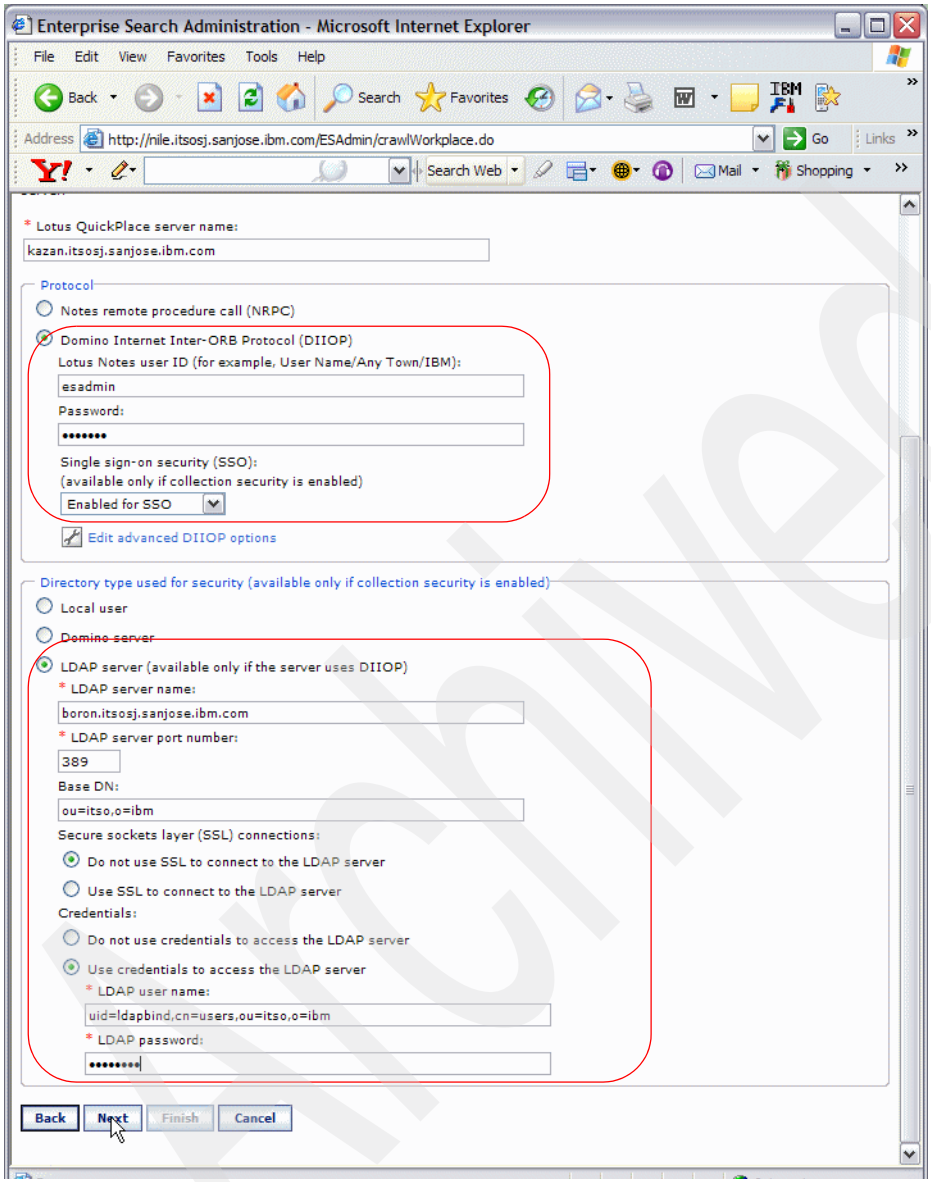

Figure 2-30 Crawler details 2/2

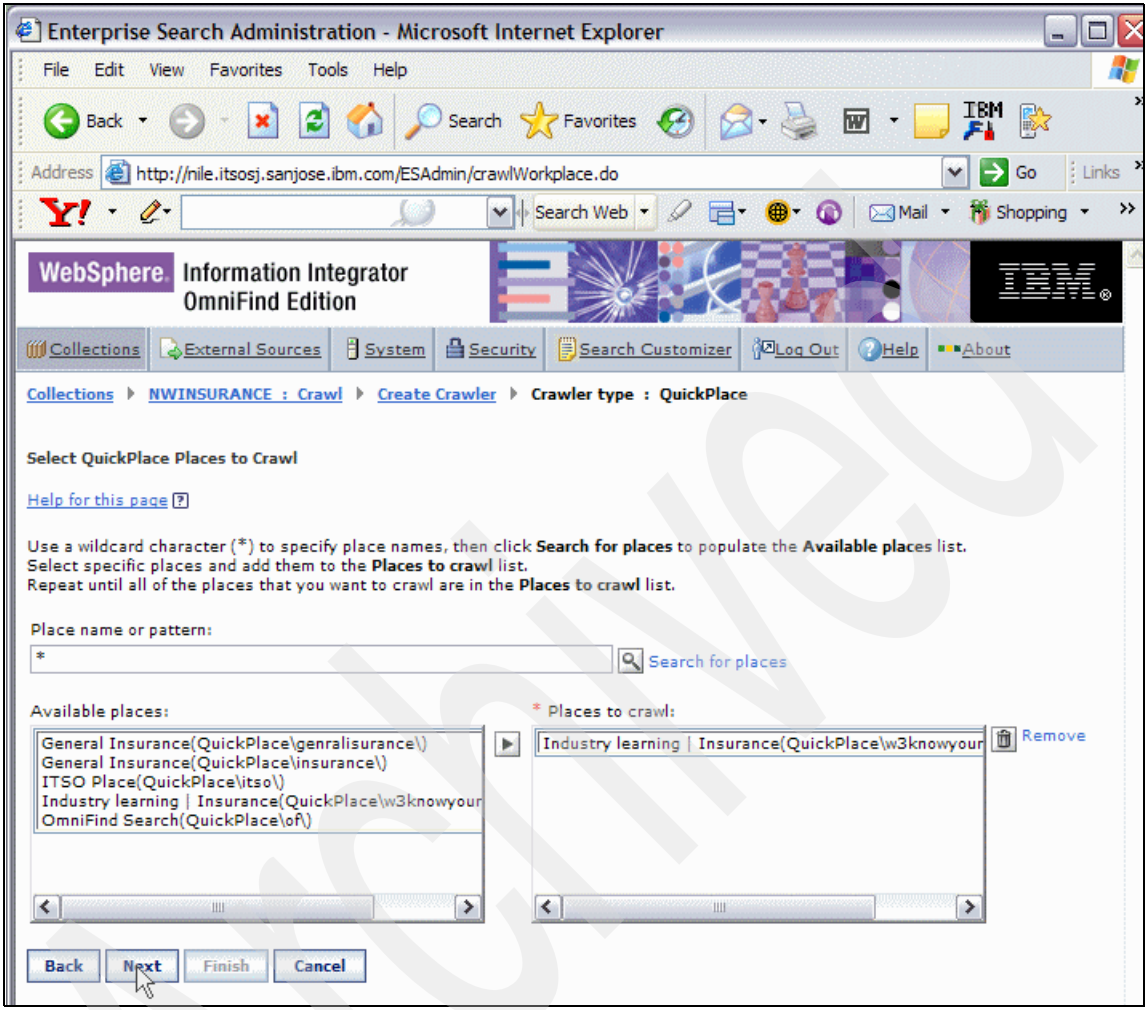

Figure 2-31 QuickPlace Places to Crawl

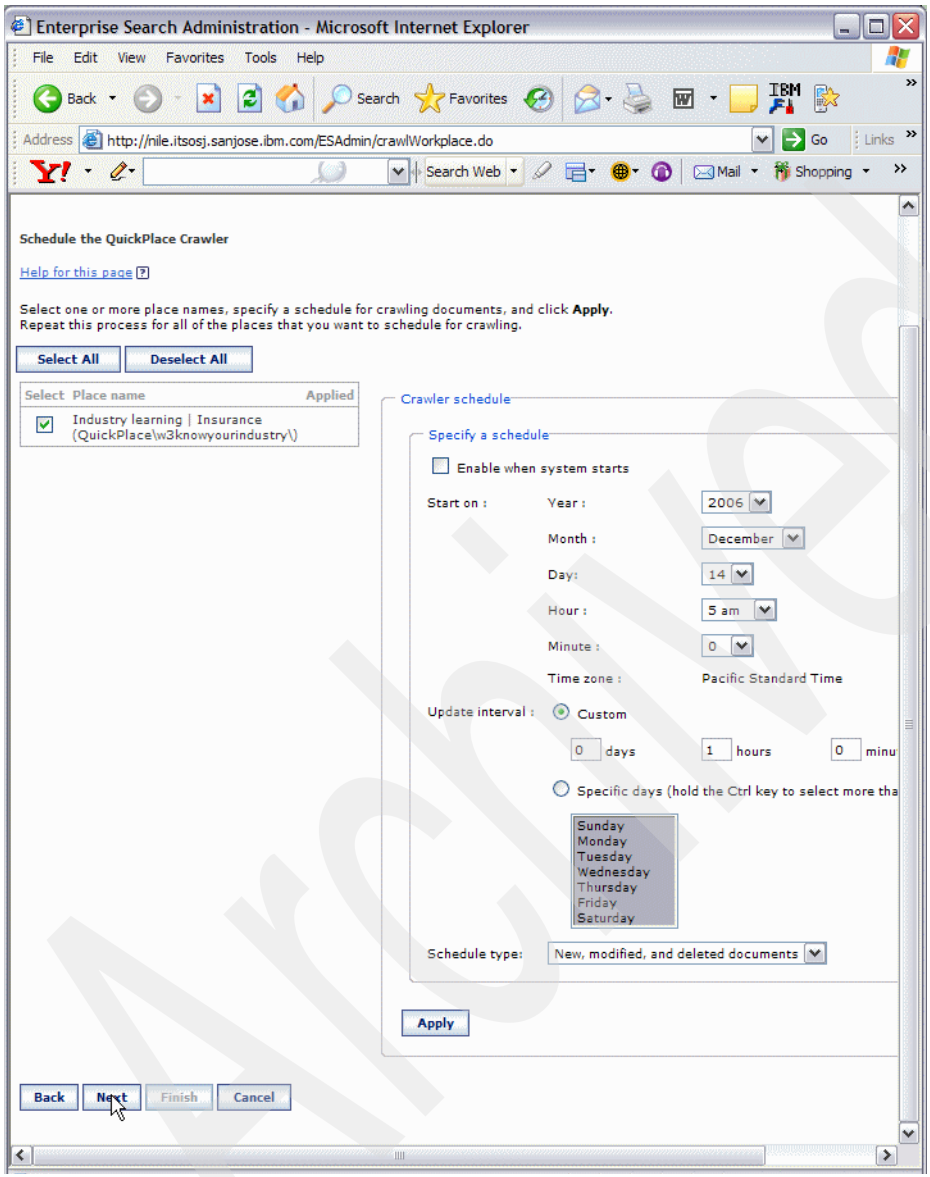

Figure 2-32 Crawl schedule

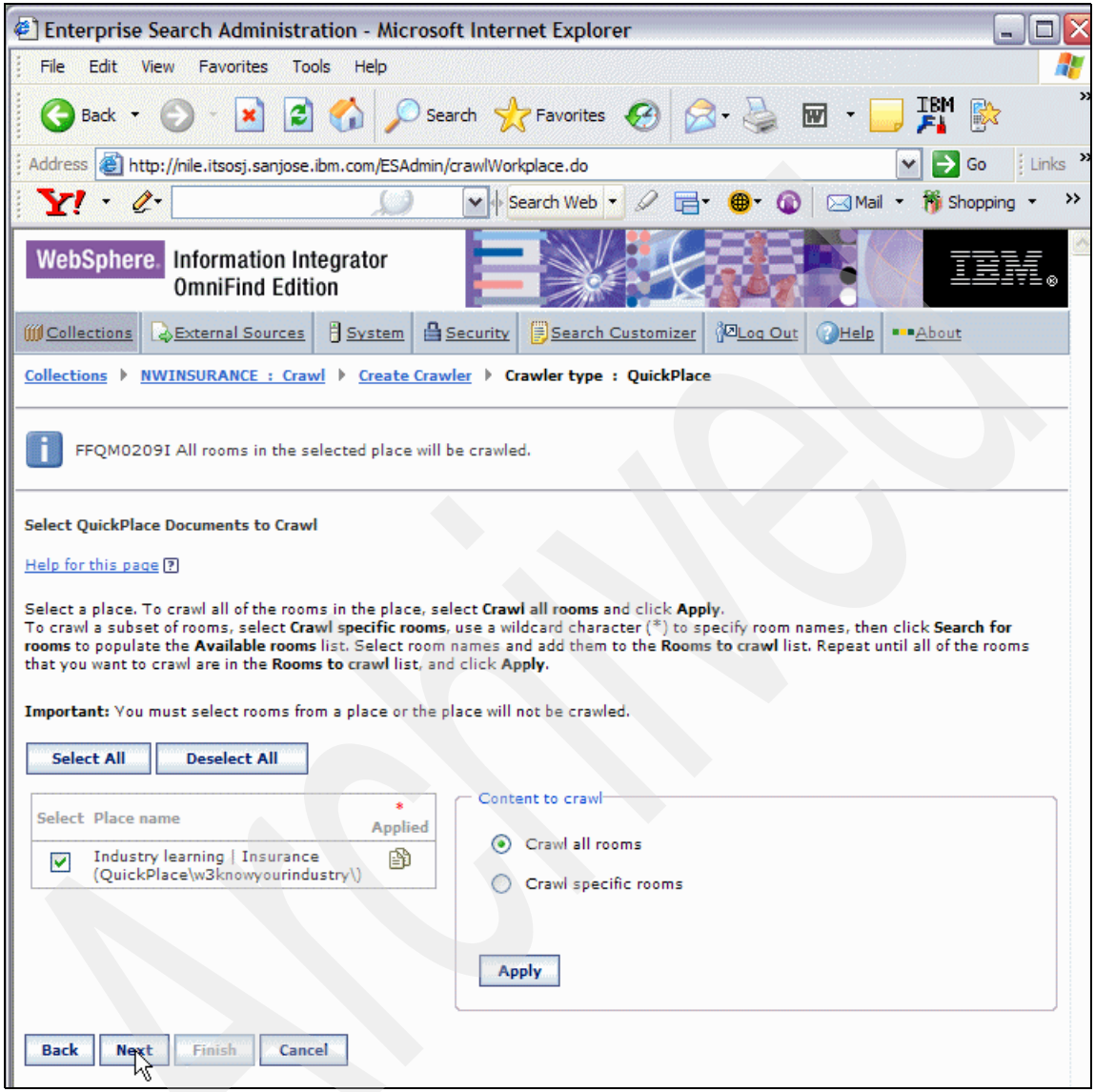

Figure 2-33 QuickPlace Documents to Crawl

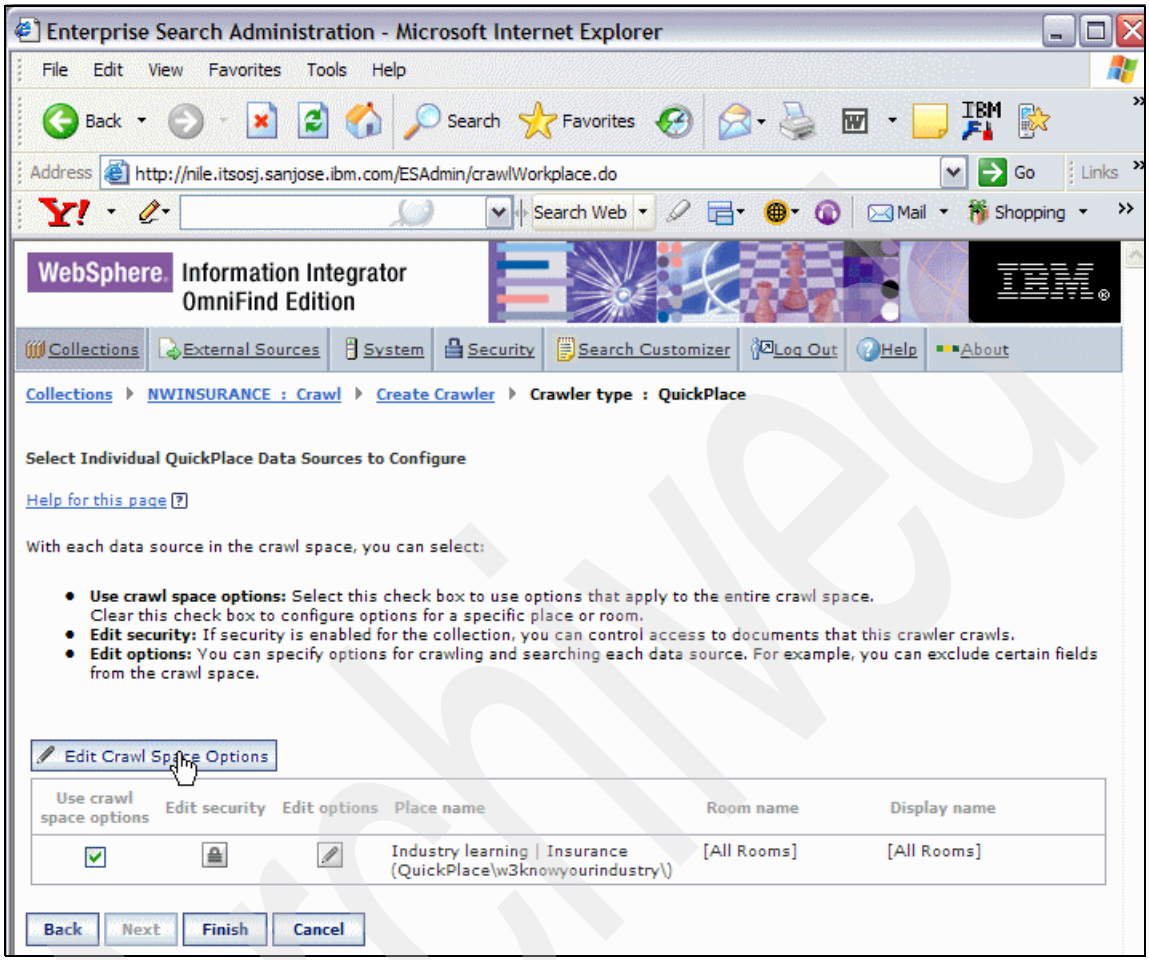

Figure 2-34 Edit Crawl Space Options 1/5

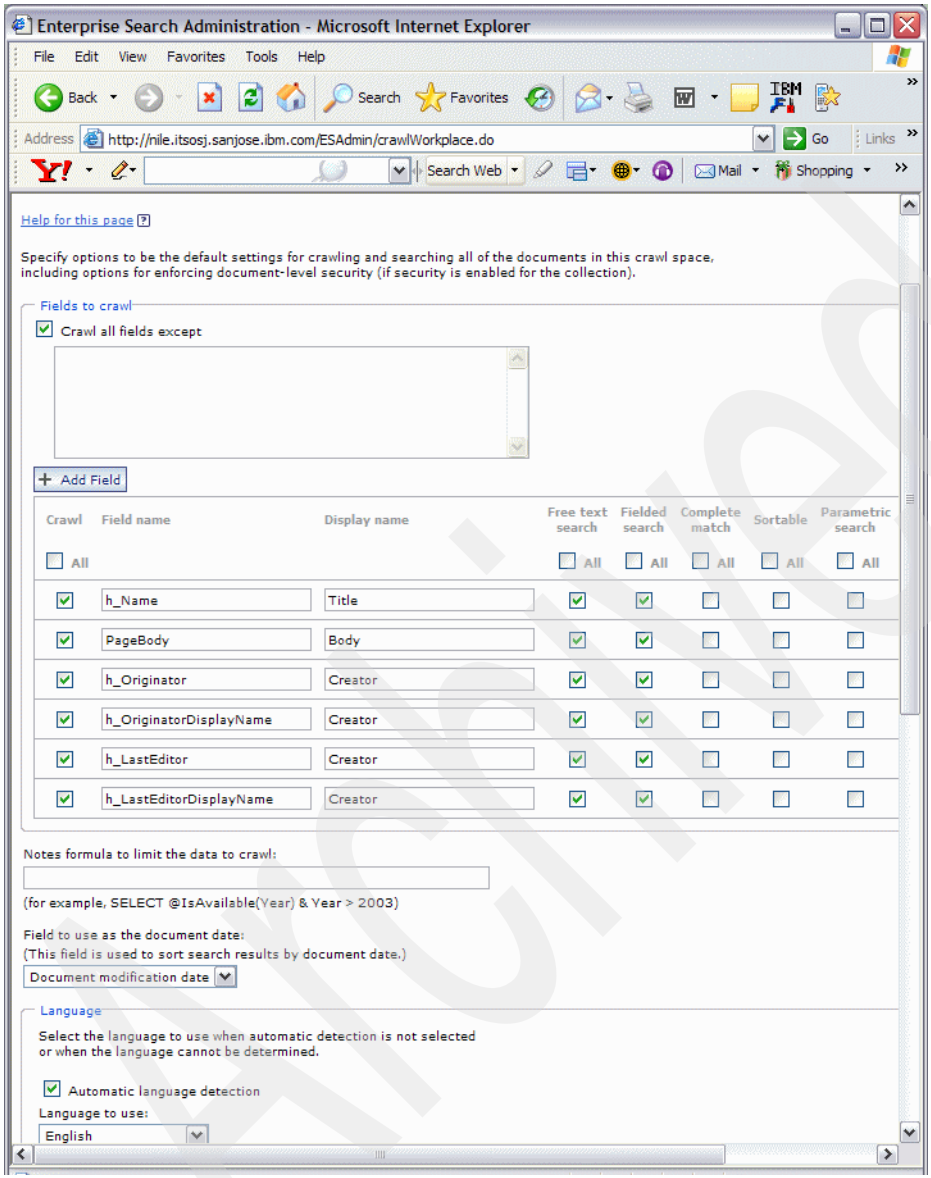

Figure 2-35 Edit Crawl Space Options 2/5

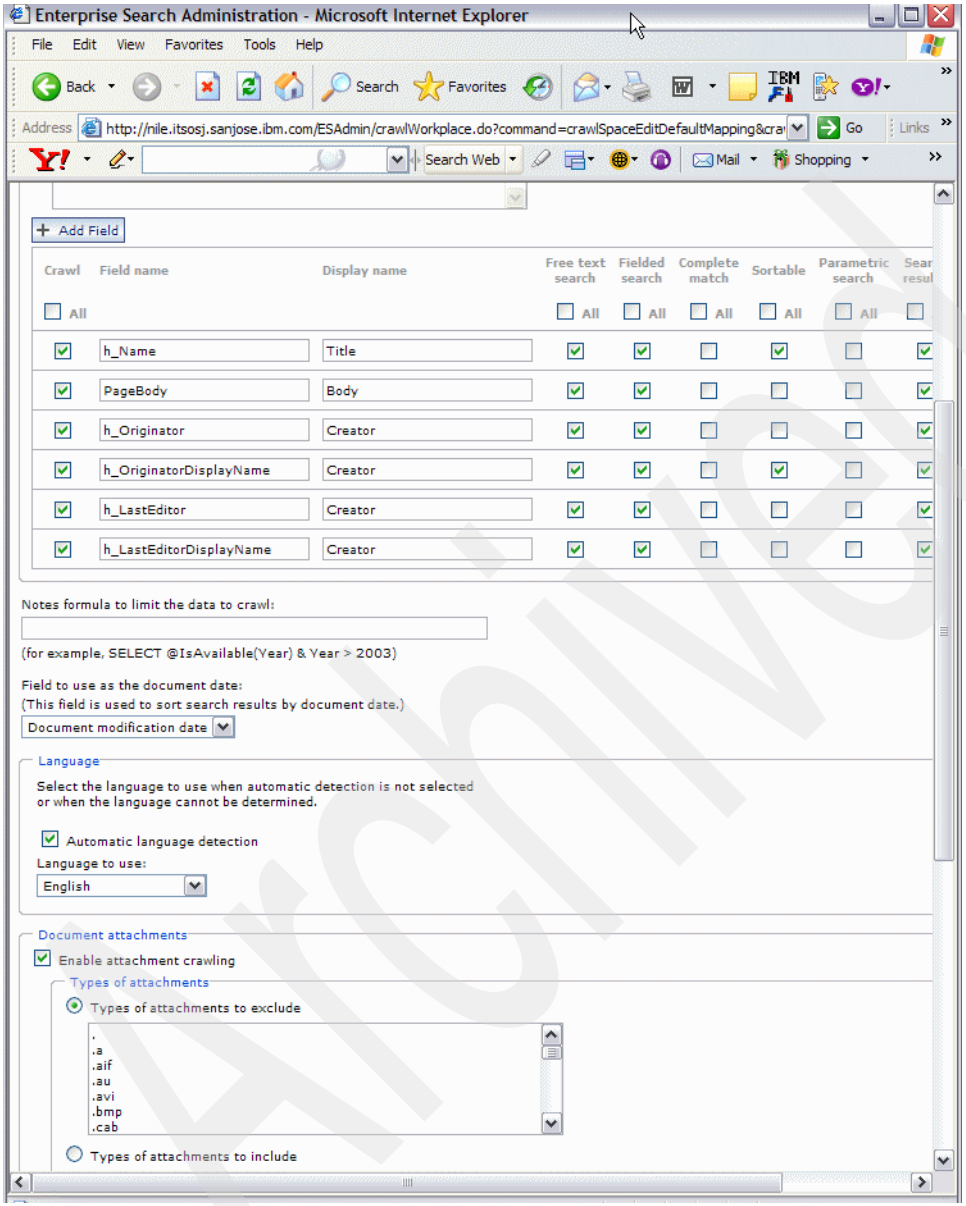

Figure 2-36 Edit Crawl Space Options 3/5

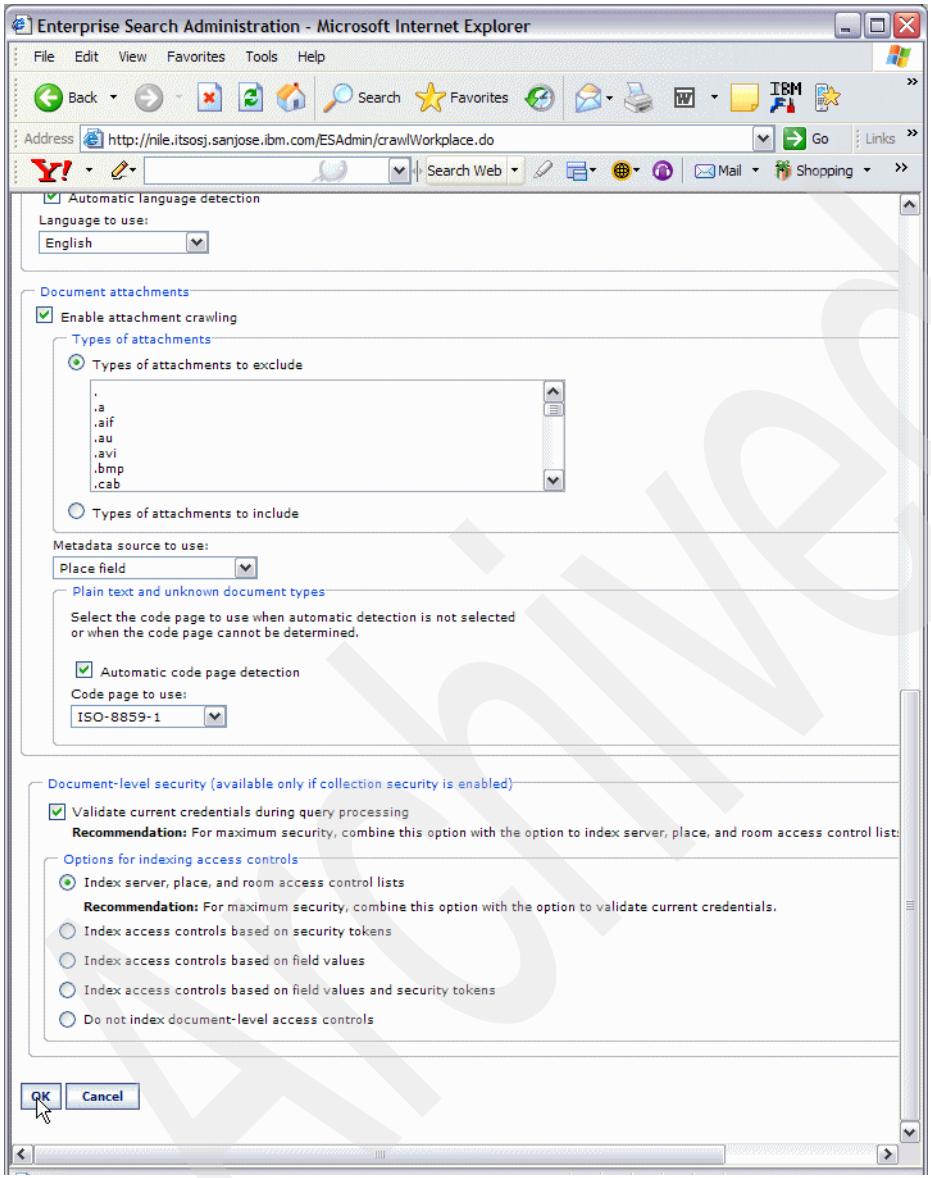

Figure 2-37 Edit Crawl Space Options 4/5

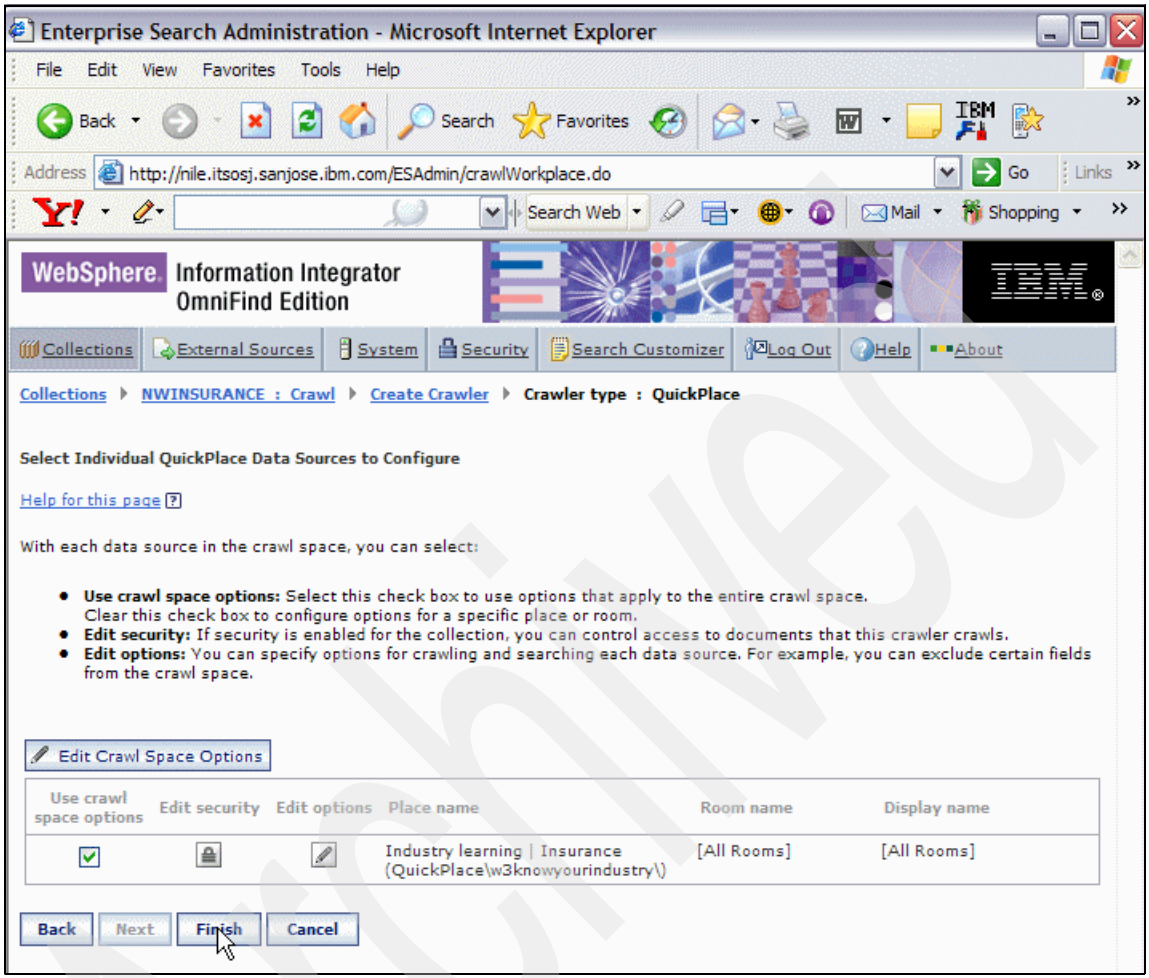

Figure 2-38 Edit Crawl Space Options 5/5

- Windows file system crawler

  Figure 2-39 on page 97 through Figure 2-55 on page 113 describe the creation and configuration of the Windows file system crawler, followed by a full crawl of the subdirectories defined.

  From the Crawl tab in Figure 2-39 on page 97, click **Create Crawler**. Select **Windows file system** from the drop-down list for Crawler type and click **Next**, as shown in Figure 2-40 on page 98.

  Provide details of the Windows file system crawler in Figure 2-41 on page 99, such as the Crawler name (NW_INSU_WF) and Maximum number of documents to crawl (20000). Click **Next**.

Click **Next** to specify the crawl schedule. Since we chose to schedule the crawls manually, click **Next** in Figure 2-42 on page 100. The next step is to select the Windows subdirectories to crawl. Figure 2-43 on page 101 shows the subdirectory (C:\Insurancedata) selected to crawl, which was obtained by first discovering local sources ('*' in the Main directory name or pattern followed by a click of **Search for subdirectories**, which lists all those found with the matching criteria in the Available subdirectories box and then copying those of interest to the Subdirectories to crawl box). Click **Next** in Figure 2-43 on page 101 to finish processing or **Edit security** or **Edit options**, as shown in Figure 2-44 on page 102.

Click the **Edit security** icon in Figure 2-44 on page 102 to view or modify the security options, as shown in Figure 2-45 on page 103, which shows the Validate current credentials during query processing and the Index file system access control lists options being selected. Click **OK**. In Figure 2-46 on page 104, click the **Edit options** icon to specify options for crawling and searching each subdirectory, as shown in Figure 2-47 on page 105, such as changing the levels of subdirectories to crawl and whether to use the file name as the document title. Click **OK** and then **Finish** on the subsequent window (Figure 2-48 on page 106) to complete the process of creating and configuring the Windows file system crawler.

We can now proceed to crawl the data sources, as described in "WSTEP4c: Crawl the data sources" on page 106.

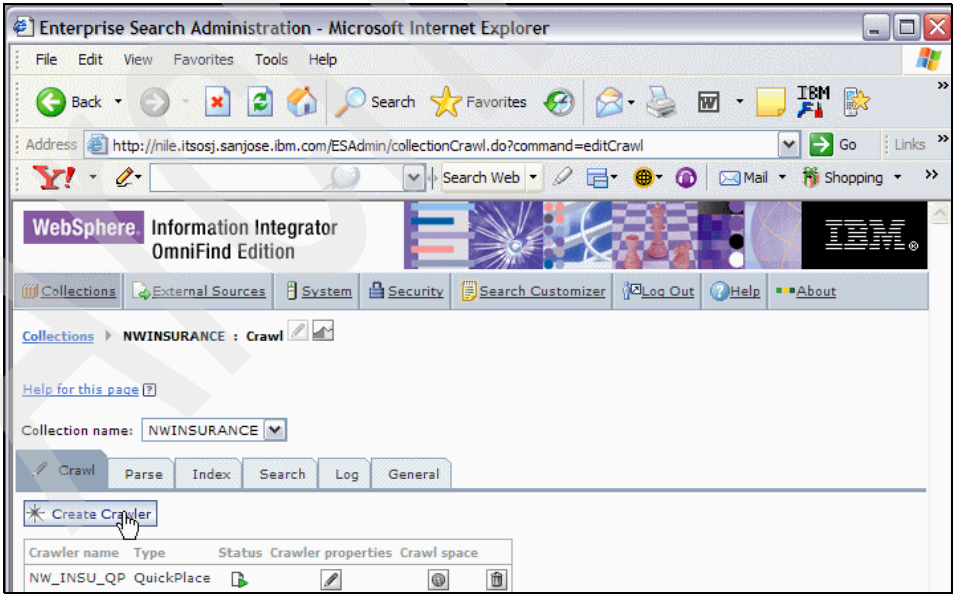

Figure 2-39 Create Crawler

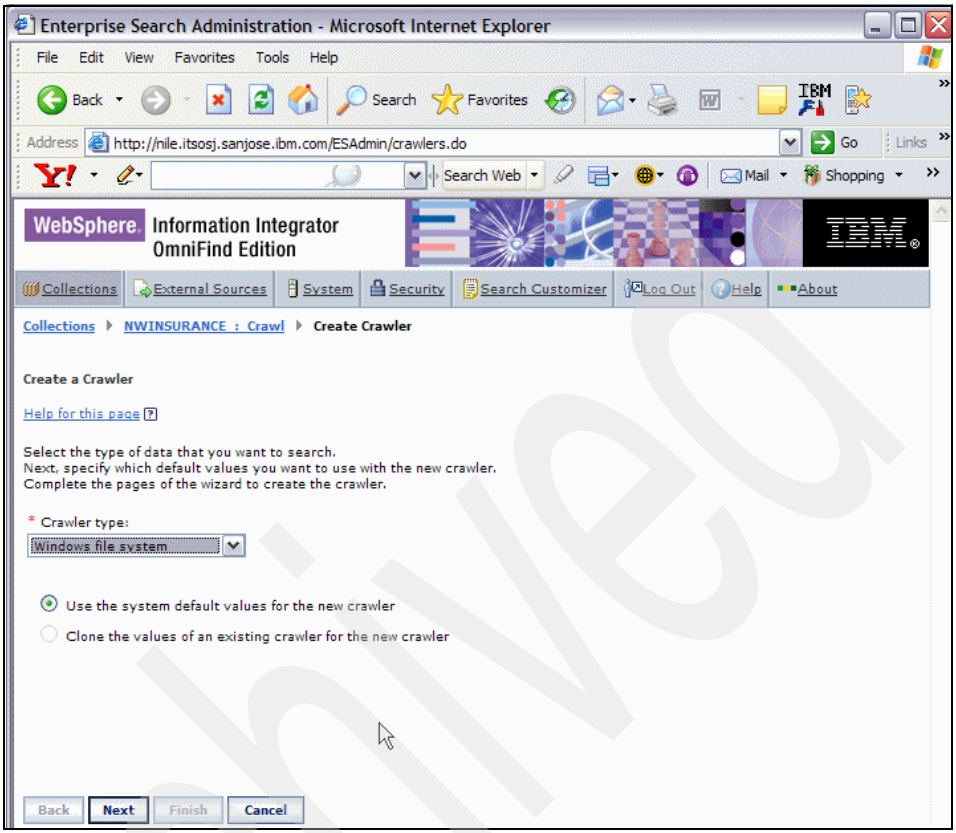

Figure 2-40 Windows file system Crawler type

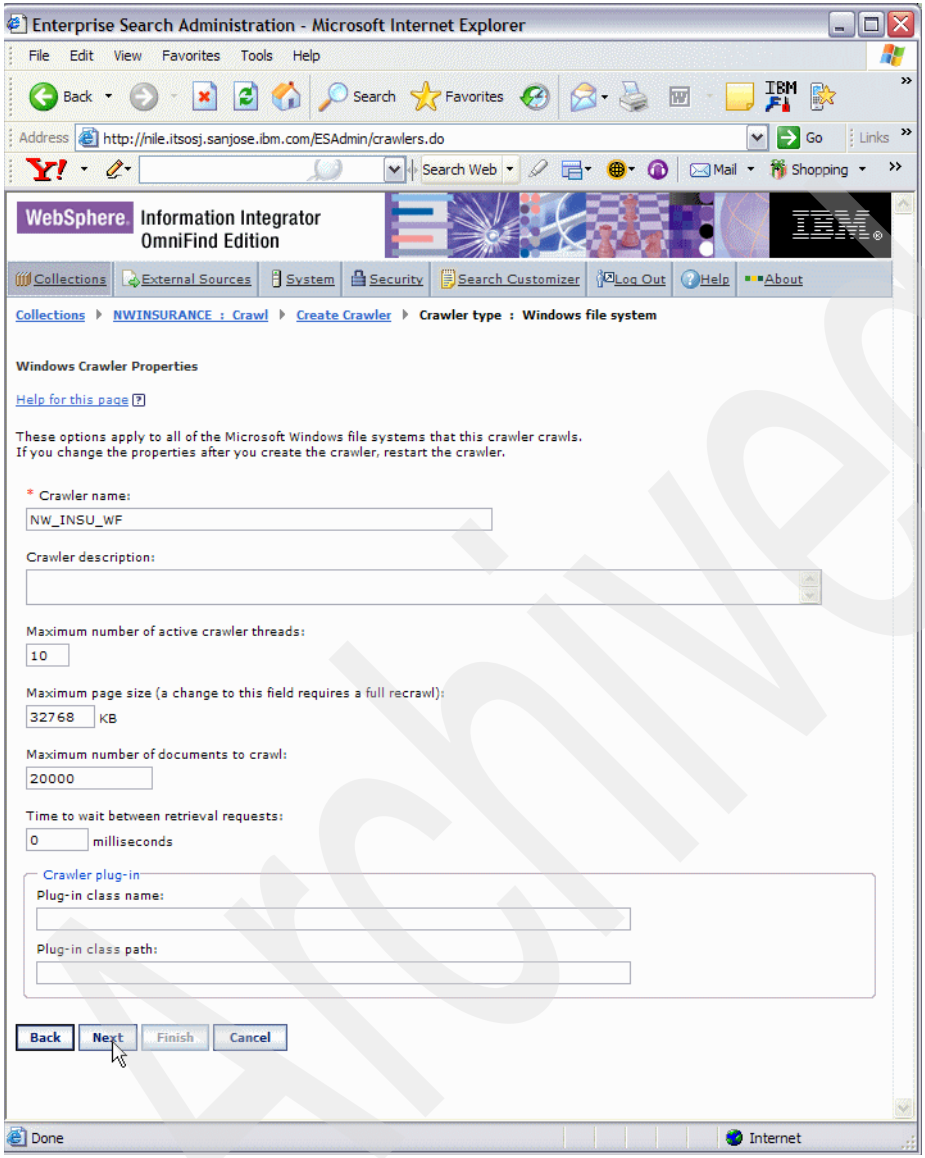

Figure 2-41 Specify Crawler details

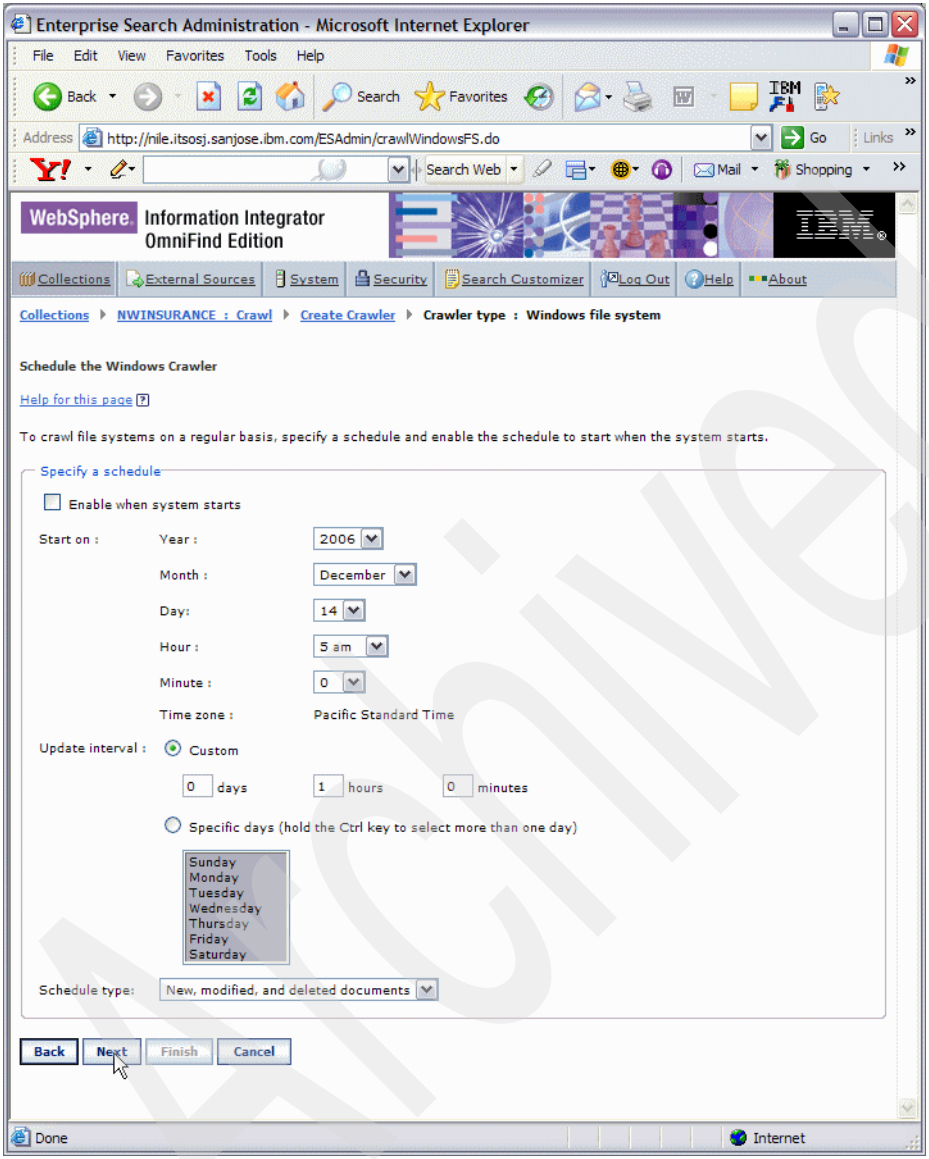

Figure 2-42 Specify Crawler schedule

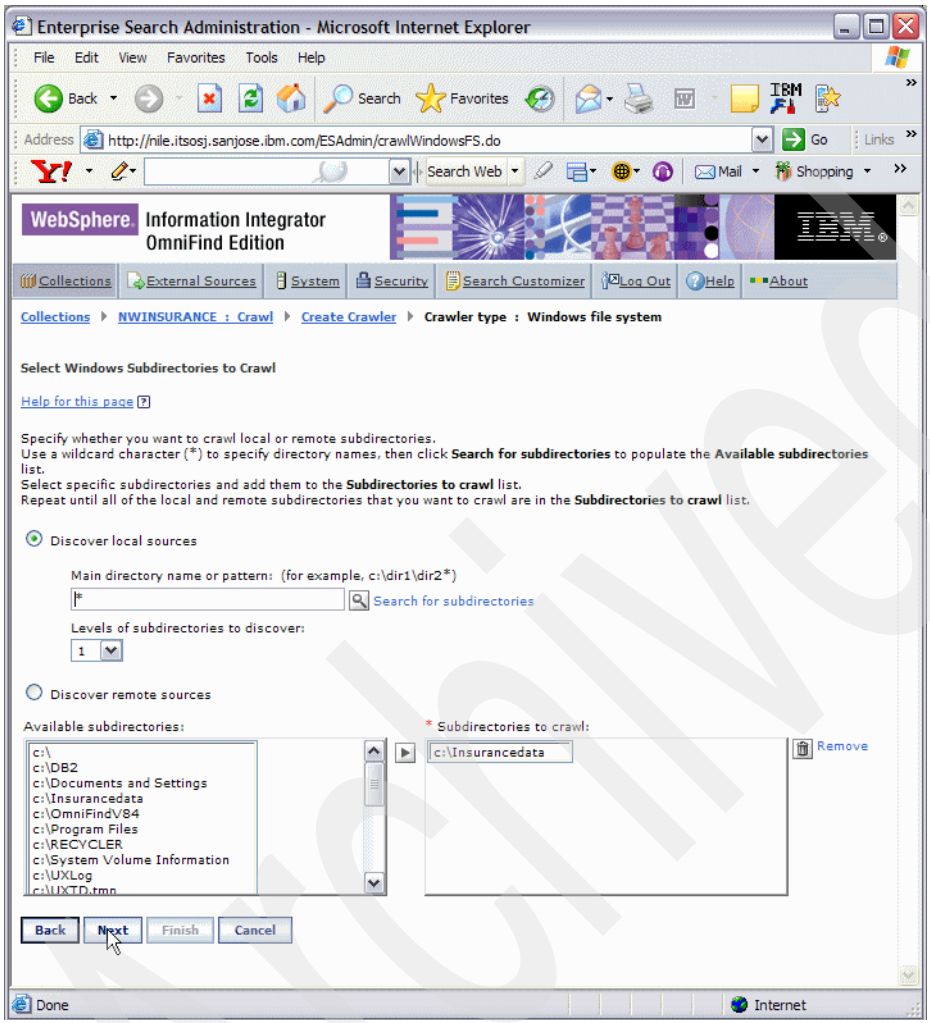

Figure 2-43 Specify Windows subdirectories to crawl 1/2

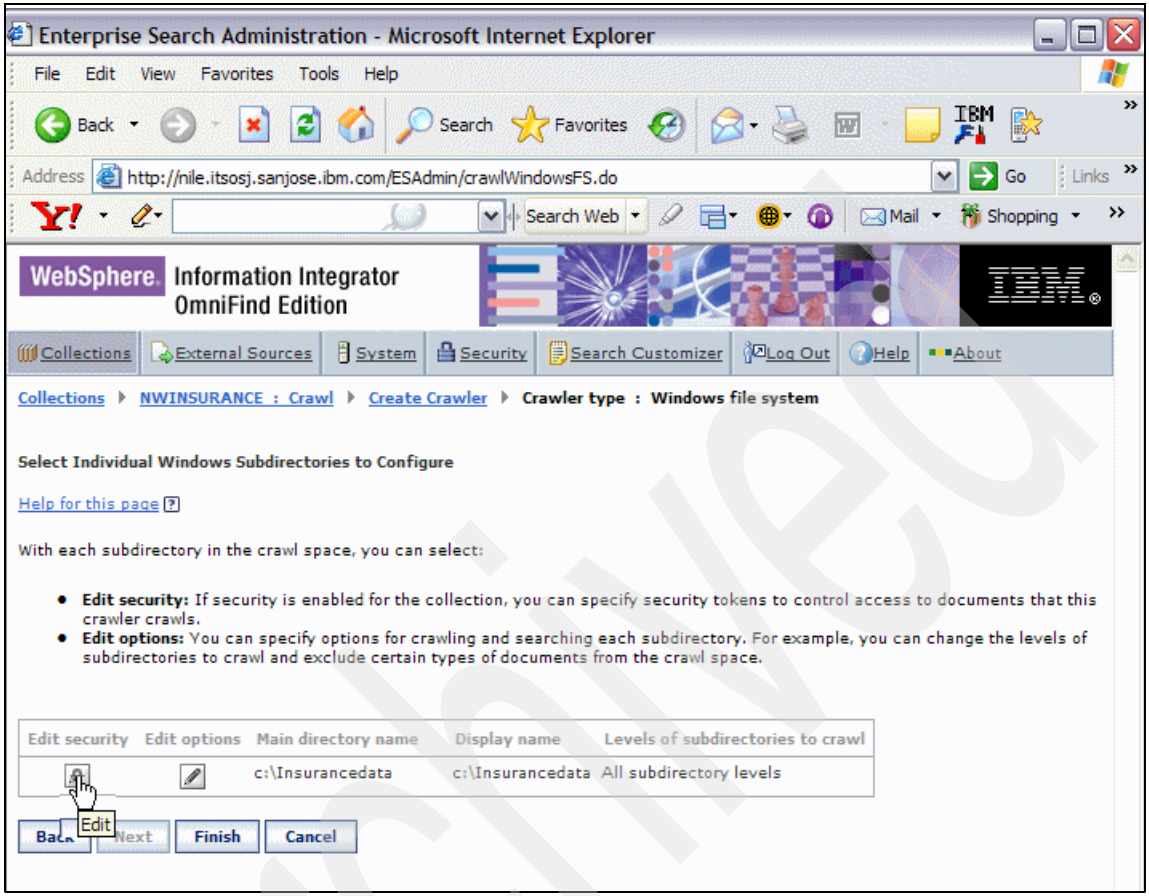

Figure 2-44 Specify Windows subdirectories to crawl 2/2

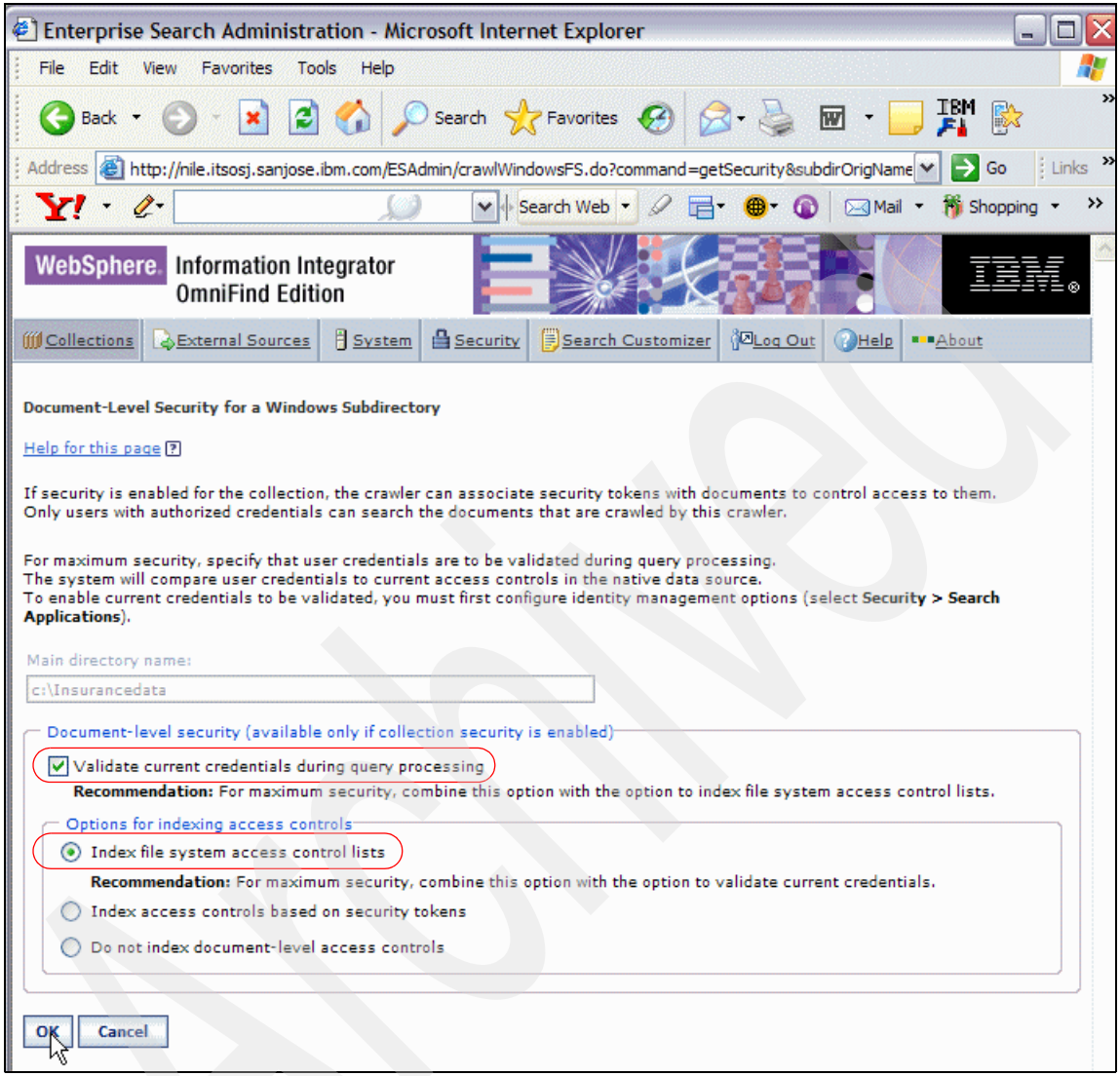

Figure 2-45 Document-level security options

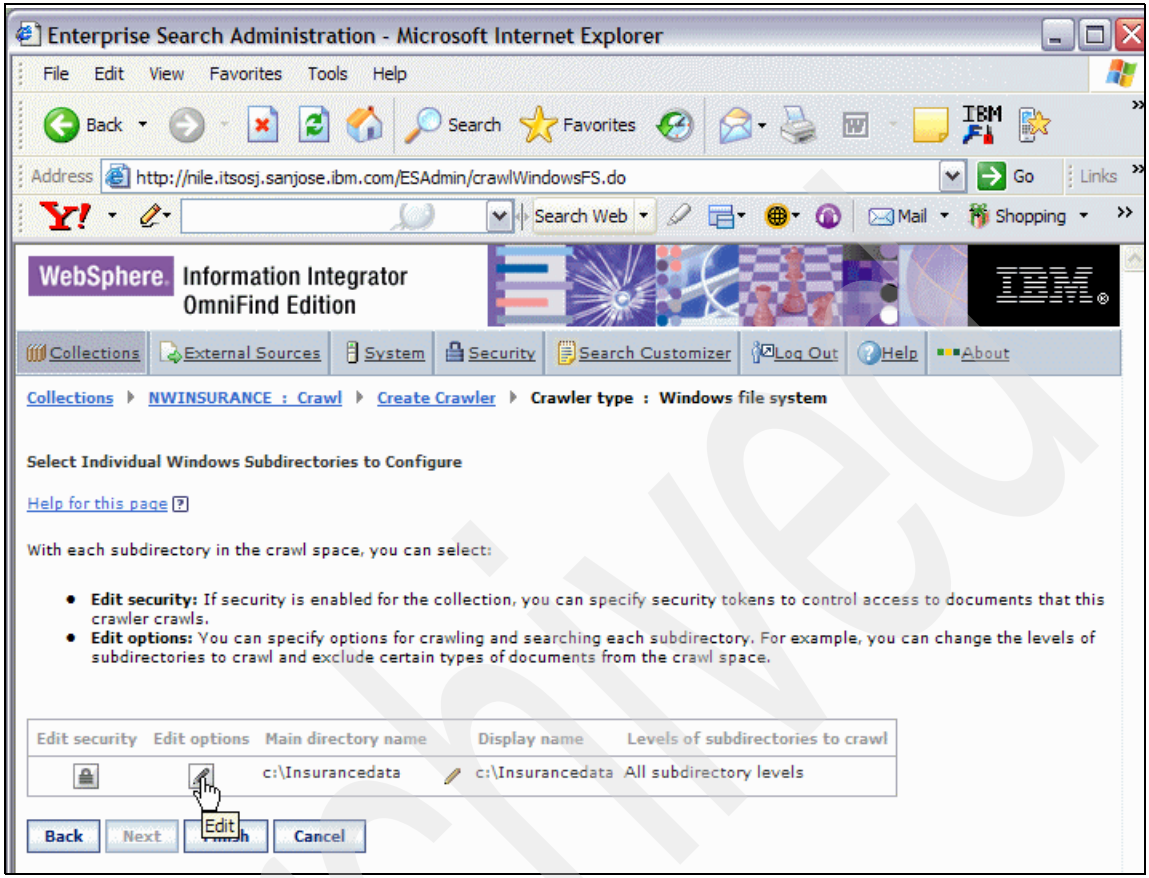

Figure 2-46 Click Edit options icon

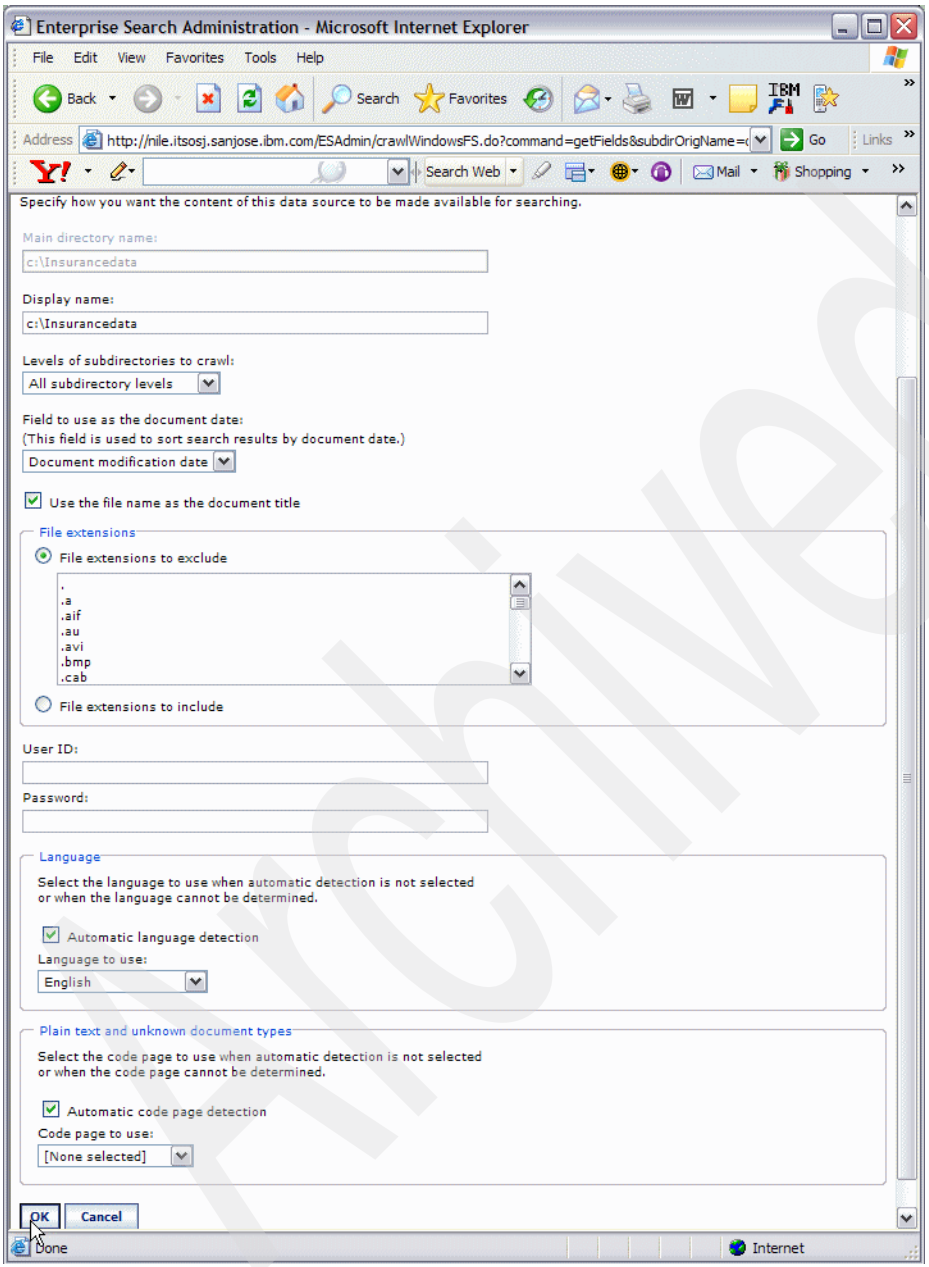

Figure 2-47 Edit options

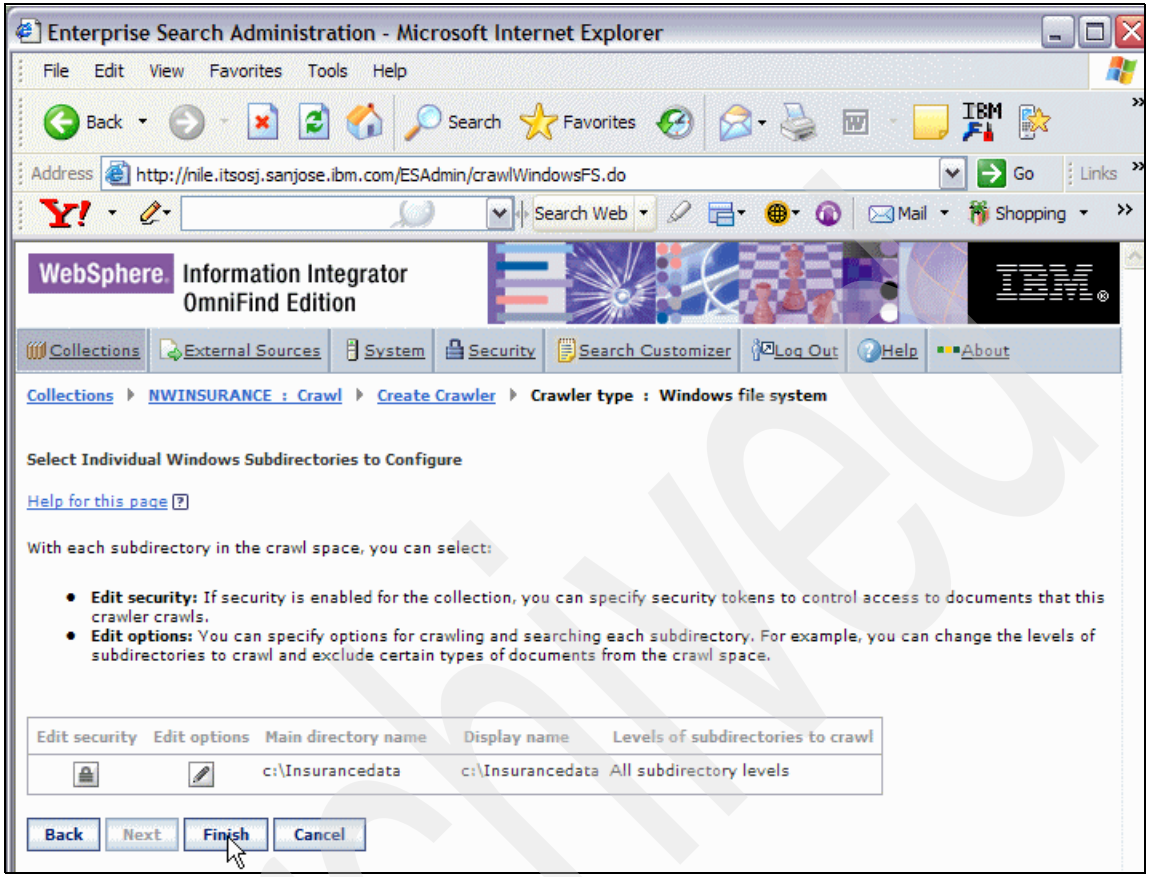

Figure 2-48 Finish creation and configuration of the Windows file system crawler

WSTEP4c: Crawl the data sources

In this step, we can initiate a crawl of the QuickPlace and Windows file system data sources described earlier.

> **Attention:** When running crawlers in OmniFind V8.4, we strongly recommend that you run the parser at the same time. This is because a file queue is used to store crawled data instead of the DB2 table used in OmniFind V8.3. The file queue can fill up if the parser is not used to parse and delete the documents in the file queue while the crawler is running.

- Initiate a full crawl of the QuickPlace places defined earlier, as described in Figure 2-49 on page 107 through Figure 2-52 on page 110.

Figure 2-49 shows the switch to Monitor mode for the Crawl component of the NWINSURANCE collection by clicking the **Monitor** icon. Start the crawler session by clicking the start button for the NW_INSU_QP QuickPlace crawler, as shown in Figure 2-50 on page 108. Once the Status icon turns green, click **Details**. Start a full crawl by clicking the appropriate icon, as shown in Figure 2-51 on page 109. Monitor the status of the crawl by clicking the **Refresh** button intermittently, until the completion of the crawl, as shown in Figure 2-52 on page 110.

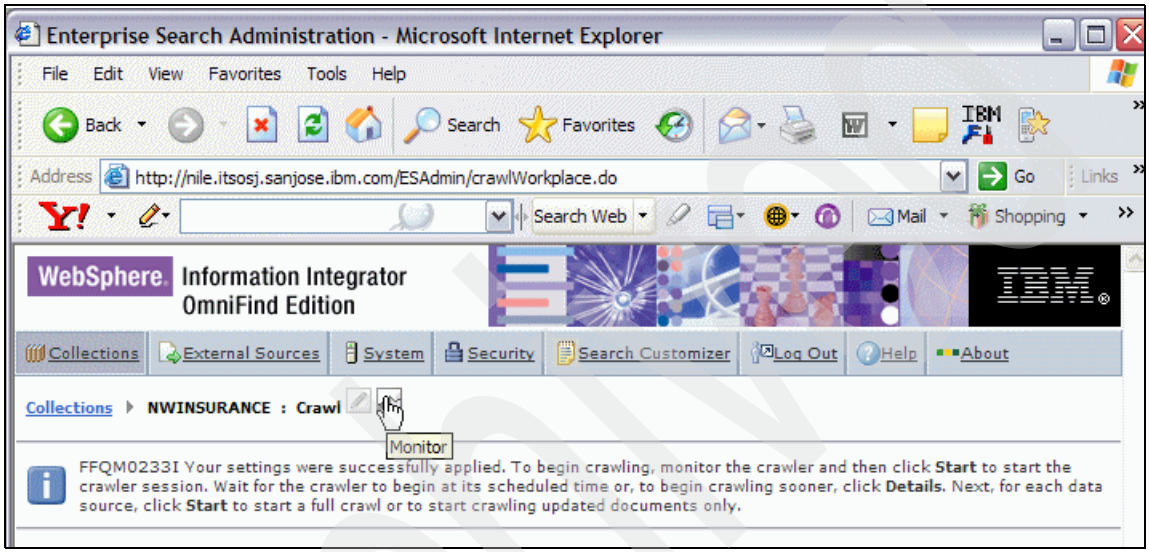

Figure 2-49 Monitor mode

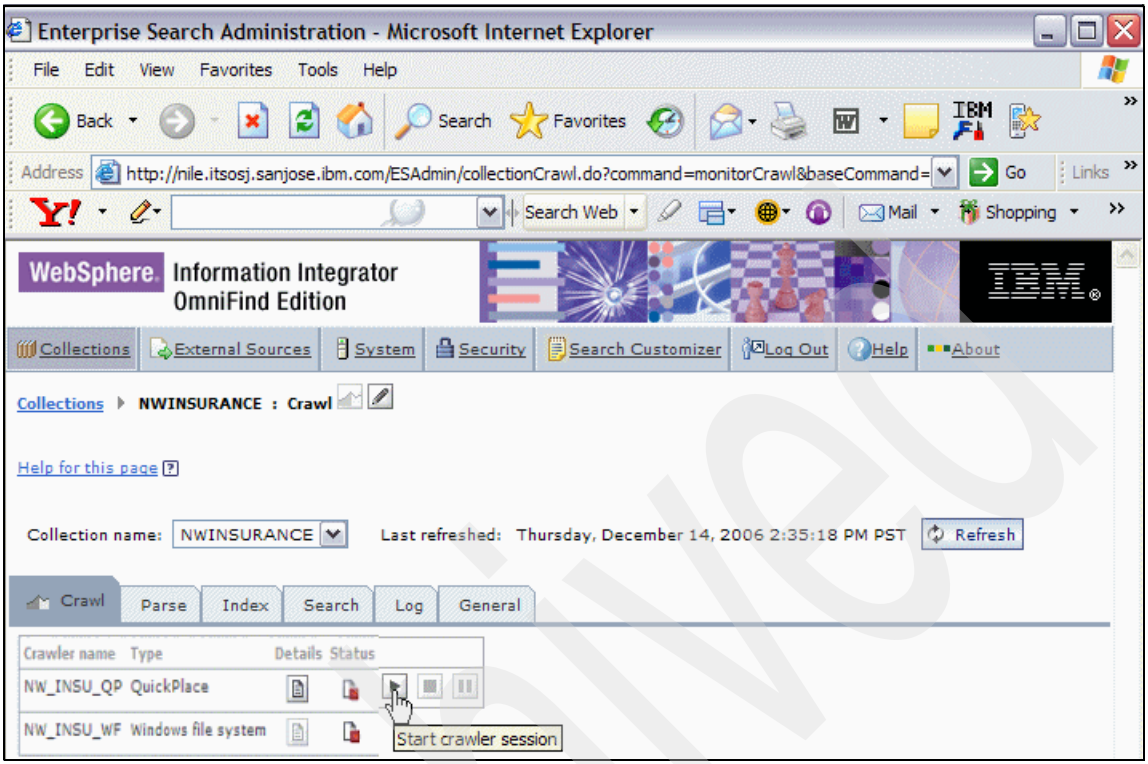

Figure 2-50 Start crawler session

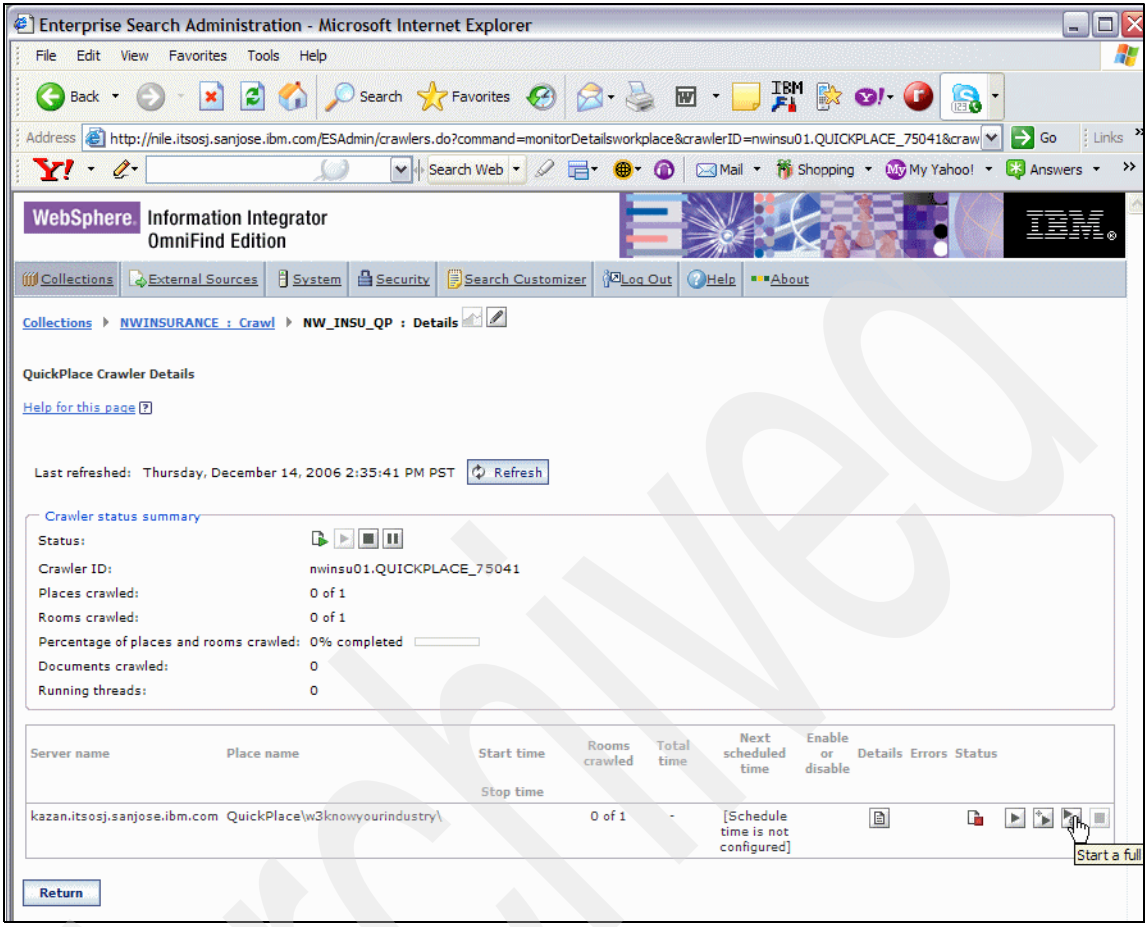

Figure 2-51 Start a full crawl

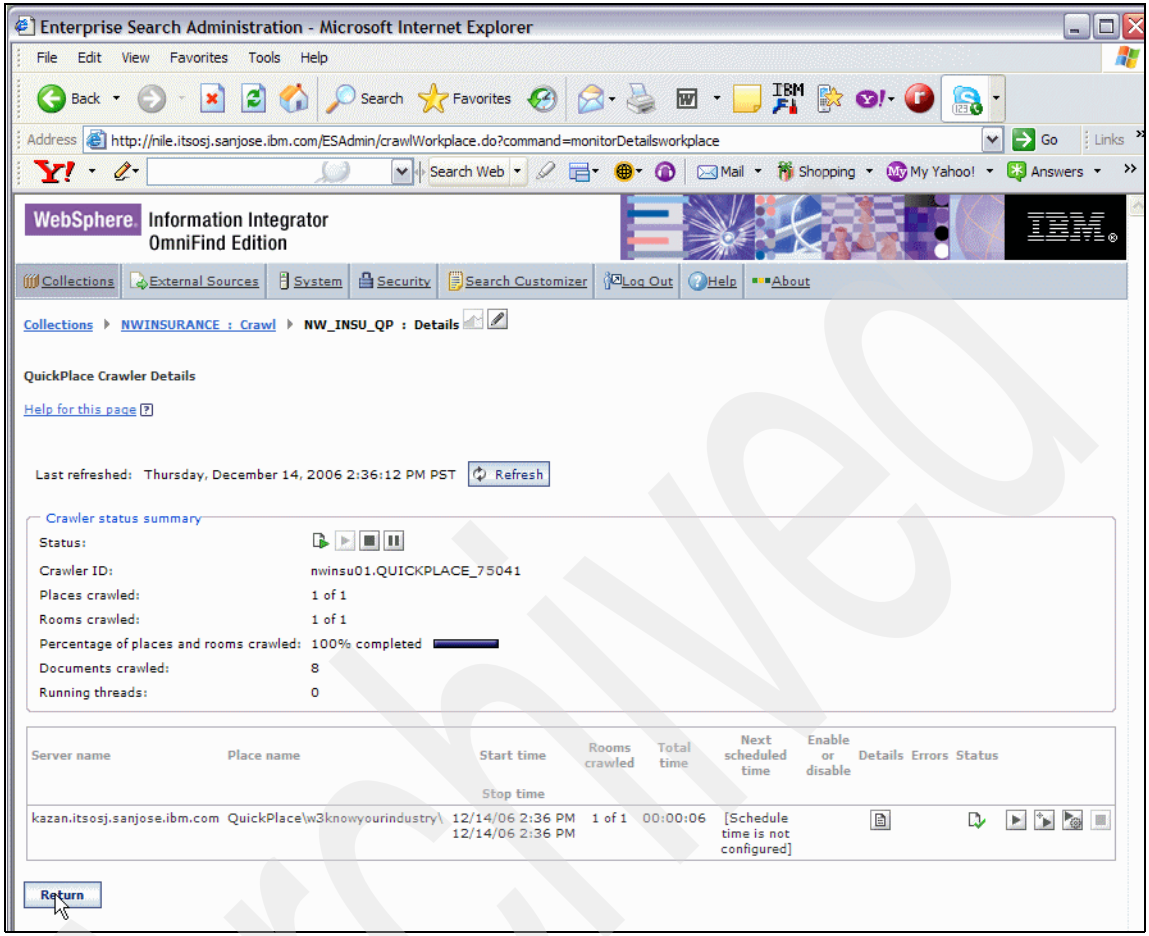

Figure 2-52 Full crawl completion status

- Initiate a full crawl of the Windows file system crawl space defined earlier, as described in Figure 2-53 on page 111 through Figure 2-55 on page 113.

  Start the crawler session by clicking the start button for the NW_INSU_WF Windows file system crawler, as shown in Figure 2-53 on page 111. Once the Status icon turns green, click **Details**. Start a full crawl by clicking the appropriate icon, as shown in Figure 2-54 on page 112. Monitor the status of the crawl by clicking the **Refresh** button intermittently, until the completion of the crawl, as shown in Figure 2-55 on page 113.

We can now proceed to parse the crawled data, as described in “WSTEP4d: Parse the crawled data” on page 113.

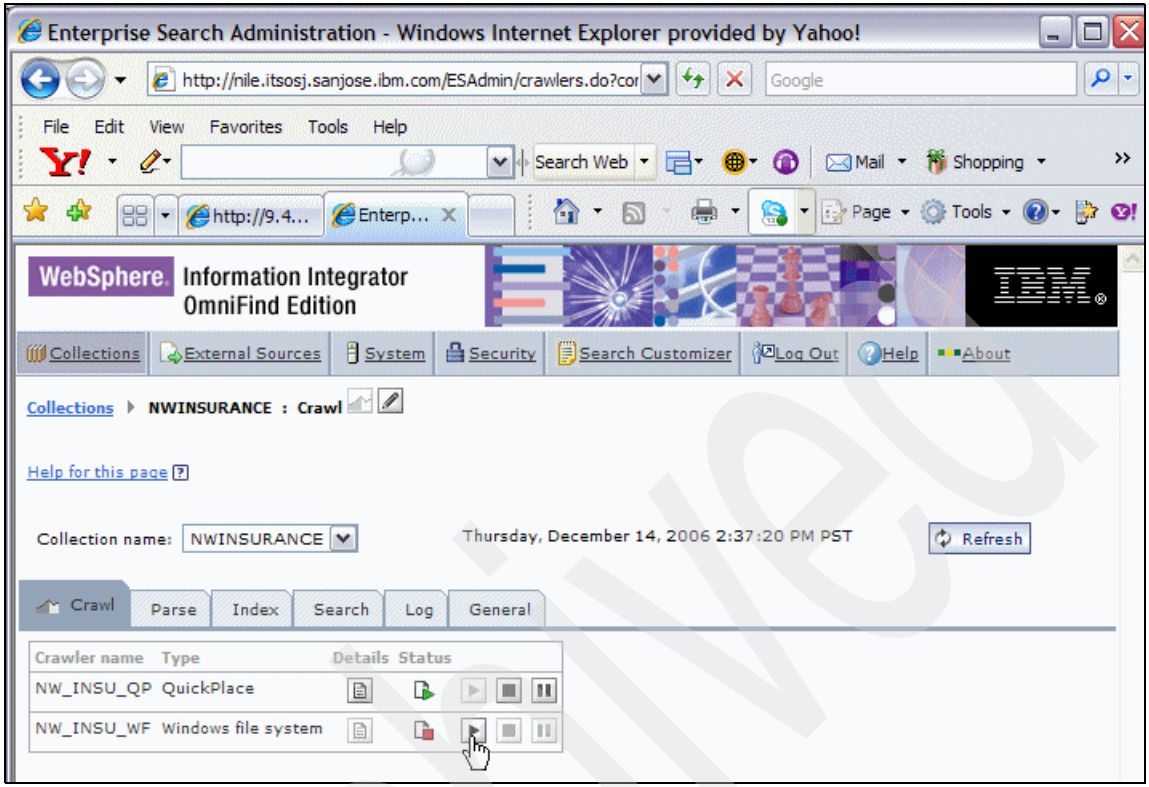

Figure 2-53 Start the crawler session

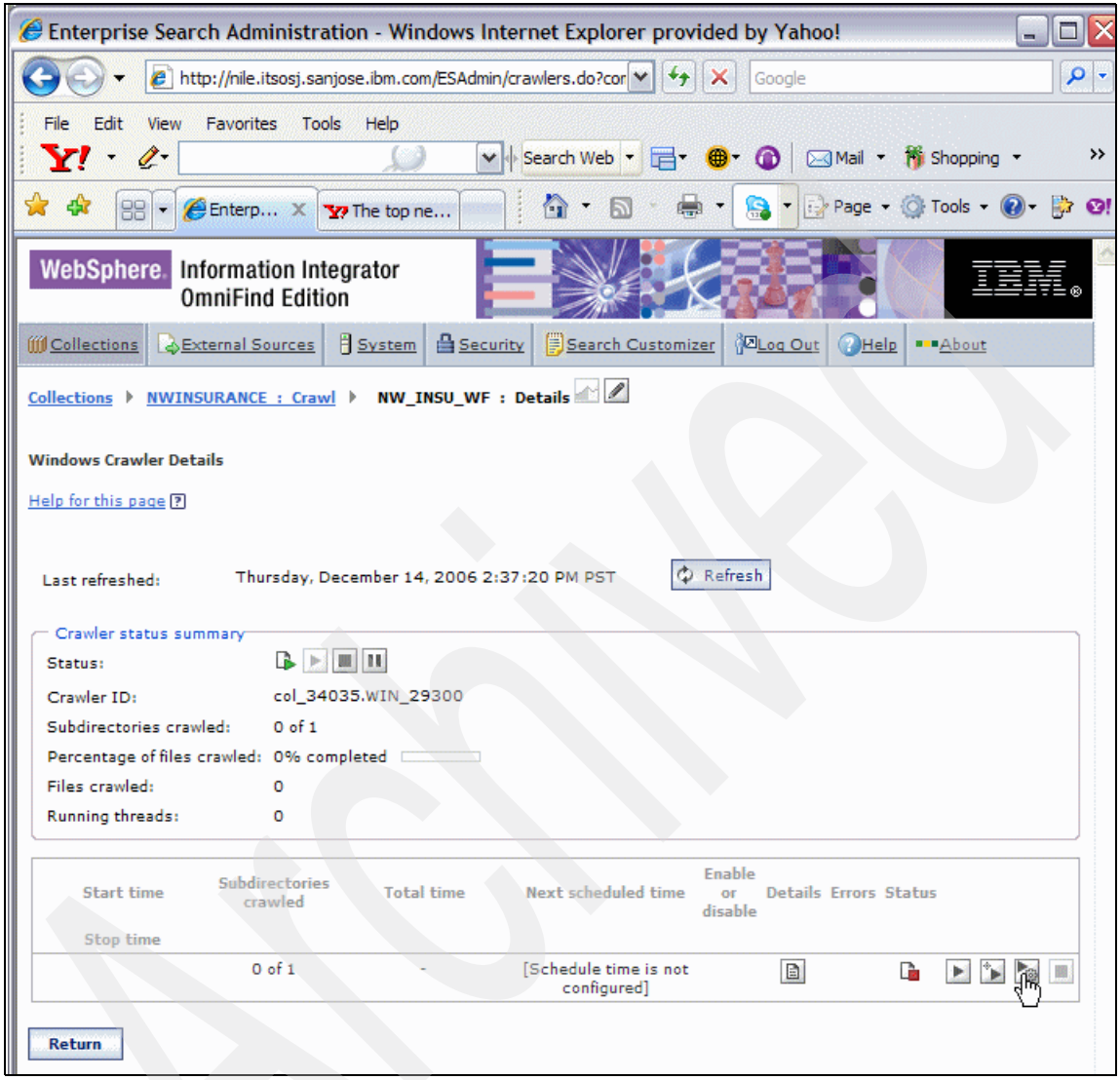

Figure 2-54 Start a full crawl of the Windows file system crawl space

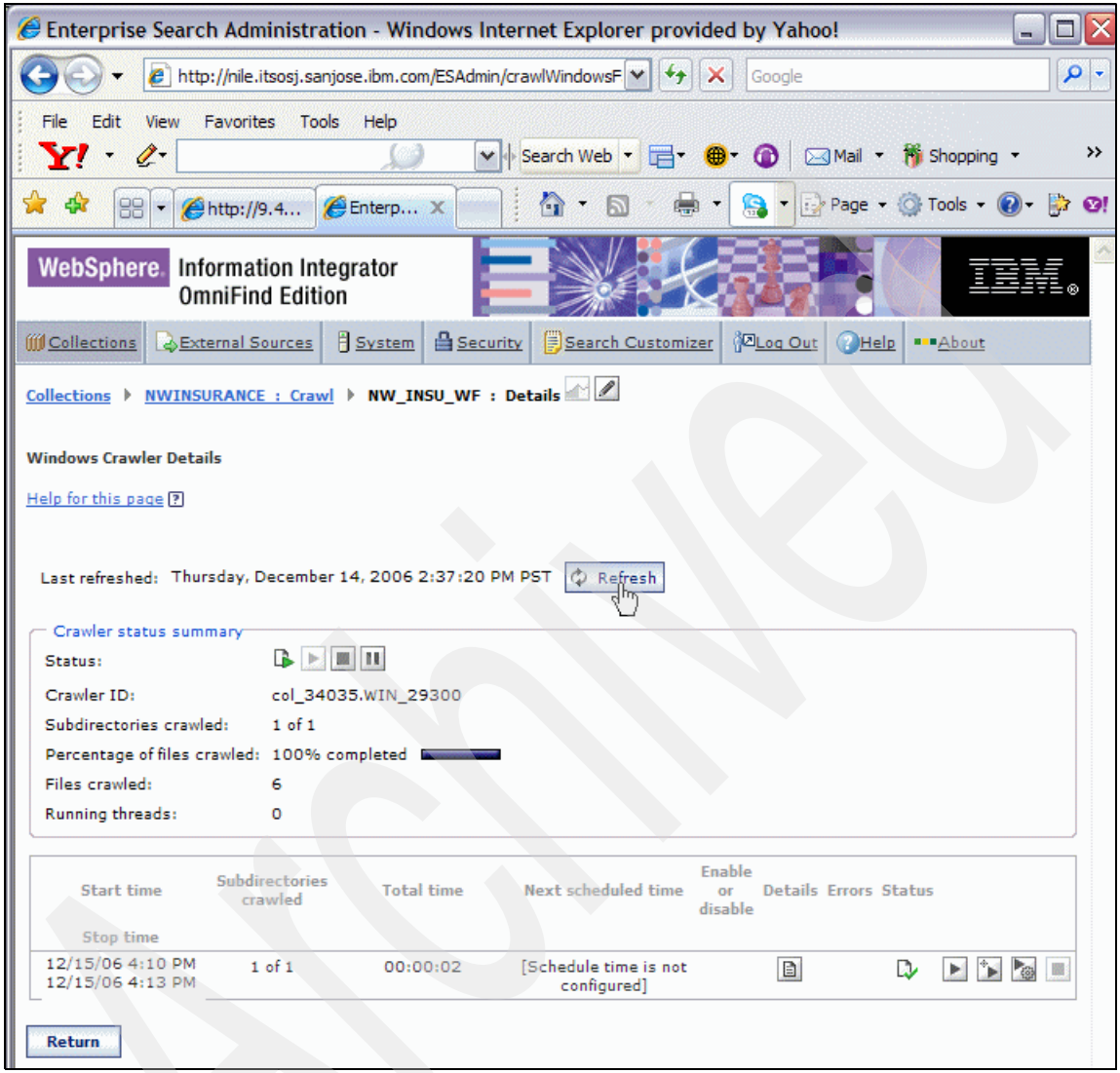

Figure 2-55 Full crawl completion status

WSTEP4d: Parse the crawled data

Figure 2-56 on page 114 through Figure 2-58 on page 116 describe the steps in parsing the crawled data.

From the Parse tab in Monitor mode, start the parser by clicking the start button, as shown in Figure 2-56 on page 114. After the Status icon turns green, you can monitor the progress of parsing by clicking **Details**, as shown in Figure 2-57 on

page 115. Periodically click the **Refresh** button until the parser completes processing, as shown in Figure 2-58 on page 116. Review the parsing statistics.

We can now proceed to build the main index, as described in “WSTEP4e: Build the main index” on page 116.

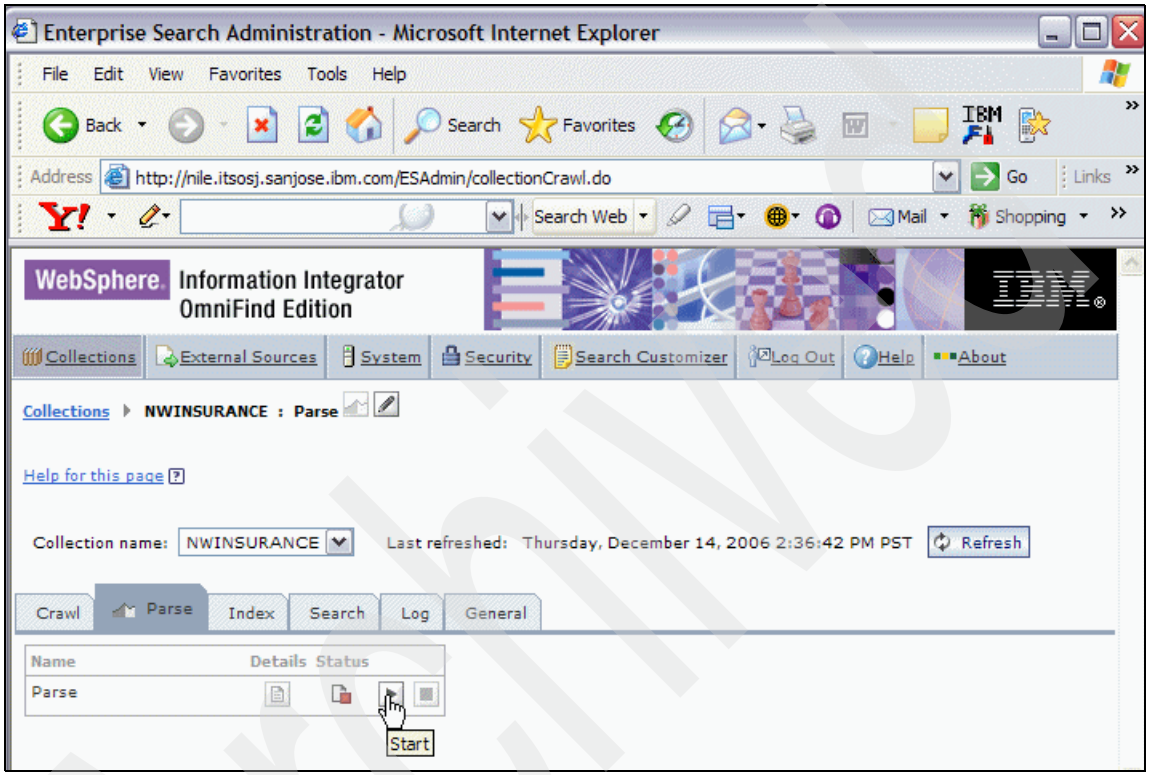

Figure 2-56 Start the parser

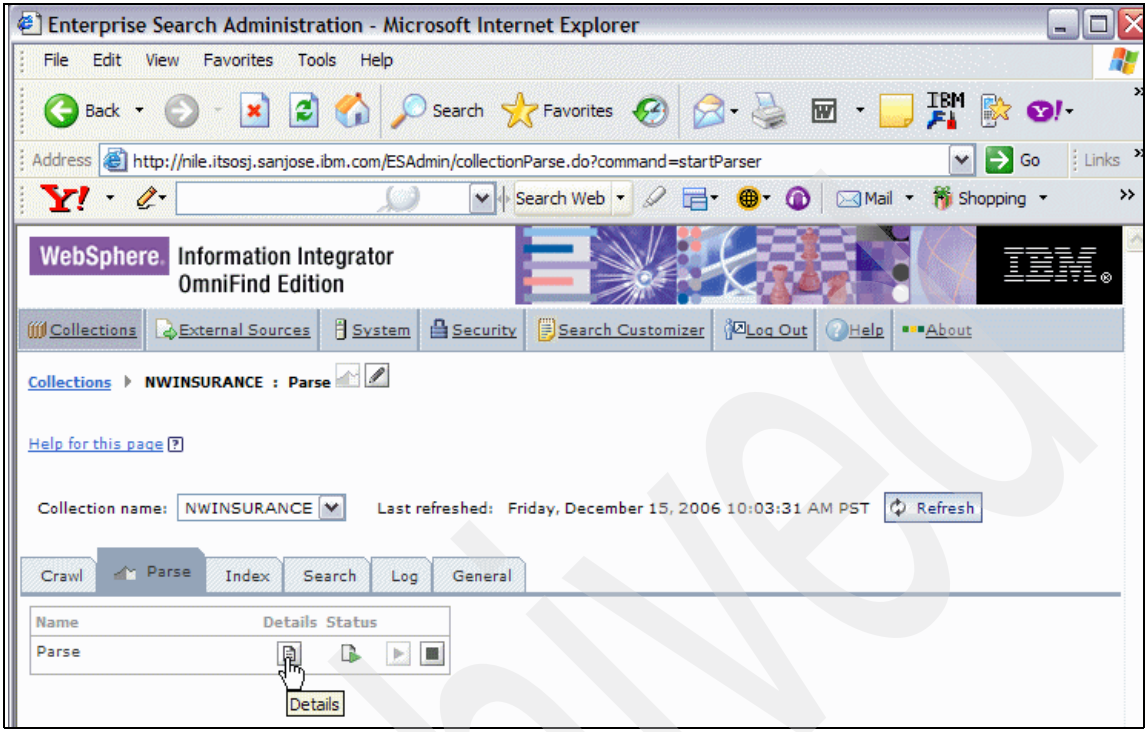

Figure 2-57 View parser details 1/2

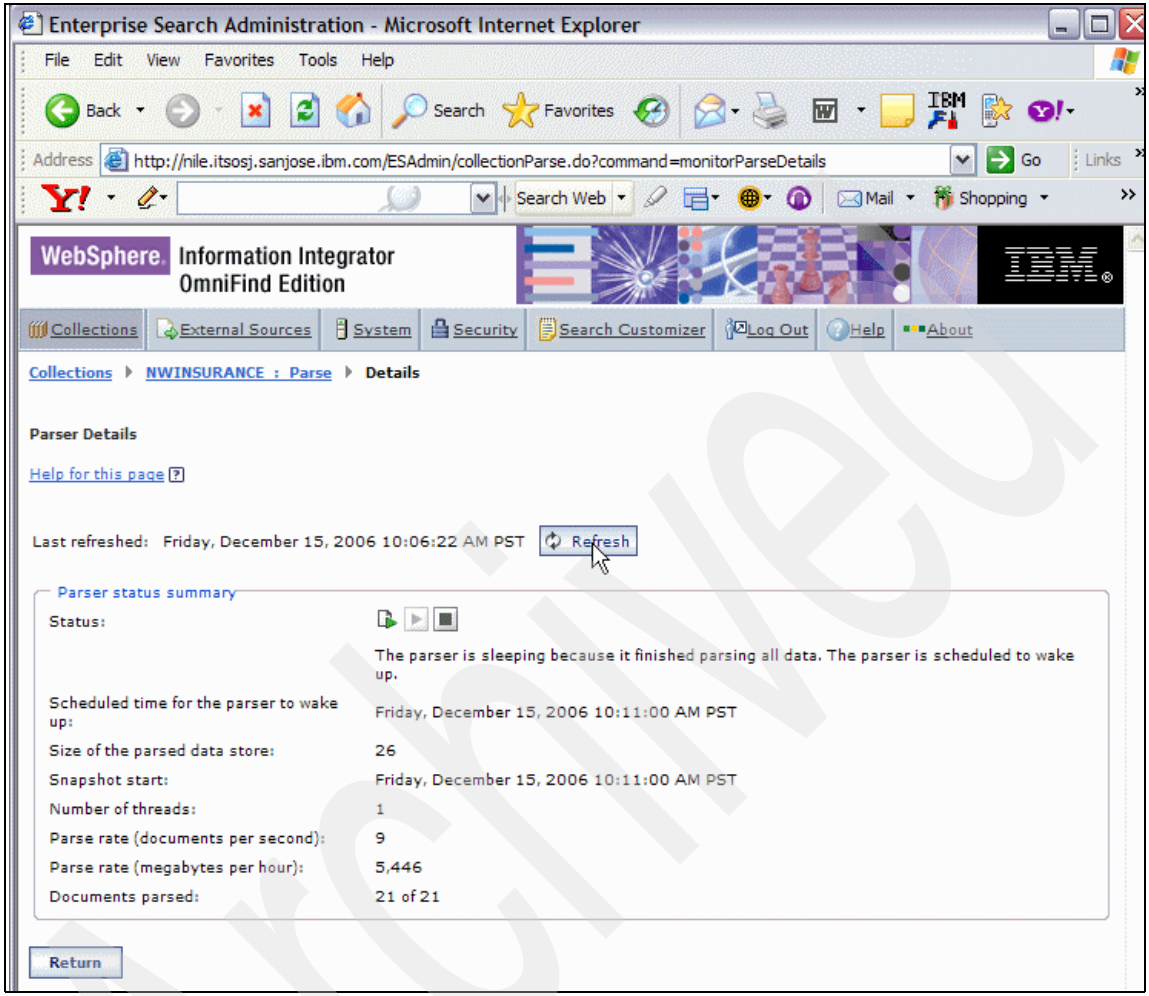

Figure 2-58 View parser details 2/2

WSTEP4e: Build the main index

From the Index tab in Monitor mode, start the main index build by clicking the start button, as shown in Figure 2-59 on page 117. Periodically click the **Refresh** button until the index build completes processing, as shown in Figure 2-60 on page 118. Review the index statistics.

We can now proceed to define the security settings, as described in “WSTEP4f: Define security settings” on page 119.

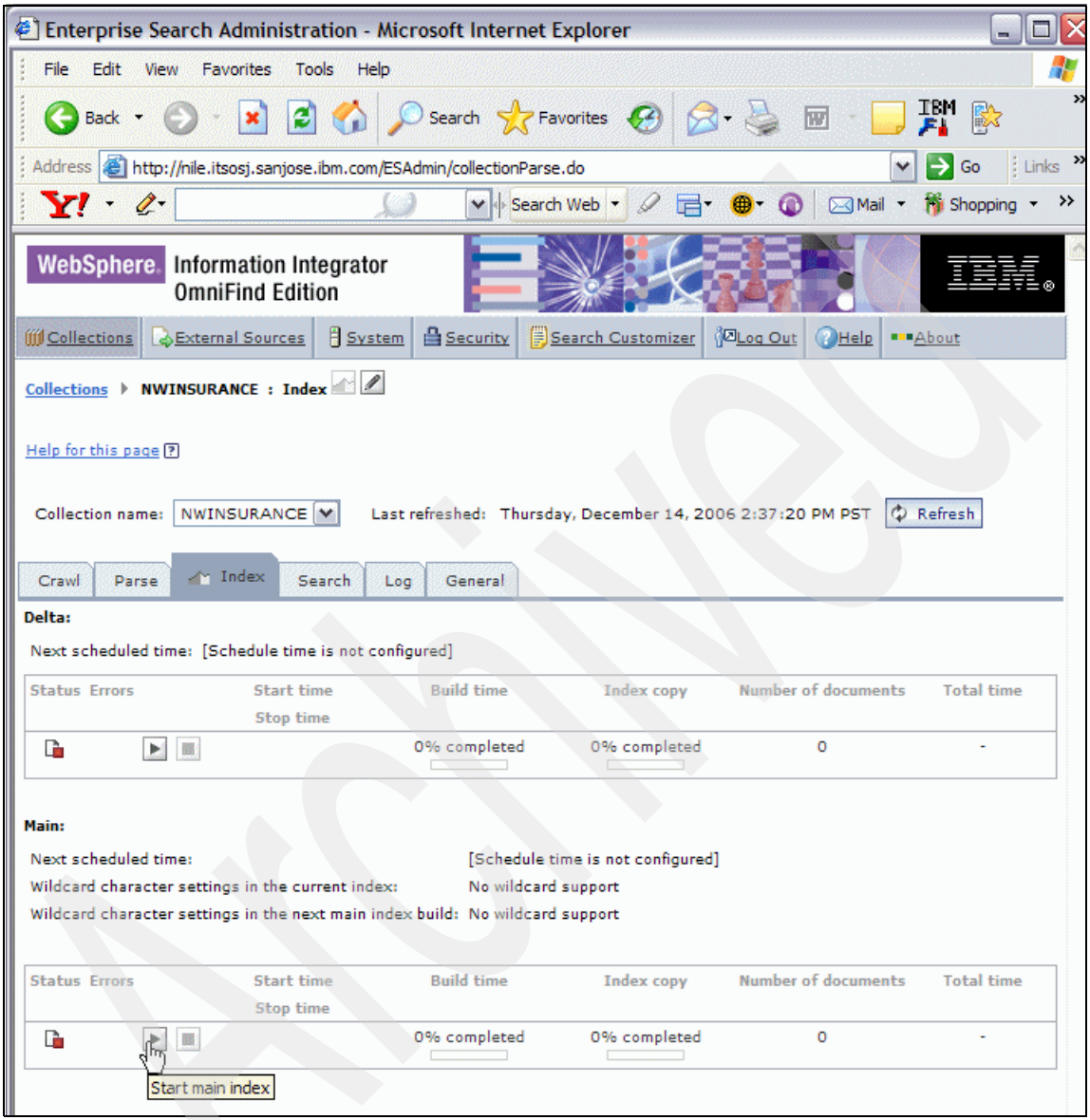

Figure 2-59 Start main index build

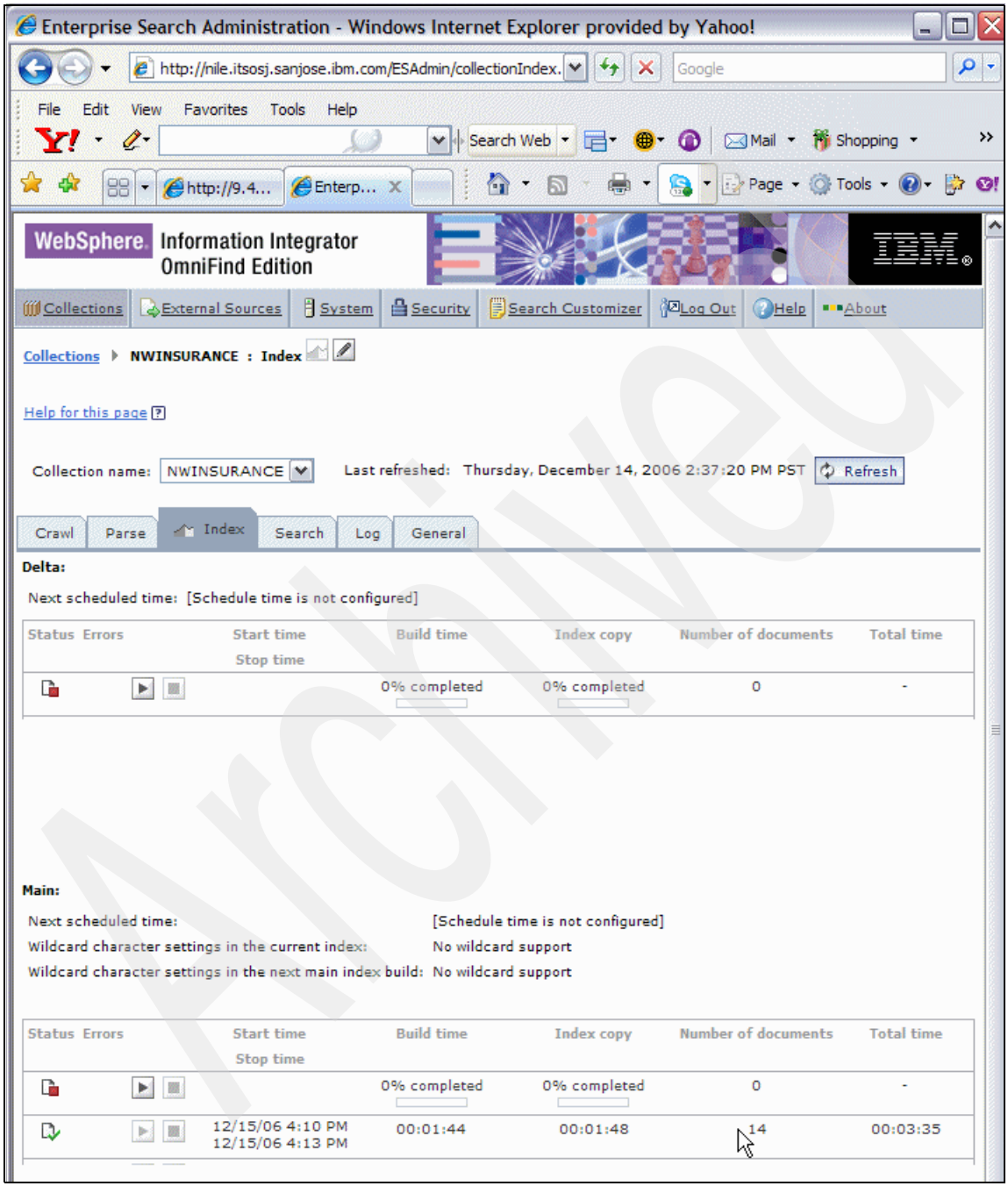

Figure 2-60 Main index build completion status

WSTEP4f: Define security settings

In this step, we configure identity management, single sign-on, and the search application to access the NWINSURANCE collection, as described in Figure 2-61 on page 120 through Figure 2-65 on page 122.

Click the **Security** link in the administration console to configure security settings for a collection, as shown in Figure 2-61 on page 120. In the subsequent window shown in Figure 2-62 on page 120, under the Search Applications tab, click **Configure identity management** to configure it, as shown in Figure 2-63 on page 121. Options that can be specified include how user credentials are refreshed, and whether all or specific crawlers should use SSO methods to authenticate users for the duration of a search session. Click **OK** to complete the configuration.

Figure 2-64 on page 122 through Figure 2-65 on page 122 show the definition of a search application that is only allowed to access the NWINSURANCE collection.

Note: Unless specific action is taken, the Default search application name automatically has access to the NWINSURANCE collection. This is desirable at least until one has tested the NWINSURANCE collection using the sample search Web application or portlet. After successful testing, you can disable access to the NWINSURANCE collection by the Default search application name.

Click the **Configure Search Applications** link under the Search Applications tab (see Figure 2-62 on page 120) to proceed to defining a search application. In Figure 2-64 on page 122, click **Add Search Application**. In Figure 2-65 on page 122, specify the Search Application name (NWINSURANCE) as having access to all the collections defined. Since there is only one collection defined, this search application name (NWINSURANCE) has access to it. In practice, you would explicitly identify the collections that a search application would have access to.

We can now proceed to query the NWINSURANCE collection, as described in 2.3.5, “WSTEP5: Query NWINSURANCE collection” on page 123.

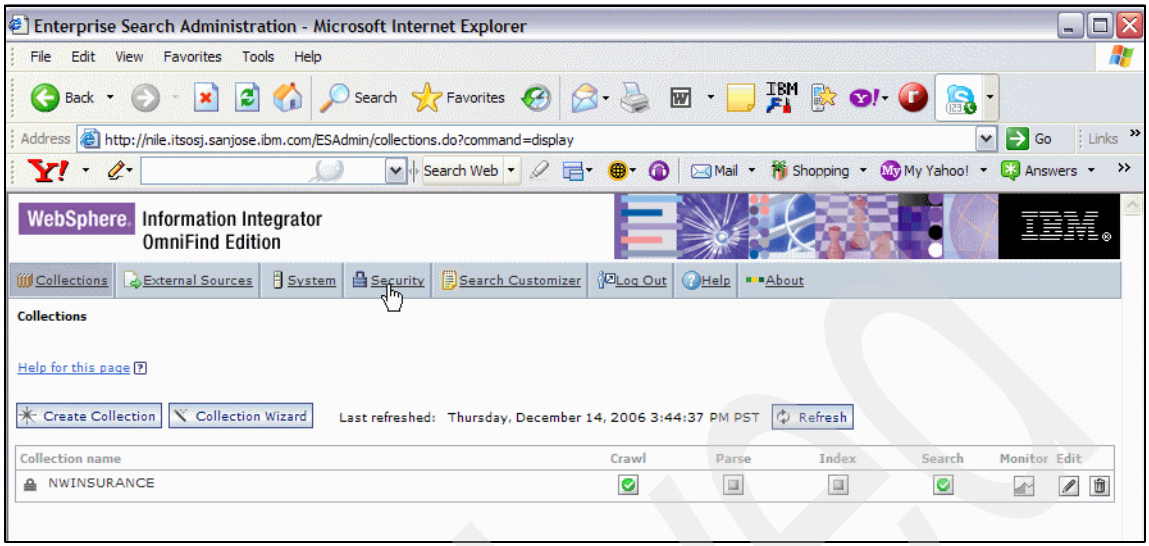

Figure 2-61 Security settings

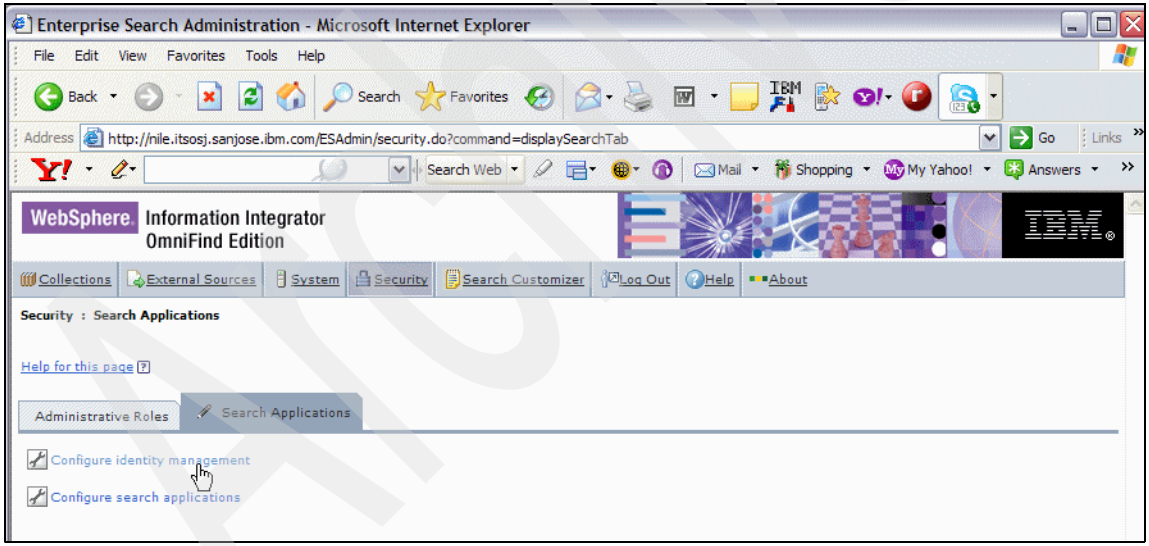

Figure 2-62 Configure identity management 1/2

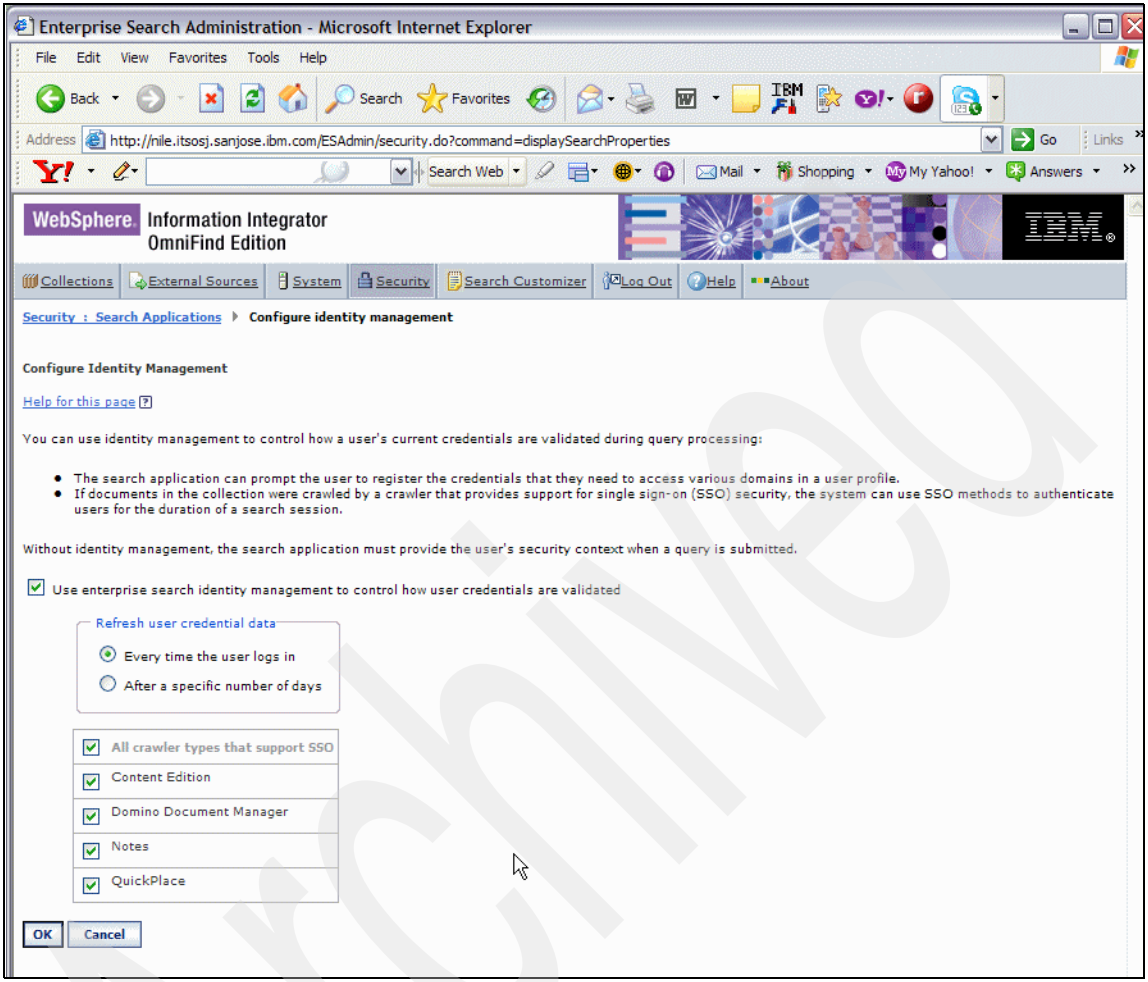

Figure 2-63 Configure identity management 2/2

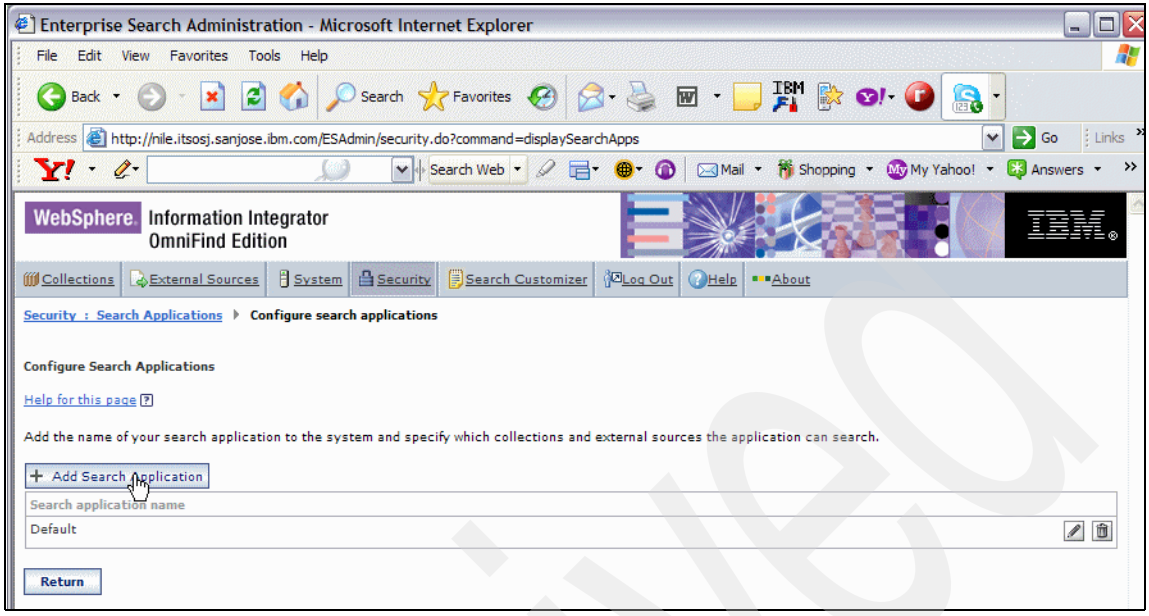

Figure 2-64 Add Search Application

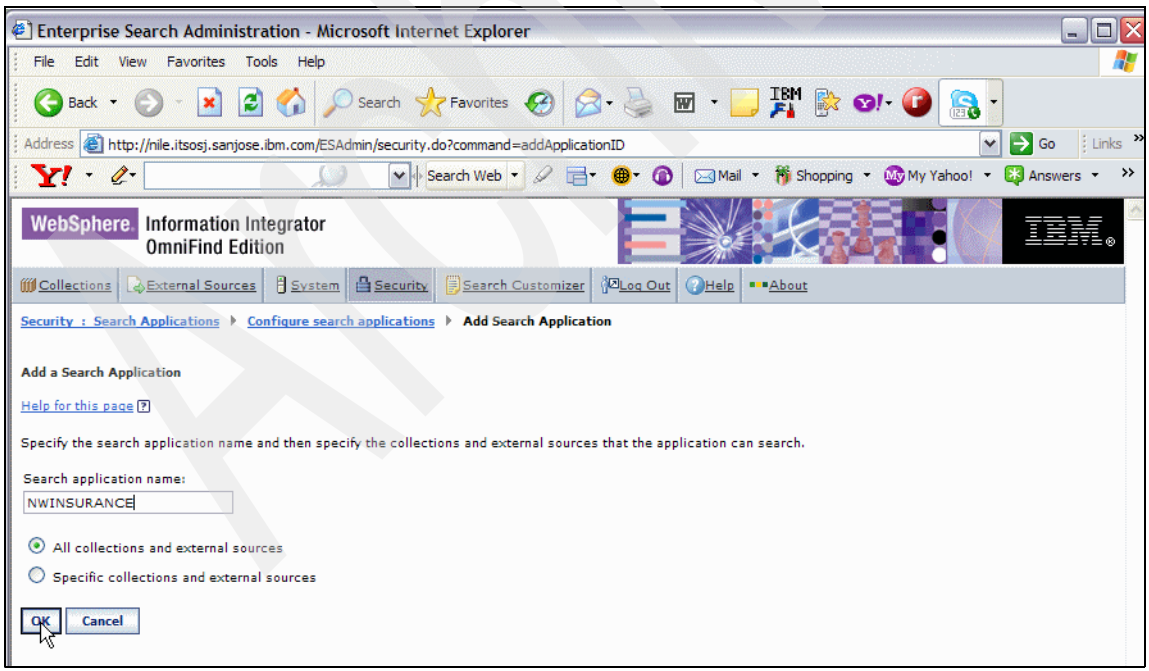

Figure 2-65 Specify search application name and accessible collections

2.3.5 WSTEP5: Query NWINSURANCE collection

In this step, we query the NWINSURANCE collection using the sample search Web application and the sample search portlet application with a modified config.properties file specifying the NWINSURANCE search application ID.

Using the Web sample application

The sample search Web application has the Default search application name and still has access to the NWINSURANCE collection, since the privilege was not taken away.

Figure 2-66 on page 124 through Figure 2-76 on page 133 describe the user interactions searching the NWINSURANCE collection.

After logging in to the sample search Web application with the esadmin user ID (Figure 2-66 on page 124), since IMC and SSO are enabled, the search runtime prompts the user for the Windows domain credentials, as shown in Figure 2-67 on page 125. Provide the credentials and click **Apply**, which takes you to the search window (Figure 2-68 on page 125). Since single sign-on support for Lotus QuickPlace is enabled, no prompts of it are presented, as the search runtime has the required credentials in the LTPA token.

Click the **Preferences** link in Figure 2-68 on page 125 to view the default preferences and modify it as required. Figure 2-69 on page 126 shows a number of options that can be modified for the search session, including inclusion of quick links, the data sources to be searched, file types to be excluded/included, number of results per page, and whether you should search for synonyms. By default, if multiple collections exist, the sample search application automatically searches across all the collections using a “remote” federator. In this case, since only one collection (NWINSURANCE) has been defined, only that collection appears in the list of choices in the Preferences window. Click **Apply** to save the choices made.

In Figure 2-70 on page 127, enter the string “insurance” in the search box and click **Search**. The results of this query is shown in Figure 2-71 on page 128. These search results can be requested in sorted order of date by selecting Date[2] from the Sort by drop-down list, as shown in Figure 2-72 on page 129, which shows the newly sorted results by date in Figure 2-73 on page 130. You can also alter the sort order from descending to ascending from the Sort order drop-down list, as shown in Figure 2-74 on page 131.

[2] The other “date” item in the drop-down list corresponds to a metadata field in one of the data sources being crawled.

You can also filter the search results by the QuickPlace source type by clicking the **QuickPlace** link in the Source type filter field, as shown in Figure 2-74 on page 131, with the resulting action shown in Figure 2-75 on page 132. You can also filter the (original) search results by the Windows file system data source by clicking the **Windows file system** link in the Source type filter field, as shown in Figure 2-75 on page 132, with the resulting action shown in Figure 2-76 on page 133.

Since the NWINSURANCE collection appears to have been created correctly, we can choose to withdraw access to the NWINSURANCE collection from the Default search application name, as shown in Figure 2-77 on page 134 and Figure 2-78 on page 135. Click the **Edit** icon for the Default search application name in Figure 2-77 on page 134, and then deselect the NWINSURANCE collection from the specific collections available for access, as shown in Figure 2-78 on page 135. Click **OK** to complete the security changes.

The sample search portlet was modified to only access the NWINSURANCE collection, as described in “Using the sample search portlet” on page 135.

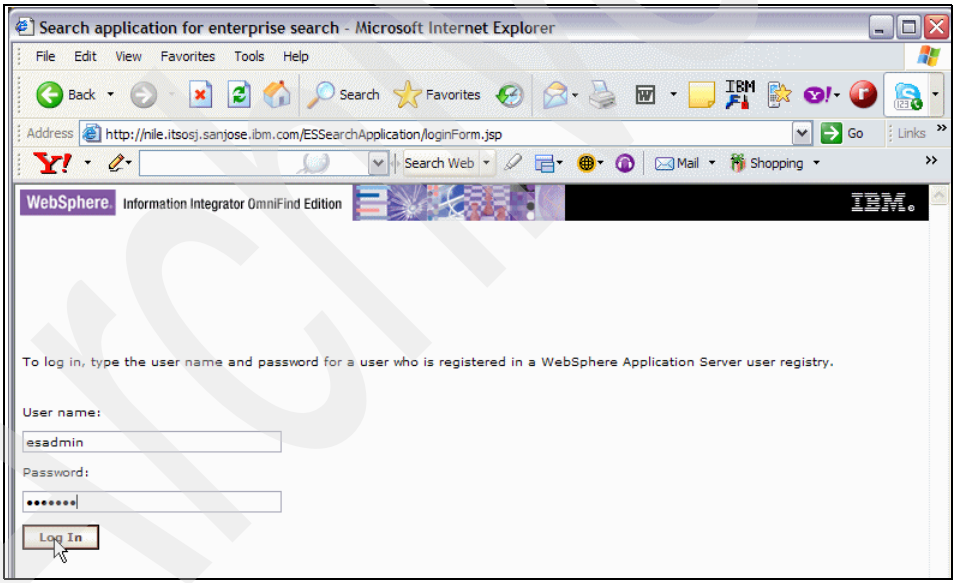

Figure 2-66 Log in to Web sample search application

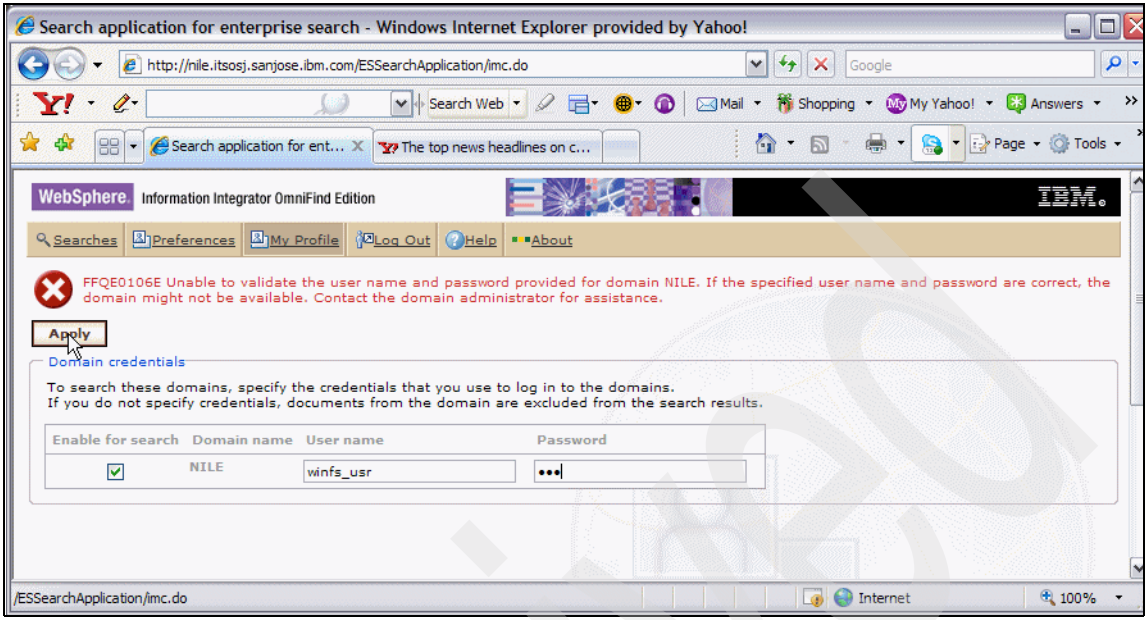

Figure 2-67 Identity Management Component prompt for credentials for Windows

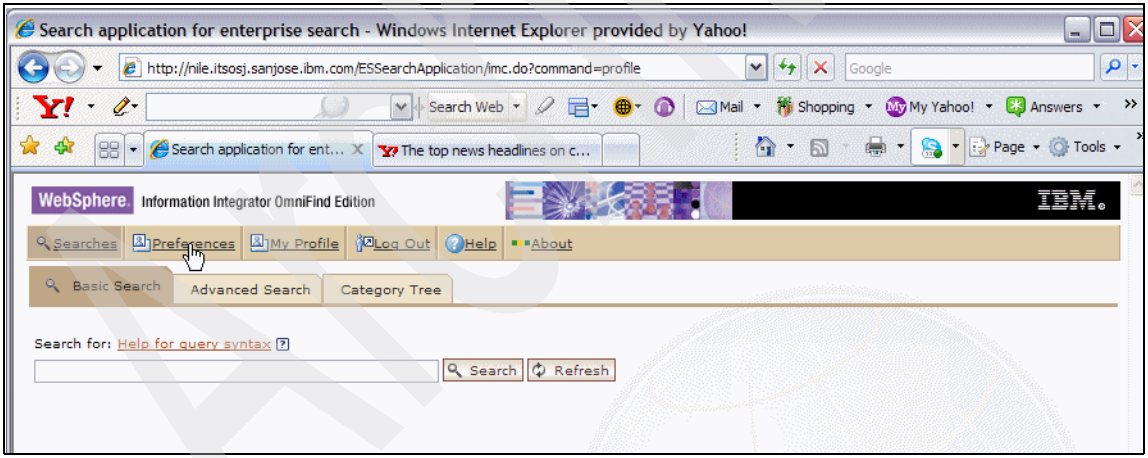

Figure 2-68 Click Preferences

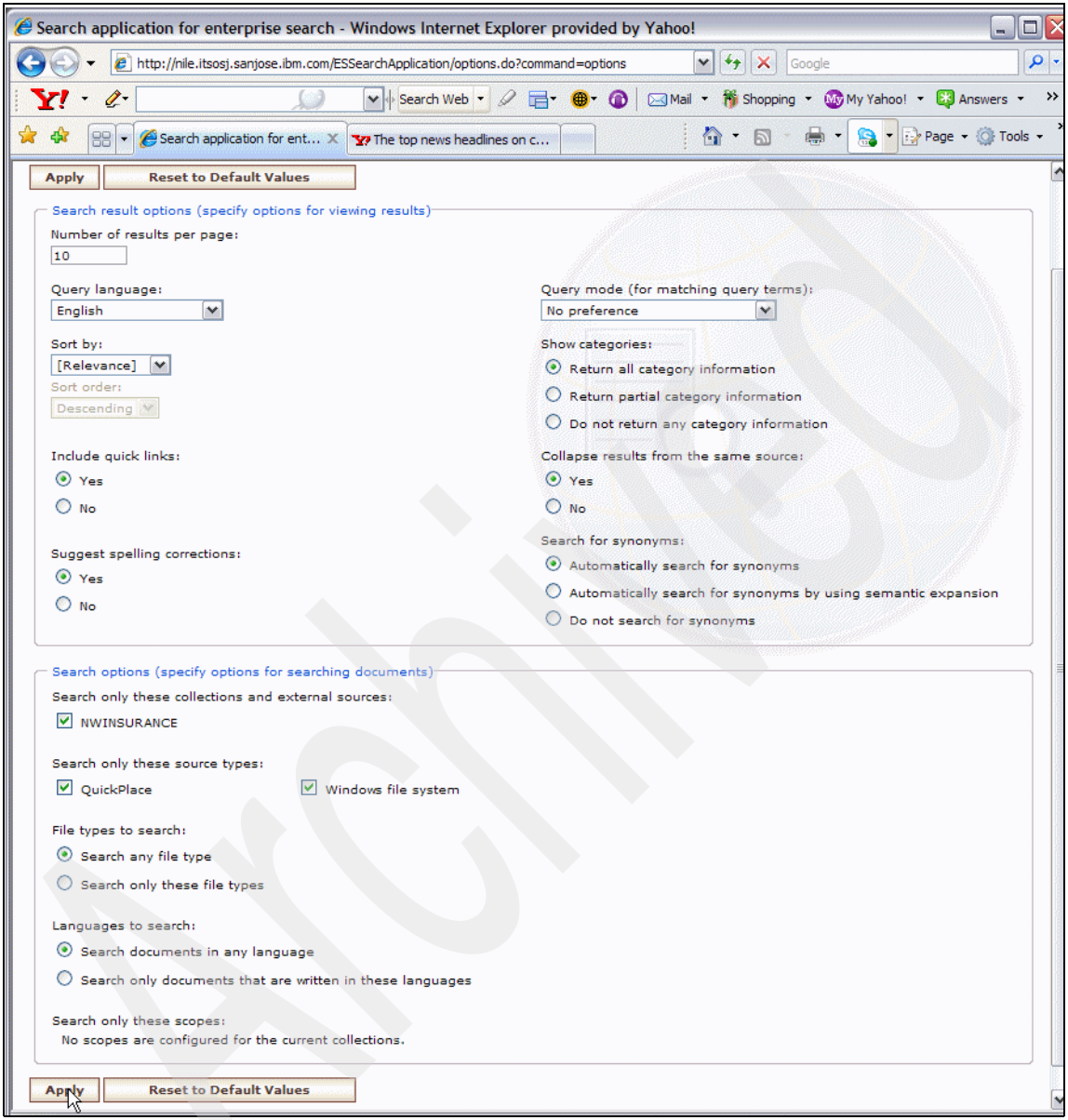

Figure 2-69 Preferences details

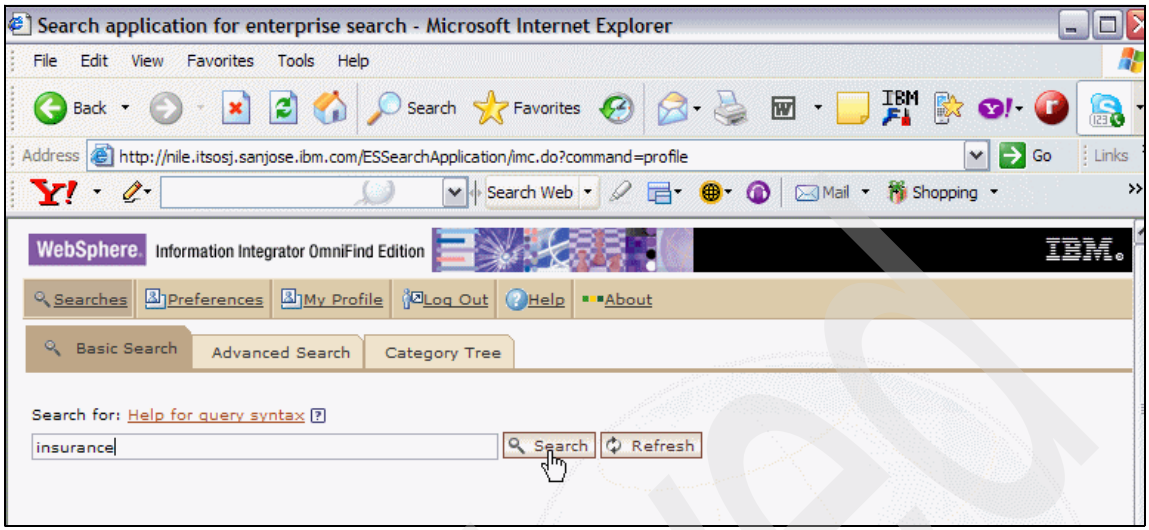

Figure 2-70 Search for “insurance”

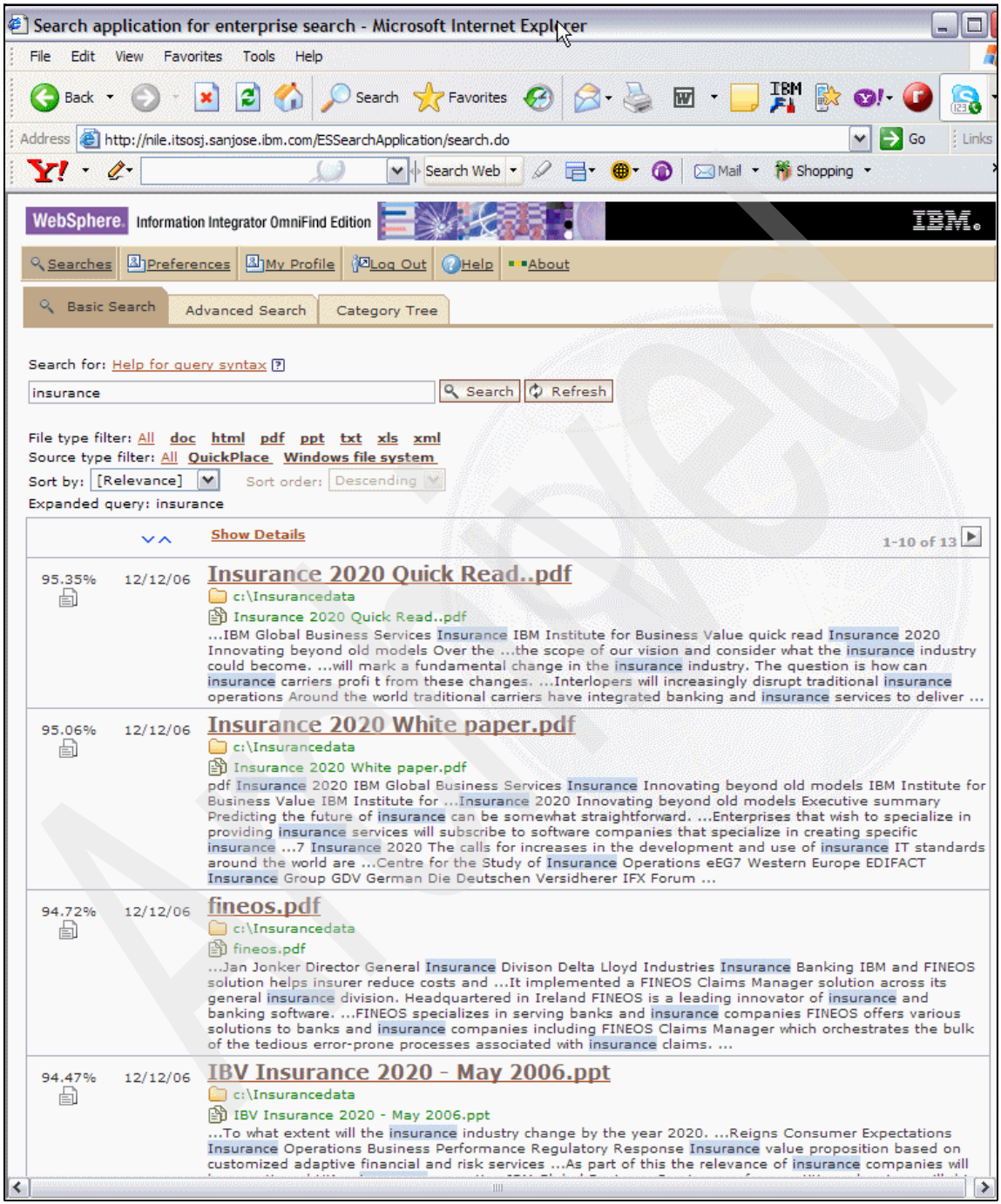

Figure 2-71 Search results for “insurance”

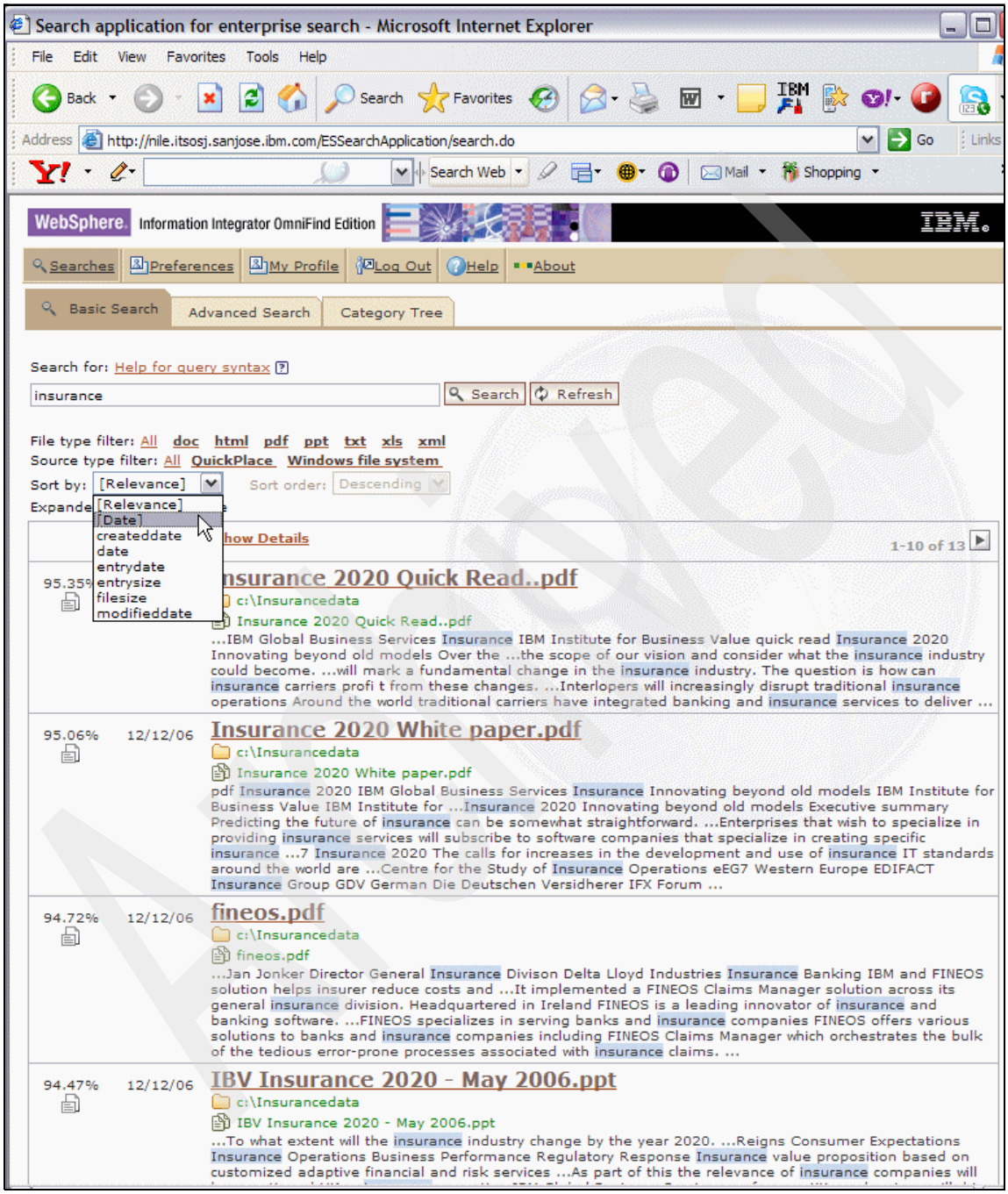

Figure 2-72 Sort search results for “insurance” by date descending

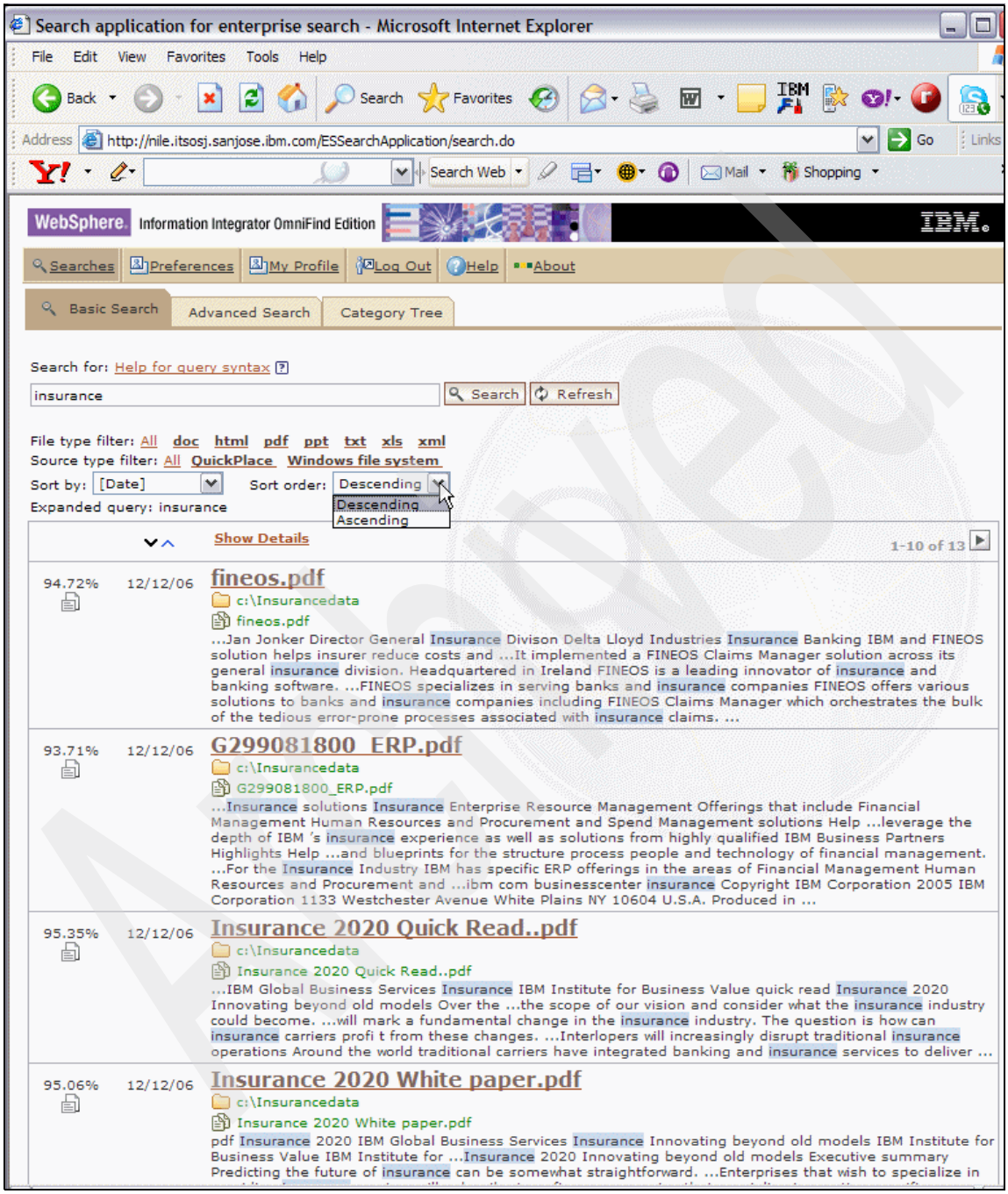

Figure 2-73 Search results for “insurance” sorted by date descending

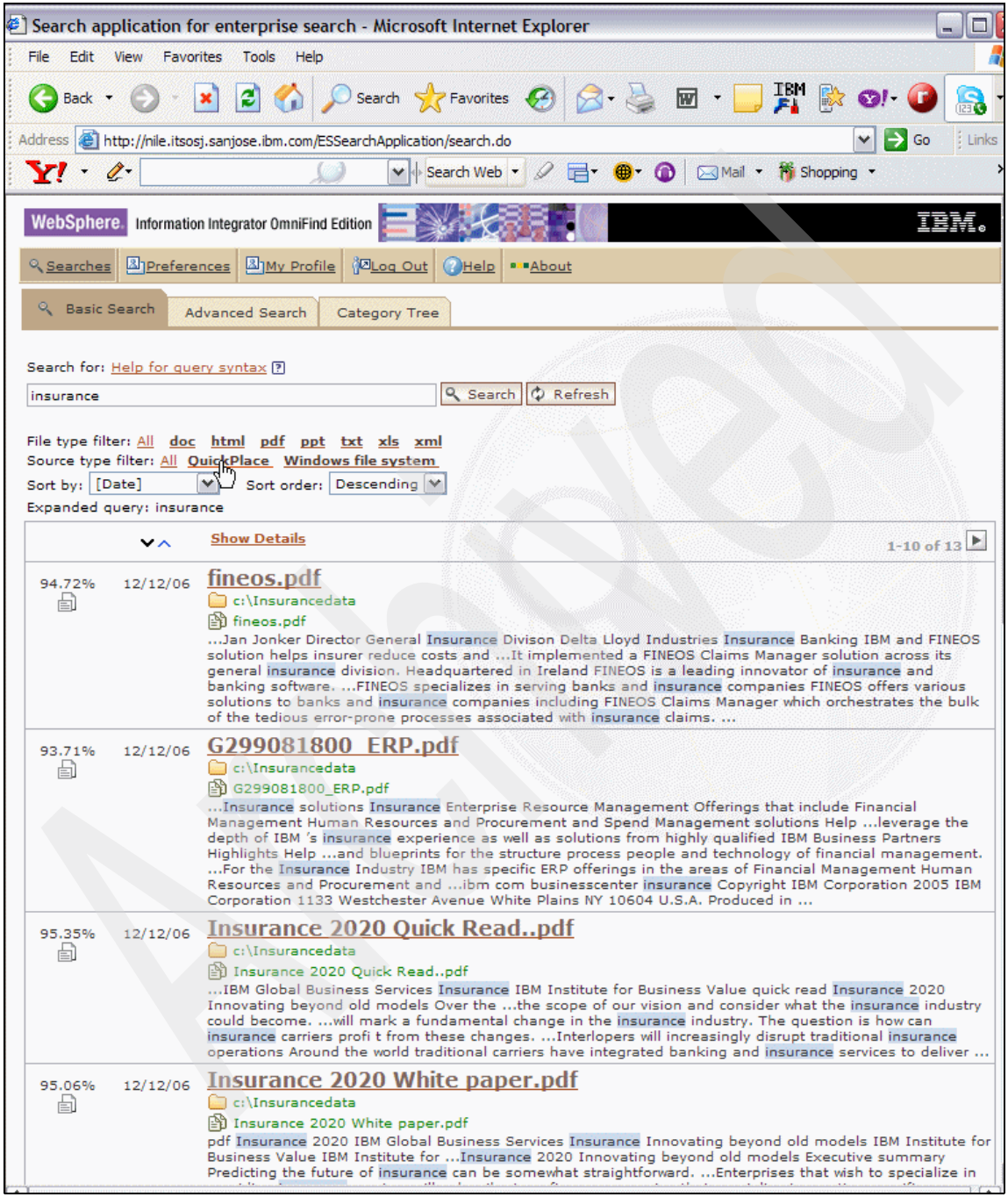

Figure 2-74 Filter search results for "insurance" by QuickPlace source only 1/2

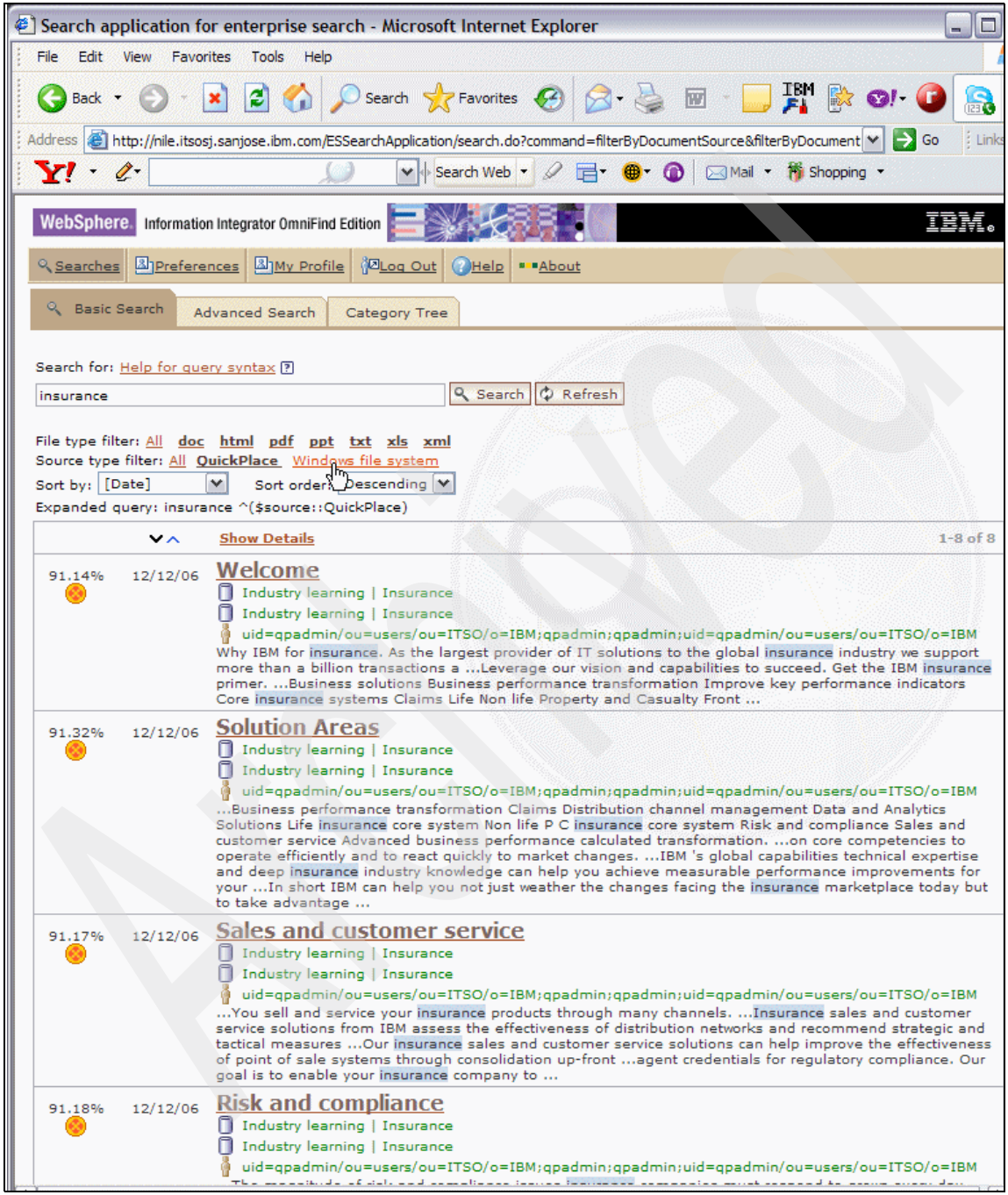

Figure 2-75 Filter search results for “insurance” by QuickPlace source only 2/2

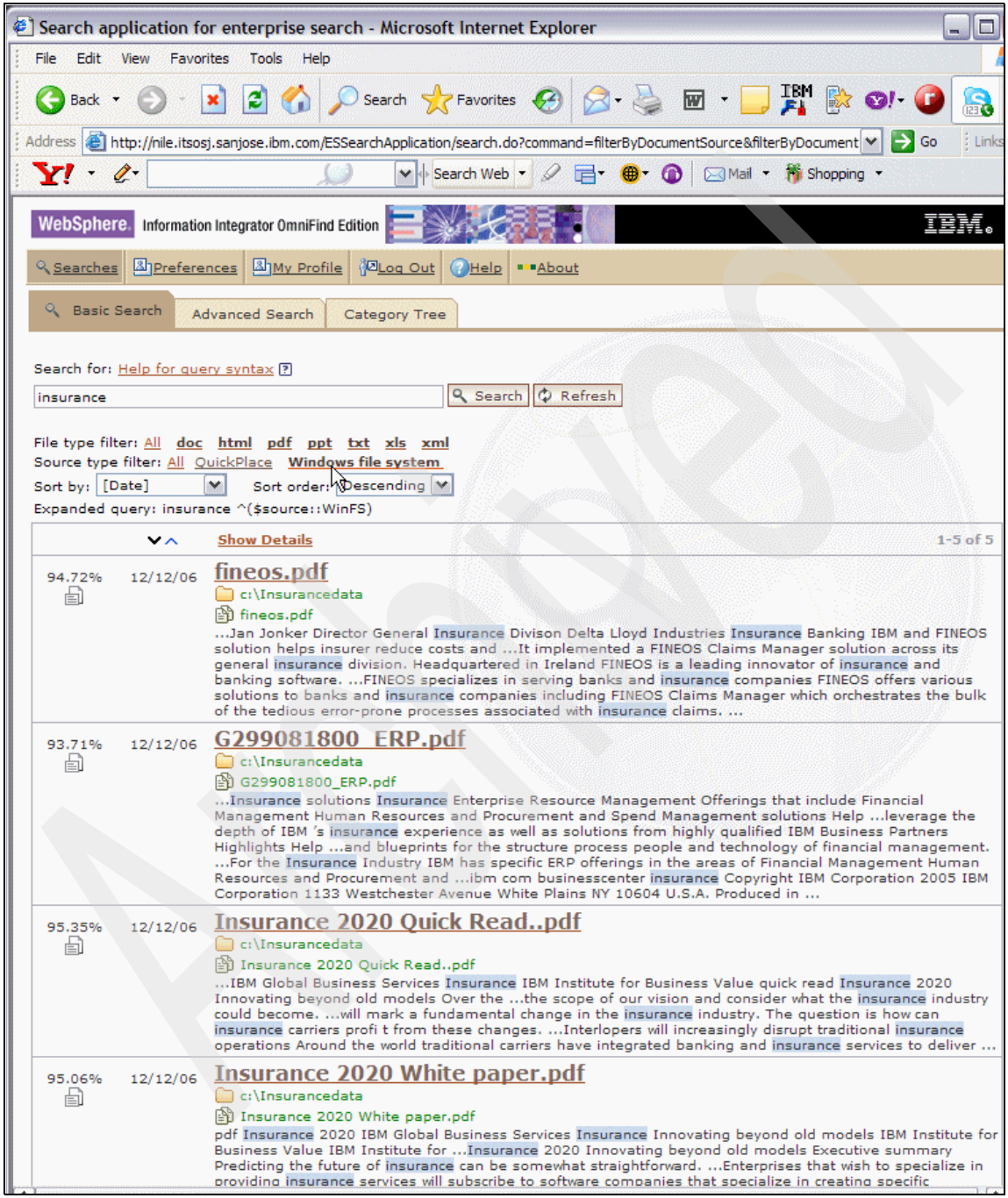

Figure 2-76 Filter search results for “insurance” by Windows file system source

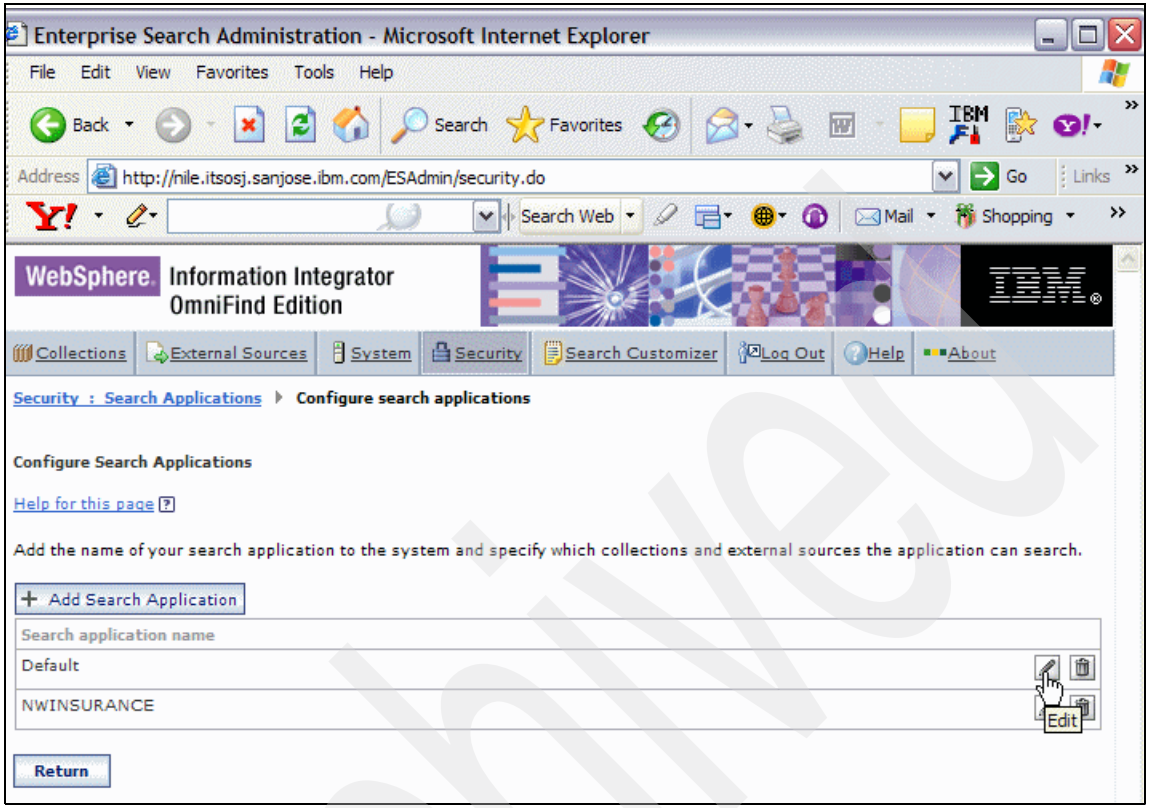

Figure 2-77 Edit Default search application name

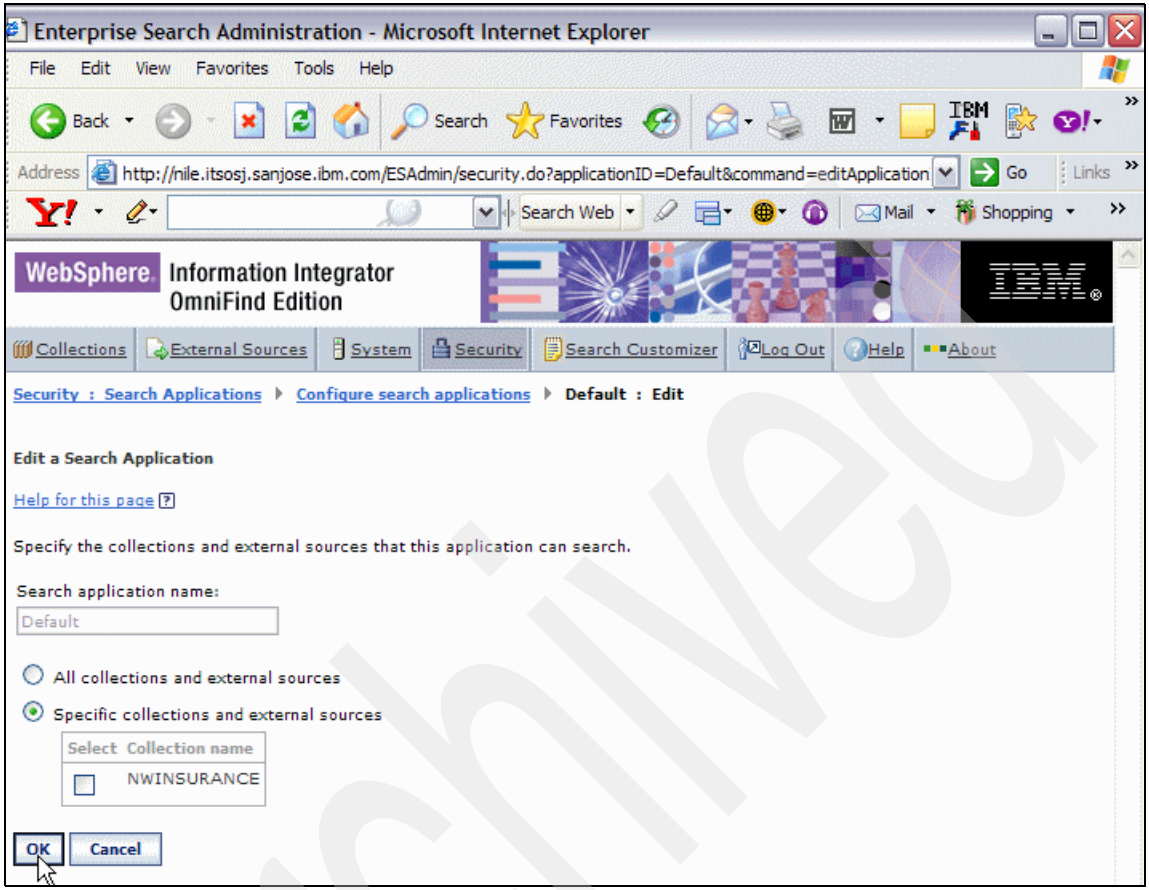

Figure 2-78 Disallow Default search application name access to NWINSURANCE

Using the sample search portlet

In this step, we modified the sample search portlet to access the NWINSURANCE collection. Appendix A, “Install Sample Search application portlet” on page 431 describes the installation of the sample search portlet in WebSphere Portal Server.

Since we had only permitted the NWINSURANCE search application name to access the NWINSURANCE collection, the config.properties file has to be modified to change the applicationName property to NWINSURANCE, as shown in Example 2-2 on page 136. Also shown here are the credentials for authenticating to the search runtime in the user name (wasadmin) and password (wasadmin).

Note: In OmniFind V8.4, the user name / password no longer needs to be updated in the config.properties file. If you are using the search portlet and you are sharing the same LtpaToken between WebSphere Portal and OmniFind, then you do not need a user name / password combination, since the LtpaToken can be used by the underlying API to communicate securely to the search server.

Figure 2-79 on page 137 through Figure 2-81 on page 139 describe some user interactions with the sample search portlet. After the user logs in to the WebSphere Portal, you can access the sample search portlet by clicking the **OmniFind - Windows** tab, as shown in Figure 2-79 on page 137.

Figure 2-80 on page 138 shows the search query “insurance” and its corresponding search results, while Figure 2-81 on page 139 shows the Preferences details for this sample search portlet, which is identical to the one shown with the sample search Web application shown in Figure 2-69 on page 126.

Note: At this point, we have the functional requirements met for the enterprise search solution for Northwest Insurance Inc. Further tests, measurements, and tuning need to be performed to ensure that the solution fully addresses the capacity and workload requirements as well.

Example 2-2 config.properties file contents

```
#Wed Dec 06 14:41:29 PST 2006
logoff.backgroundImage=/images/IIOF_logout.gif
username=wasadmin
.........
password=wasadmin
.............
applicationName=NWINSURANCE
.............
```

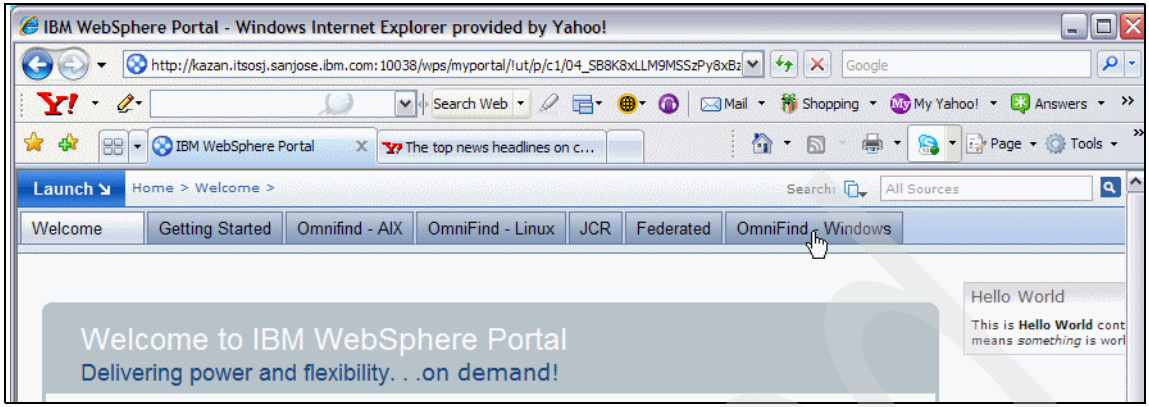

Figure 2-79 WebSphere Portal

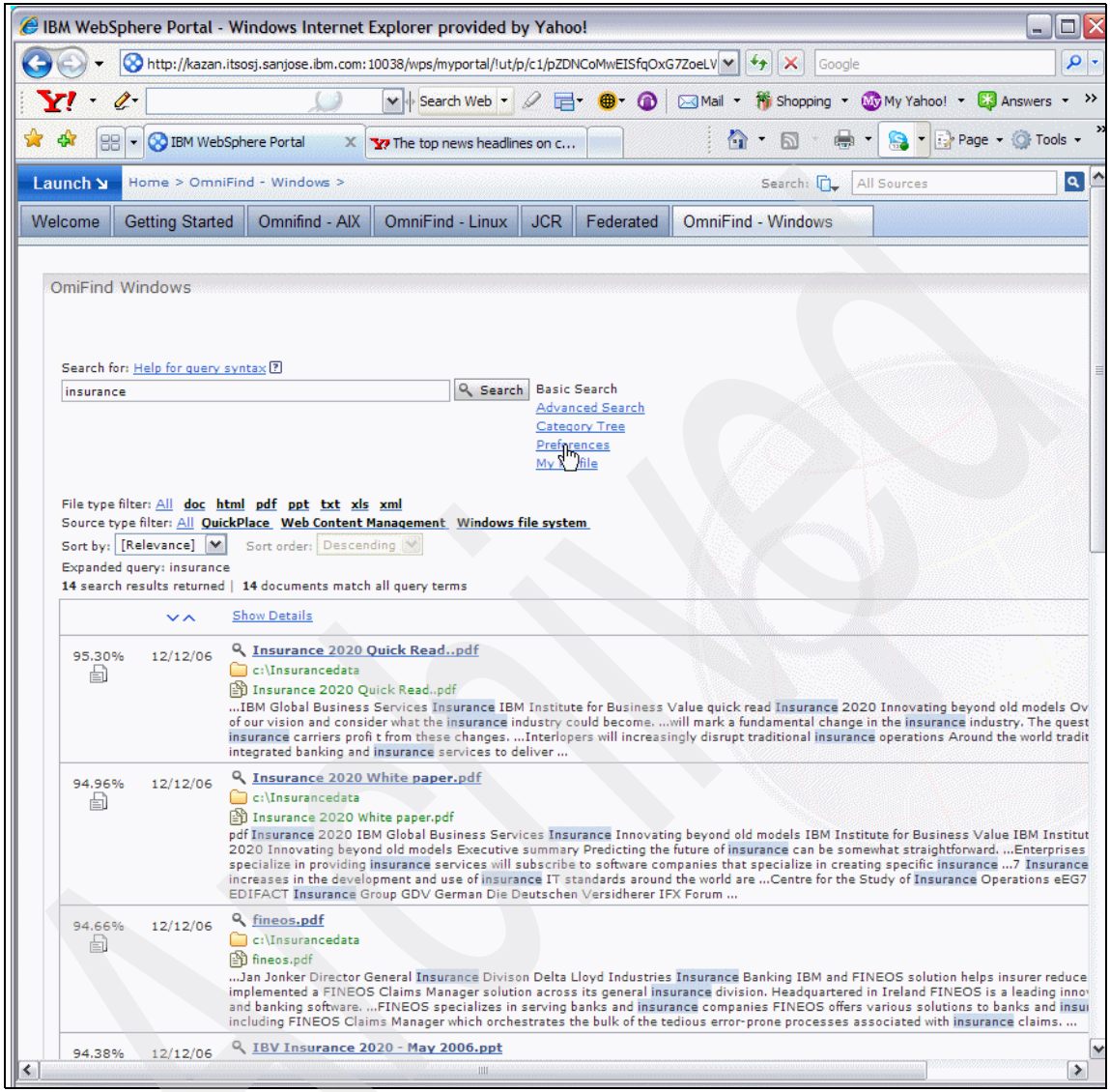

Figure 2-80 Search for “insurance” and corresponding search results

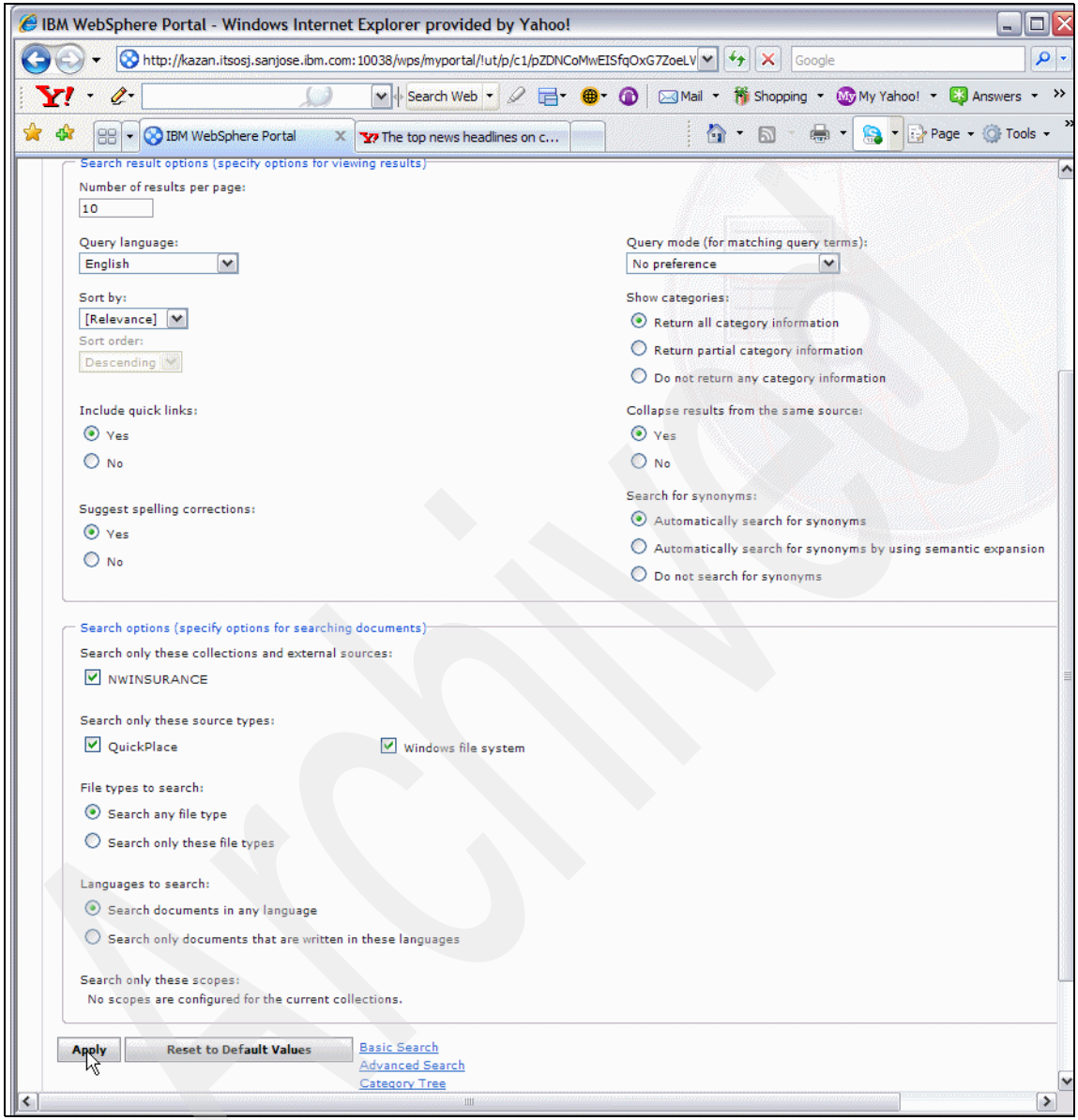

Figure 2-81 Preferences details

3

Medium-size organization OmniFind scenario on Red Hat Enterprise Linux platform

In this chapter, we describe a step-by-step approach to implementing IBM OmniFind Enterprise Edition on a two server Red Hat Enterprise Linux platform for a hypothetical insurance company.

The topics are:

- Business requirement
- Environment configuration
- Configure the environment

3.1 Business requirement

Our fictitious company Sequoia General is an auto insurance company supporting customers on the west coast of the United States, including Washington, Oregon, and California. It has under 2000 employees supporting over 200,000 customers. In a competitive marketplace, Sequoia General needed to provide superior customer service by improving the productivity of its sales, marketing, and technical personnel by making customer information spread across multiple data sources available on demand in response to customer and prospect inquiries. A high availability architecture is a critical requirement of any proposed solution.

One of the solutions aimed at achieving this objective was to implement an enterprise search system for the 1 million customer-related documents located in its Web Content Management (WCM), WebSphere Portal Server (WPS), and Portal Document Manager (PDM) systems. Given the sensitive nature of customer information stored in WPS and PDM, Sequoia General requires the enterprise search solution to support and leverage the native security capabilities of these underlying data sources; only authorized employees need to have access to secure documents. However, information stored in the WCM needs to be available to all the employees in the organization. The solution also needs to provide a simple GUI interface available as a portlet in WebSphere Portal Server.

From an IBM OmniFind Enterprise Edition implementation perspective, this translates to having:

- Two server Intel-based Linux platform implementations to address the high availability requirement, and the strategic direction of the organization to support Linux.
- Enable WebSphere global security with an LDAP repository.
- Two collections, one with no document-level security for information stored in the WCM system (GENINSINFO collection), and the other (CUSTINFO collection) with document-level security and single sign-on enabled for sensitive information stored in the WPS and PDM systems.
  - The WCM system contains general information about product offerings, and corporate news that is accessible to all employees of the organization.
  - The PDM and WPS systems contain sensitive customer related information, including policy and claim details, that is only accessible by authorized employees.

The rational for creating two collections is to split the sensitive information that needs to be restricted to authorized employees from information that is available to all the employees in the organization. Document-level security and validating credentials during query processing has a performance impact that can be avoided by not selecting this option (see Figure 2-44 on page 102).

- Enable Identity Management Component (IMC) and single sign-on for the WebSphere Portal crawler, but only IMC (with no SSO) for the PDM crawler. We wanted to demonstrate a configuration within a collection where SSO and no SSO are configured for different crawlers.
- A Web crawler for WCM system, WebSphere Portal crawler, and PDM crawler.
- The sample search portlet installed on WebSphere Portal Server.
- Use a UIMA annotator that permits search by telephone number and e-mail address.

3.2 Environment configuration

Sequoia General's workload demands (medium-size employee community), number of documents to be indexed, and the need for a high availability environment permits the adoption of a sufficiently configured two server IBM OmniFind Enterprise Edition environment.

The Red Hat Enterprise Linux platform is considered sufficient to address Sequoia General's enterprise search solution needs.

Note: In the real world, the high availability environment for IBM OmniFind Enterprise Edition requires a load balancing solution, such as Network Dispatcher, to distribute the search load to the search servers evenly. OmniFind does not require Network Deployment to provide load balancing. In our contrived environment, however, we did not implement this functionality due to time constraints.

Figure 3-1 on page 144 shows the configuration used in the Sequoia General' two server Red Hat Enterprise Linux configuration, including:

- A Windows 2003 server (kazan.itsosj.sanjose.ibm.com) provides Sequoia General's enterprise portal through which authorized users will access the enterprise search solution. This server uses LDAP (Tivoli Directory Server) to address the security requirements of the enterprise search solution.

- Tivoli Directory Server (boron.itsosj.sanjose.ibm.com) is installed on a separate server that is physically well secured, given the sensitive nature of the information it contains.
- Two server Red Hat Enterprise Linux servers (falcon.itsosj.sanjose.ibm.com and buzzard.itsosj.sanjose.ibm.com) for the IBM OmniFind Enterprise Edition search, indexer, parser, and crawler components.
- The WCM, PDM and WPS data sources are located on the server kazan.itsosj.sanjose.ibm.com.
- The various crawler types defined in this configuration.

Note: The Windows 2003 server has direct connectivity and authorization to the data sources in order to render the document that a user selects in the search result.

- IBM OmniFind Enterprise Edition administrators administer and manage the environment through the administration console GUI.

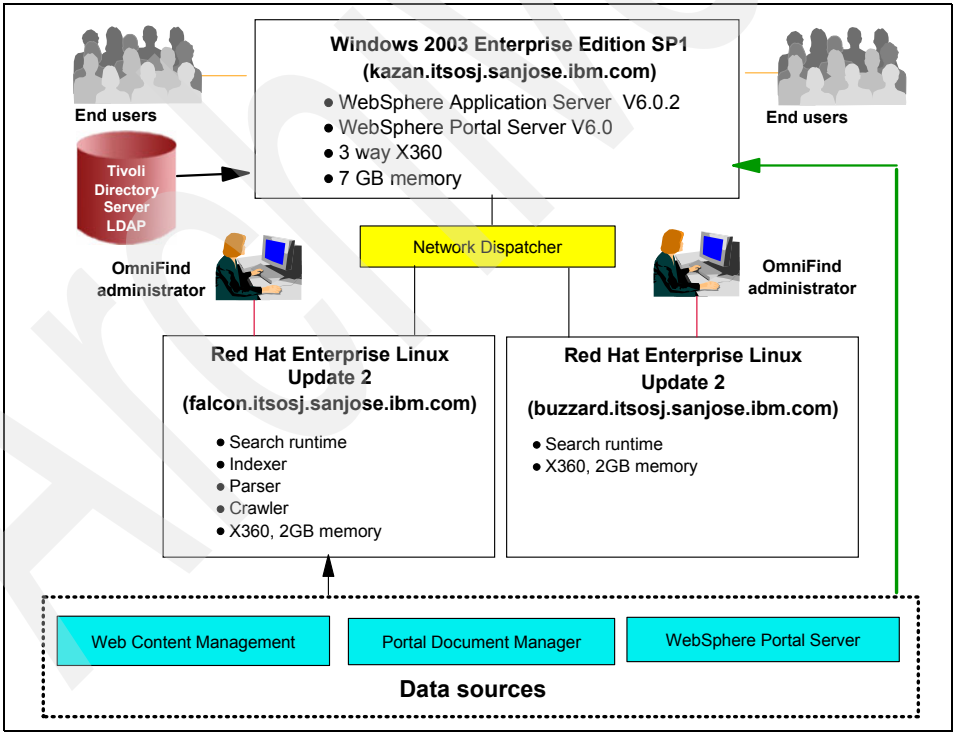

Figure 3-1 Sequoia General's two server Red Hat Enterprise Linux search solution

3.3 Configure the environment

In this section, we document the step-by-step configuration of the two server Red Hat Enterprise Linux configuration in our fictitious company Sequoia General. Figure 3-2 lists the main steps involved in configuring this environment. First, the administrator and users that need to access the IBM OmniFind Enterprise Edition environment must be defined in the Tivoli Directory Server LDAP repository. Next, global security must be enabled on each of the two WebSphere Application Servers of the IBM OmniFind Enterprise Edition search servers. Once global security is enabled, the es.cfg configuration file must be updated with the WebSphere Application Server user ID and password. Sequoia General's two collections can now be created (one with collection security disabled, and the other with collection security enabled to enforce document-level security), populated, and queried using the sample search Web application, and sample search portlet.

Each of these steps is described in detail in the following subsections.

Note: We assume that the IBM OmniFind Enterprise Edition has been verified to have been correctly installed on the two Red Hat Enterprise Linux servers with the proper prerequisites. The IBM OmniFind Enterprise Edition binaries were installed in /opt/es, while IBM OmniFind Enterprise Edition data was installed in /opt/var/es.

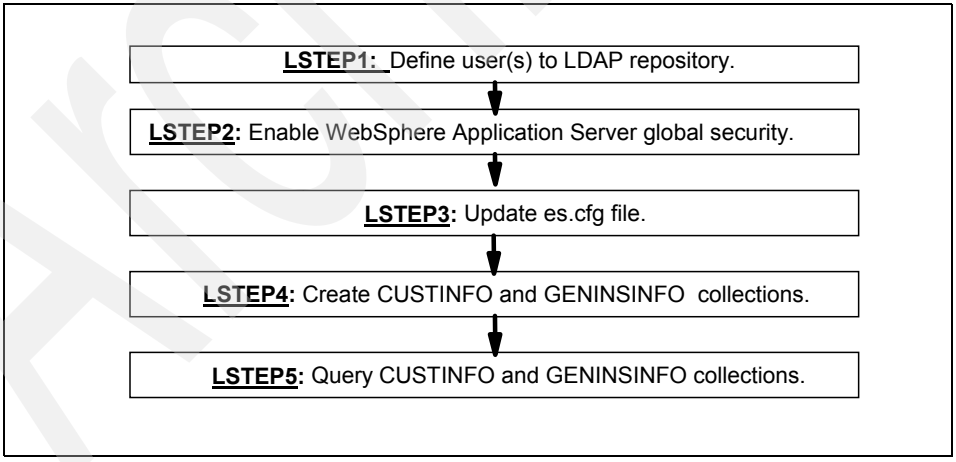

Figure 3-2 Steps to configure Sequoia General's two server configuration

Attention: In all the following sections, for the purposes of avoiding screen capture overload, we have *not* included all the windows that you would typically navigate through in order to perform the desired function. Instead, we have focused on including select windows (and in some cases portions of selected windows) that highlight the key items of interest, thereby skipping both initial as well as intervening windows in the process.

3.3.1 LSTEP1: Define user(s) to LDAP repository

In this step, we add all the users authorized to access IBM OmniFind Enterprise Edition to the Tivoli Directory Server LDAP repository using the Tivoli Directory Server Web Administration Tool.

Since the process is identical to that described in 2.3.1, “WSTEP1: Define users in LDAP repository” on page 57, it is not repeated here.

3.3.2 LSTEP2: Configure WebSphere Application Server global security

In this step, we enable global security on both the WebSphere Application Servers with the Search Runtime, and specify that they use the Tivoli Directory Server as the LDAP repository as its user registry. LTPA keys are generated and then exported to other servers participating in the single sign-on domain.

Since this process is identical to that described in 2.3.2, “WSTEP2: Enable WebSphere Application Server global security” on page 66, it is not repeated here.

Note: The same user ID / password should be used on both WebSphere Application Servers.

3.3.3 LSTEP3: Configure es.cfg properties file

Once WebSphere global security is enabled in the IBM OmniFind Enterprise Edition Search Runtime server, you must update the es.cfg file with the WebSphere Application Server user ID and password.

Note: When WebSphere global security is enabled, the Common Communications Layer (CCL) component of IBM OmniFind Enterprise Edition needs to authenticate with WebSphere Application Server in order to start the search runtime. It obtains the required user ID / password from the es.cfg file; it has the key to decrypt the WASPassword.

Since this process is identical to that described in 2.3.3, “WSTEP3: Update es.cfg file” on page 77, it is not repeated here.

3.3.4 LSTEP4: Create CUSTINFO and GENINSINFO collections

In this step, we create the CUSTINFO and GENINSINFO collections with the appropriate crawlers, then parse and index the crawled data. The individual steps involved are shown in Figure 3-3 on page 148 and described in more detail in the following subsections.

Note: Many of the steps shown in Figure 3-3 need to be performed in the sequence shown, such as preparing for crawling the data sources prior to beginning the crawl, and configuring the UIMA annotator before commencing parsing. But other steps may be performed in a different sequence, such as completing the creation, crawling, parsing, and index building of the CUSTINFO collection before doing the same with the GENINSINFO collection. We performed the steps in the sequence shown; you may choose another sequence.

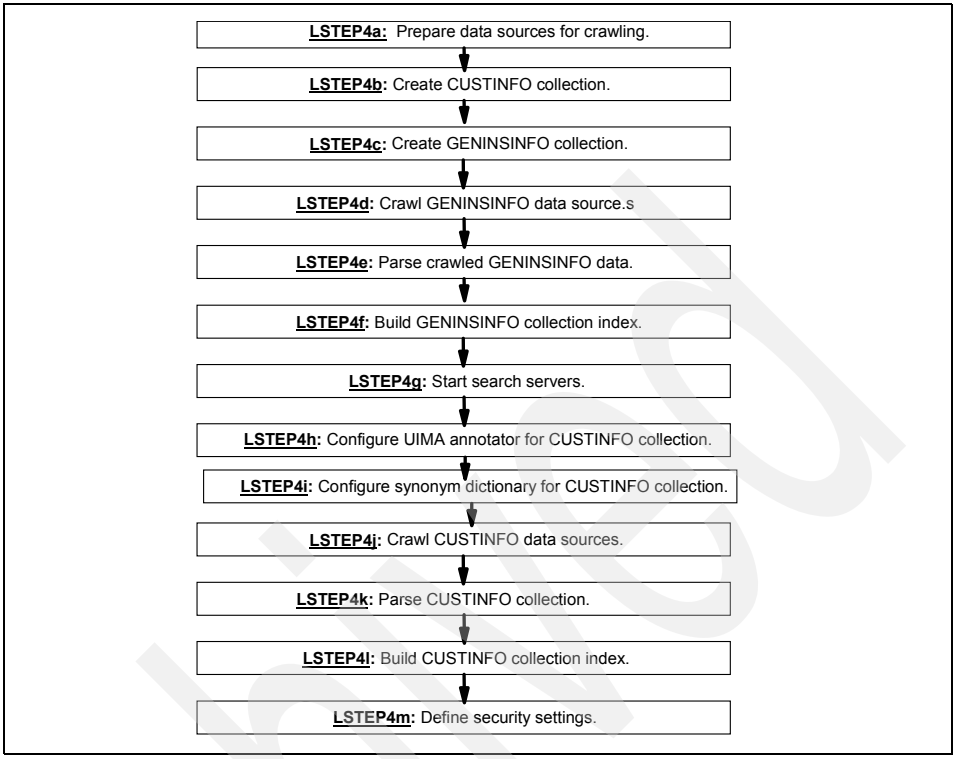

Figure 3-3 Creating and configuring the CUSTINFO and GENINSINFO collections

Note: A number of parameters can be specified during the creation and configuration of a collection and the associated crawlers. A description of these parameters is beyond the scope of this book. You are encouraged to read the product documentation or invoke **Help** on the GUI for detailed information about these parameters.

LSTEP4a: Prepare for crawling the data sources

In this step, we need to prepare the environment for crawling the WCM, PDM, and WPS data sources. Specifically, the following tasks need to be performed:

- Portal Document Manager (PDM) can only be crawled through the WebSphere Information Integrator Content Edition (IICE) PDM connector. Therefore, it needs to be installed and configured on the crawler server falcon.itsosj.sanjose.ibm.com.

Note: You can install IICE together with IBM OmniFind Enterprise Edition. We did not do this, and therefore had to manually install it later using response files.

- Web Content Management (WCM) may be crawled through the Web crawler or the Web Content Manager crawler.[1] Since our WCM content is sourced without security for the GENINSINFO collection, we will be using the Web crawler to crawl WCM. We therefore need to ensure that the WCM content crawled is accessible to all users.

Note: For WebSphere Portal Version 6, you should be crawling the WCM content using the WCM crawler. If you do not need security, you can still use the WCM crawler and simply disable security at the collection or crawler levels. We chose to use the Web crawler here in order to demonstrate a variety of crawler configurations in this book.

- WebSphere Portal Server (WPS) is crawled using the WebSphere Portal crawler. It needs a starting URL to be specified for crawling that needs to be identified.

Configure the WebSphere IICE PDM connector

The following steps need to be performed to install and configure the WebSphere IICE PDM connector:

1. Install WebSphere IICE PDM connector on the crawler server using a response file in silent mode through the following command, as shown in Figure 3-4 on page 150:

```
./setupLinux.bin -options ES_NODE_ROOT/logs/install/wiice.rsp
-silent -is:javahome ES_INSTALL_ROOT/_jvm/jre
where
ES_NODE_ROOT - /var/es/
ES_INSTALL_ROOT - /opt/es/
```

[1] New in OmniFind Enterprise Edition V8.4, which supports the security capabilities of WCM

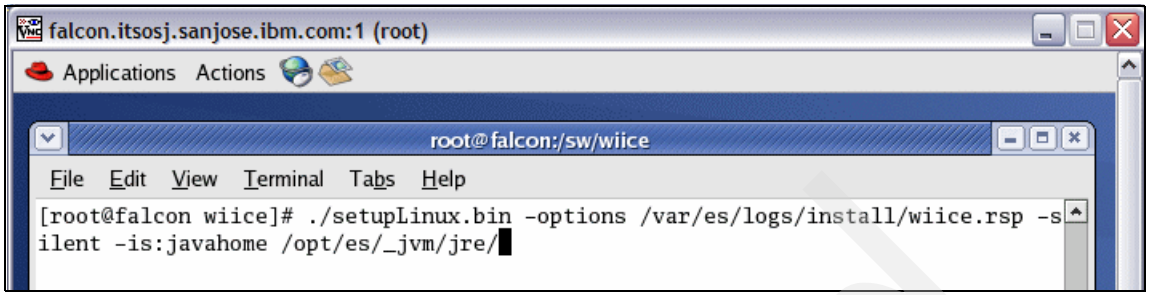

Figure 3-4 Install a WebSphere IICE PDM connector using a response file in the silent mode

2. Replace file vbr_pdm.jar in /opt/es/content_edition/ejb/ with the update one supplied by IBM OmniFind development.

> **Note:** This was a fix for a bug we encountered. A technote has been published to address this problem. It is available at:
>
> http://www-1.ibm.com/support/docview.wss?rs=0&q1=1252053&uid=swg21252053&loc=en_US&cs=utf-8&cc=us&lang=en

3. Modify the config.sh file to contain the correct WAS_HOME path (/opt/was), as shown in Example 3-1.

Example 3-1 config.sh file contents

```
#!/bin/sh
error() {
   echo "You must set the environment variable VBR_HOME"
   echo "to point to the DB2 Information Integrator Content Edition installation directory."
   echo "Please ensure that this variable is set before attempting to launch DB2 Information Integrator
Content Edition components."
   exit 1;
}

VBR_HOME=/opt/es/content_edition
export VBR_HOME

if [ "$VBR_HOME" =  "" ]
then
   error
fi

# ------------------------------
# DB2 Information Integrator Content Edition environment variables
# ------------------------------
# This file is used to set up DB2 Information Integrator Content Edition environment variables.
echo Setting WebSphere Information Integrator Content Edition environment variables
```

```
# WebSphere App Server Home
WAS_HOME=/opt/was
export WAS_HOME

# WebSphere MQ Home
MQ_HOME=/usr/mqm/java
export MQ_HOME

# A valid Java 2 installation.
JAVA_HOME=$WAS_HOME/java
export JAVA_HOME

PATH=$JAVA_HOME/bin:$VBR_HOME/datastore:$VBR_HOME/htmlconverter:$PATH
export PATH

LD_LIBRARY_PATH=$VBR_HOME/datastore:$VBR_HOME/htmlconverter:/usr/X11R6/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH

SHLIB_PATH=$VBR_HOME/datastore:$SHLIB_PATH
export SHLIB_PATH

LIBPATH=$VBR_HOME/datastore:$VBR_HOME/htmlconverter:$LIBPATH
export LIBPATH

# Some helpful variables for running client applications like the samples
JNDI_CLIENT_FACTORY=com.ibm.websphere.naming.WsnInitialContextFactory
export JNDI_CLIENT_FACTORY
JNDI_CLIENT_PROVIDER=iiop://localhost:2810
export JNDI_CLIENT_PROVIDER

# WebSphere implfactory.properties
VBR_CLASSPATH=$WAS_HOME/properties
export VBR_CLASSPATH

VBR_CLASSPATH=$VBR_CLASSPATH:$VBR_HOME/docs/examples/java
export VBR_CLASSPATH

VBR_CLASSPATH=$VBR_CLASSPATH:$VBR_HOME/docs/examples/wsapi/java
export VBR_CLASSPATH

# vbr_subscription_api.jar needed for RepoBrowser
VBR_CLASSPATH=$VBR_CLASSPATH:$VBR_HOME/opt/vbr_subscription.jar
VBR_CLASSPATH=$VBR_CLASSPATH:$VBR_HOME/datastore/datastore.jar
VBR_CLASSPATH=$VBR_CLASSPATH:$VBR_HOME/opt/vbr_vrepo.jar
VBR_CLASSPATH=$VBR_CLASSPATH:$VBR_HOME/opt/vbr_wc.jar
export VBR_CLASSPATH

# The VBR_CLASSPATH environment variable
for i in $VBR_HOME/lib/*.jar
do
VBR_CLASSPATH=$VBR_CLASSPATH:$i
export VBR_CLASSPATH
done
```

```
# Make sure that vbr.jar is first in the classpath
VBR_CLASSPATH=$VBR_HOME/lib/vbr.jar:$VBR_CLASSPATH
export VBR_CLASSPATH

# Set the portion of the classpath needed to enable an EJB client like Admin
# Tool to access app server EJBs
EJB_CLIENT_CLASSPATH=$WAS_HOME/lib/bootstrap.jar:$WAS_HOME/lib/ejbportable.jar
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$WAS_HOME/lib/disthub.jar:$WAS_HOME/lib/ecutils.jar
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$WAS_HOME/lib/ffdc.jar:$WAS_HOME/lib/idl.jar
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$WAS_HOME/lib/iwsorb.jar:$WAS_HOME/lib/j2ee.jar
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$WAS_HOME/lib/logbrjface.jar:$WAS_HOME/lib/naming.jar
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$WAS_HOME/lib/namingclient.jar:$WAS_HOME/lib/ras.jar
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$WAS_HOME/lib/tcljava.jar:$WAS_HOME/lib/tx.jar
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$WAS_HOME/lib/utils.jar:$WAS_HOME/lib/wsexception.jar
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$WAS_HOME/lib/txClientPrivate.jar
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$WAS_HOME/lib/messagingClient.jar

# These three jar files are required for WebSphere Application Server 5.1.x, but not WebSphere Application
Server 6.x.
# Additionally, the mqbind jar file has been removed in WebSphere Application Server 6.x and the
# other two jar files have moved in the WebSphere Application Server installation LIB directory.
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$MQ_HOME/lib/com.ibm.mq.jar
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$MQ_HOME/lib/com.ibm.mqbind.jar
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$MQ_HOME/lib/com.ibm.mqjms.jar

# This jar file is only available on, and required for WebSphere Application Server 6.x
EJB_CLIENT_CLASSPATH=$EJB_CLIENT_CLASSPATH:$WAS_HOME/lib/emf.jar
export EJB_CLIENT_CLASSPATH

VBR_CLASSPATH=$VBR_CLASSPATH:$EJB_CLIENT_CLASSPATH
```

4. Set WebSphere IICE Admin to run in a direct mode by editing the Admin.sh file to include the following entry, as shown in Example 3-2.

   ```
   -Dvbr.as.operationMode=direct \
   ```

Example 3-2 Admin.sh file contents

```
#!/bin/sh
# Runs the Administration Tool Console.

if [ "$VBR_HOME" = "" ]
then
. ./config.sh
  else
. $VBR_HOME/bin/config.sh
fi

# To run in direct mode, add the following Java system property
# -Dvbr.as.operationMode=direct \

java -classpath \
"$VBR_CLASSPATH" \
-Dvbr.home="$VBR_HOME" \
-Dvbr.as.operationMode=direct \
-Dlog4j.category.com.venetica.vbr.tools.admin=WARN \
com.venetica.vbr.tools.admin.AdminFrame $1 $2 $3 $4
```

5. Start ./Admin.sh to bring up the Administration tool (Figure 3-5).

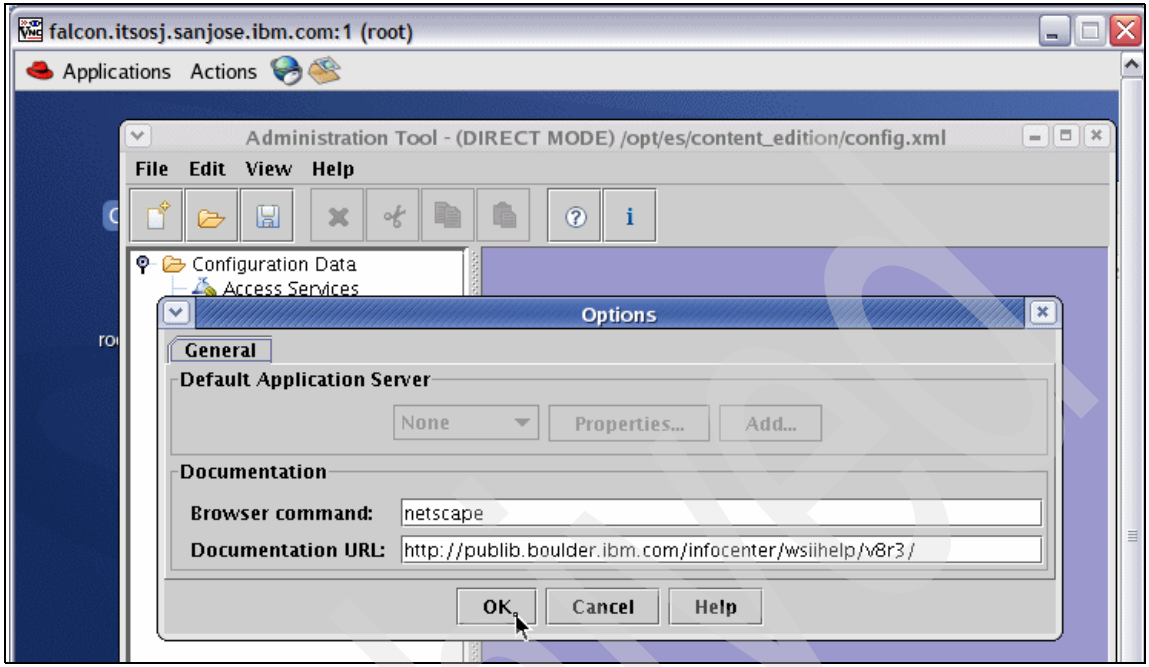

Figure 3-5 WebSphere IICE Administration Tool

6. To create a new connector, right click **Connectors** in the navigation pane and choose **New IBM WebSphere Portal Document Manager Connector**, as shown in Figure 3-6.

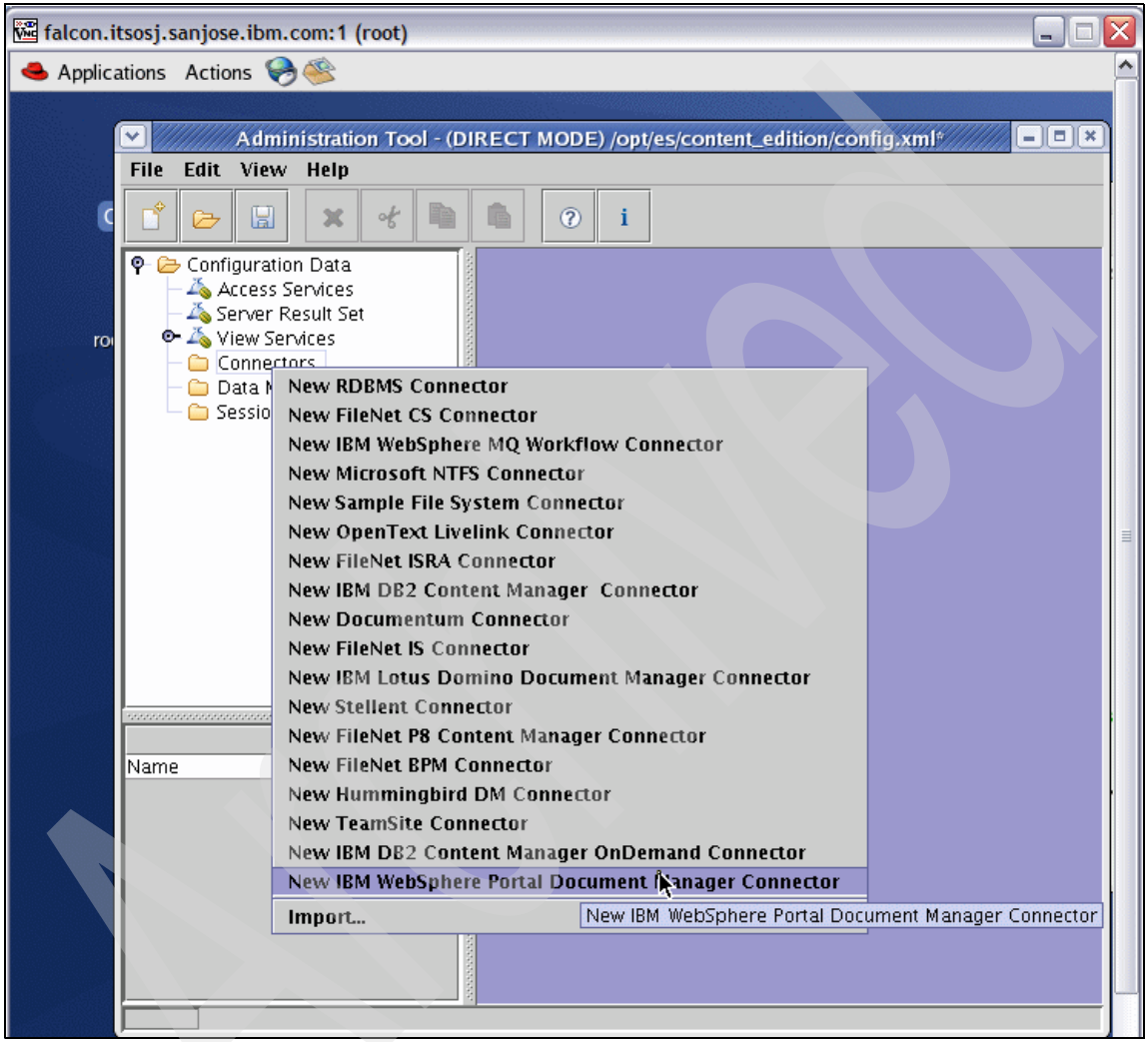

Figure 3-6 Choose New IBM WebSphere Portal Document Manager Connector

7. Modify the RMI proxy connector URL to reference the following location of the WebSphere IICE RMIBridgeServer, as shown in Figure 3-7 through Figure 3-10 on page 159:

   `http://kazan.itsosj.sanjose.ibm.com:1250/RMIBridgeServer2`

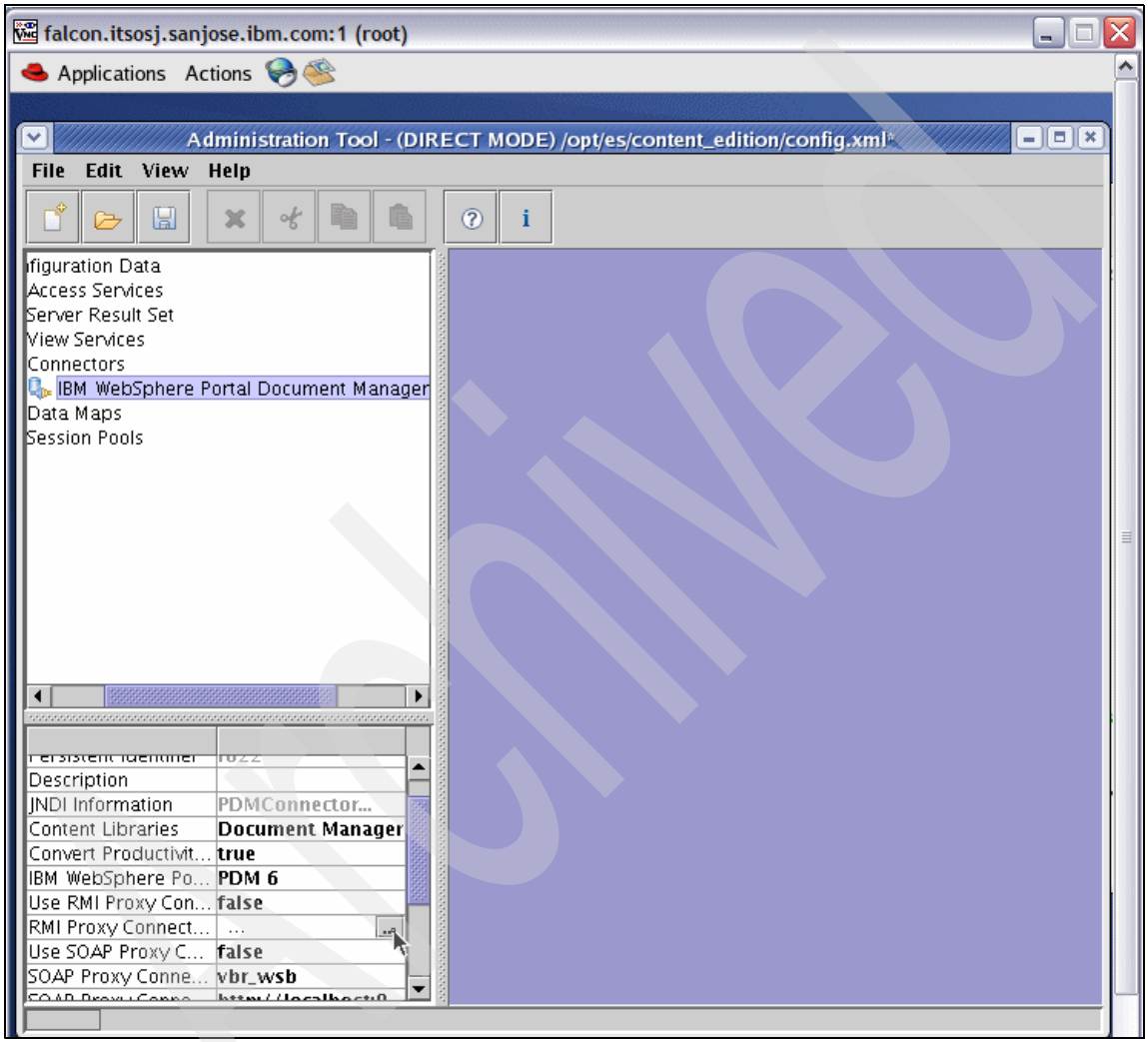

Figure 3-7 Modify RMI Proxy Connector URL 1/4

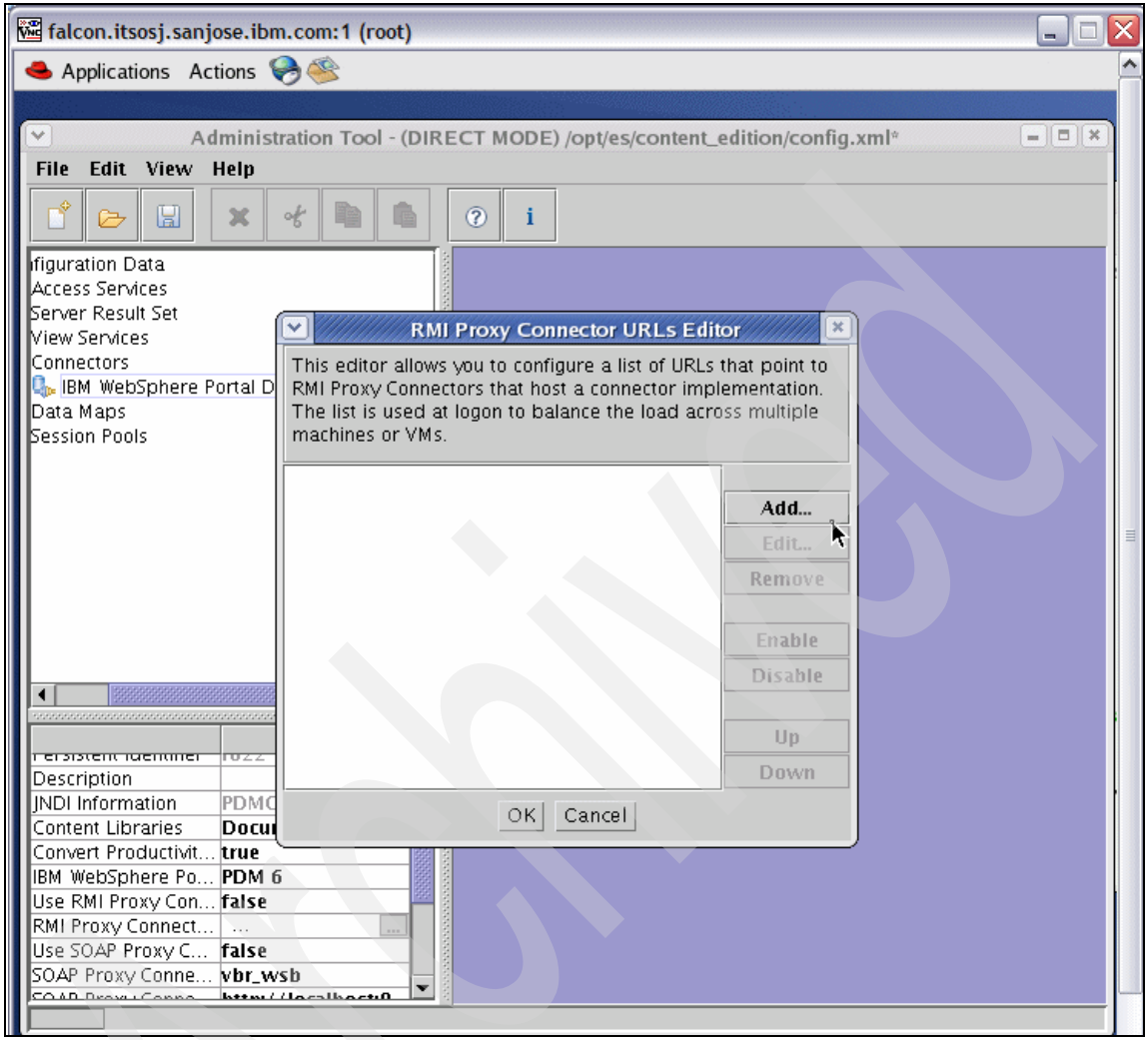

Figure 3-8 Modify RMI Proxy Connector URL 2/4

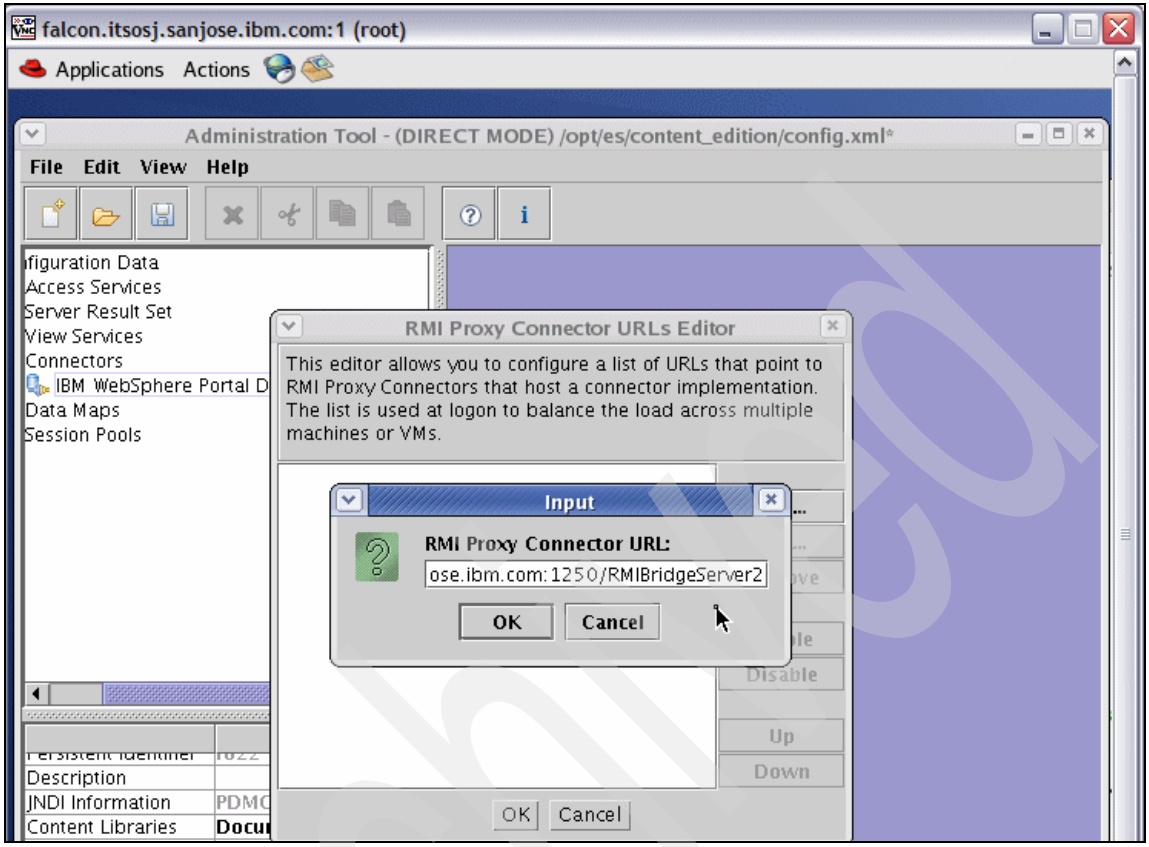

Figure 3-9 Modify RMI Proxy Connector URL 3/4

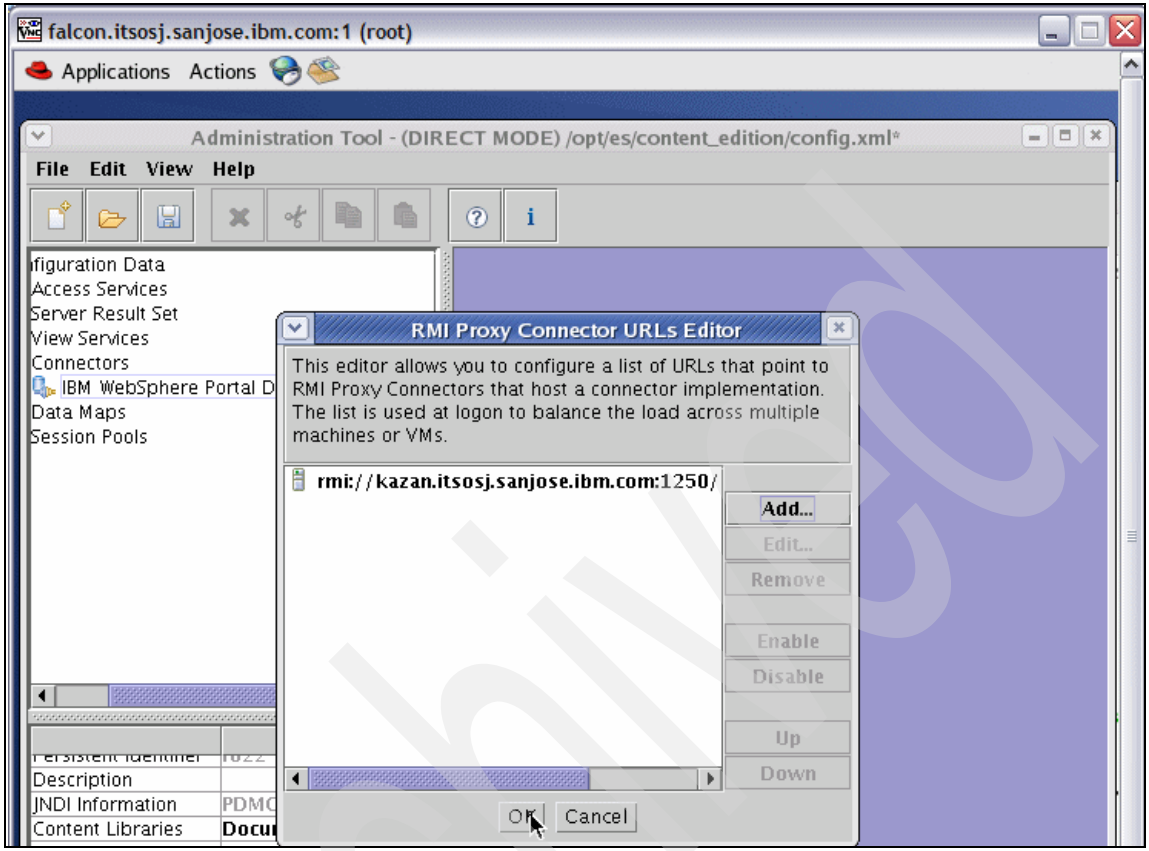

Figure 3-10 Modify RMI Proxy Connector URL 4/4

8. Modify the Use RMI Proxy Connector property value to true, as shown in Figure 3-11, and then save the configuration changes, as shown in Figure 3-12 on page 161.

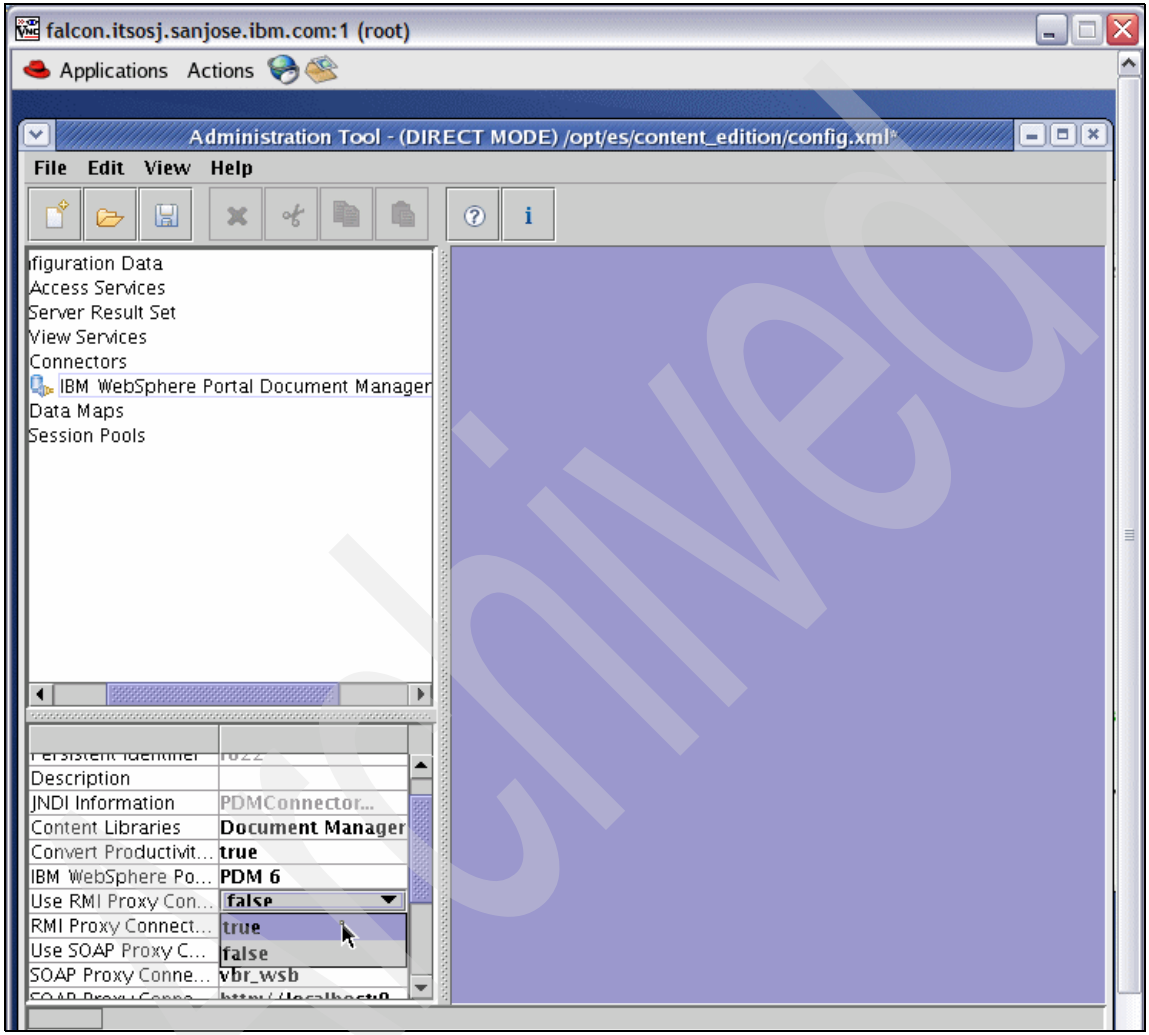

Figure 3-11 Modify the Use RMI Proxy Connector property value to true

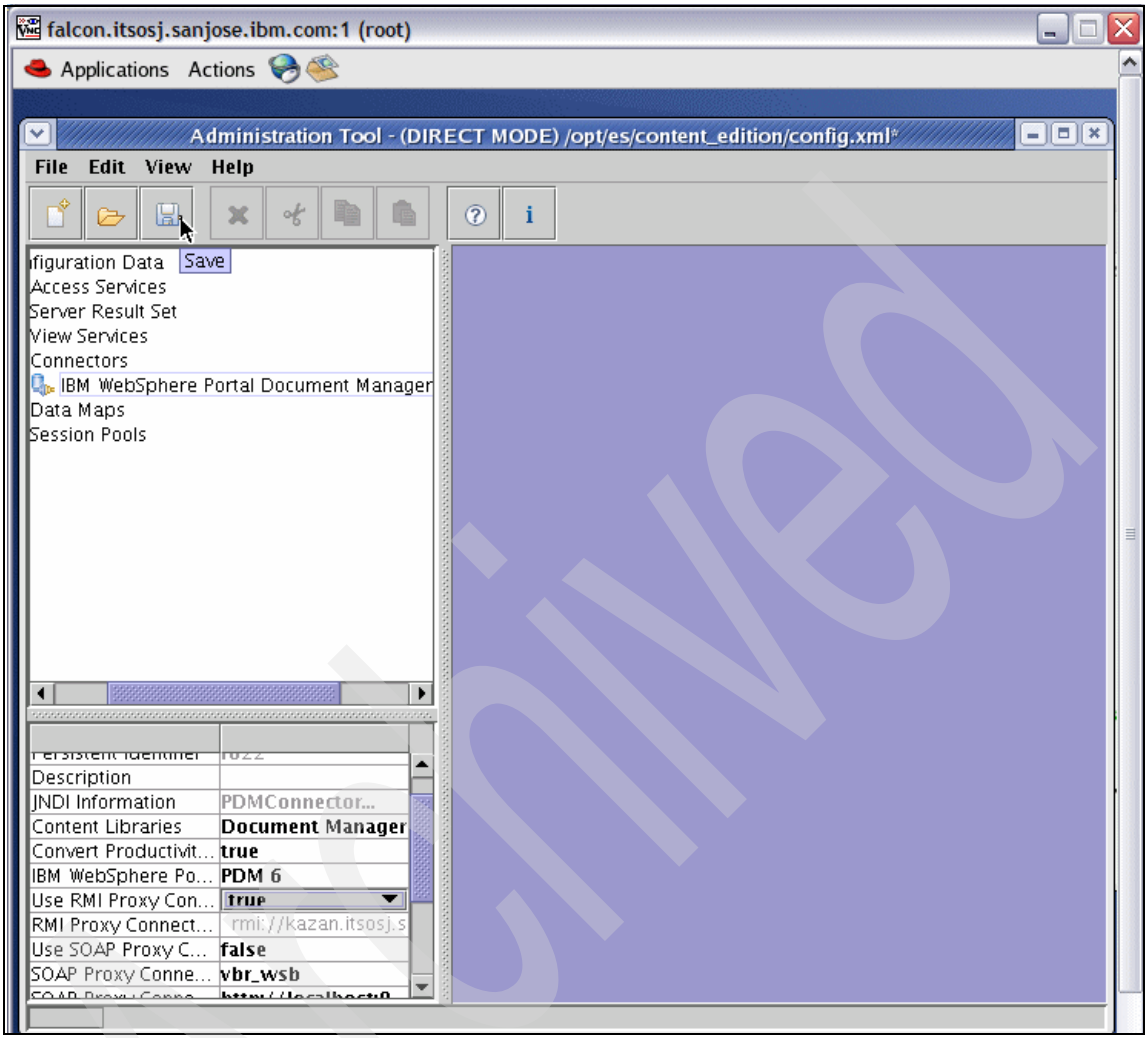

Figure 3-12 Save all the configuration changes

9. Test the newly configured connector by right clicking the **WebSphere IBM Portal Document Manager Connector** and selecting **Test Connection**, as shown in Figure 3-13. Provide a user name and password (wpsadmin) at the prompt, as shown in Figure 3-14 on page 163. When the connection is successful, you get the connection successful message, as shown in Figure 3-15 on page 164.

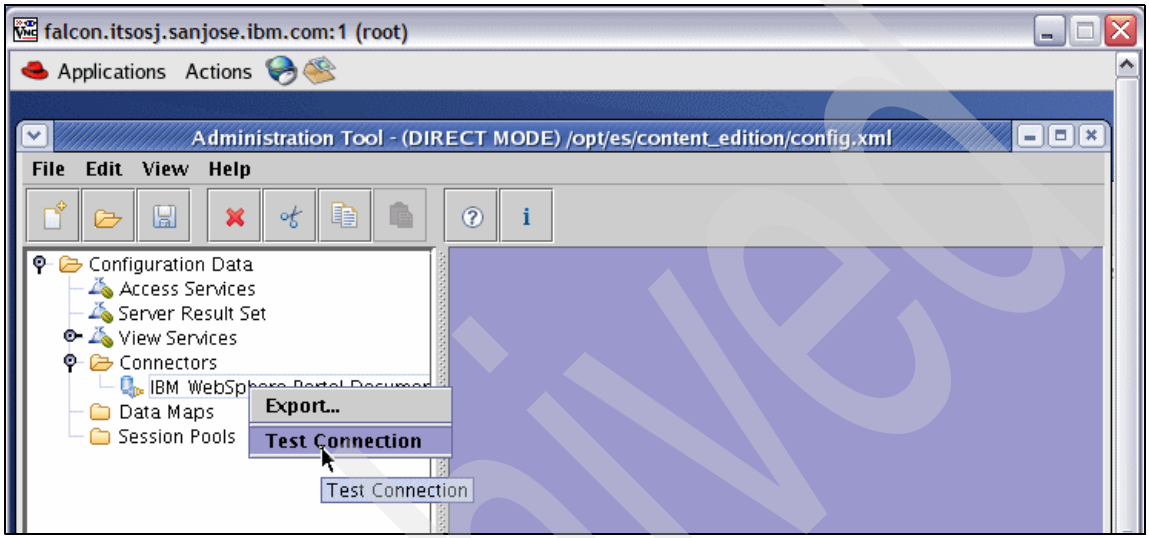

Figure 3-13 Test the IBM WebSphere Portal Document Manager Connector 1/3

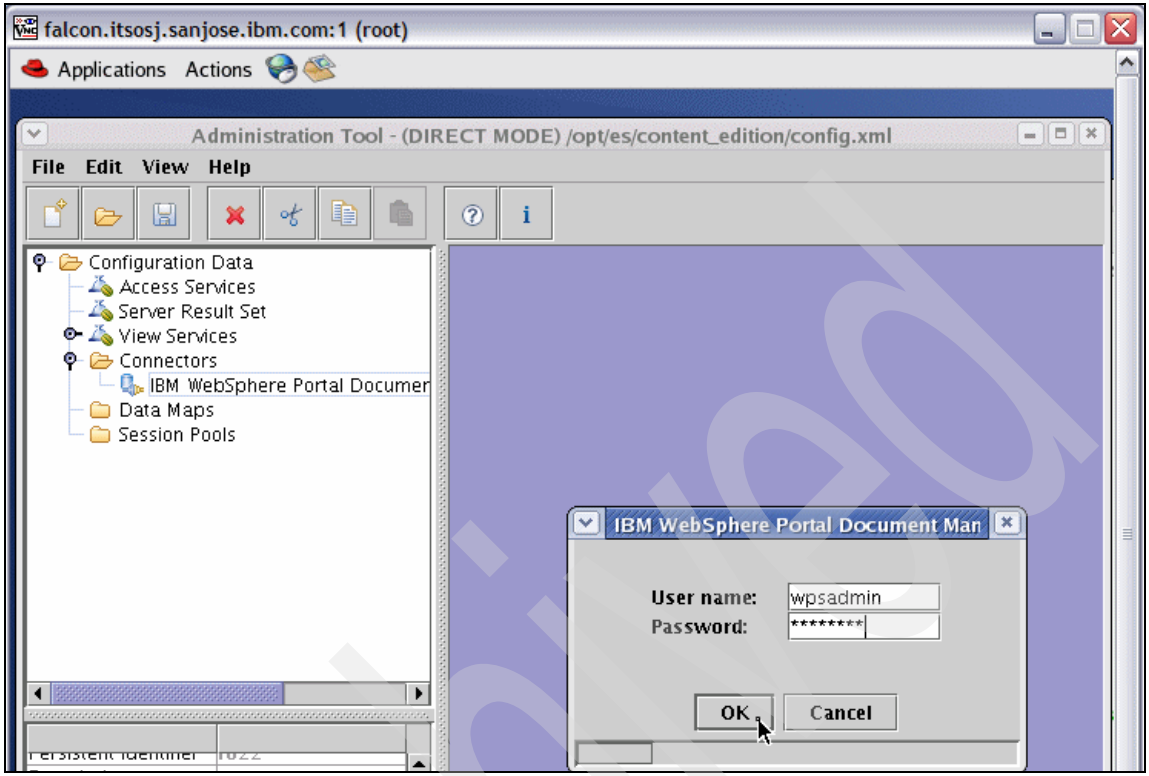

Figure 3-14 Test the IBM WebSphere Portal Document Manager Connector 2/3

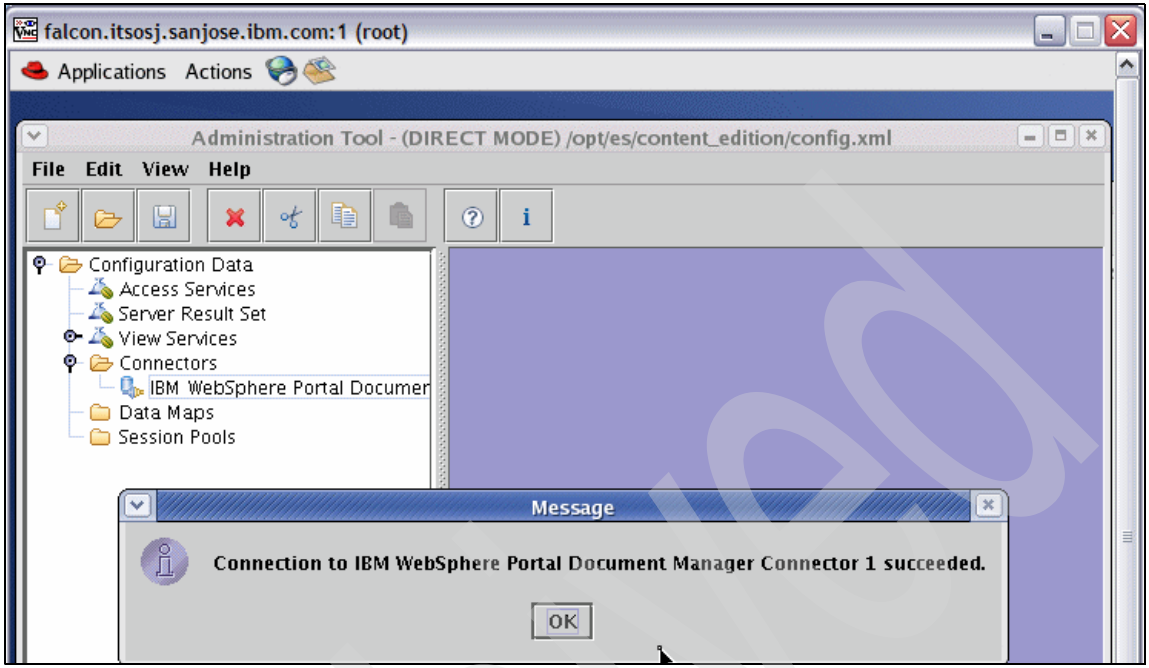

Figure 3-15 Test the IBM WebSphere Portal Document Manager Connector 3/3

Configure WCM for the Web crawler

Since we will be using the Web crawler to access WCM without security, we need to ensure that the WCM content to be crawled has public access granted (resources have read access of [all users] group granted); this includes all sites, site areas, library components, and presentation templates. We also need to identify the starting URL for the Web crawler for WCM content.

Note: We created Web Content named “menu_content” (displayed as Content Menu in Figure 3-16) that links to all the WCM content we want to crawl rather than the built-in facility that crawls all the content. It is the starting link for the Web crawler for the WCM content).

Log in to the WebSphere Portal Server, launch Web Content Management, expand to and highlight **Insurance definitions** in the navigation pane, select **Content Menu**, and click **Edit**, as shown in Figure 3-16 on page 166, to view details.

Figure 3-17 on page 167 through Figure 3-23 on page 172 show the navigation to the appropriate fields in order to grant read access to “menu_content” to the group [all users]. Click **Show Hidden Fields** in Figure 3-17 on page 167 and use the scroll bar to view the **Access** field in Figure 3-18 on page 168. Expand it and click **Grant Read Access** in Figure 3-19 on page 169 to edit the list of users with Read access to “menu_content”. Select Groups from the drop-down list in Figure 3-20 on page 170 and click **Search** to obtain the available groups. Select the **(all users)** group and click **OK**, as shown in Figure 3-21 on page 171 and Figure 3-22 on page 172 to complete the “all users” group access to “menu_content”. This is confirmed in Figure 3-23 on page 172.

Figure 3-24 on page 173 shows the changes made to “menu_content” resource permissions being saved.

Ensure that the crawler has access to the WCM libraries. Figure 3-25 on page 174 through Figure 3-27 on page 176 show Web Content Libraries being granted access to the public. After logging in to the WebSphere Portal, expand Portal Content in the navigation pane and click Web Content Libraries, as shown in Figure 3-25 on page 174. Click the key icon for Web Content, as shown in Figure 3-26 on page 175, to view the resource permissions for it. Select all the boxes under Allow Propagation and Allow Inheritance for all the Roles and click **Apply** to essentially grant public access to all of the Web Content Libraries, as shown in Figure 3-27 on page 176.

Figure 3-28 on page 177 through Figure 3-31 on page 179 show how the starting URL for the Web crawler for WCM can be obtained by previewing “menu_content” in WCM. In this case, as shown in Figure 3-31 on page 179, it is:

```
http://kazan.itsosj.sanjose.ibm.com:10038/wps/wcm/connect/web+content/insurance/definitions/menu_content
```

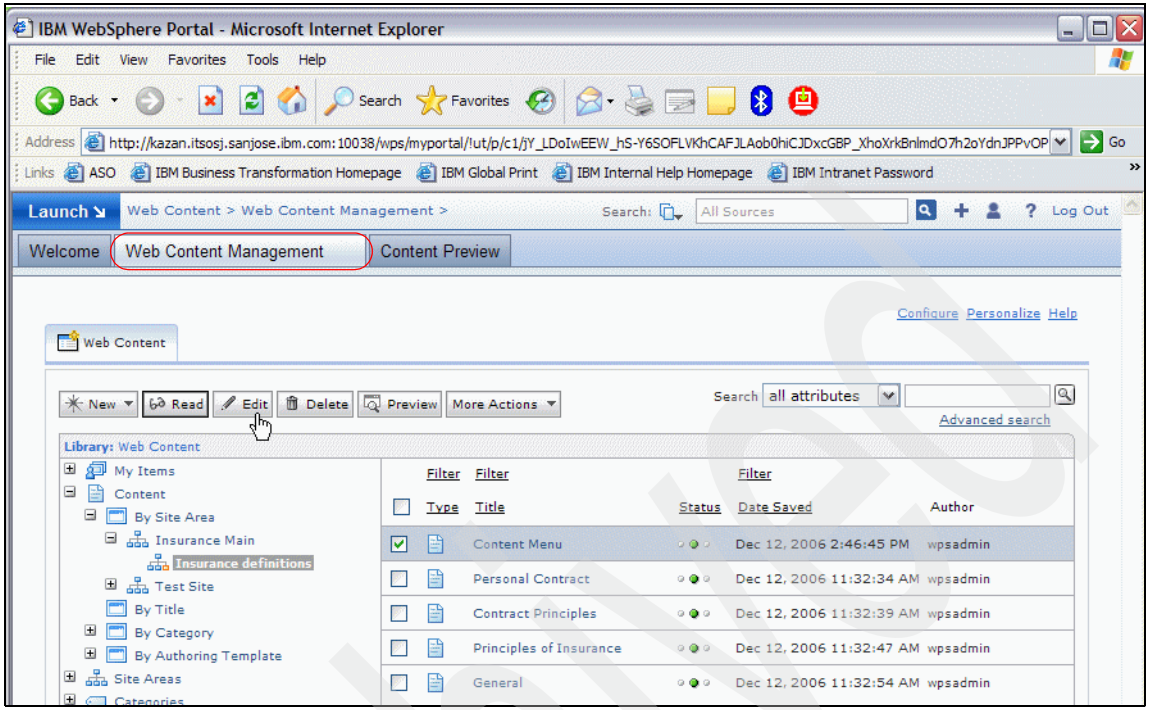

Figure 3-16 WCM content to be crawled

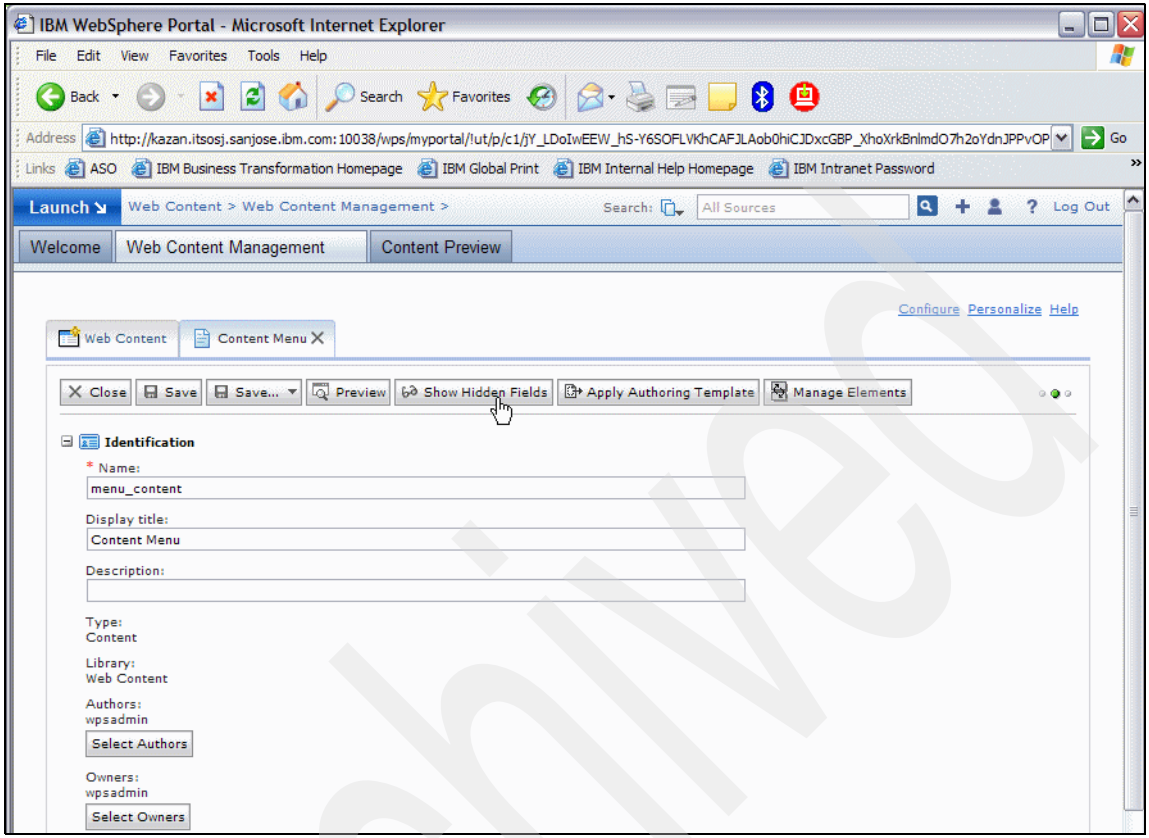

Figure 3-17 Grant access to all users 1/7

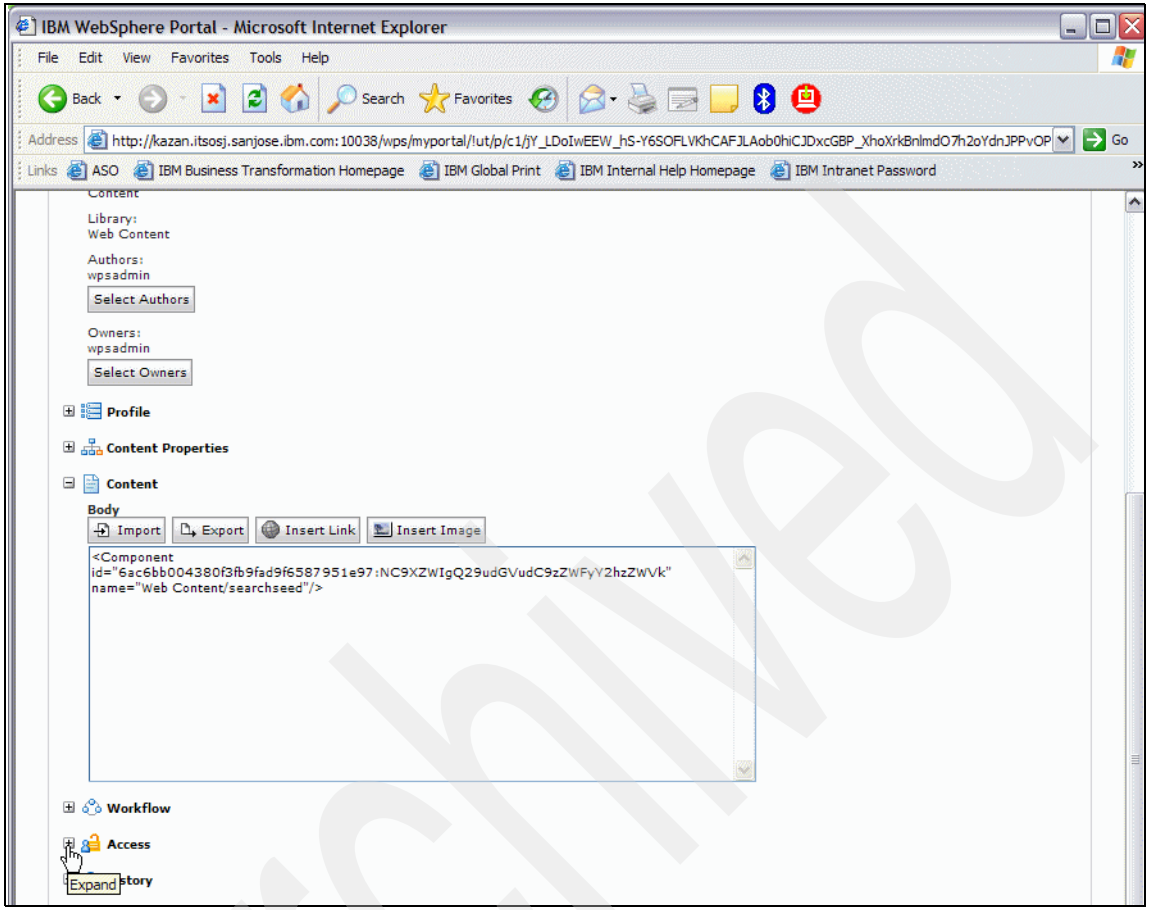

Figure 3-18 Grant access to all users 2/7

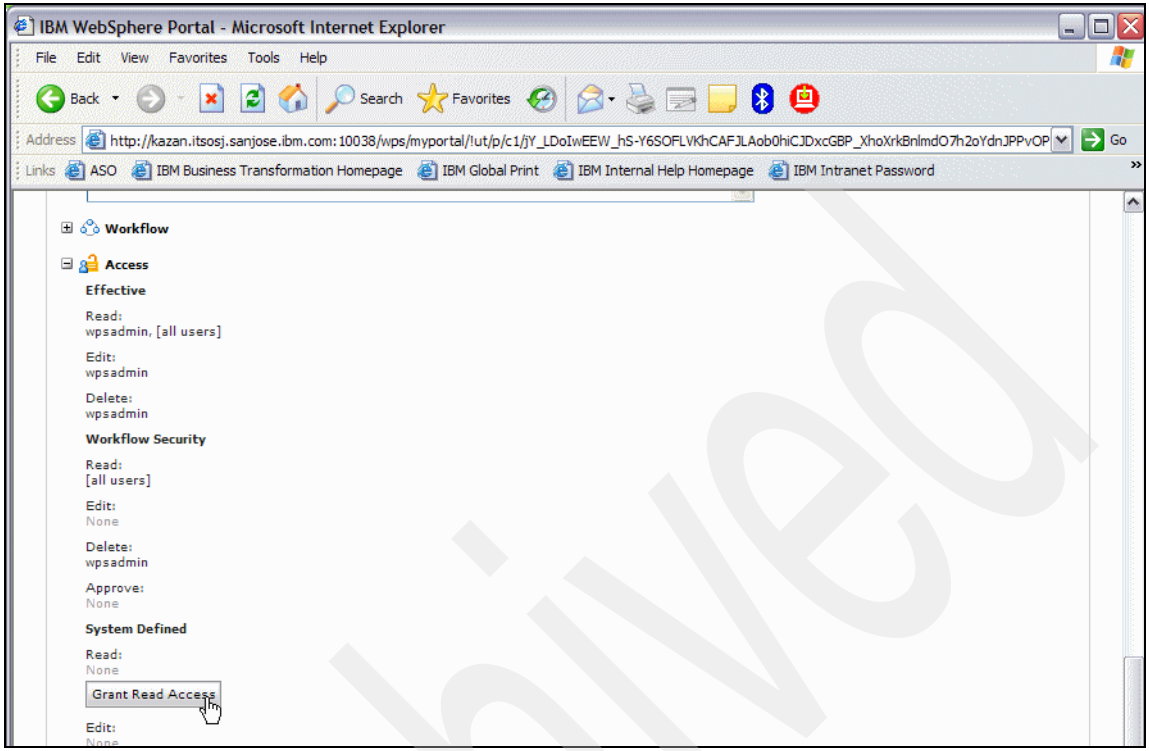

Figure 3-19 Grant access to all users 3/7

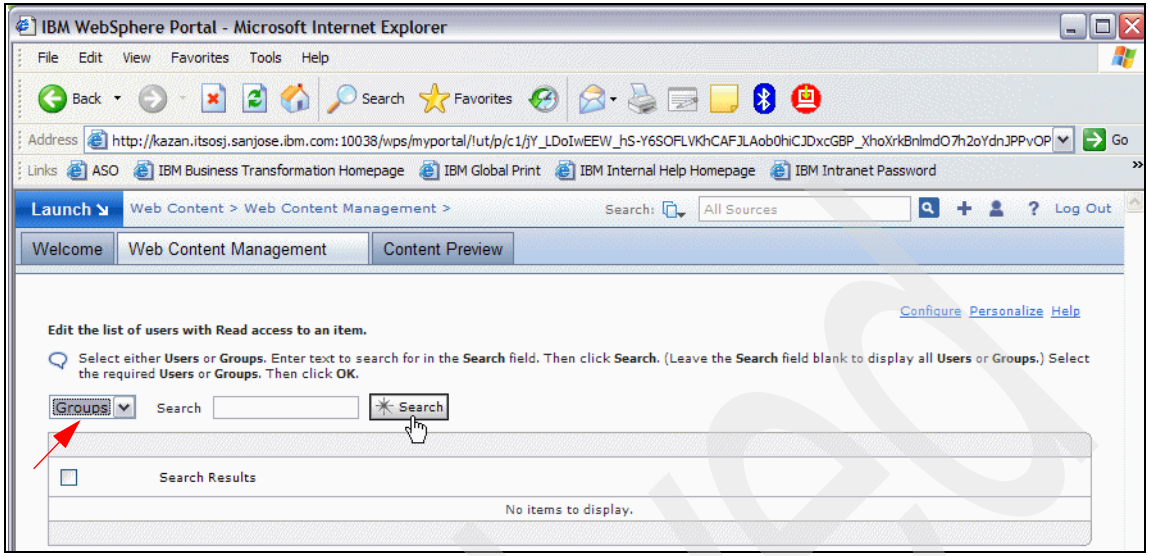

Figure 3-20 Grant access to all users 4/7

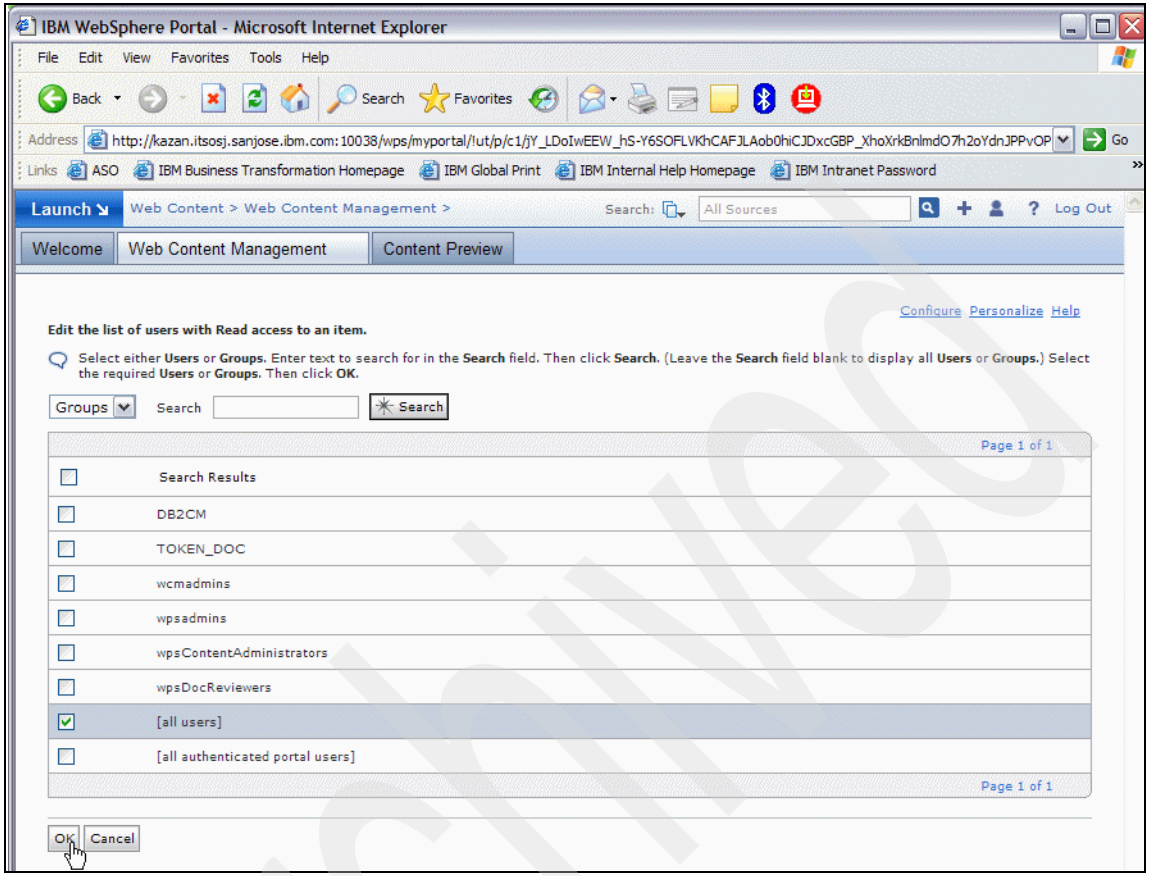

Figure 3-21 Grant access to all users 5/7

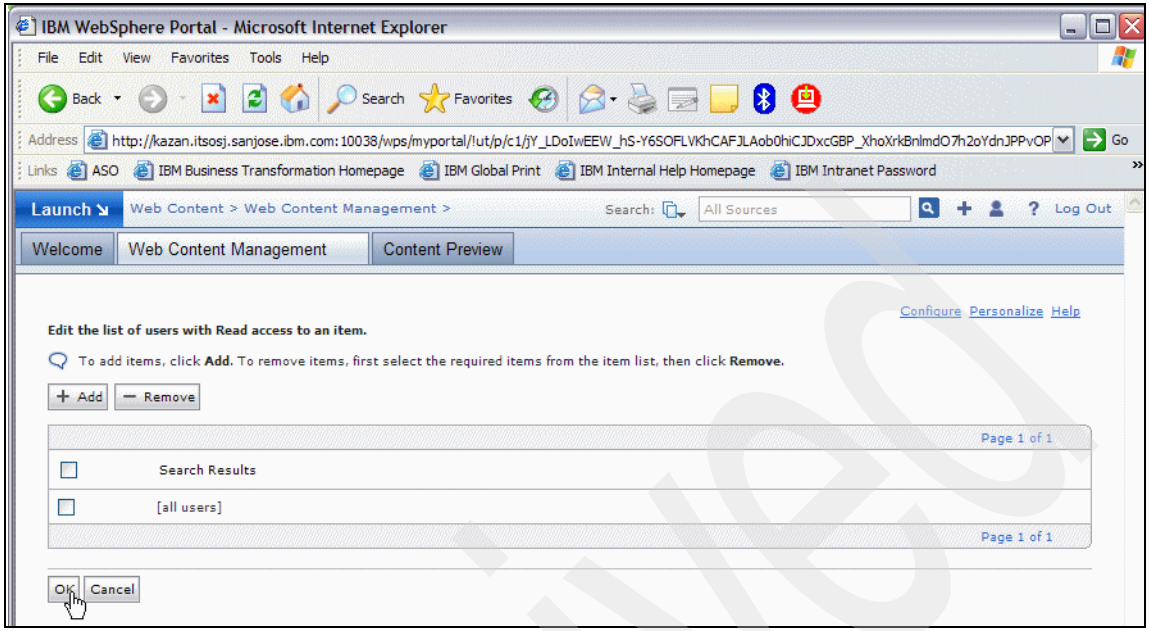

Figure 3-22 Grant access to all users 6/7

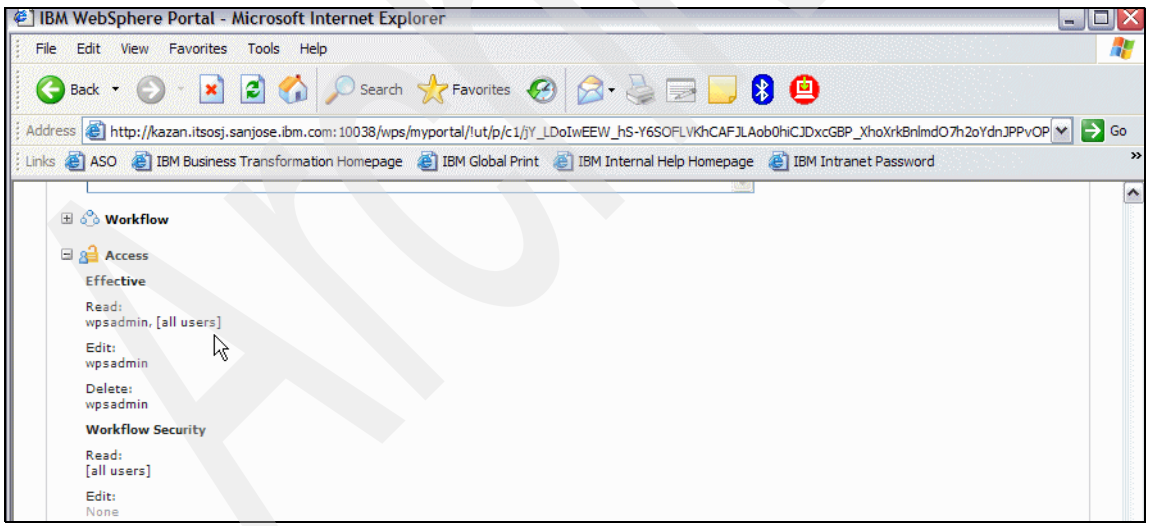

Figure 3-23 Grant access to all users 7/7

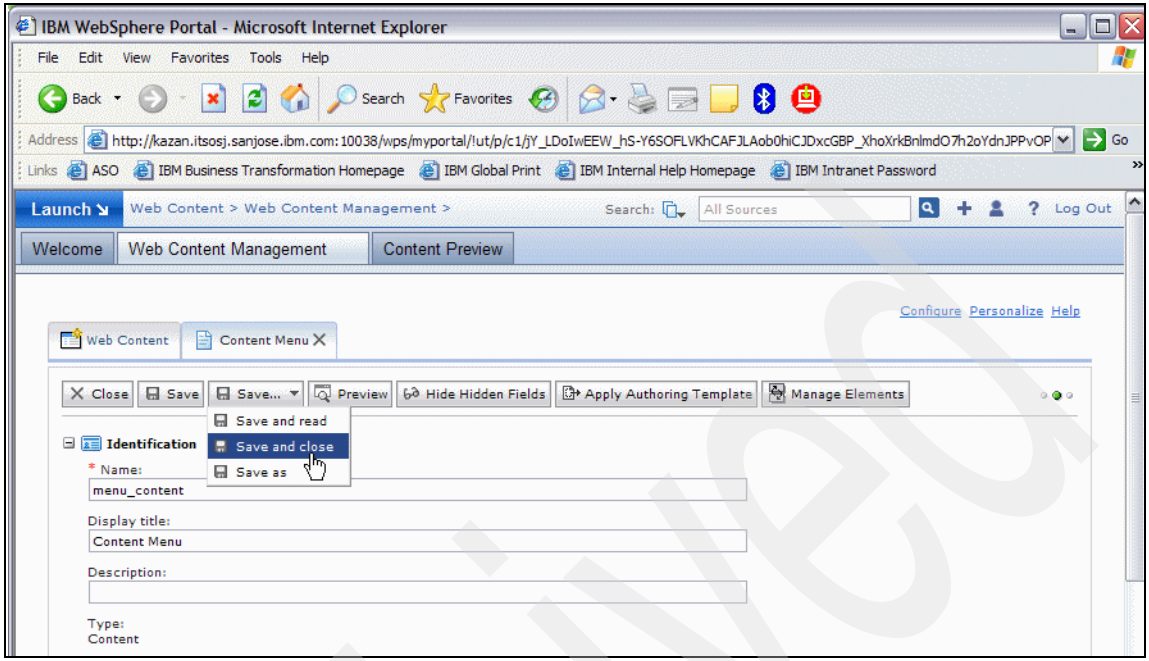

Figure 3-24 Save the changes

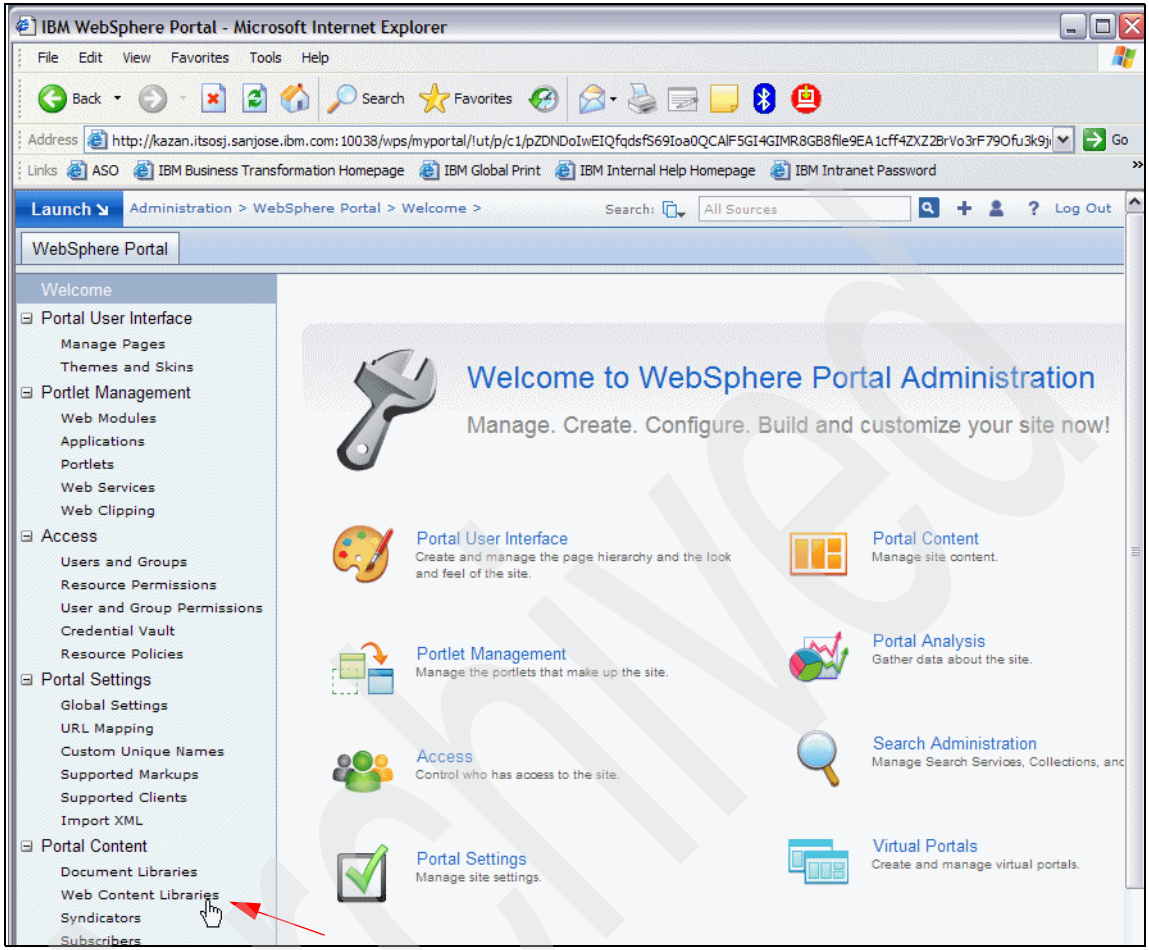

Figure 3-25 Grant total access to Web Content Libraries to all 1/3

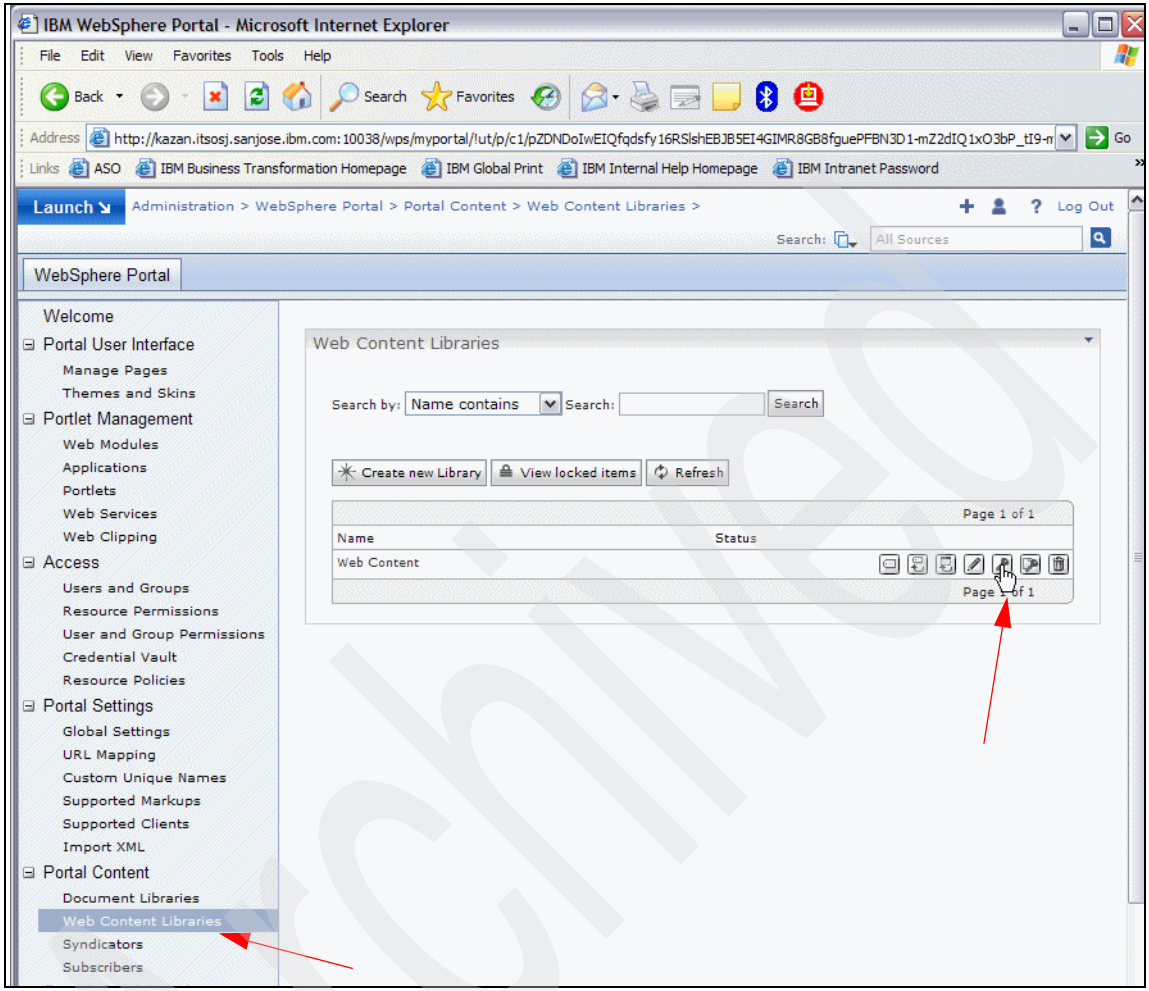

Figure 3-26 Grant total access to Web Content Libraries to all 2/3

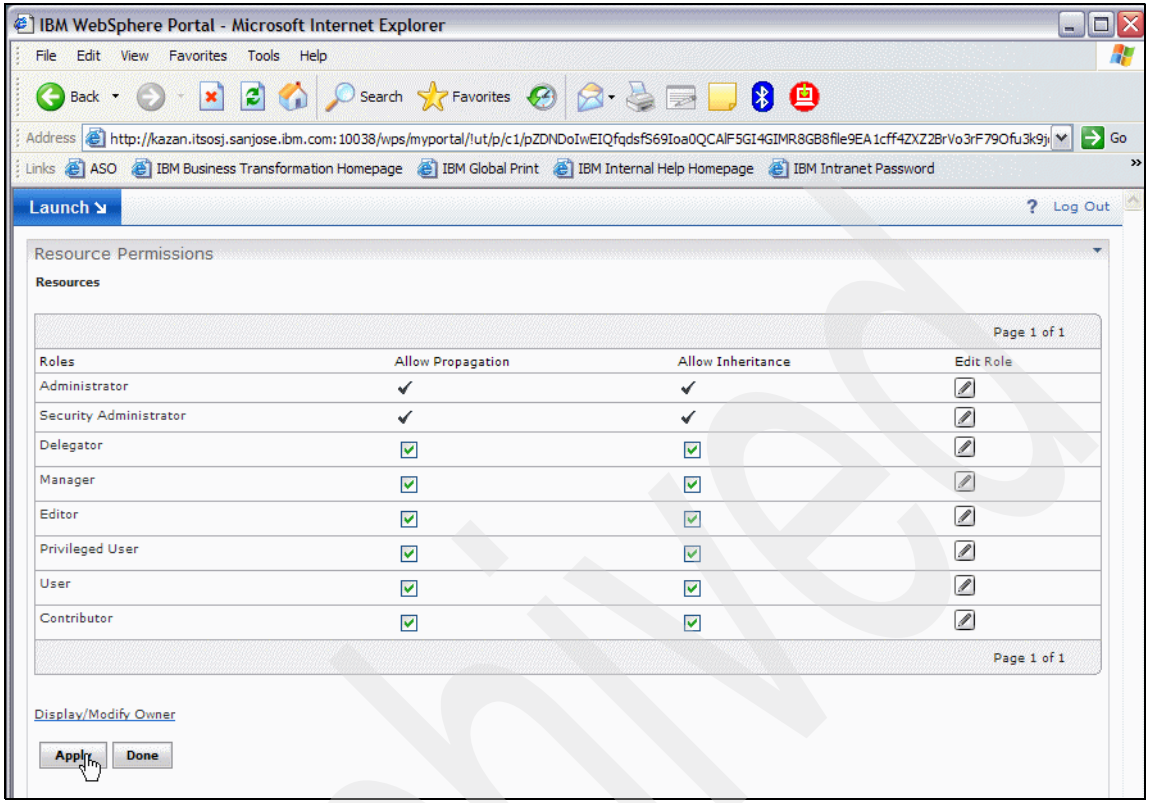

Figure 3-27 Grant total access to Web Content Libraries to all 3/3

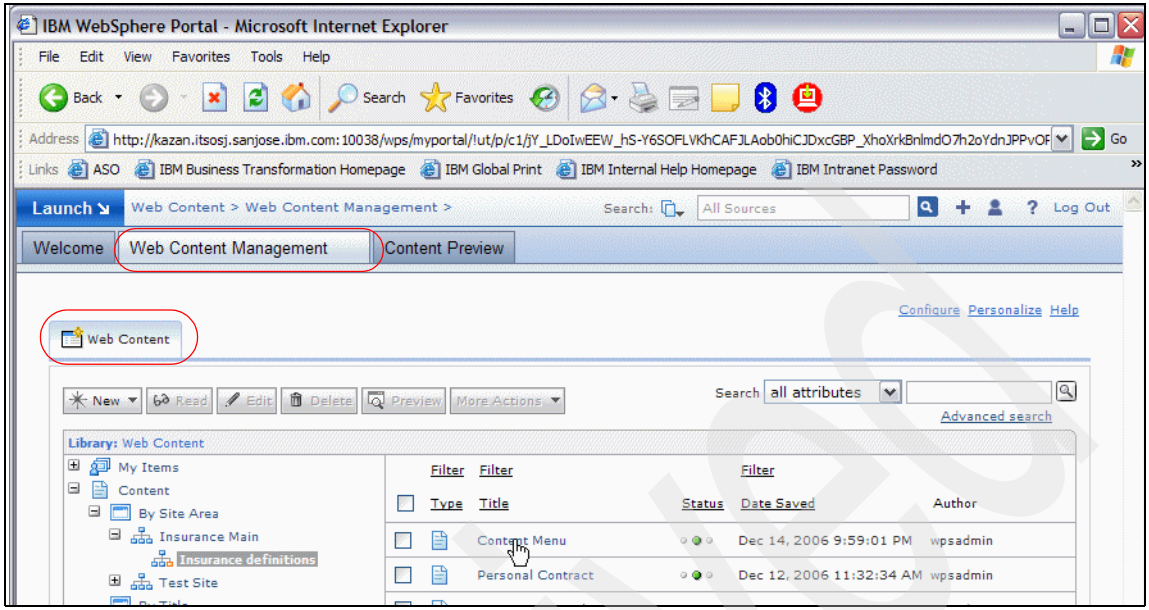

Figure 3-28 Preview menu_content in WCM to obtain the starting URL link for the Web crawler 1/4

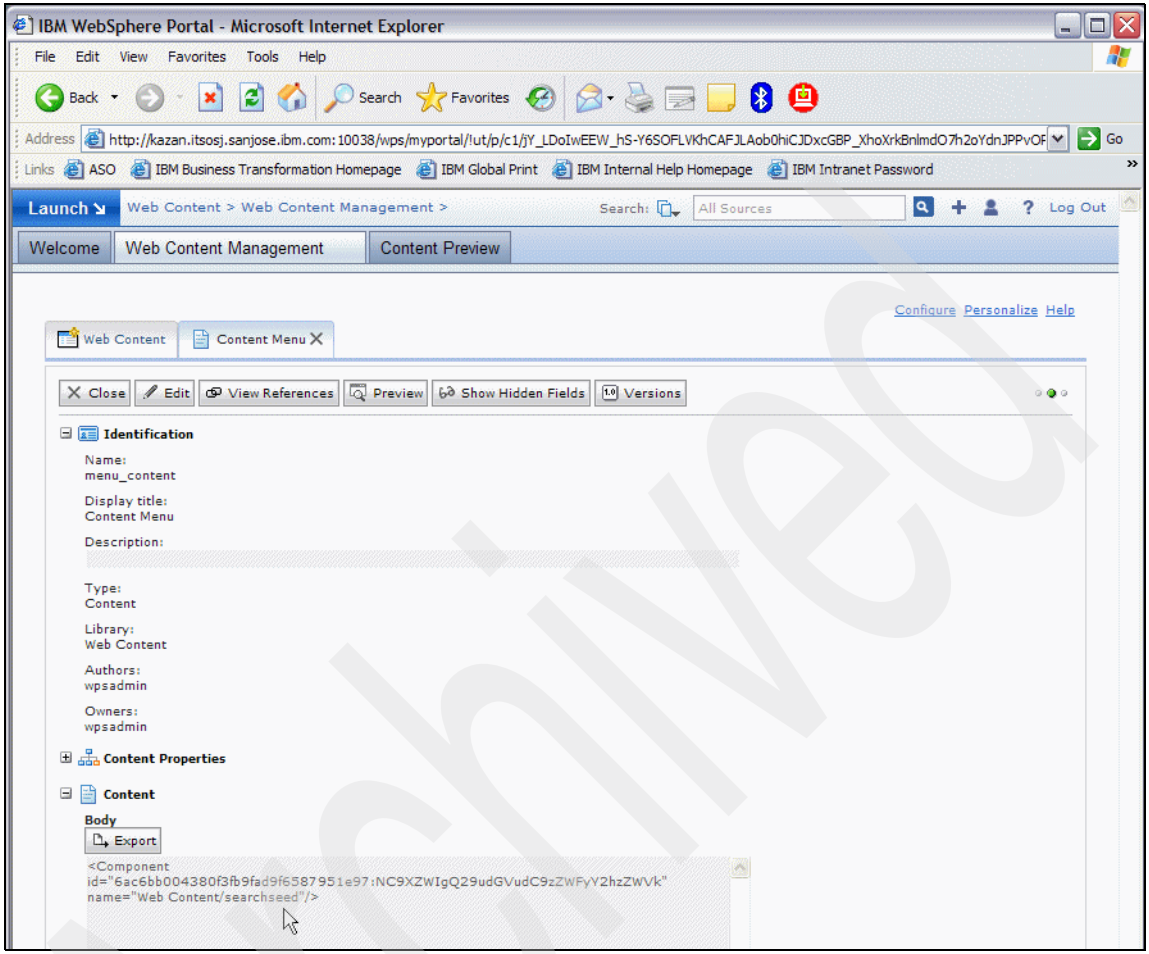

Figure 3-29 Preview menu_content in WCM to obtain the starting URL link for the Web crawler 2/4

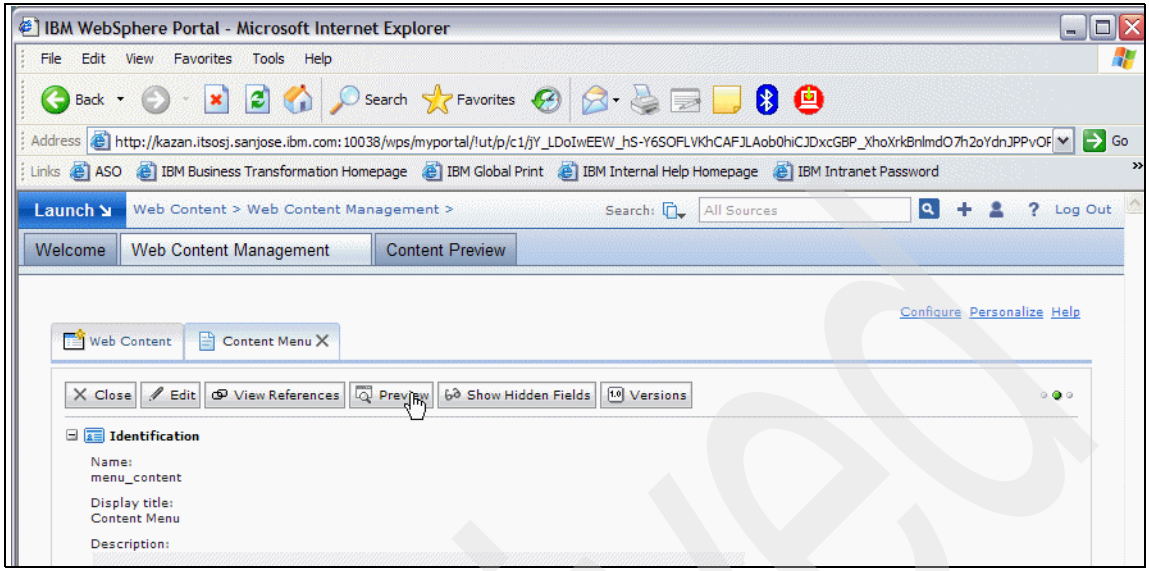

Figure 3-30 Preview menu_content in WCM to obtain the starting URL link for the Web crawler 3/4

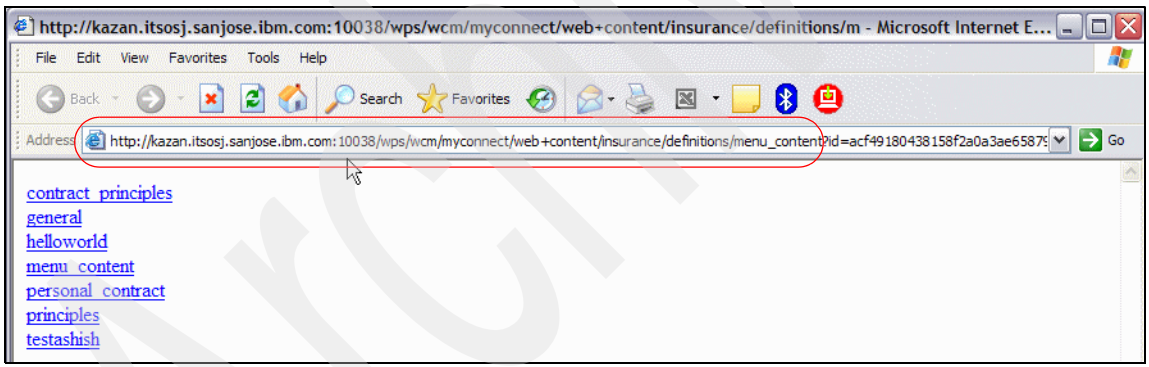

Figure 3-31 Preview menu_content in WCM to obtain the starting URL link for the Web crawler 4/4

Determine the URL for WebSphere Portal crawler

The WebSphere Portal crawler must be provided a starting URL for its crawl similar to that of the Web crawler.

Figure 3-32 on page 181 through Figure 3-37 on page 186 show the steps involved in determining the starting URL for the WebSphere Portal crawler.

After logging in to the WebSphere Portal, expand **Search Administration** and click **Manage Search**, as shown in Figure 3-32 on page 181, to create or manage Search Services. Click **Search Collections** to display and manage all search collections across various Search Services, as shown in Figure 3-33 on page 182. Search for collections in the Default Portal Search Service, as shown in Figure 3-34 on page 183, and select the Portal Content collection, which contains the content to be crawled, as shown in Figure 3-35 on page 184. Click the Edit Content Source icon for the Portal Content Source in Figure 3-36 on page 185 to view the details of this content source, which is the target of the WebSphere Portal crawler. The Collect documents linked from this URL field has the value (http://kazan.itsosj.sanjose.ibm.com:10038/wps/portal/....) as the starting URL for the WebSphere Portal crawler.

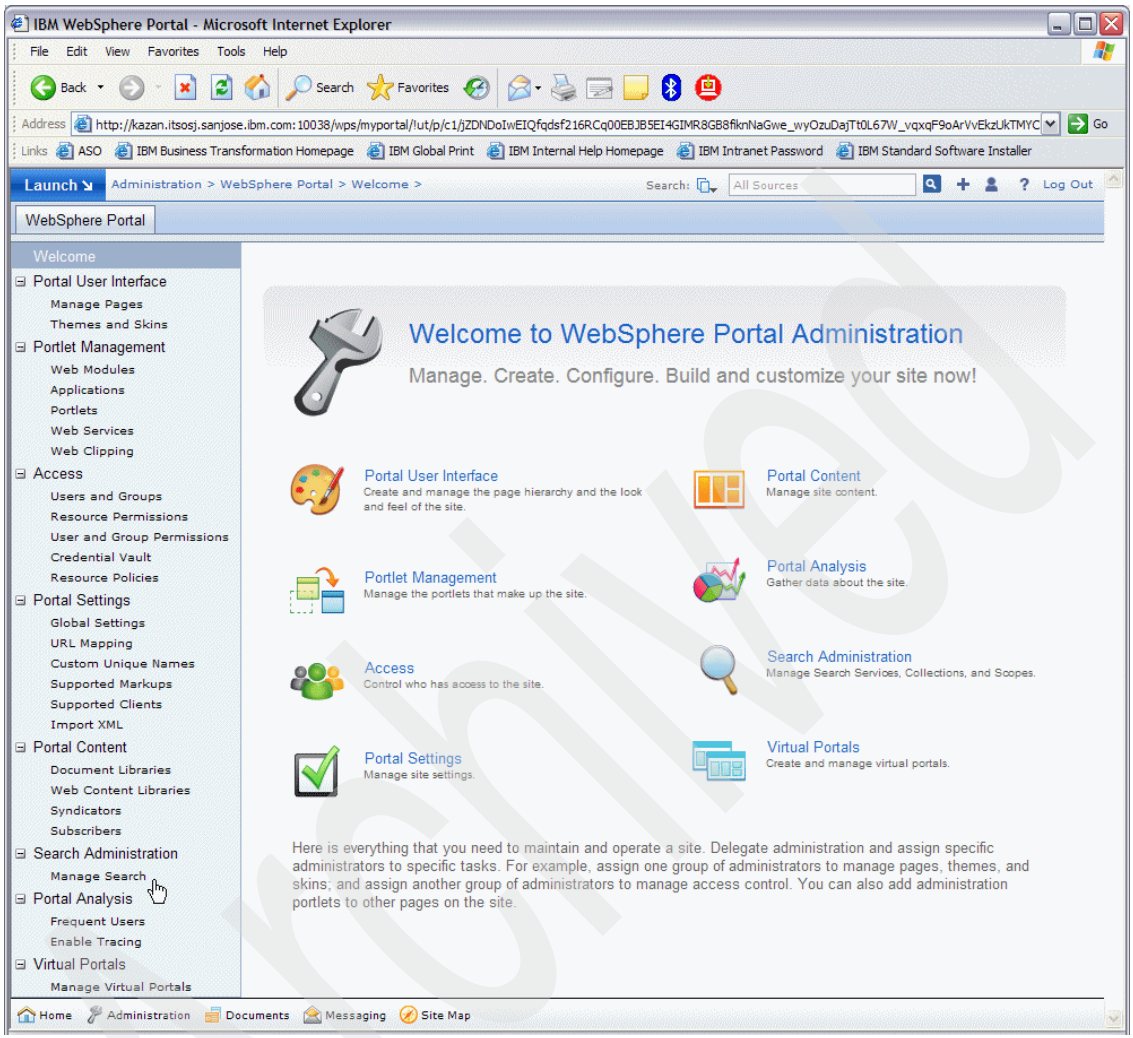

Figure 3-32 Determine the starting URL for the WebSphere Portal crawler 1/6

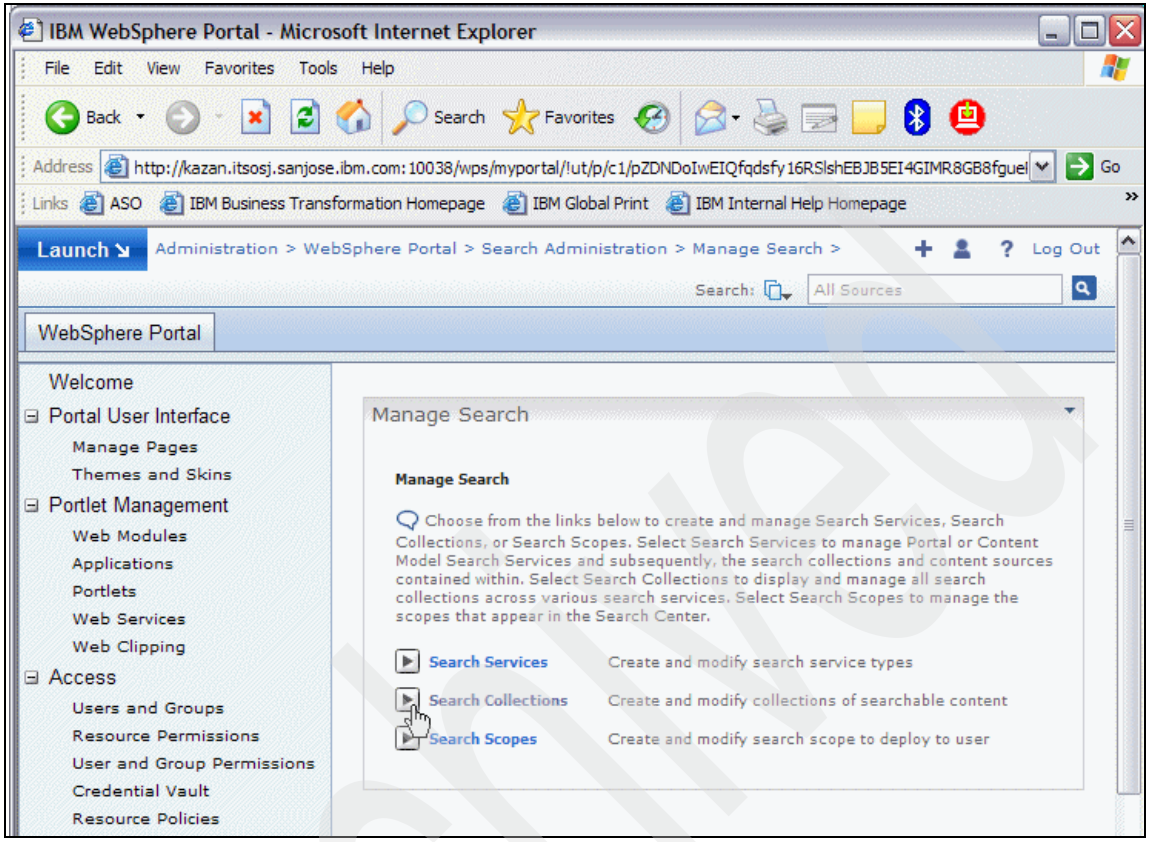

Figure 3-33 Determine the starting URL for the WebSphere Portal crawler 2/6

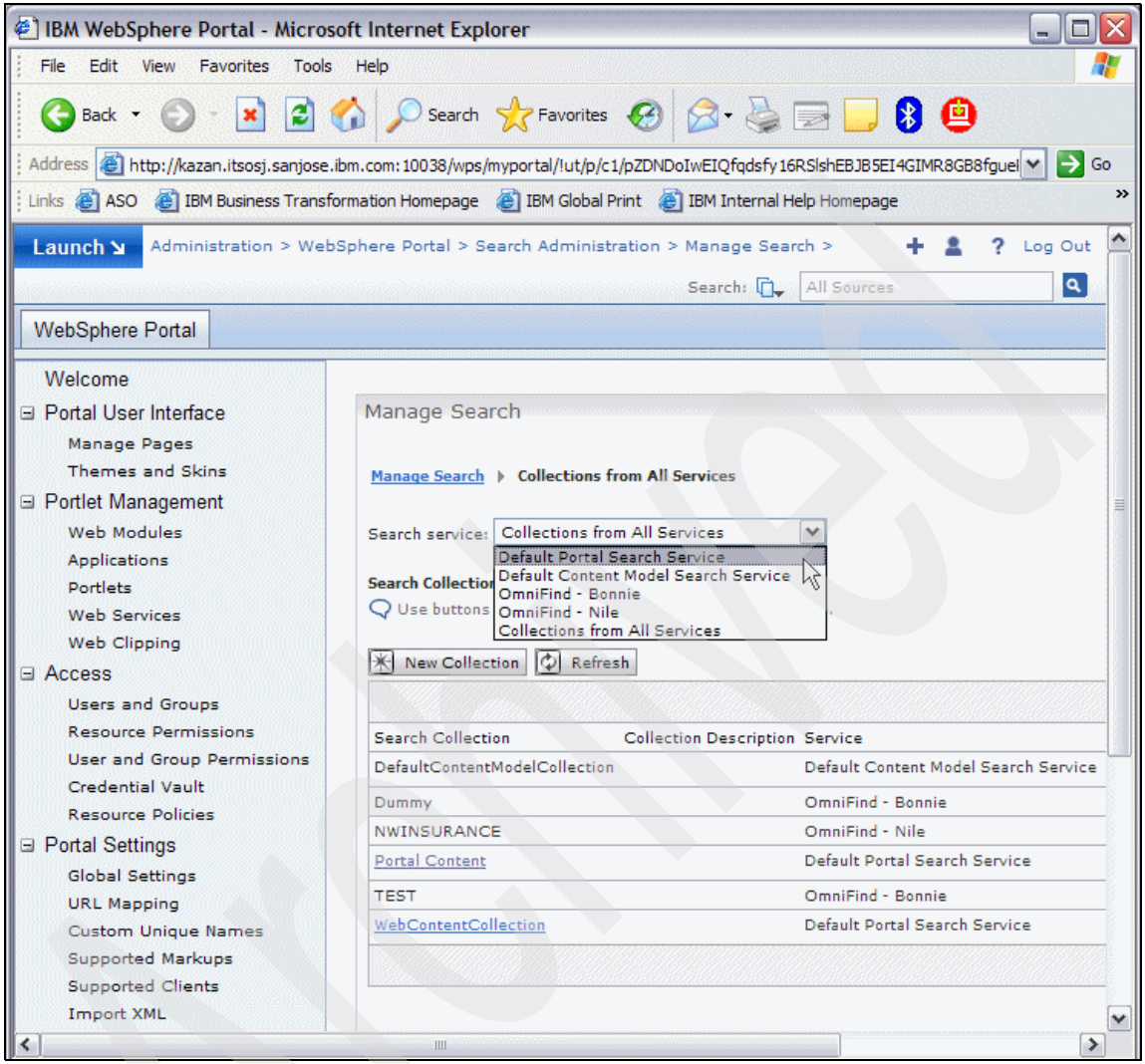

Figure 3-34 Determine the starting URL for the WebSphere Portal crawler 3/6

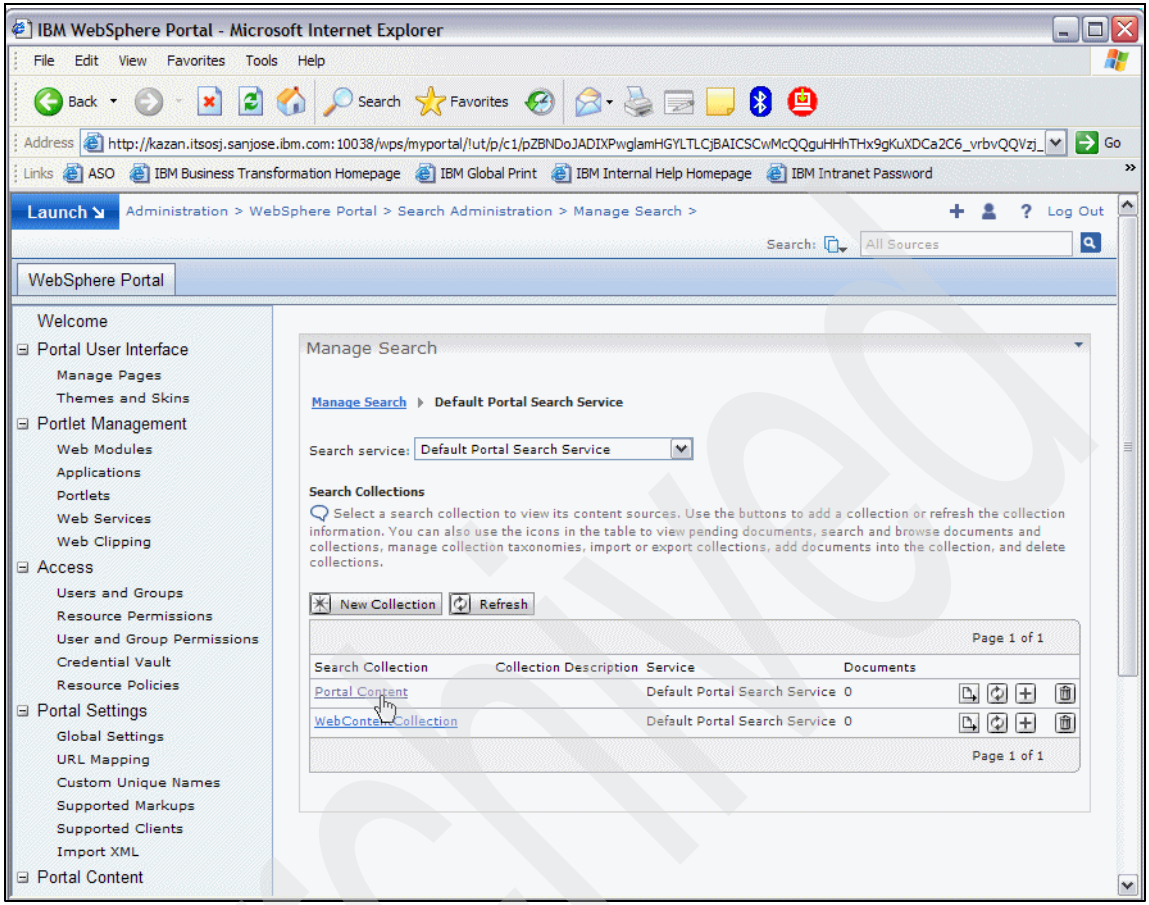

Figure 3-35 Determine the starting URL for the WebSphere Portal crawler 4/6

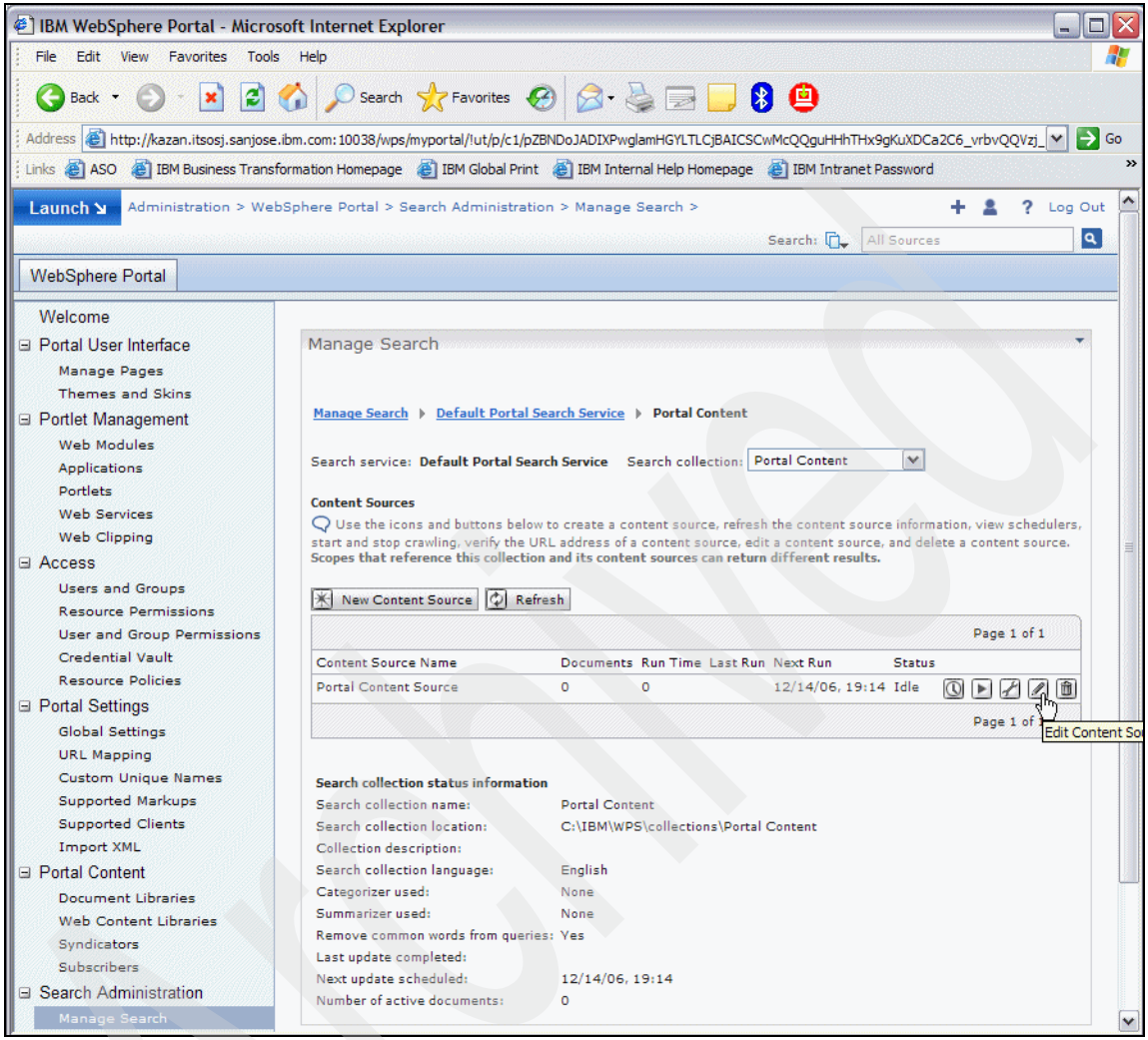

Figure 3-36 Determine the starting URL for the WebSphere Portal crawler 5/6

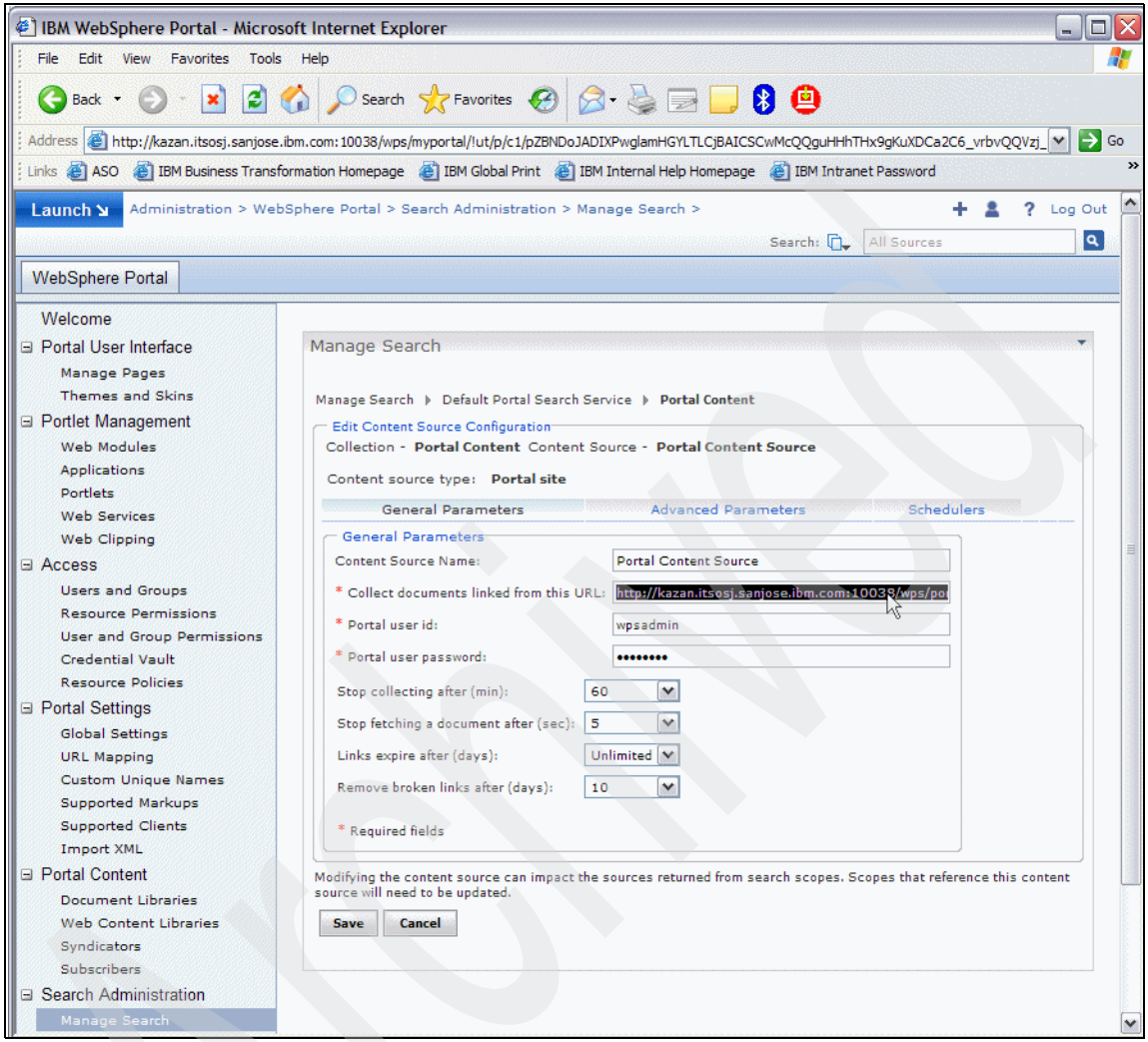

Figure 3-37 Determine the starting URL for the WebSphere Portal crawler 6/6

LSTEP4b: Create CUSTINFO collection

In this step, we create the CUSTINFO collection with the WebSphere Portal and Portal Document Manager crawlers.

Note: A number of parameters can be specified during the creation and configuration of a collection and the associated crawlers. A description of these parameters is beyond the scope of this book. You are encouraged to read the product documentation or invoke **Help** on the GUI for detailed information about these parameters.

Create the collection

After logging in to the GUI administration console as the enterprise search administrator, click the **Collections** view and click **Create Collection**, as shown in Figure 3-38 on page 188. Provide details in Figure 3-39 on page 189 about the collection, such as the Collection name (CUSTINFO), Collection security[2] (Enable security for the collection), Document importance (Rank by the document date[3]), and Categorization type (None) during parsing.

Note: We also chose to explicitly name the Collection ID to be the same as the collection name CUSTINFO. We recommend explicitly specifying this ID rather than let it default to the format col-nnnn, which is difficult to memorize when used in custom applications and command-line tools.

Click **OK** to complete the creation of the collection.

Note: The key point here is to enable security for the collection so that document-level security can be enforced, since this option cannot be changed once the collection is created.

Once the collection is created, we can proceed to the creation of the WebSphere Portal and Content Edition (for the PDM) crawlers, as described in “Create and configure the crawlers” on page 190.

[2] Required for enforcing document-level security.

[3] This is normally used for NNTP crawlers. We chose this option here to demonstrate this functionality.

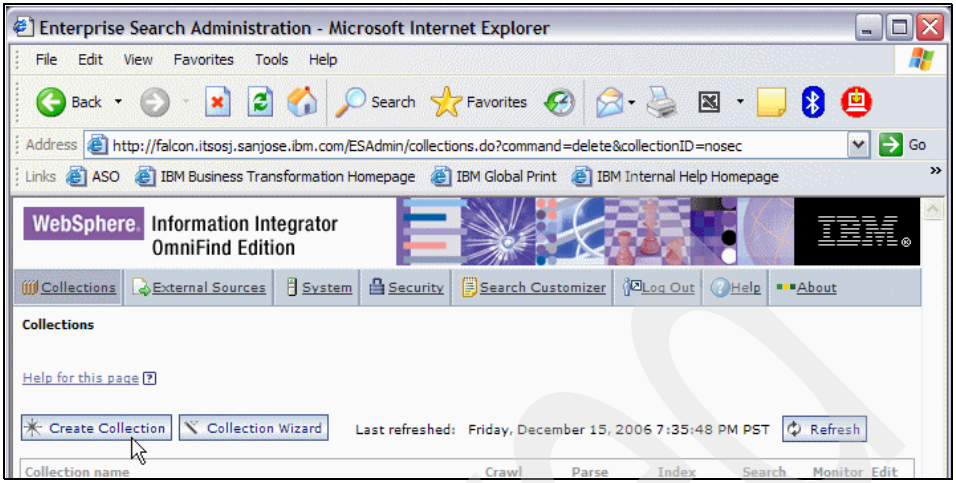

Figure 3-38 Create Collection

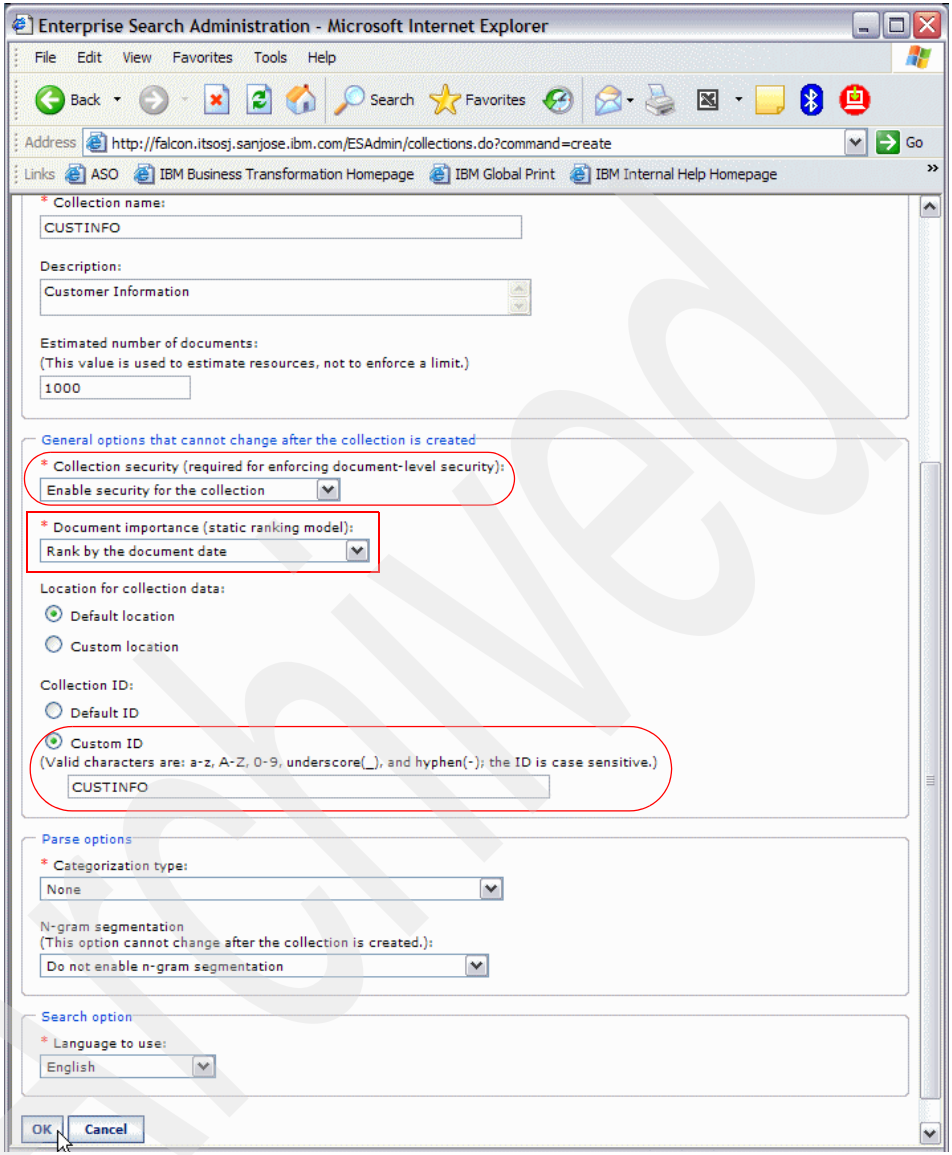

Figure 3-39 CUSTINFO collection details

Create and configure the crawlers

In this step, the WebSphere Portal and Content Edition crawlers are defined with the appropriate security configurations.

- Create and configure the WebSphere Portal crawler.

  Figure 3-40 on page 191 through Figure 3-56 on page 205 describe the creation and configuration of the WebSphere Portal crawler.

  After logging in to the administration console, select the **Collections** view and click the **Crawl** icon, as shown in Figure 3-40 on page 191. In the following window (Figure 3-41 on page 192), switch to Edit mode by clicking the **Edit** icon. From the Crawl tab in Figure 3-42 on page 192, click **Create Crawler**. Select WebSphere Portal for Crawler type and click **Next** in Figure 3-43 on page 193.

  Provide details of the WebSphere Portal crawler in Figure 3-44 on page 194, such as the Crawler name (portal) and Maximum number of documents to crawl (2000). Click **Next** to provide further details in Figure 3-45 on page 195, such as the URL of the WebSphere Portal site (http://kazan.itsosj.sanjose.ibm.com:10038/wps/portal/....)[4] to be crawled, which was determined in "Determine the URL for WebSphere Portal crawler" on page 179, and the credentials for the crawler to authenticate to WPS documents in the User DN (uid=esadmin,cn=users,ou=itso,o=ibm) and Password fields. Click **Edit document-level security** in Figure 3-45 on page 195 to view, in Figure 3-46 on page 196, the default options, which are Validate current credentials during query processing, and Index page and associated portlet access control lists. Click **OK** to accept the defaults.

  Since we want to leverage WebSphere Portal support for SSO, click **Specify the SSO authentication type** in Figure 3-47 on page 197 to configure the required information to access WPS documents that are protected by single sign-on (SSO) security, as described in Figure 3-48 on page 198 through Figure 3-53 on page 203. Select **Form-based authentication** from the drop-down list for SSO authentication type and the Login form URL[5] in Figure 3-48 on page 198. Click **Add Field** to identify each field in the form that the crawler must provide to be authenticated; in this case, it is the user ID and password fields with the corresponding values, as shown in Figure 3-49 on page 199 through Figure 3-52 on page 202. Click **OK** in Figure 3-52 on page 202 and **Next** in Figure 3-53 on page 203 to proceed to test the success of the SSO authentication configuration.

---

[4] Since this URL can be long and include encoded non-ASCII characters, you should copy the URL from the WebSphere Portal server and paste it here.

[5] Obtained by entering the URL `http://kazan.itsosj.sanjose.ibm.com:10038/wps/myportal` in a Web browser, which displays the Login Form with a slightly changed URL; this modified URL should be copied and pasted into the field.

Note: In the case of the standard LTPA configuration, it is not necessary to configure SSO support for WebSphere Portal crawling. We did so here to demonstrate the process involved.

The only time that Form Based Authentication (FBA) with SSO support configuration is required is when you secure your portal server with Siteminder with FBA or Tivoli Access Manager or some other type of external authentication that blocks access to the portal server.

Click **Test the configuration** in Figure 3-54 on page 204 to test the crawler's ability to connect to the WebSphere Portal URL specified. A successful connection results in the "FFQM0350I Success..." shown in Figure 3-55 on page 204. Click **Next** to specify the crawl schedule.

Since we are going to manually schedule this crawler, we chose the defaults and clicked **Finish** in Figure 3-56 on page 205 to complete the configuration of the WebSphere Portal Server.

We can now proceed to create and configure the Content Edition crawler on "Create and configure the Content Edition crawler." on page 206.

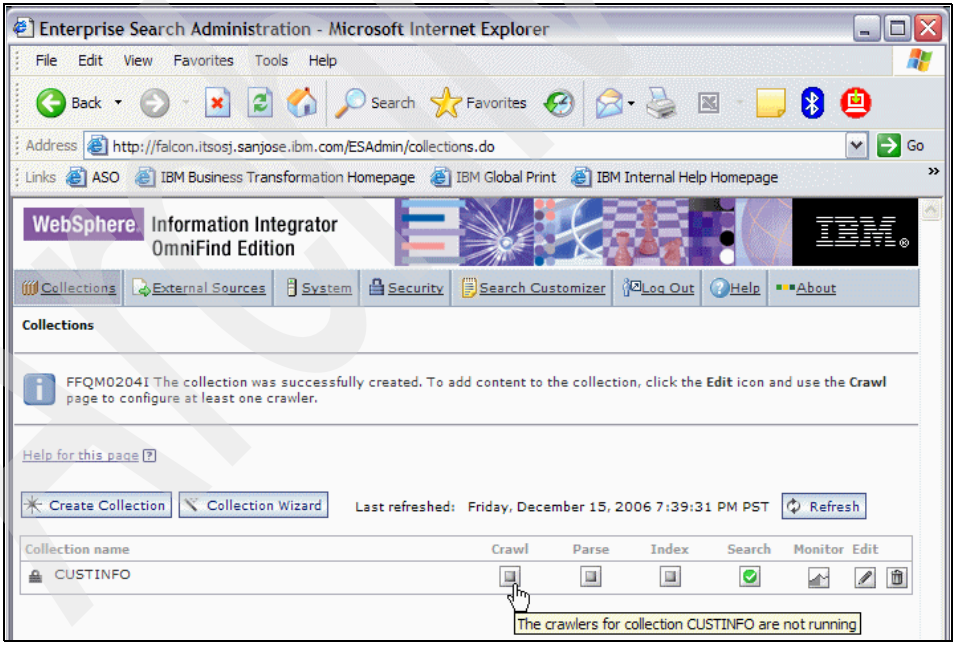

Figure 3-40 Click Crawl icon

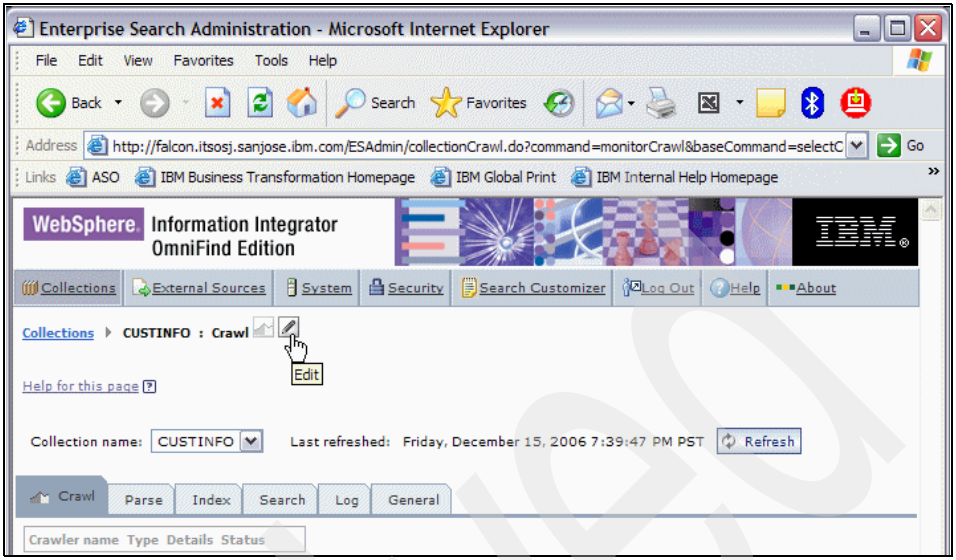

Figure 3-41 Click Edit icon

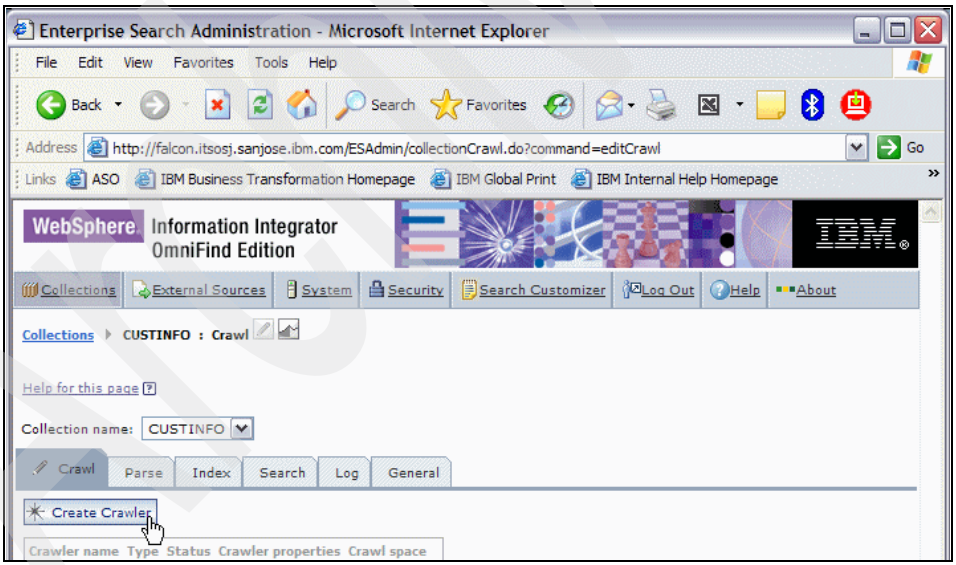

Figure 3-42 Create Crawler

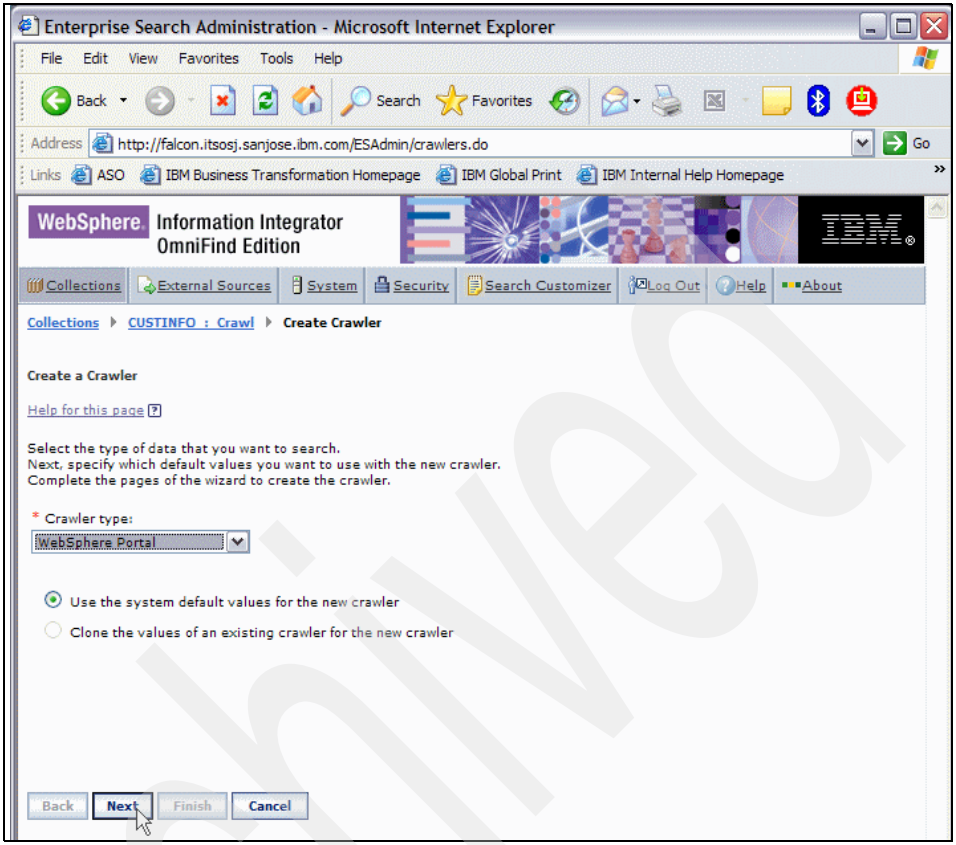

Figure 3-43 WebSphere Portal crawler type

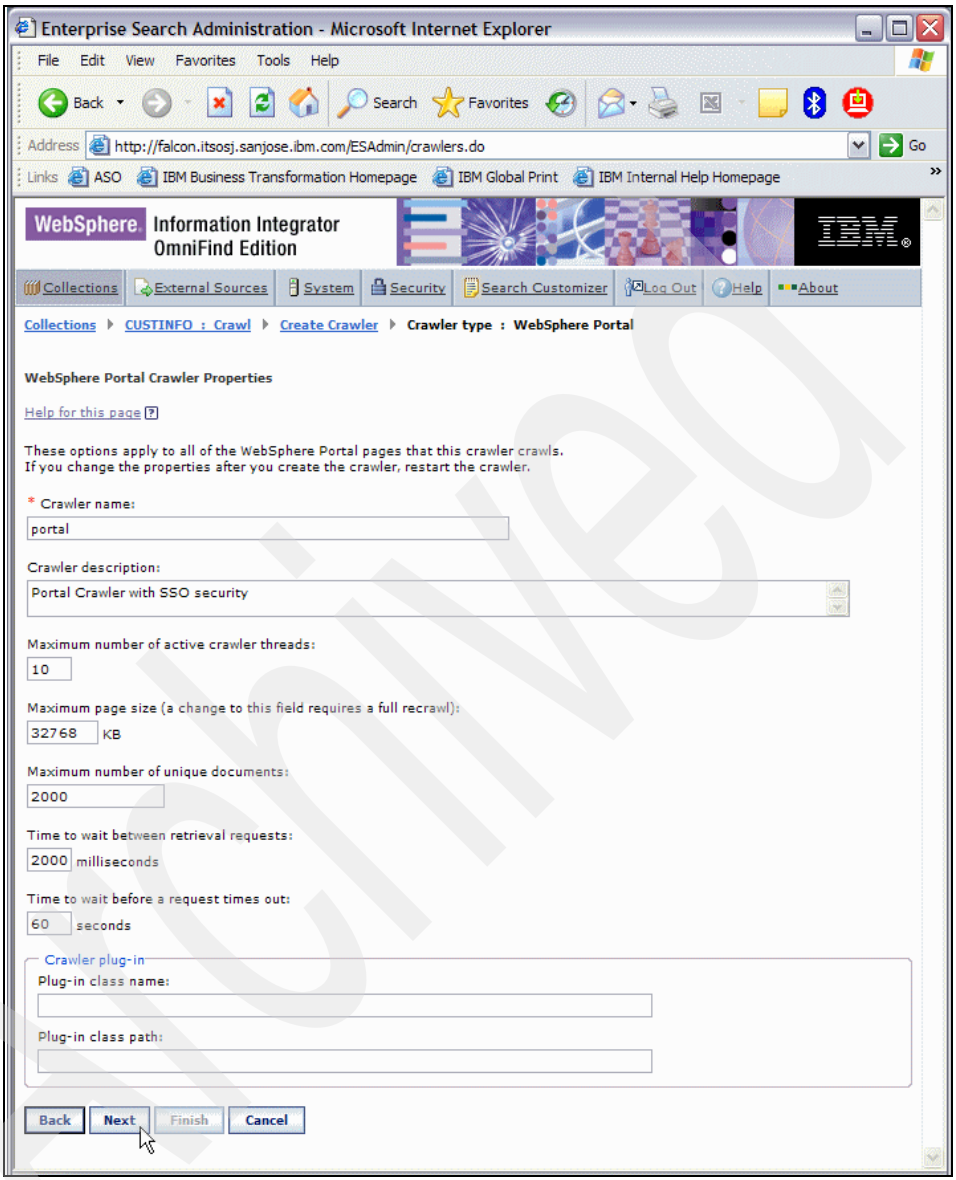

Figure 3-44 Crawler details 1/2

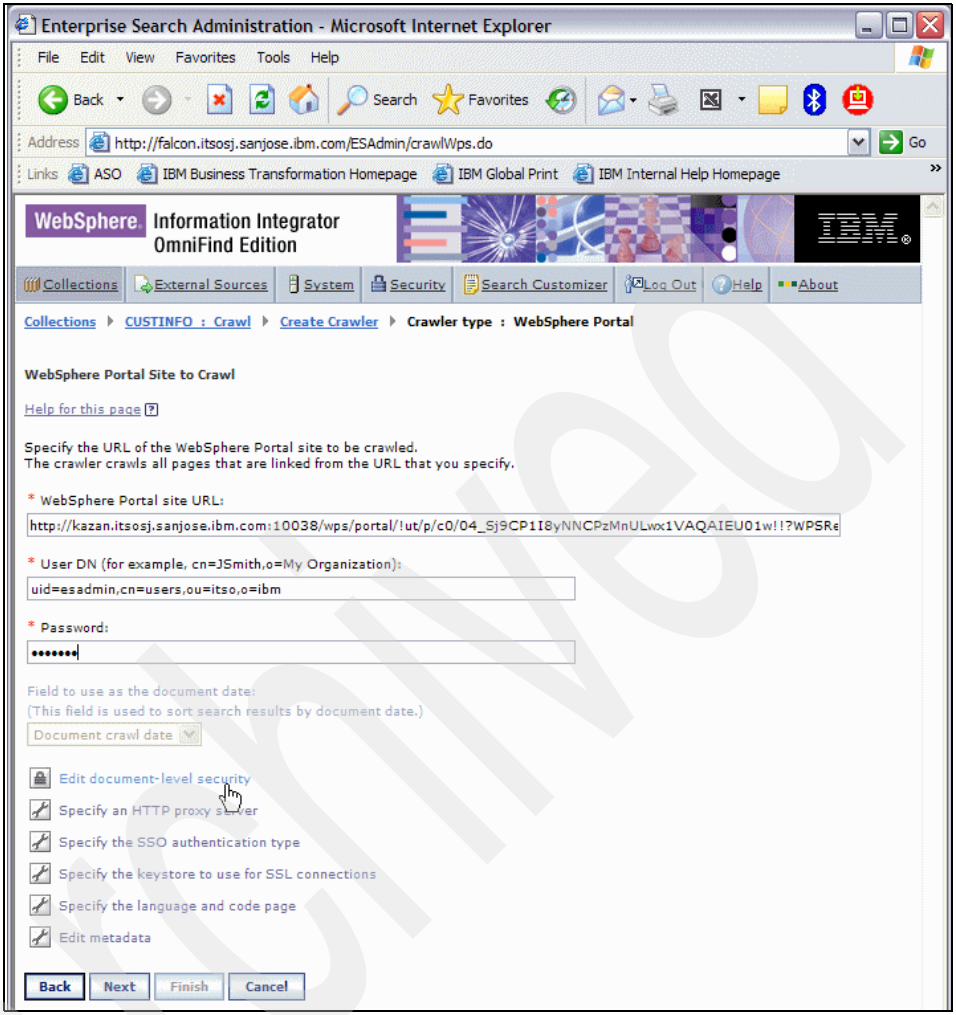

Figure 3-45 Crawler details 2/2

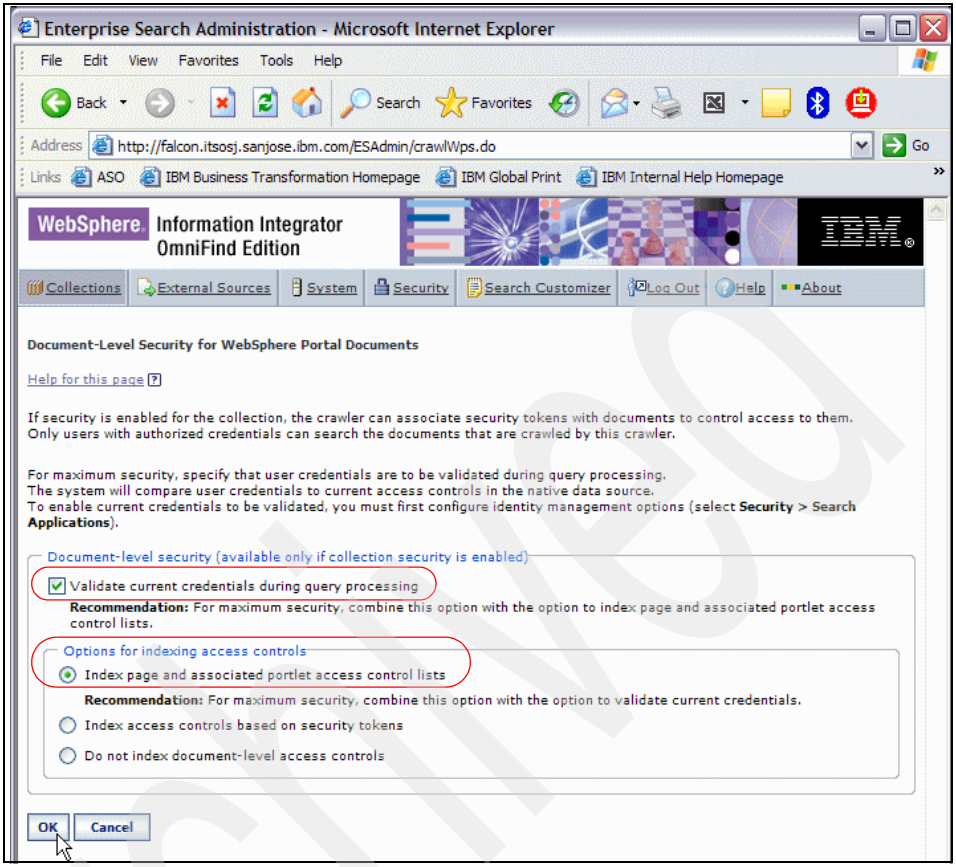

Figure 3-46 Document-Level Security for WebSphere Portal Documents

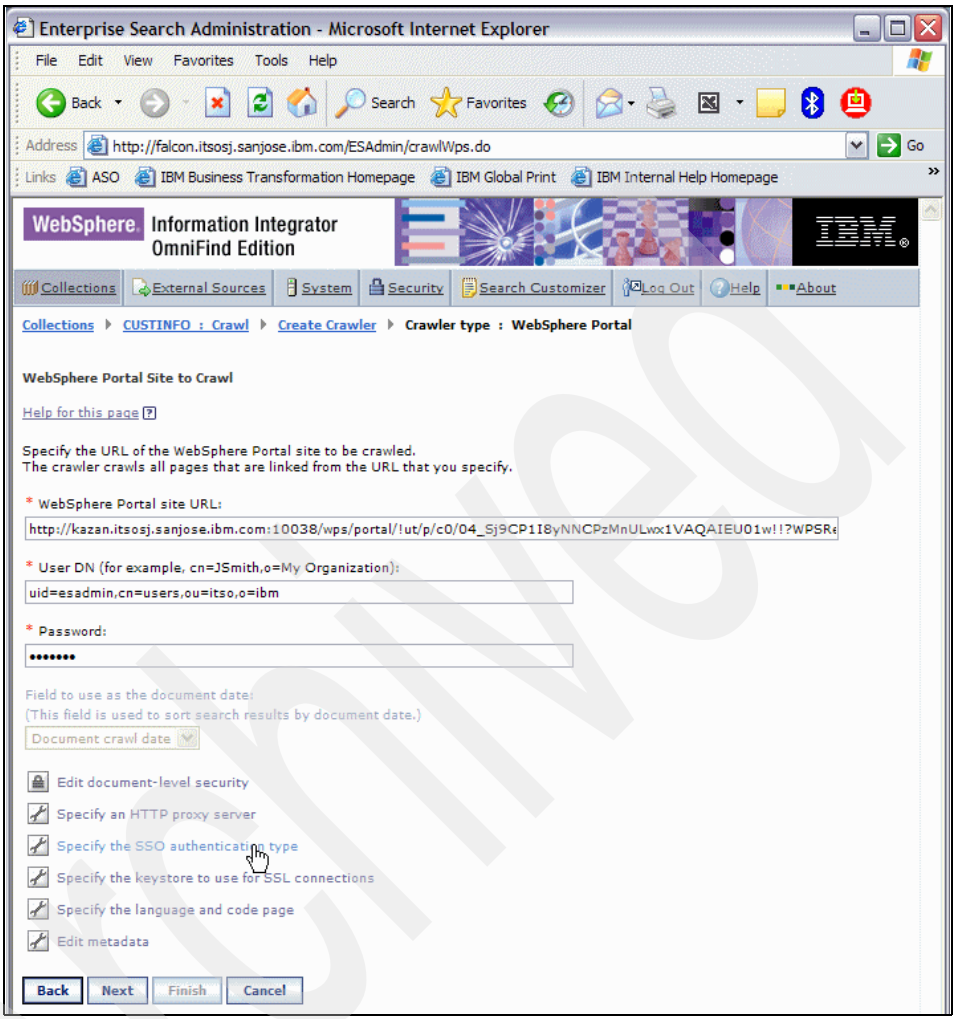

Figure 3-47 Specify the SSO authentication type 1/7

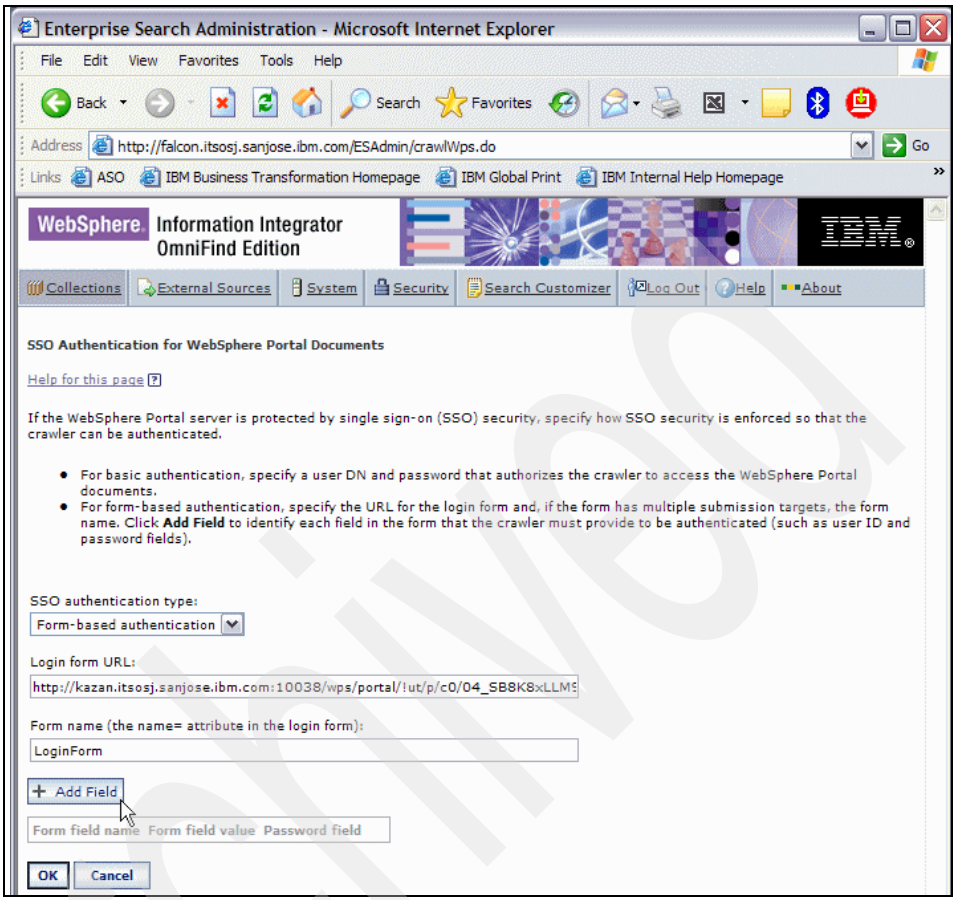

Figure 3-48 Specify the SSO authentication type 2/7

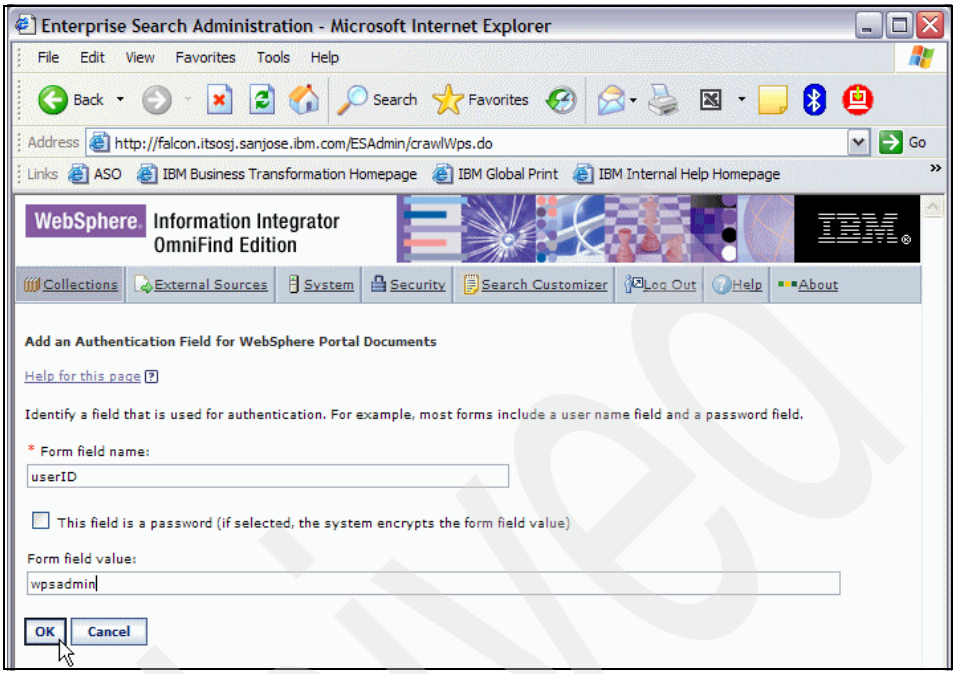

Figure 3-49 Specify the SSO authentication type 3/7

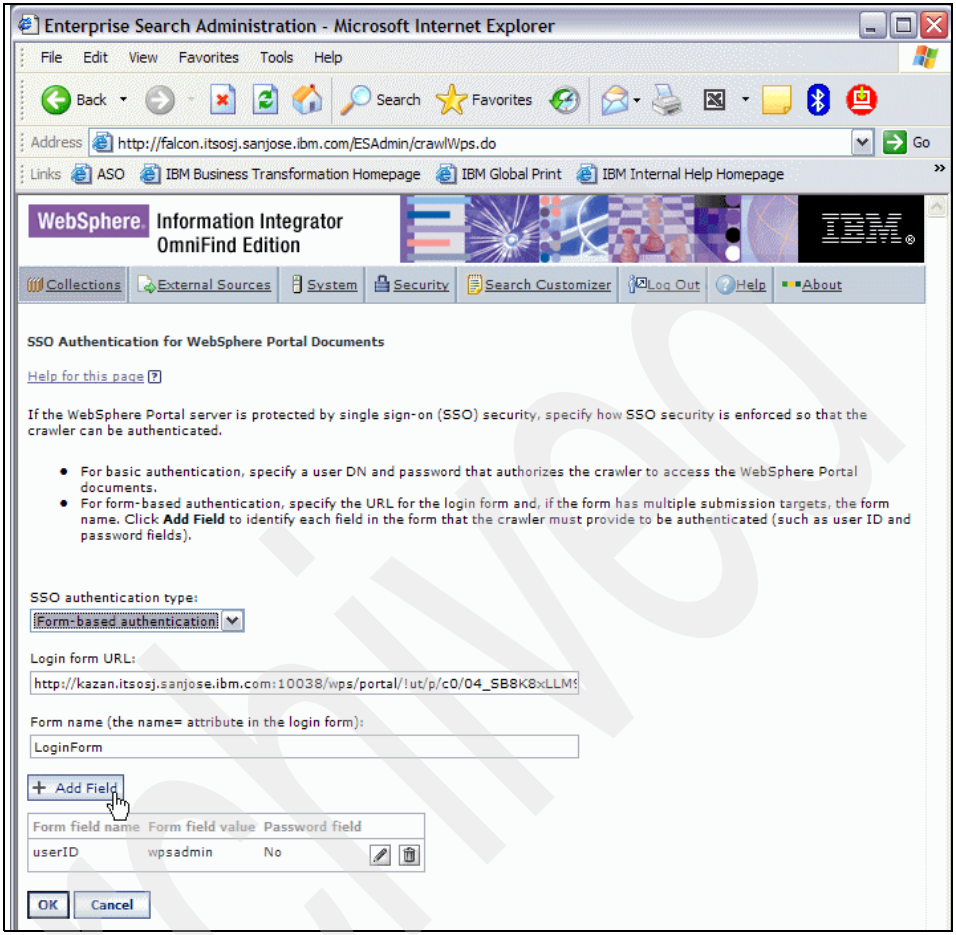

Figure 3-50 Specify the SSO authentication type 4/7

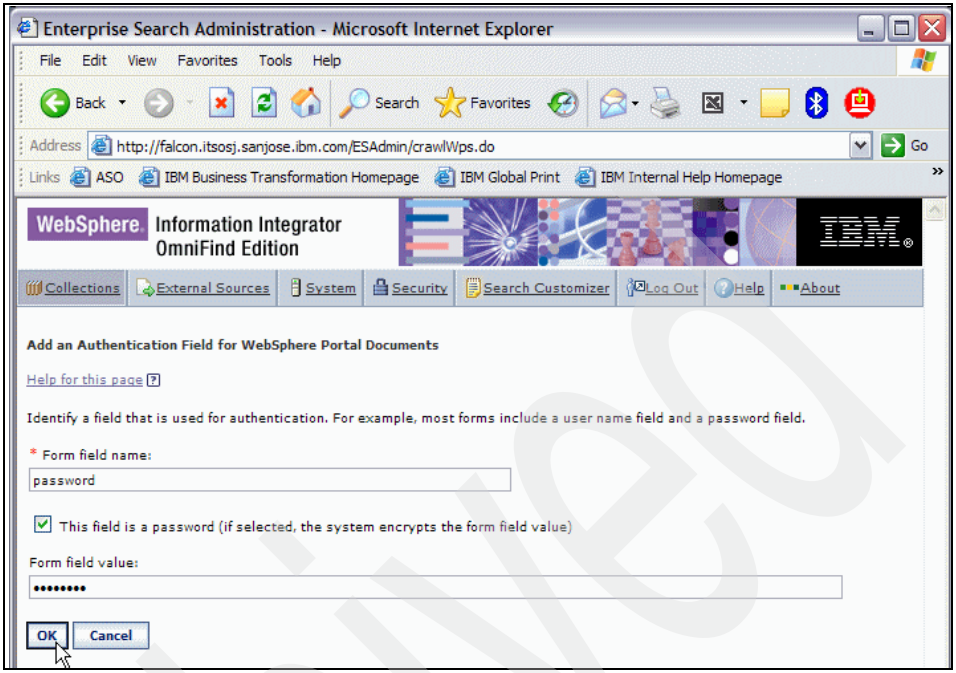

Figure 3-51 Specify the SSO authentication type 5/7

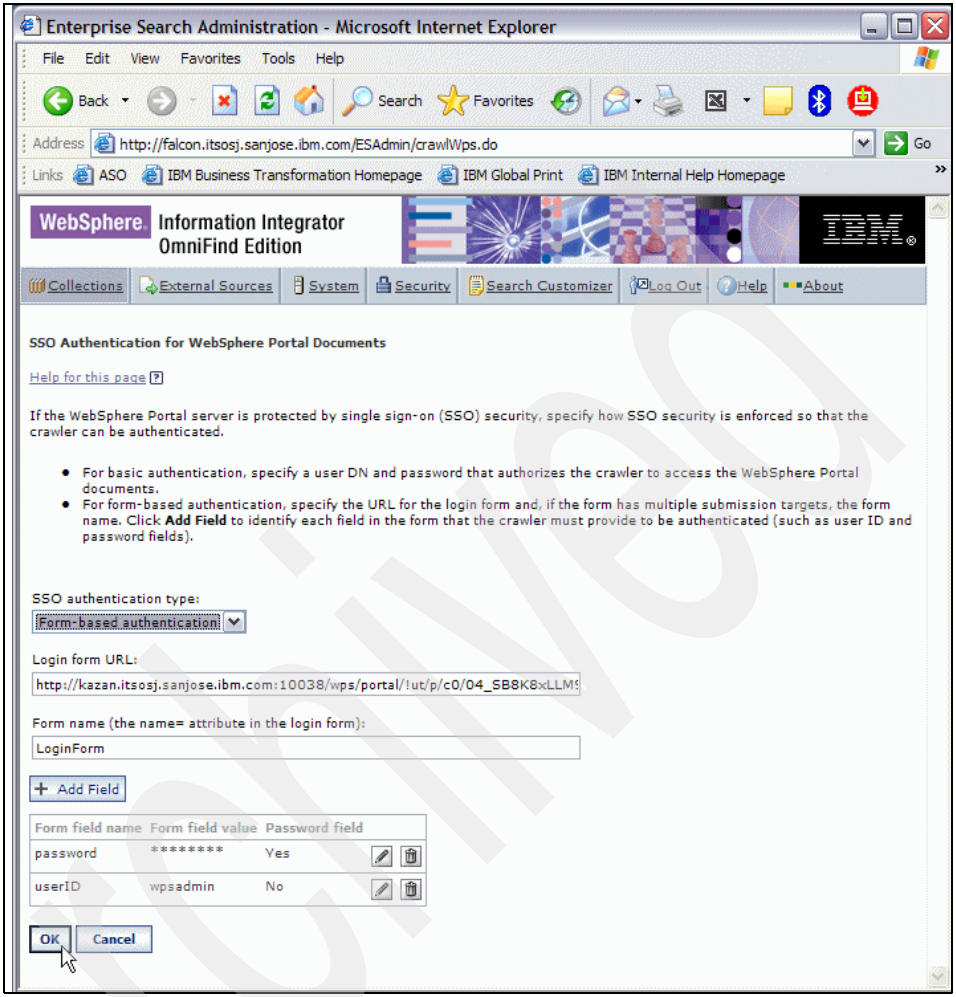

Figure 3-52 Specify the SSO authentication type 6/7

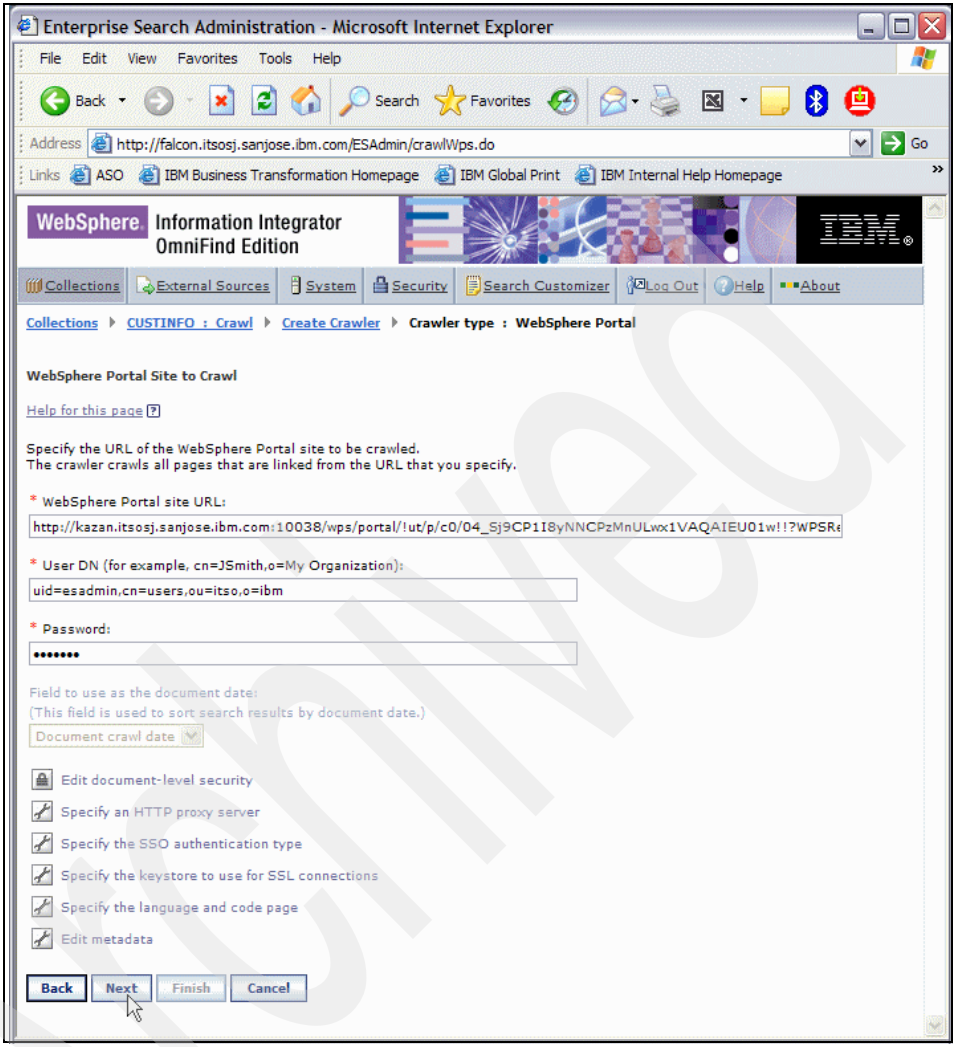

Figure 3-53 Specify the SSO authentication type 7/7

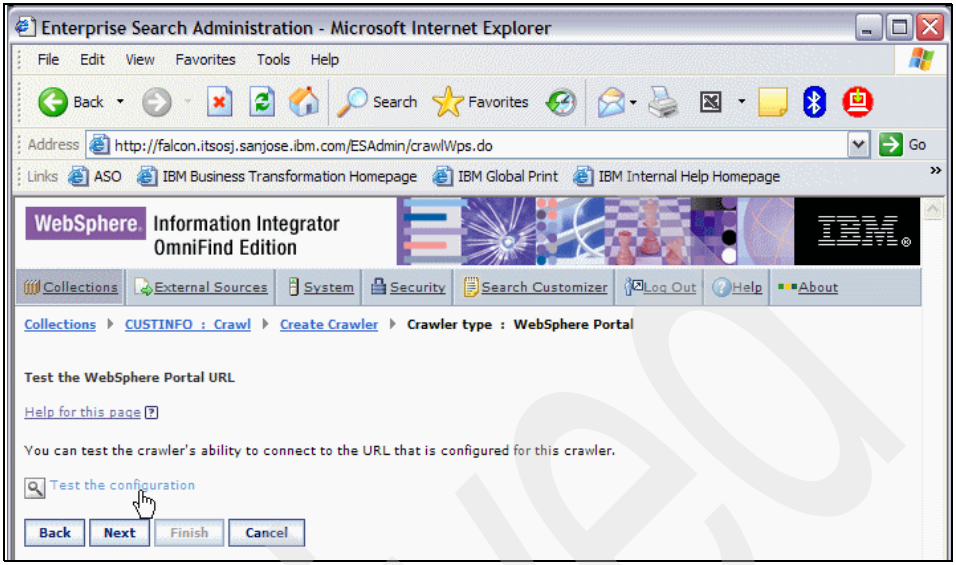

Figure 3-54 Test the configuration 1/2

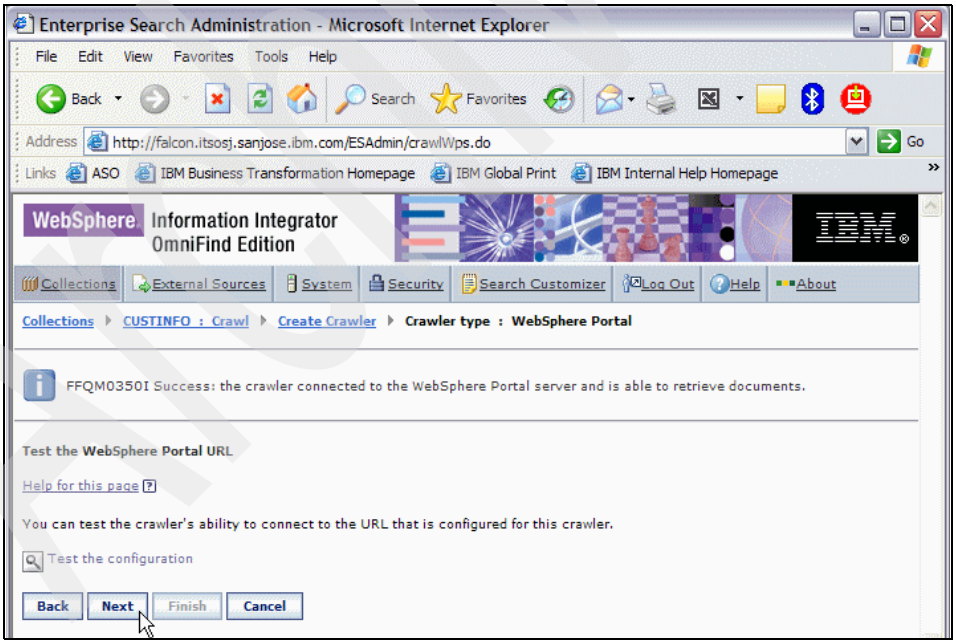

Figure 3-55 Test the configuration 2/2

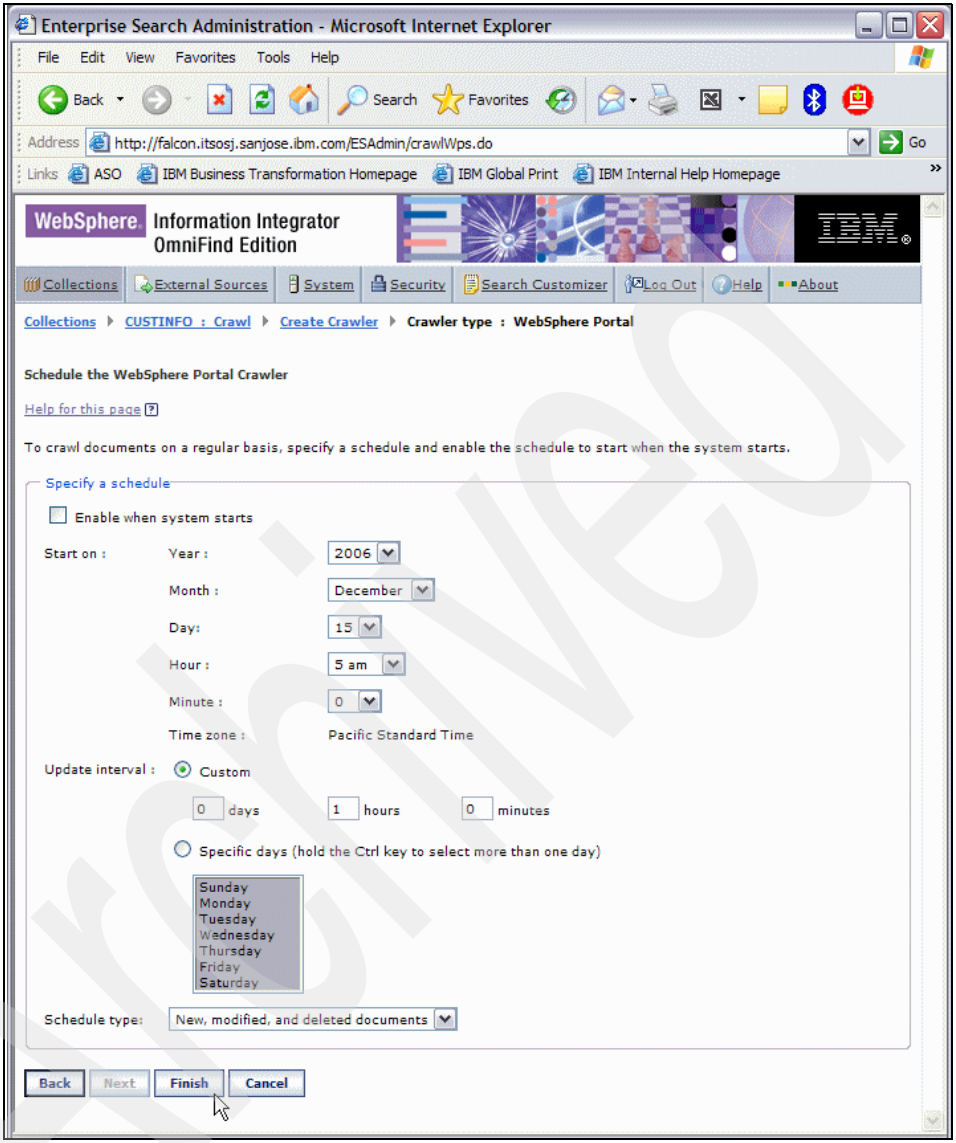

Figure 3-56 Crawl schedule

- Create and configure the Content Edition crawler.

  Figure 3-57 on page 207 through Figure 3-66 on page 216 describe the creation and configuration of the Content Edition crawler.

  After logging in to the administration console, navigate to the **Collections** view in Edit mode, and from the Crawl tab in Figure 3-57 on page 207, click **Create Crawler**. Select **Content Edition Crawler type** and click **Next** in Figure 3-58 on page 208.

  Provide details of the Content Edition crawler in Figure 3-59 on page 209, such as the Crawler name (pdm) and Maximum number of documents to crawl (2000). Click **Next** to specify the Content Edition access mode. Select **Direct mode (access repositories through a connector on the crawler server)** in Figure 3-60 on page 210 and click **Next** to select the Content Edition repositories to crawl**.**

  Figure 3-61 on page 211 shows the selected Repositories to crawl (IBM WebSphere Portal Document Manager Connector Portal Document Manager) obtained by first discovering available repositories ('*' in the Repository name or pattern followed by clicking **Search for repositories**, which lists all those found with the matching criteria in the Available repositories box and then copying those of interest to the Repositories to crawl box). Click **Next** in Figure 3-61 on page 211 to specify Content Edition Repository User IDs (and password) to access the selected repository, as shown in Figure 3-62 on page 212. The Single sign-on security (SSO) drop-down list has Not enabled for SSO selected. Click **Next** to specify the crawl schedule (Figure 3-63 on page 213). Since we chose to schedule the crawls manually, click **Next** in Figure 3-63 on page 213. The next step is to identify all the item classes to be crawled. Figure 3-64 on page 214 shows the selected Item Classes to crawl [lotus:collaborativeDocument(Content) and icm:documentLibrary[6](Folder)] obtained by first discovering available item classes ('*' in the Item class name or pattern followed by a click of **Search for item classes**, which lists all those found with the matching criteria in the Available item classes box and then copying those of interest to the Item classes to crawl box). Click **Next** in Figure 3-64 on page 214 to view and optionally modify the Edit options and Edit security parameters for the selected item classes to crawl in Figure 3-65 on page 215. Click **Finish** to view the CUSTINFO collection's crawlers and their status, as shown in Figure 3-66 on page 216.

  We can now proceed to create the GENINSINFO collection, as described in "LSTEP4c: Create GENINSINFO collection" on page 216.

[6] This is not required, but there is no harm done by specifying it.

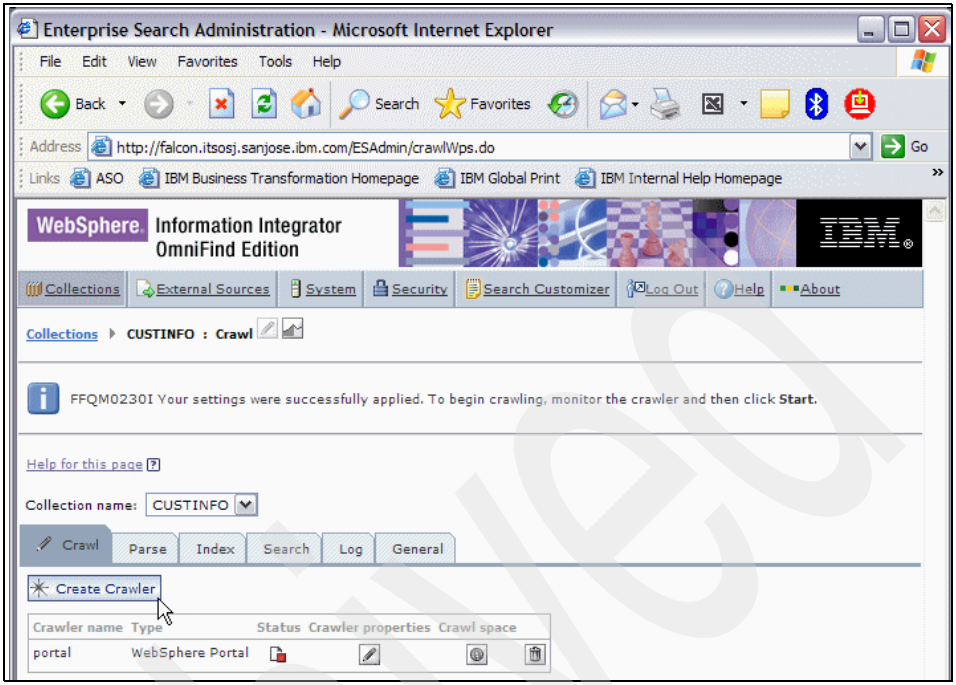

Figure 3-57 Create Crawler

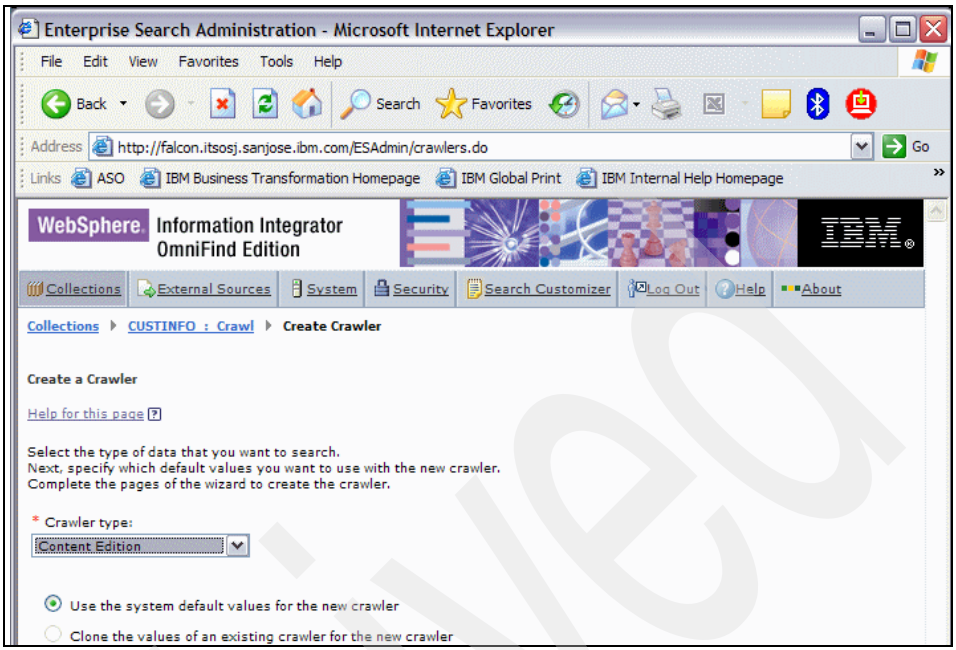

Figure 3-58 Content Edition crawler type

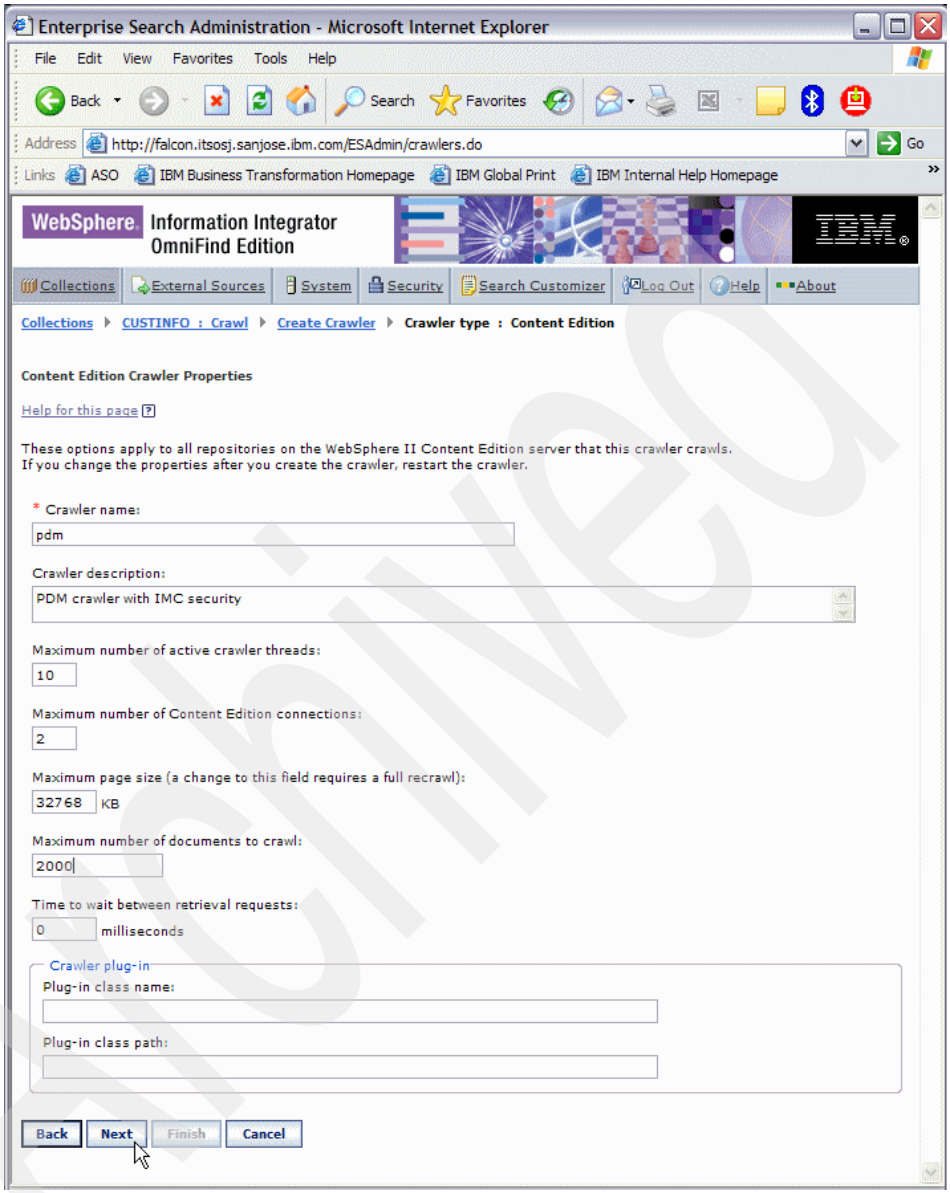

Figure 3-59 Crawler details 1/2

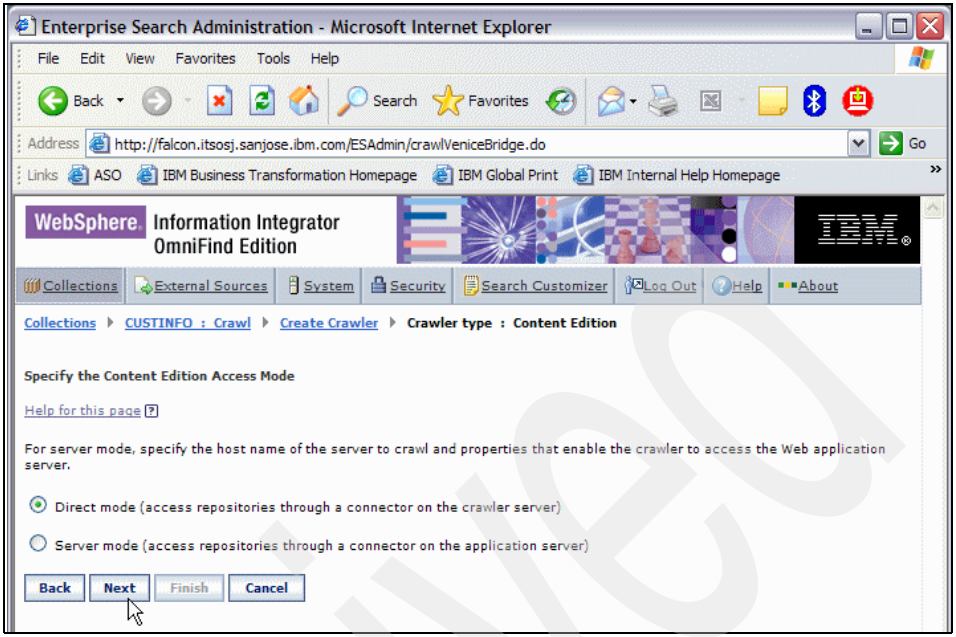

Figure 3-60 Crawler details 2/2

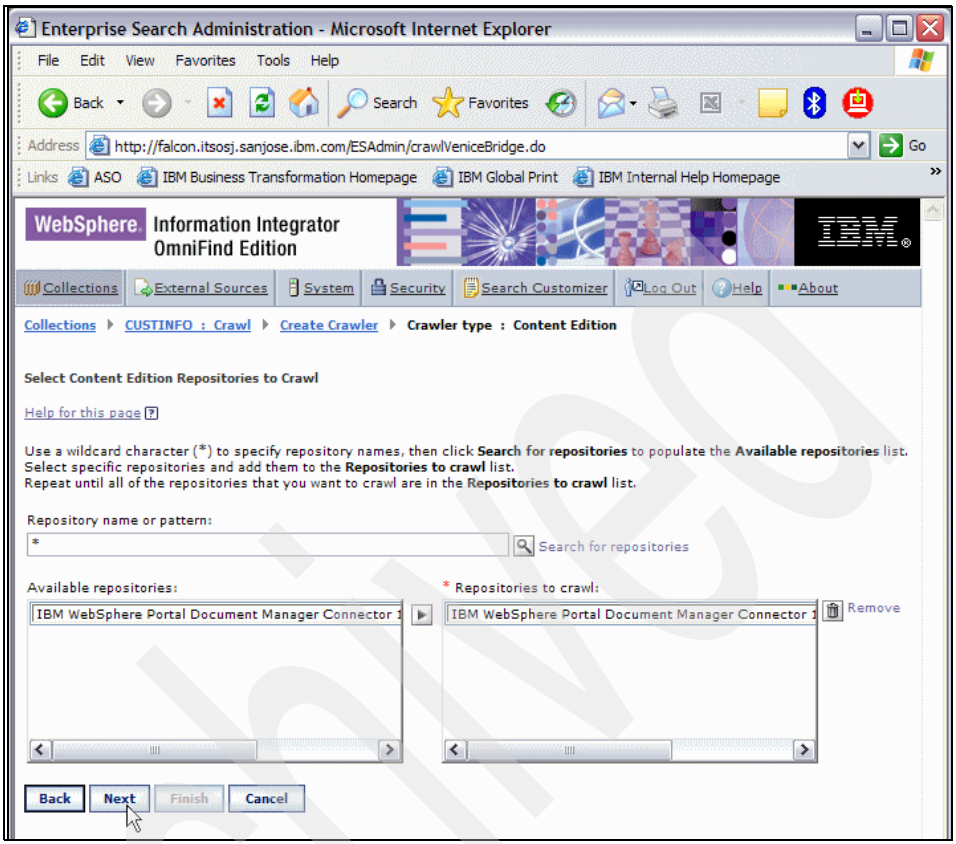

Figure 3-61 Content Edition Repositories to Crawl

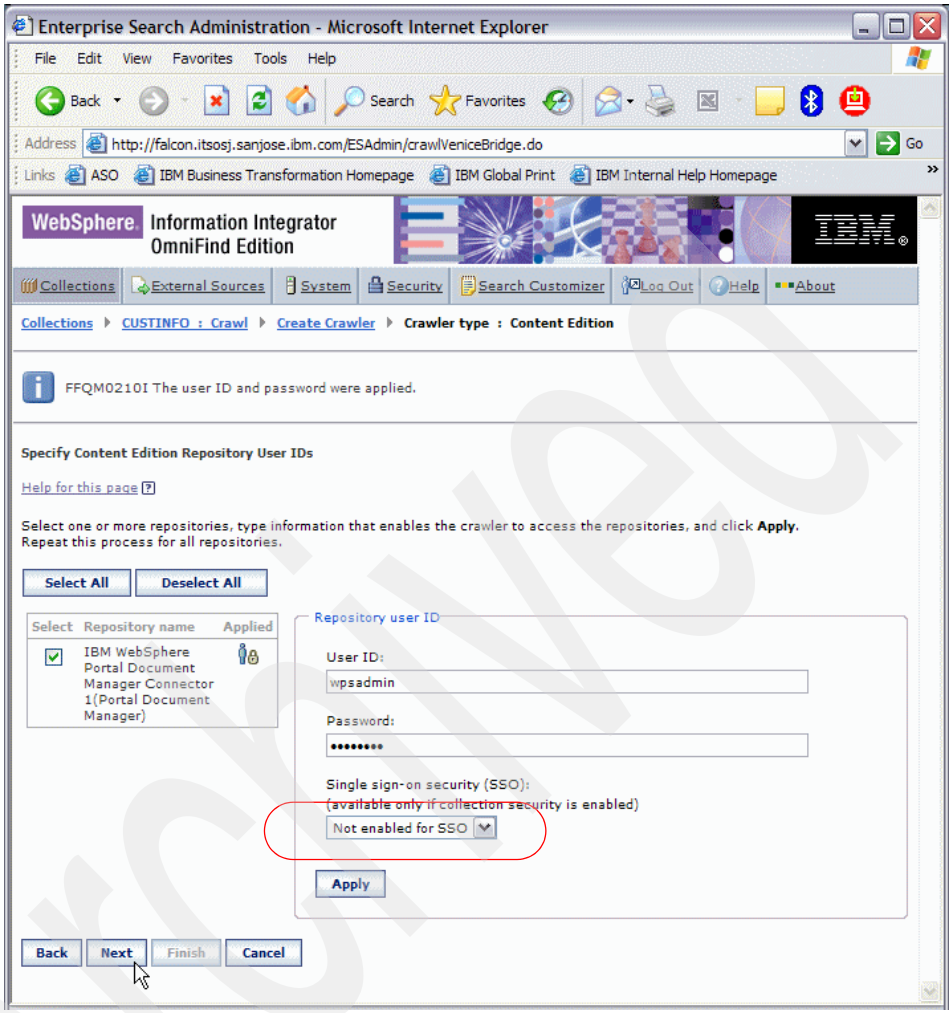

Figure 3-62 Specify Content Edition Repository User IDs

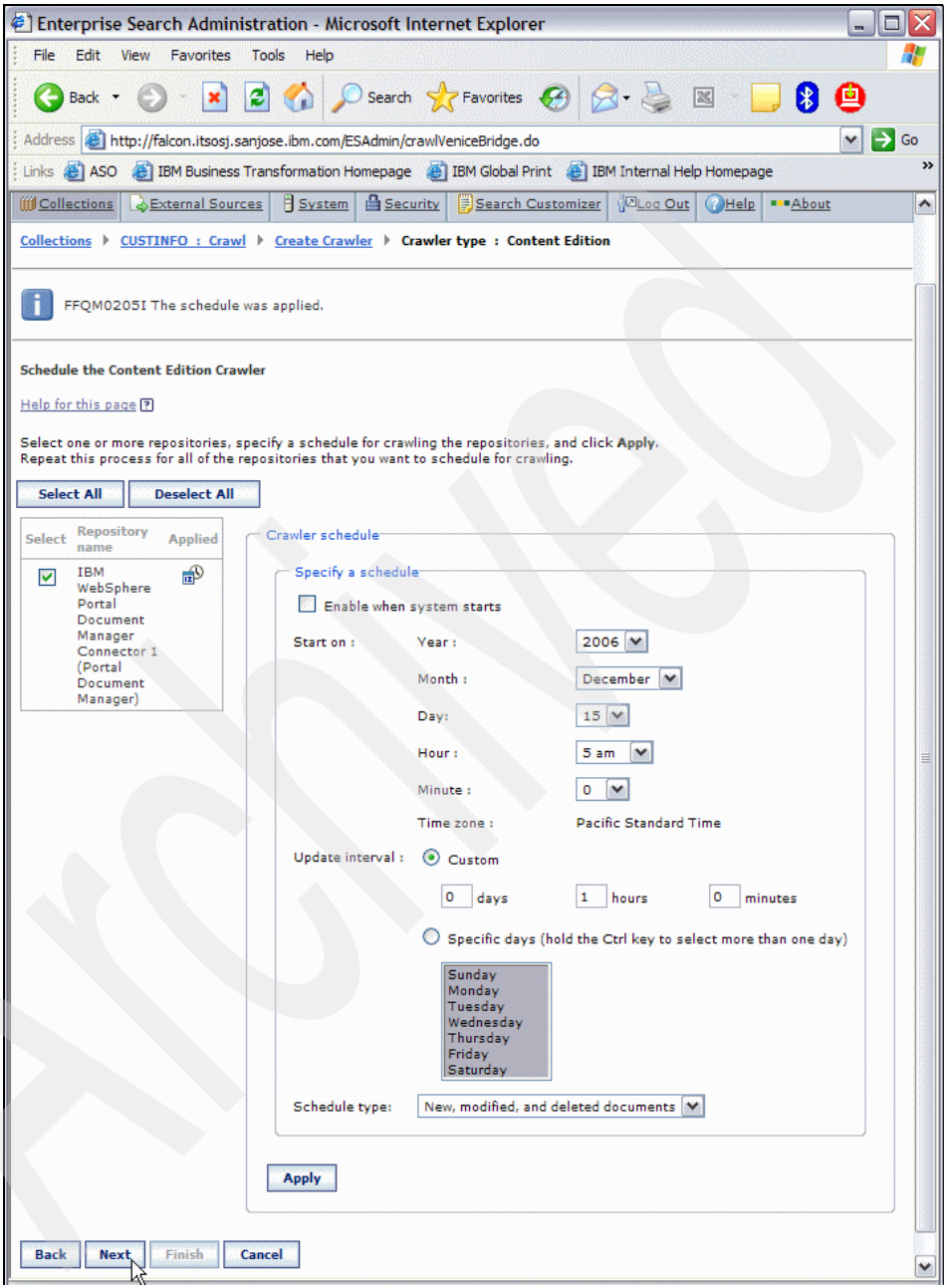

Figure 3-63 Crawl schedule

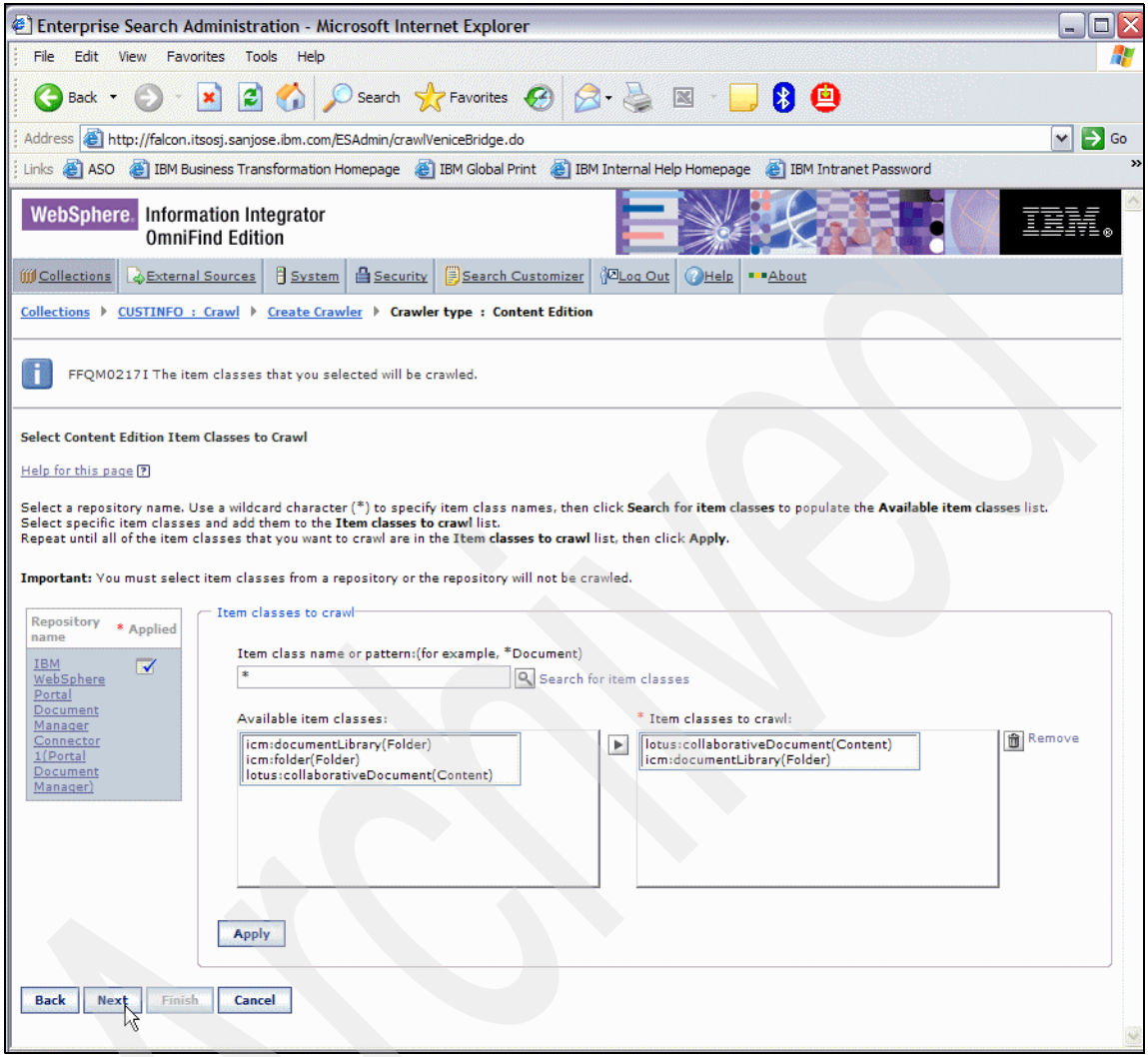

Figure 3-64 Content Edition Item Classes to Crawl

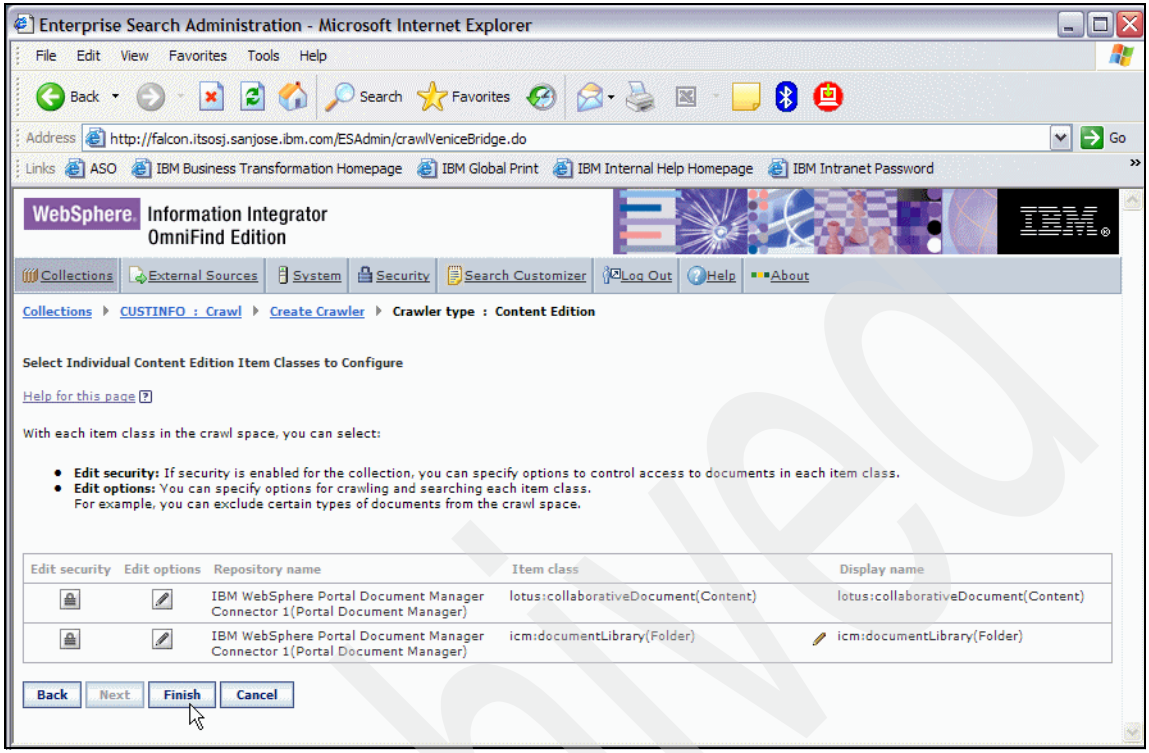

Figure 3-65 Item classes selected

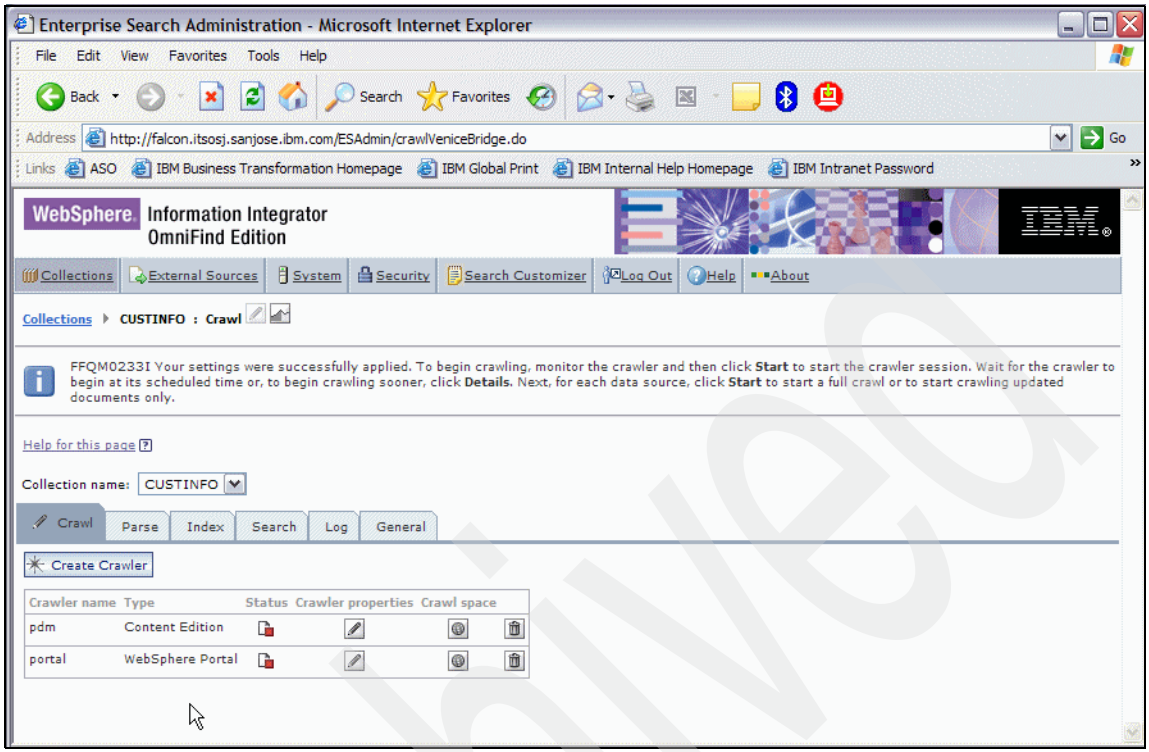

Figure 3-66 Crawlers in the CUSTINFO collection

LSTEP4c: Create GENINSINFO collection

In this step, we create the GENINSINFO collection with the Web crawler that crawls the Web Content Management data source.

Note: A number of parameters can be specified during the creation and configuration of a collection and the associated crawlers. A description of these parameters is beyond the scope of this book. You are encouraged to read the product documentation or invoke **Help** on the GUI for detailed information about these parameters.

Create the collection

After logging in to the GUI administration console as the enterprise search administrator, click the **Collections** view and click **Create Collection**, as shown in Figure 3-67 on page 217. Provide details in Figure 3-68 on page 218 about the collection, such as the Collection name (GENINSINFO), Collection security (Do not enable security for the collection), Document importance (Rank by the document date), and Categorization type (None) during parsing.

Note: We also chose to explicitly name the Collection ID to be the same as the collection name GENINSINFO. We recommend explicitly specifying this ID rather than let it default to the format col-nnnn, which is difficult to memorize when used in custom applications and command-line tools.

Click **OK** to complete the creation of the collection and proceed to the creation of the Web crawler, as described in “Create and configure the crawlers” on page 219.

Note: The key point here is to not enable security for the collection, since we want all the documents in this collection to be available to all employees within the organization.

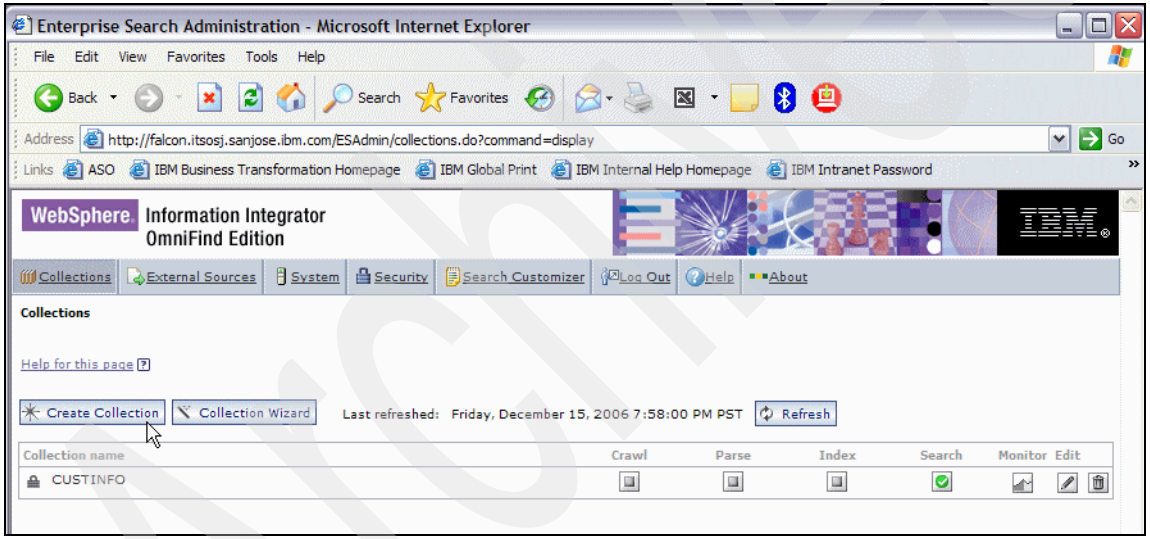

Figure 3-67 Create Collection

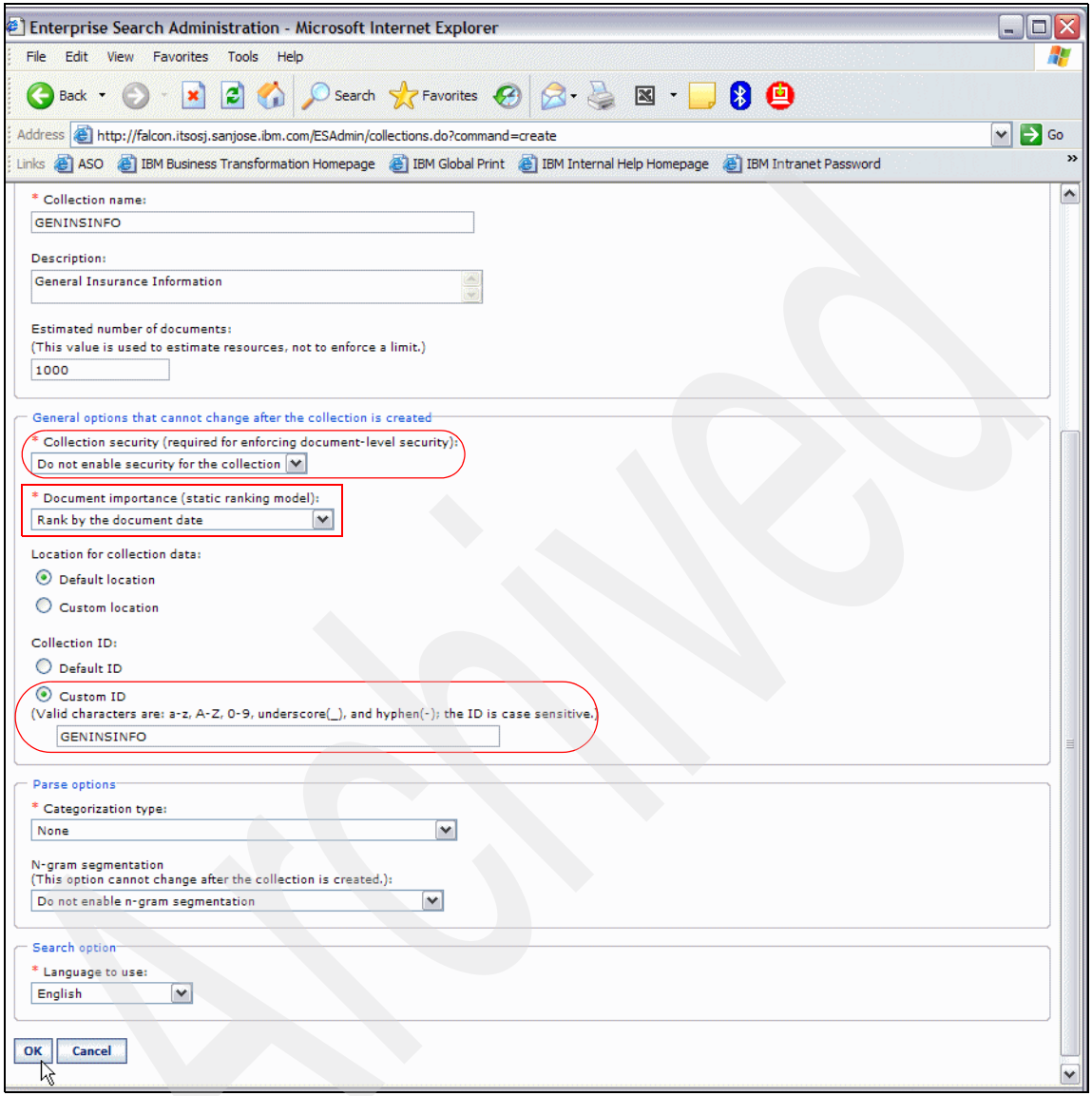

Figure 3-68 Collection details

Create and configure the crawlers

In this step, we define and configure the Web crawler to access the WCM data source.

Figure 3-69 on page 220 through Figure 3-76 on page 226 describe the creation and configuration of the Web crawler.

After logging in to the administration console, navigate to the **Collections** view in Edit mode, and from the Crawl tab in Figure 3-69 on page 220, click **Create Crawler**. Select **Web Crawler type** and click **Next** in Figure 3-70 on page 220.

Provide details of the Web crawler in Figure 3-71 on page 221, such as the Crawler name (wcm), E-mail address for comments about the crawler (crawleradmin@sequoia.com) to be sent, and User agent (crawler)[7]. Click **Next** to provide further details in Figure 3-72 on page 222, such as the URL of the Web site (http://kazan.itsosj.sanjose.ibm.com:10038/wps/wcm/connect/web+content/insurance/definitions/menu_content) to be crawled, which was determined in "Configure WCM for the Web crawler" on page 164. Click **Next** to apply rules to crawl domains and HTTP Prefixes. Figure 3-73 on page 223 limits the crawling to only the http://kazan.itsosj.sanjose.ibm.com:10038 domain and nothing else (established with the forbid domain *). Figure 3-74 on page 224 limits the crawling of the prefixes listed.Click **Next** to proceed to test the crawler's ability to to connect to the URLs with the user agent configured.

Select **Test the start URLs** button and click **Test** in Figure 3-75 on page 225 to test the crawler's ability to connect to the starting URL specified. A successful connection is indicated, as shown in Figure 3-76 on page 226. Click **Next** to complete the configuration of the Web crawler.

We can now proceed to crawl the WCM data source, as described in create and configure the Content Edition crawler in "LSTEP4d: Crawl GENINSINFO data sources" on page 226.

[7] You need to ensure that the robots.txt file on the Web site to be crawled allows the user agent configured here to crawl the Web site. If the robots.txt file does not exist, the Web site is open to unrestricted crawling. If the robots.txt file exists, it specifies what areas of the site (directories) are off limits to the crawler. The robots.txt file specifies permissions for crawlers by identifying their user agent name.

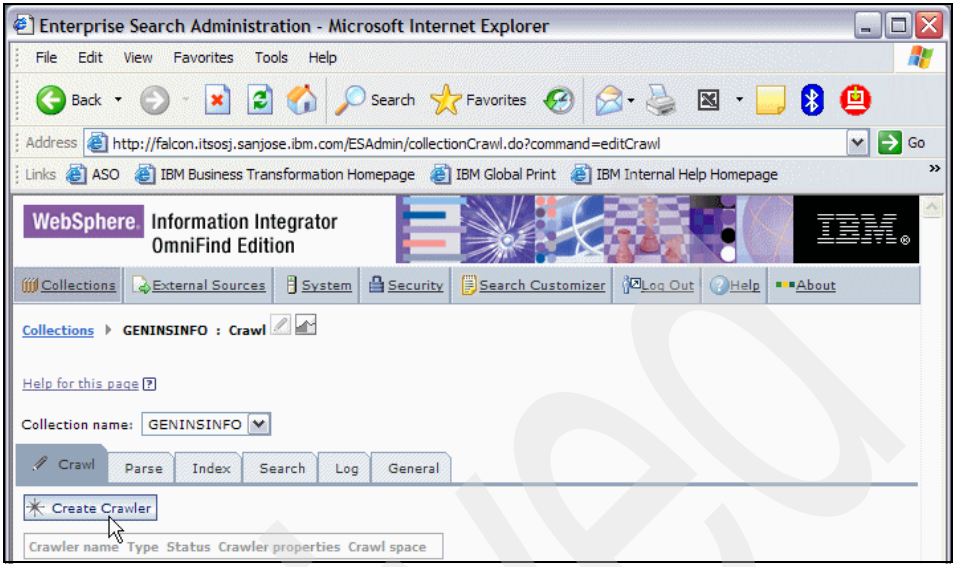

Figure 3-69 Create Crawler

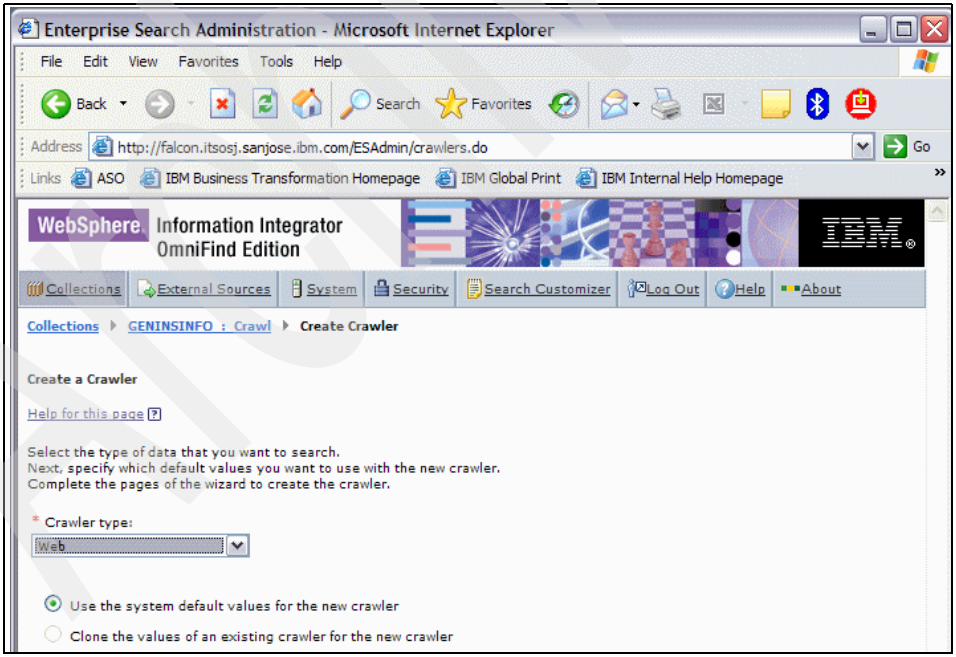

Figure 3-70 Web crawler type

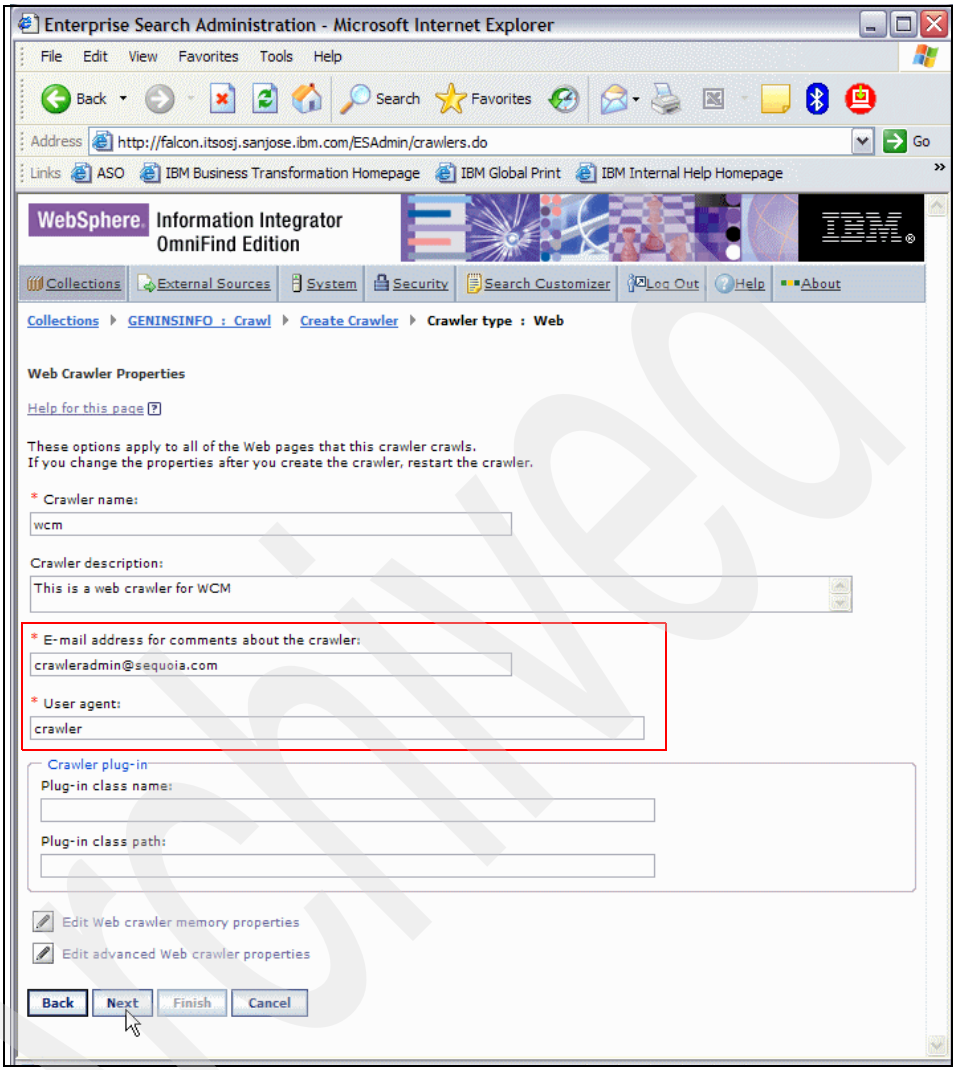

Figure 3-71 Crawler details 1/2

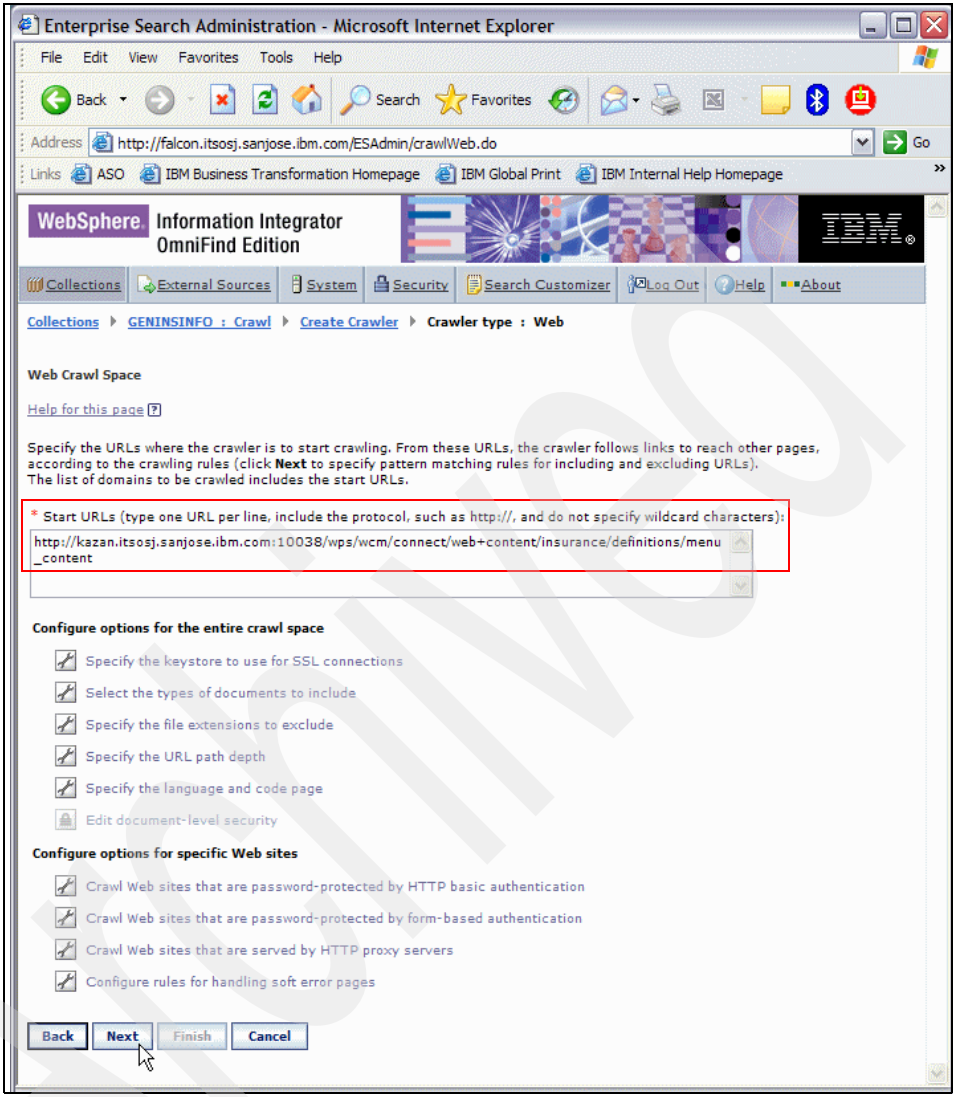

Figure 3-72 Crawler details 2/2

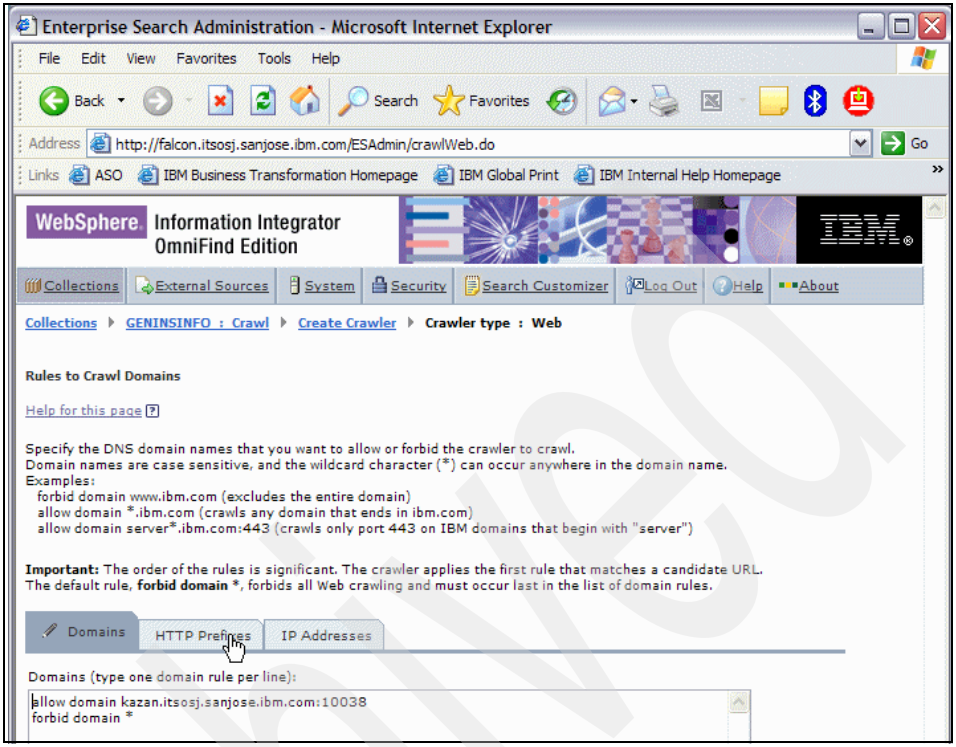

Figure 3-73 Rules to Crawl Domains

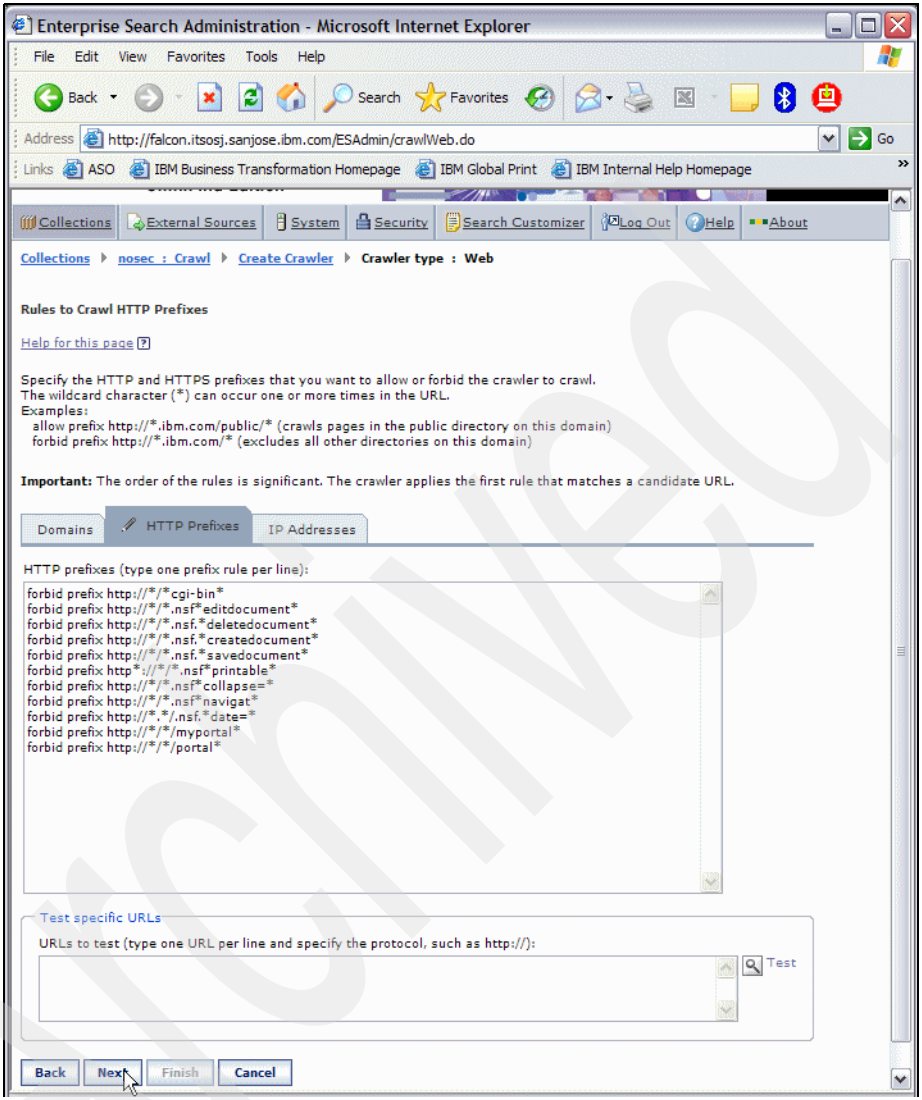

Figure 3-74 Rules to Crawl HTTP Prefixes

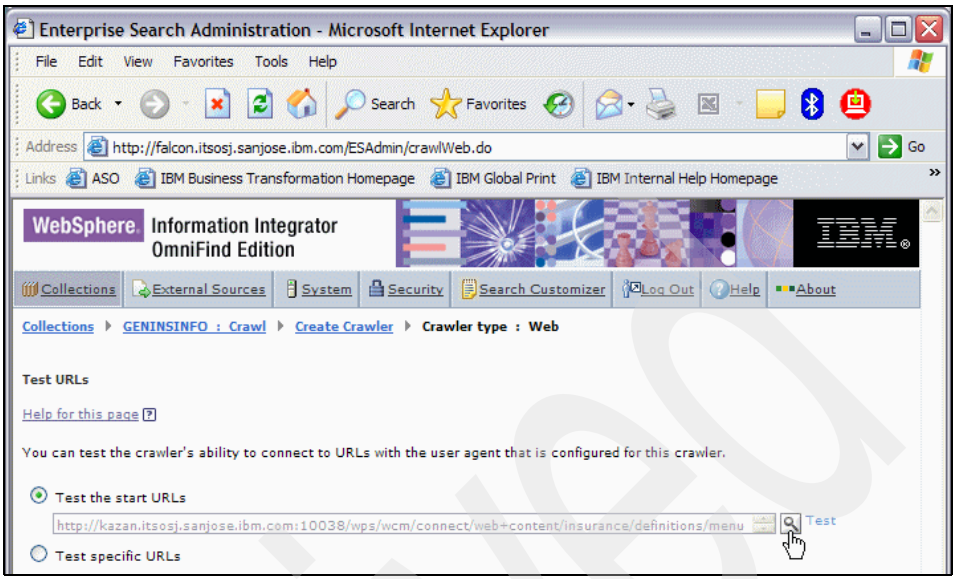

Figure 3-75 Test URL 1/2

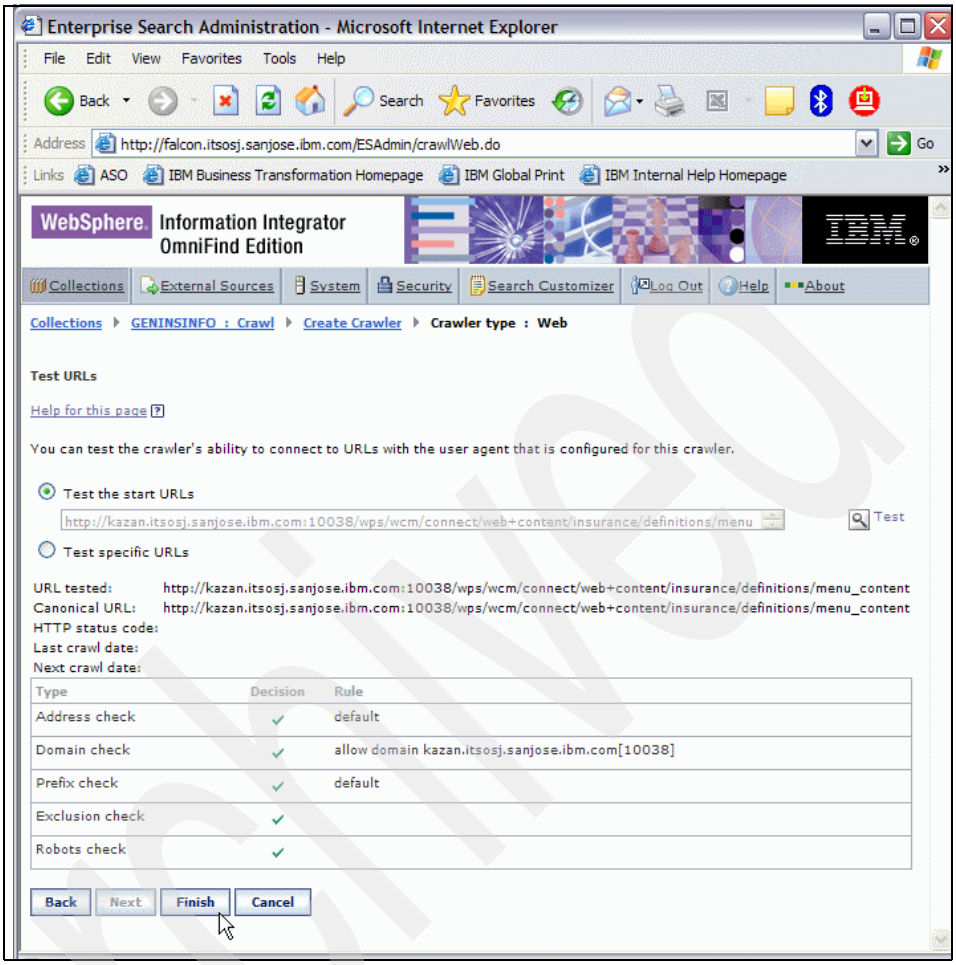

Figure 3-76 Test URL 2/2

LSTEP4d: Crawl GENINSINFO data sources

In this step, we implicitly initiate the crawl of the Web site by starting the crawler session, as shown in Figure 3-77 on page 227.

Note: With a Web crawler, a crawl schedule is not specifiable, since it is determined heuristically based on bounding parameters (for which we took the defaults).

When the Status icon changes to green, it indicates that the crawler session has started, as shown in Figure 3-78 on page 228. Click the **Details** icon to view the progress of crawling (not shown here).

Attention: When running crawlers in OmniFind V8.4, we strongly recommend that you run the parser at the same time. This is because a file queue is used to store crawled data instead of a DB2 table used in OmniFind V8.3. The file queue can fill up if the parser is not used to parse and delete the documents in the file queue while the crawler is running.

When it completes, we can now proceed to parse the crawled data, as described in Figure on page 228.

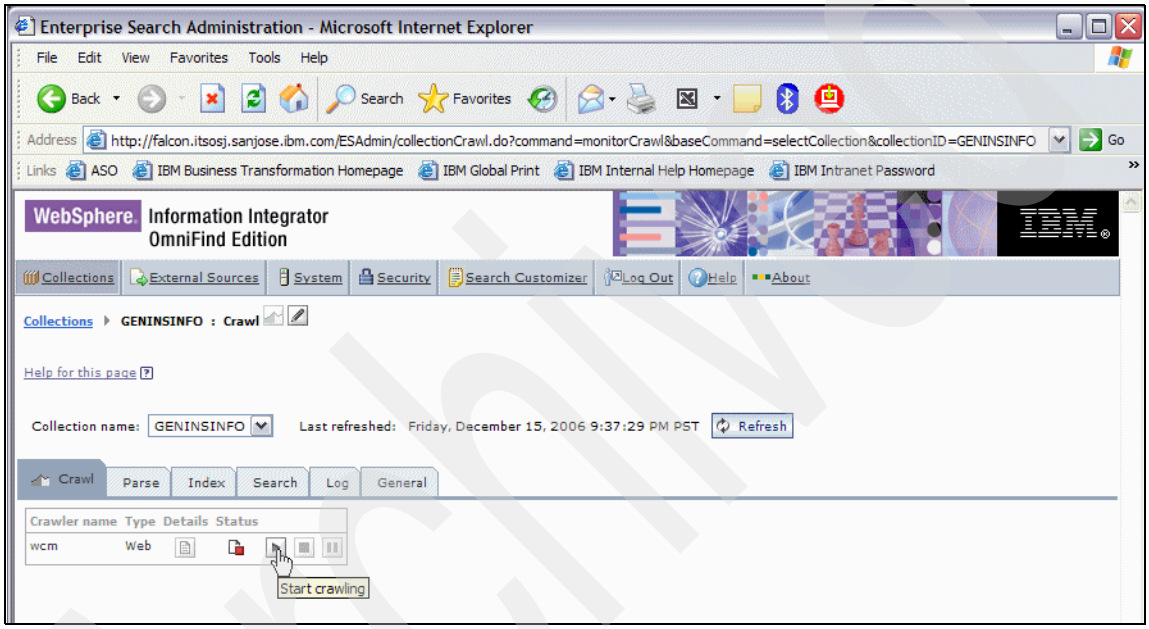

Figure 3-77 Start crawling

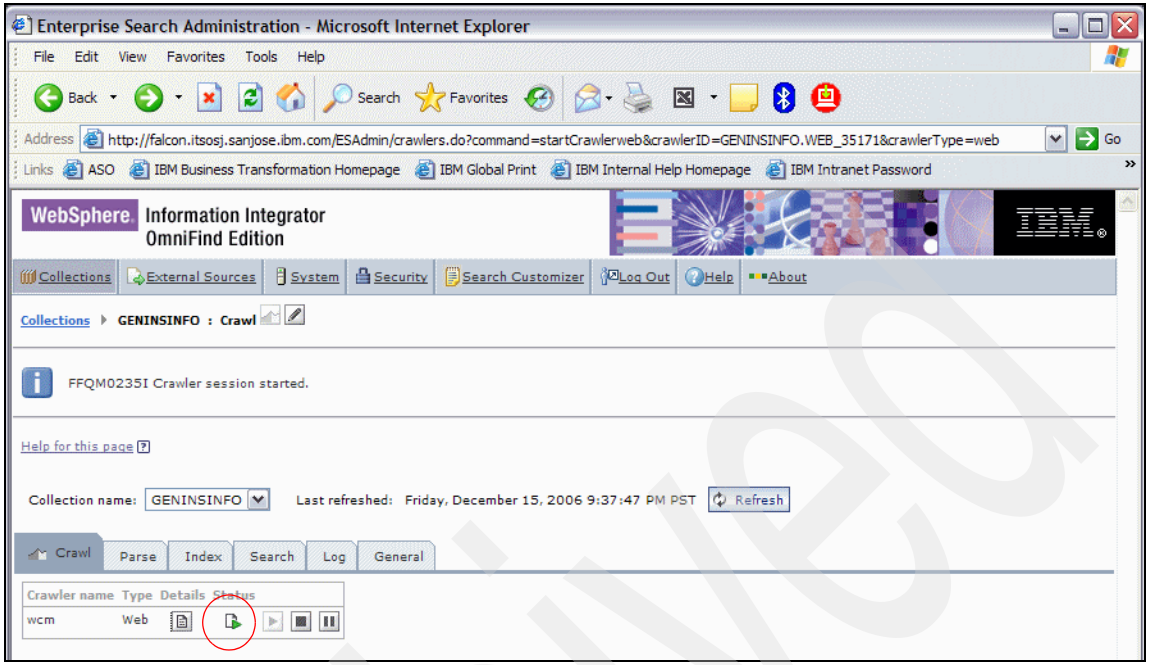

Figure 3-78 Crawler session started

LSTEP4e: Parse crawled GENINSINFO data

Figure 3-79 on page 229 through Figure 3-81 on page 230 describe the steps in parsing the crawled data.

From the Parse tab in Monitor mode, start the parser by clicking the start button, as shown in Figure 3-79 on page 229. After the Status icon turns green, you can monitor the progress of parsing by clicking **Details**, as shown in Figure 3-80 on page 229. Periodically click the **Refresh** button until the parser completes processing, as shown in Figure 3-81 on page 230. Review the parsing statistics.

Note: We stopped the parser to conserve resources in our constrained environment. In a real world environment, you would have the parser running continuously.

We can now proceed to build the main index, as described in “LSTEP4f: Build GENINSINFO collection index” on page 230.

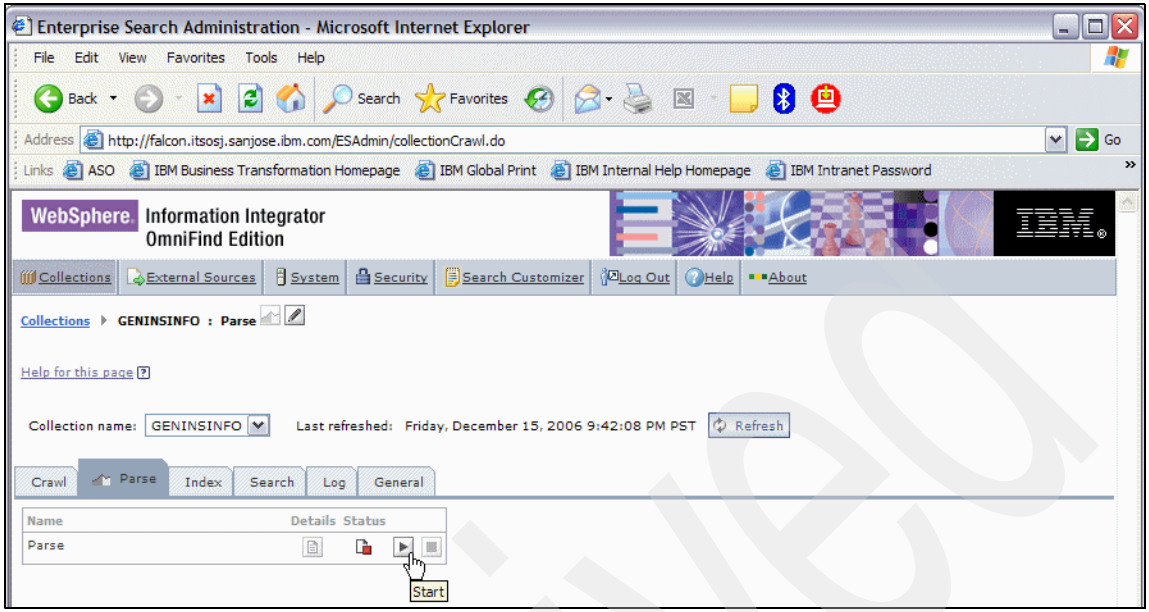

Figure 3-79 Start the parser

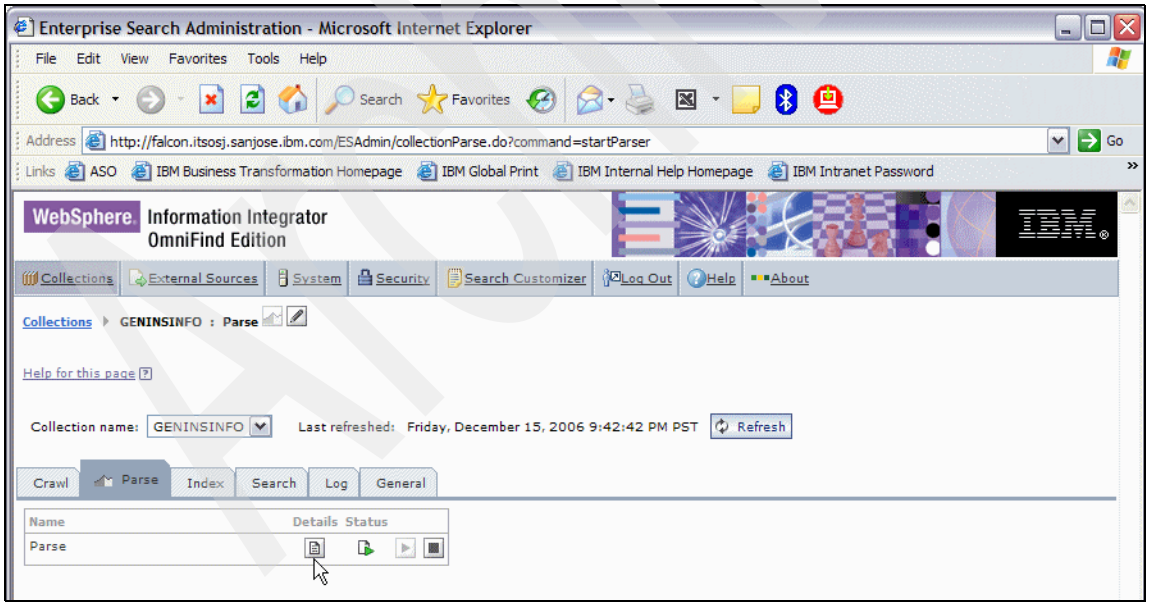

Figure 3-80 Click Details

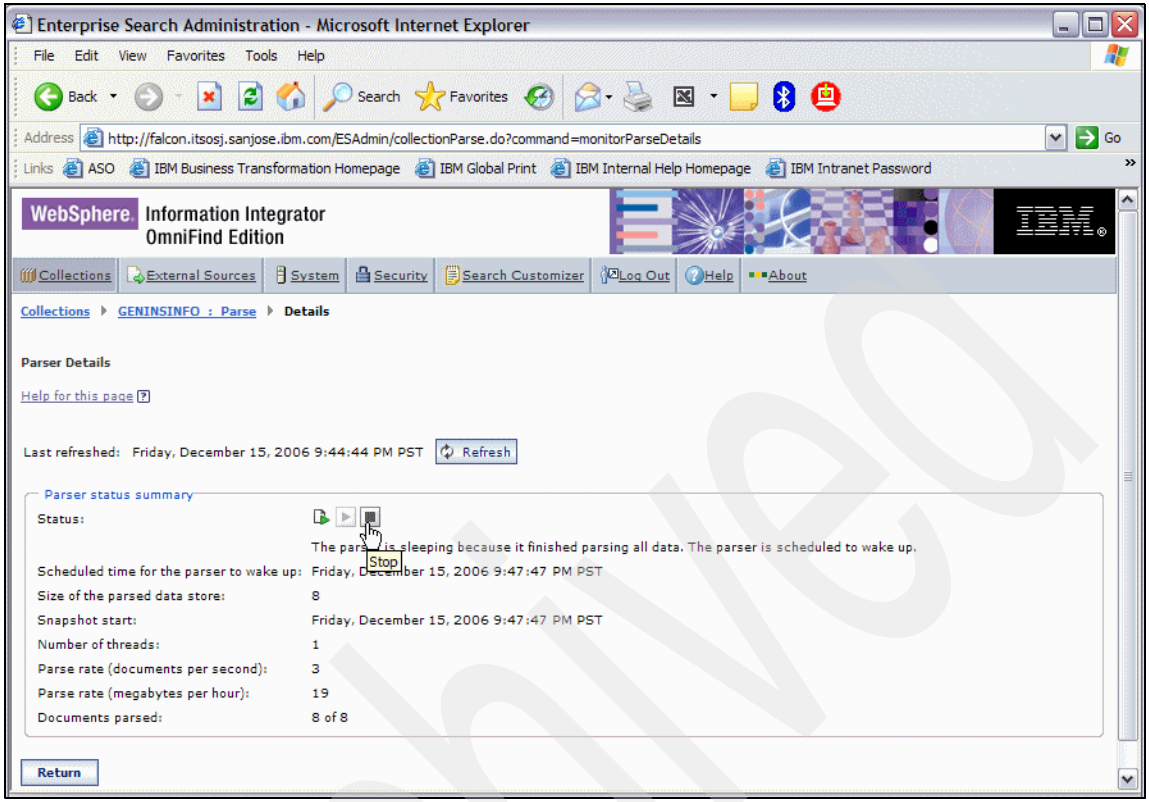

Figure 3-81 Parser completion

LSTEP4f: Build GENINSINFO collection index

From the Index tab in Monitor mode, start the main index build by clicking the start button, as shown in Figure 3-82 on page 231. Periodically click the **Refresh** button until the index build completes processing as shown in Figure 3-83 on page 232. Review the index statistics.

We can now proceed to configure start the search servers if they are not already running, as described in “LSTEP4g: Start search servers” on page 232.

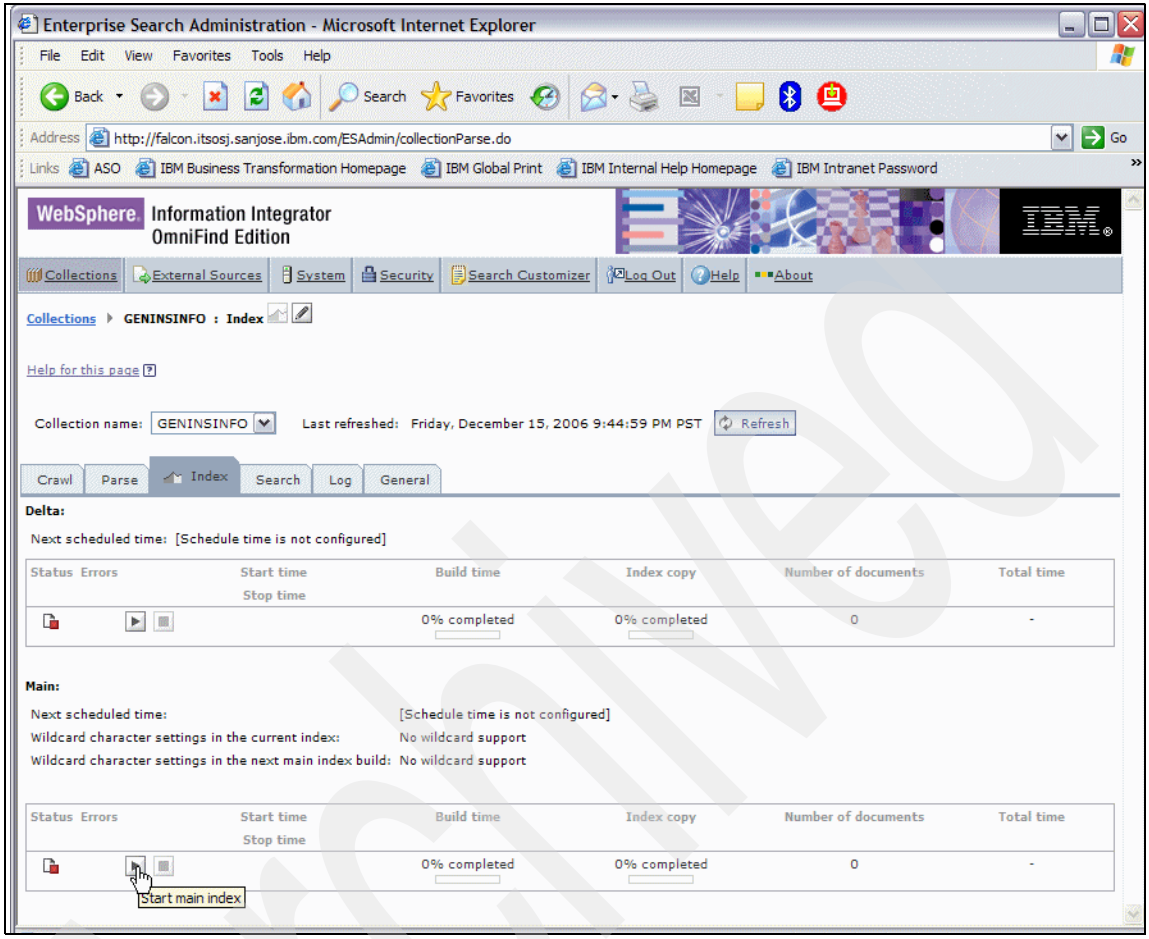

Figure 3-82 Start the main index build

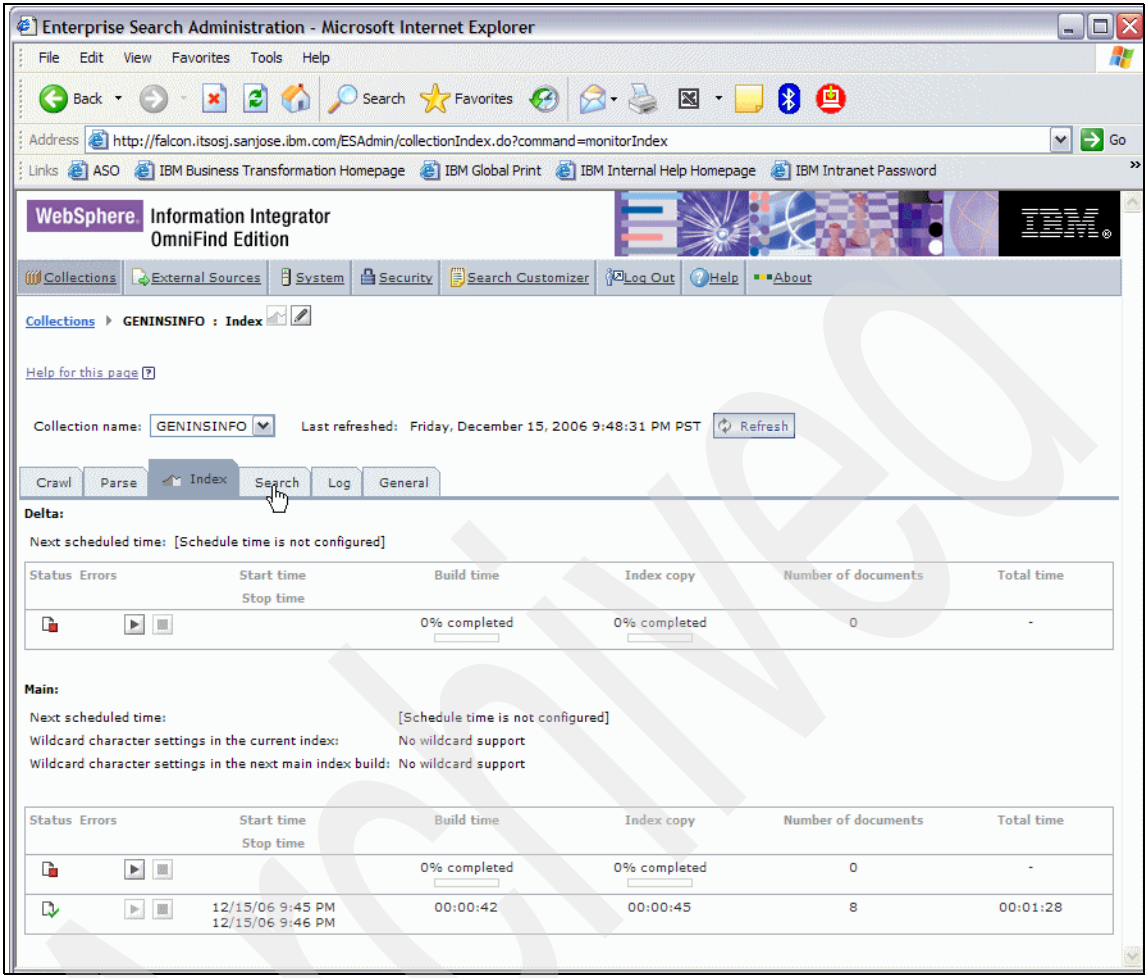

Figure 3-83 Main index build completion

LSTEP4g: Start search servers

Review the status of the search servers by navigating to the Collections view under the Search tab in Monitor mode, and click the start button if the Status icon is not green. Figure 3-84 on page 233 shows the Status icon to be green with the search servers running.

Before proceeding to parse the data collected by the CUSTINFO crawlers, we need to configure the UIMA annotator and synonym dictionary to enable searching of telephone numbers and e-mail addresses, as described in “LSTEP4h: Configure UIMA annotator for CUSTINFO collection” on page 233

and “LSTEP4i: Configure synonym dictionary for CUSTINFO collection” on page 247.

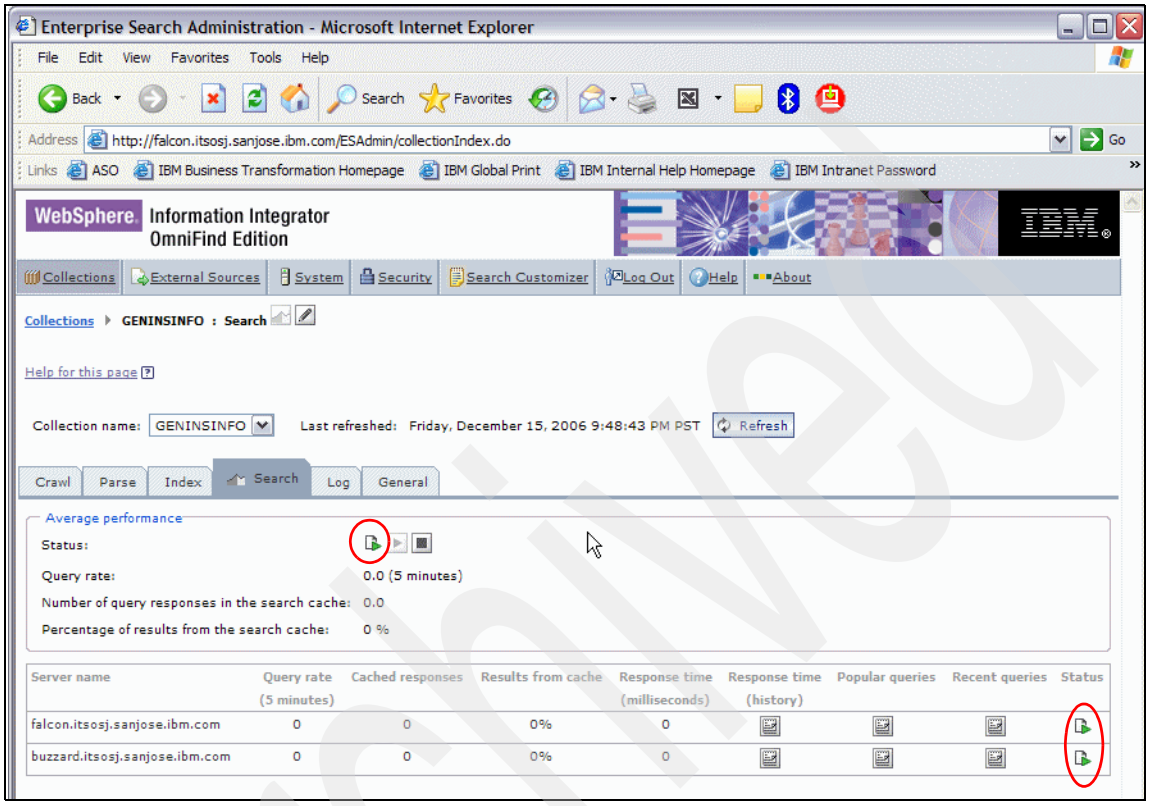

Figure 3-84 Search status of both search servers

LSTEP4h: Configure UIMA annotator for CUSTINFO collection

In this step, we need to define a UIMA text analysis engine and synonym dictionary to the system, and then associate them with the CUSTINFO collection.

The quality and precision of search results can be improved by integrating custom text processing algorithms with enterprise search collections. As mentioned earlier, IBM OmniFind Edition supports UIMA, which is a framework for creating, discovering, composing, and deploying text analysis functions. Application developers create and test analysis algorithms for the content to be searched, then create a processing engine archive (.pear file) that includes all of the resources required to use the archive for enterprise search. To be able to search collections with your custom analysis algorithms, you must add the archive (which contains the text analysis engine) to the enterprise search system. The analysis logic component in a text analysis engine is called an

annotator. Each annotator performs specific linguistic analysis tasks. A text processing engine can contain any number of annotators, or it can be a composite of several text analysis engines, each of which contain their own custom annotators. The information produced by the annotators is referred to as the analysis results. Analysis results, which correspond to the information that you want to search for, are written to a data structure called a common analysis structure (CAS).

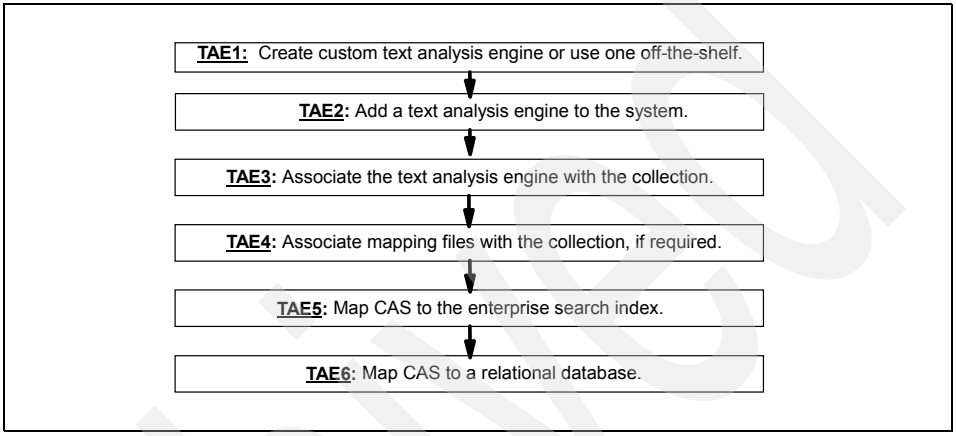

Figure 3-85 Steps to configure text processing engine with a collection

The steps involved in configuring the text processing options for a collection are shown in Figure 3-85 and are described briefly as follows:

1. In step TAE1, create a custom text analysis engine if necessary, or use one provided by a vendor. We are going to use the one provided with OmniFind (of_regex.pear) that performs telephone number, e-mail addresses, and URL annotations.
2. In step TAE2, add the text analysis engine to the system before it can be used by a collection.

   Figure 3-86 on page 235 through Figure 3-92 on page 240 describe the addition of the of_regex.pear text analysis engine to the system.

   From the System view, switch to Edit mode by clicking the **Edit** icon, as shown in Figure 3-86 on page 235. Under the Parse tab, click **Configure text analysis engines**, as shown in Figure 3-87 on page 235. Click **Add Text Analysis Engine** in Figure 3-88 on page 236 to provide details. In Figure 3-89 on page 237, provide the Text analysis engine name (IBM_TAE) and the location and name of the pear file (/opt/es/packages/uima/regex/of_regex.pear), and click **OK** to complete the addition of the text analysis engine to the system. To view details of the text

analysis engine, click the **Details** icon in Figure 3-90 on page 238 to view its XML source, as shown in Figure 3-91 on page 239.

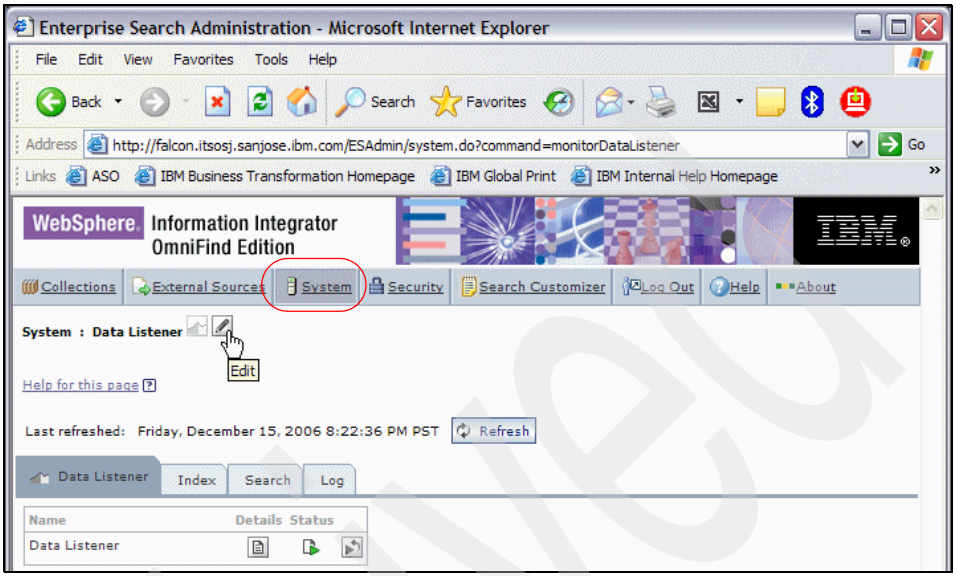

Figure 3-86 Click Edit icon under System view

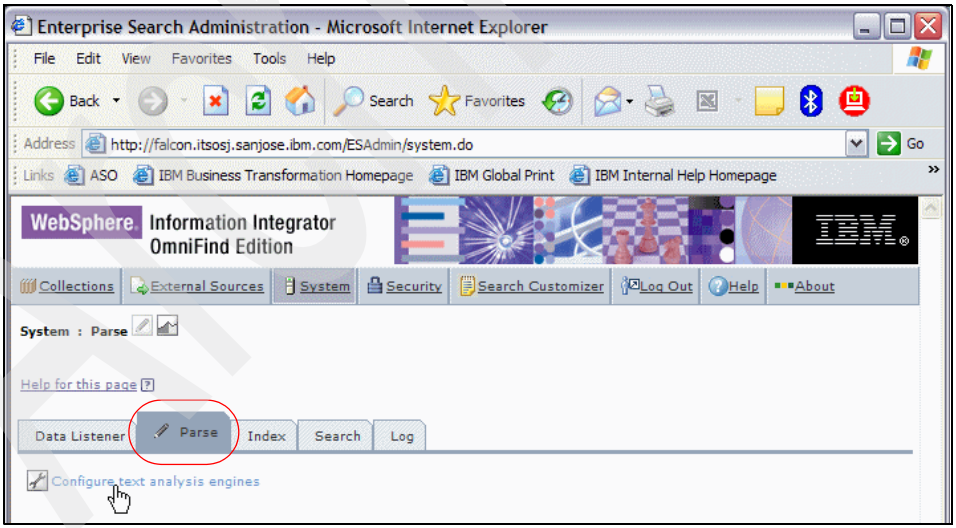

Figure 3-87 Configure text analysis engines

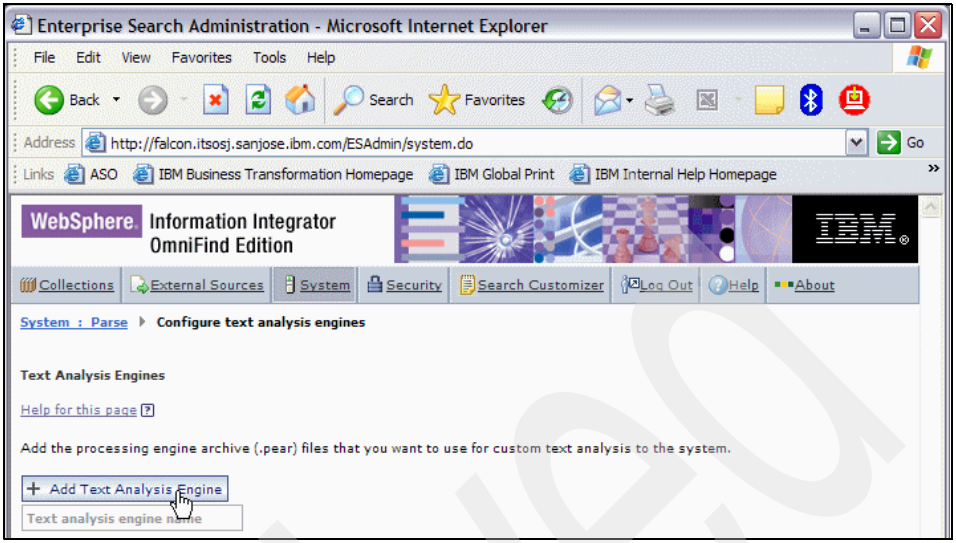

Figure 3-88 Add Text Analysis Engine 1/5

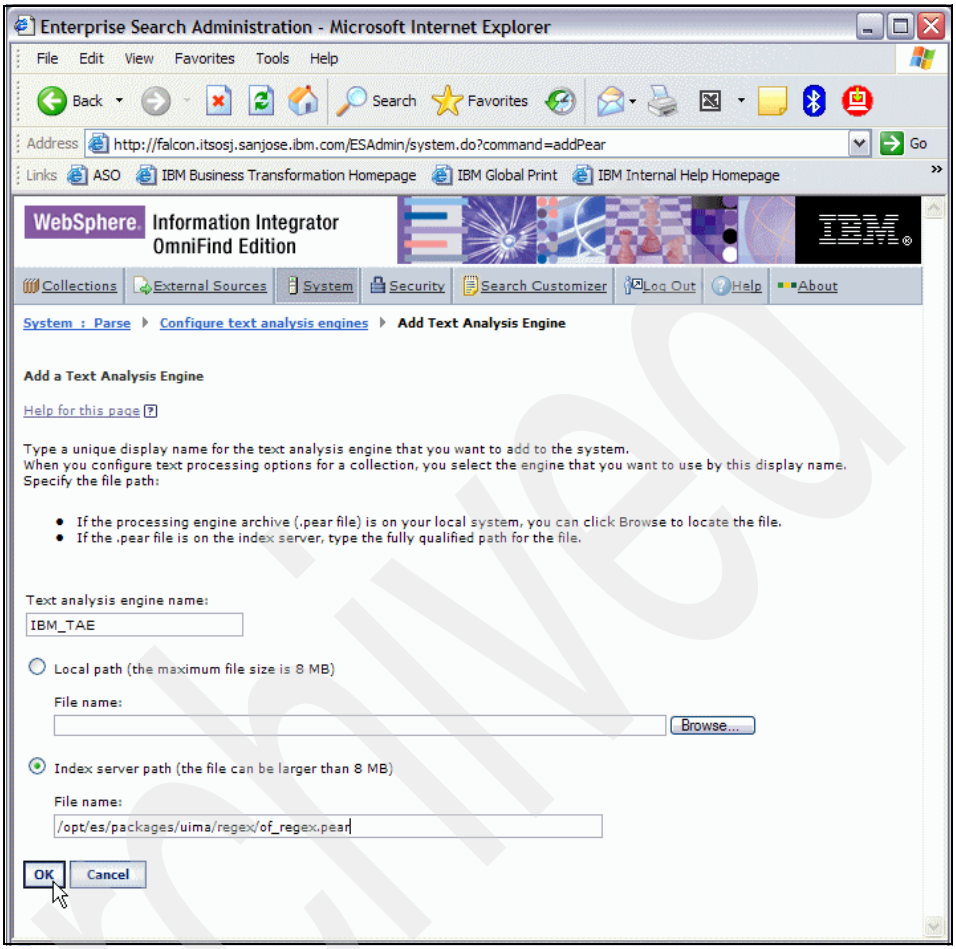

Figure 3-89 Add Text Analysis Engine 2/5

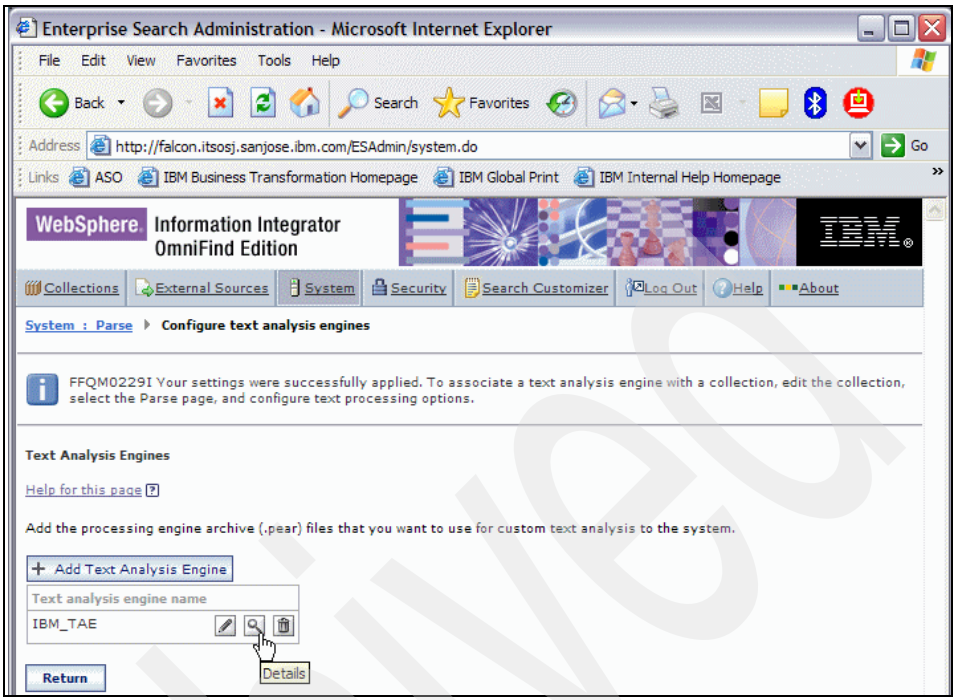

Figure 3-90 Add Text Analysis Engine 3/5

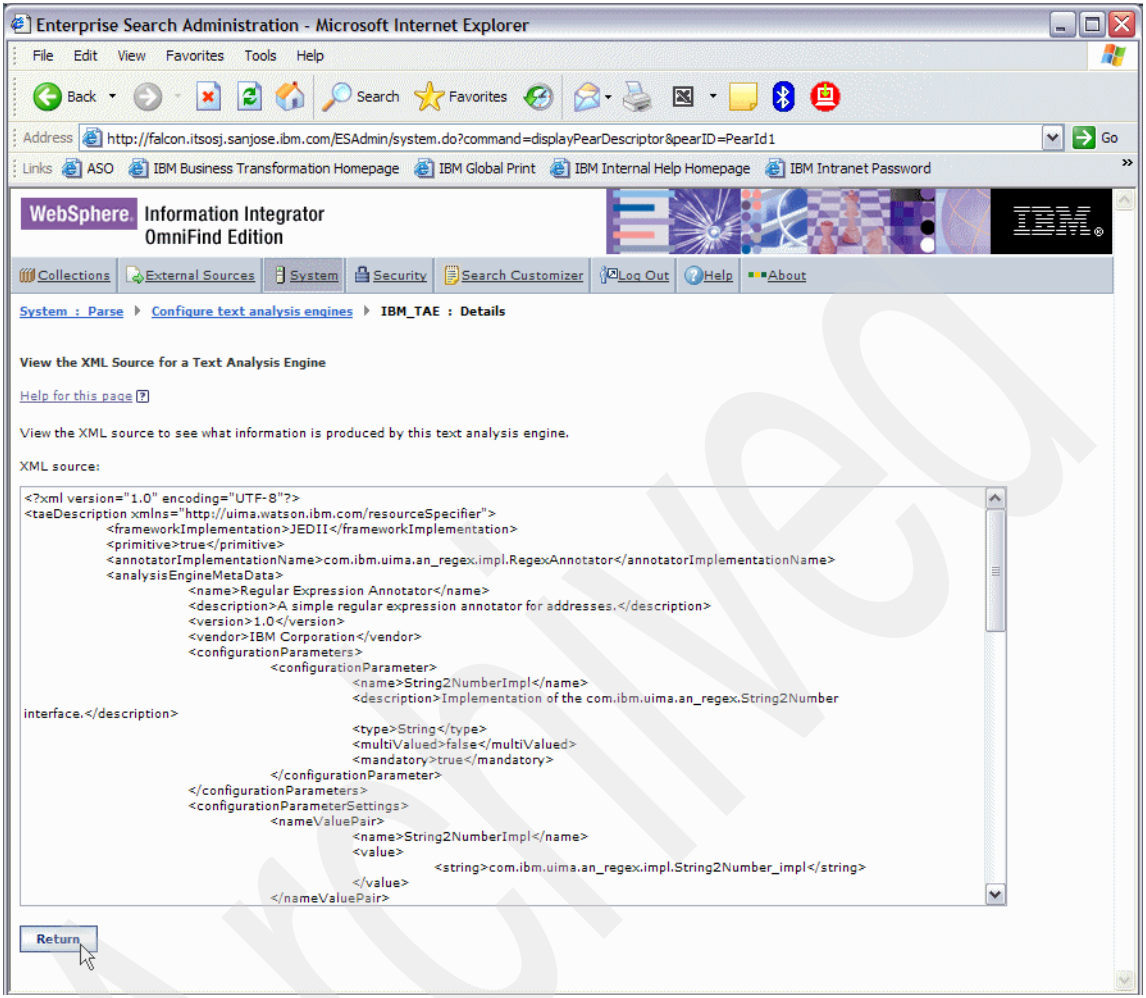

Figure 3-91 Add Text Analysis Engine 4/5

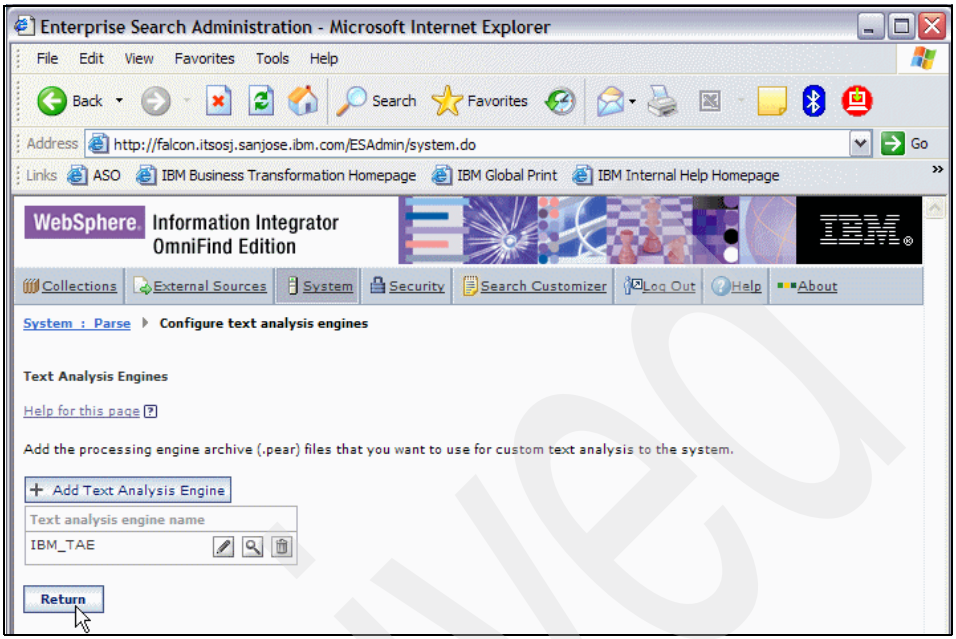

Figure 3-92 Add Text Analysis Engine 5/5

3. In step TAE3, associate the text analysis engine with a collection.

   In the Collections view, under the Parse tab in Edit mode for the CUSTINFO collection, click **Configure the text processing options**, as shown in Figure 3-93 on page 241. Select **IBM_TAE** from the Text analysis engine name drop-down list and click **OK** in Figure 3-94 on page 242 to confirm the association of the IBM_TAE text analysis engine with the CUSTINFO collection.

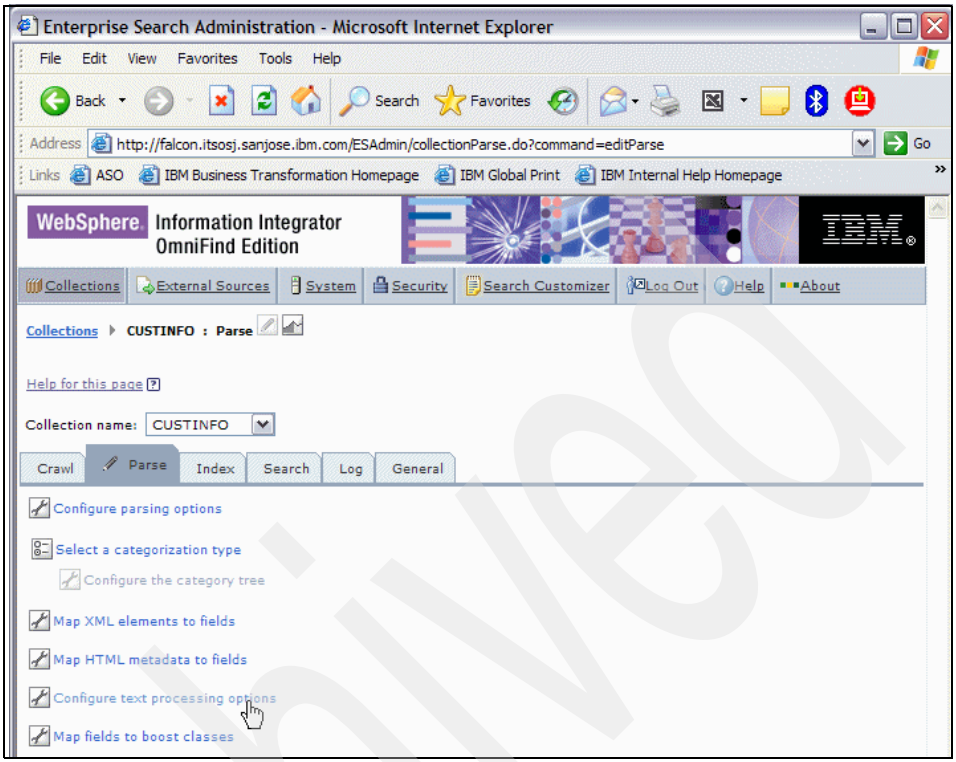

Figure 3-93 Configure text processing options for the CUSTINFO collection

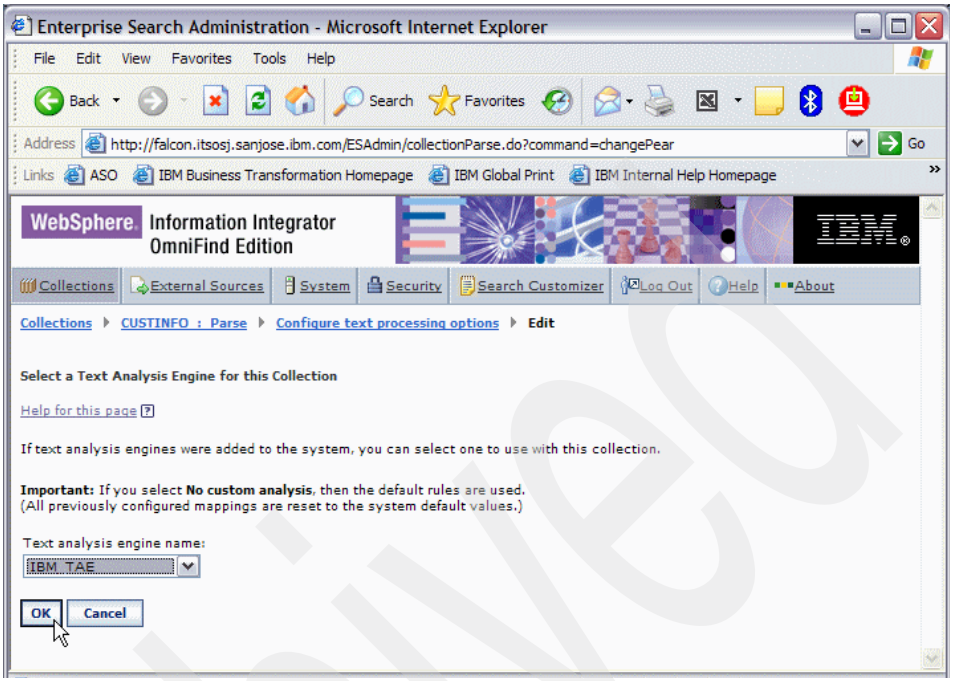

Figure 3-94 Associate IBM TAE text analysis engine with CUSTINFO collection

4. In step TAE4, if your collection contains XML documents with meaningful markup, and you want to use this markup in your custom text analysis, you can associate mapping files with the collection and map the output of the XML mapping to the common analysis structure (CAS).

   For example, you can map the content of <addressee> and <customer> elements to Person annotations in the common analysis structure. These annotations can then be accessed by your custom annotators, which might detect additional information (for example, they might detect the gender of the Person). You can also map Person annotations to the enterprise search index, which allows users to search for Persons without having to know the original XML elements.

   If you want to allow users to specify the original XML elements in queries, then you do not need to define any XML mappings. Instead, you can configure parsing options and enable native XML search for the collection.

   Note: Since the CUSTINFO collection does not include XML documents, this step does not apply in our case.

5. In step TAE5, map the CAS to the enterprise search index, which enables the annotated documents to be searched with semantic search. For example, depending on the entities and relationships that are detected by the annotators, users can search for concepts that occur in the same sentence (such as a specific person and any competitor name), or a keyword and a concept (such the name Alex and a phone number).

   Figure 3-95 on page 244 through Figure 3-98 on page 247 describes the mapping of the CAS to the enterprise search index. Click **Select a mapping file** in the Map the common analysis structure to the index section in Figure 3-95 on page 244, and specify the location and file name of the mapping file (\opt\es\packages\uima\regex\of_sample_regex_cas2index.xml) in the Index server path in Figure 3-96 on page 245. Click **OK** to confirm the mapping.

   To view the XML source to see how the CAS maps to the enterprise search index, click **View XML source** in the Map the common analysis structure to the index section in Figure 3-97 on page 246 to view it, as shown in Figure 3-98 on page 247. Click **Return**.

6. In step TAE6, map the common analysis structure to a relational database. You can map data to IBM DB2 Universal Database (DB2 UDB) or Oracle tables. This type of mapping enables the results of analysis to be used in database applications such as data mining. It also enables you to use SQL queries to search the data outside of enterprise search. We are not using this capability in our enterprise search solution.

The next step is to configure the synonym dictionary, as described in “LSTEP4i: Configure synonym dictionary for CUSTINFO collection” on page 247.

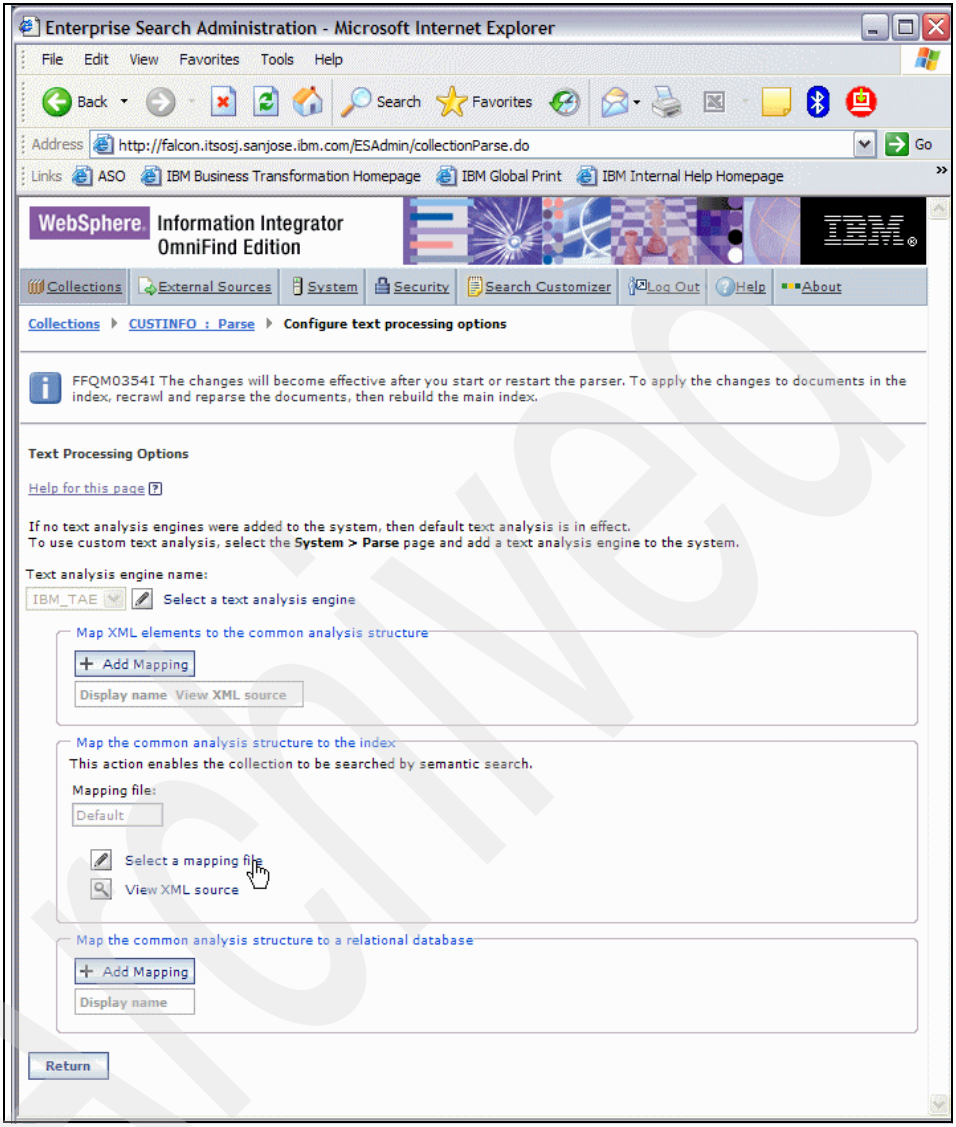

Figure 3-95 Select a mapping file

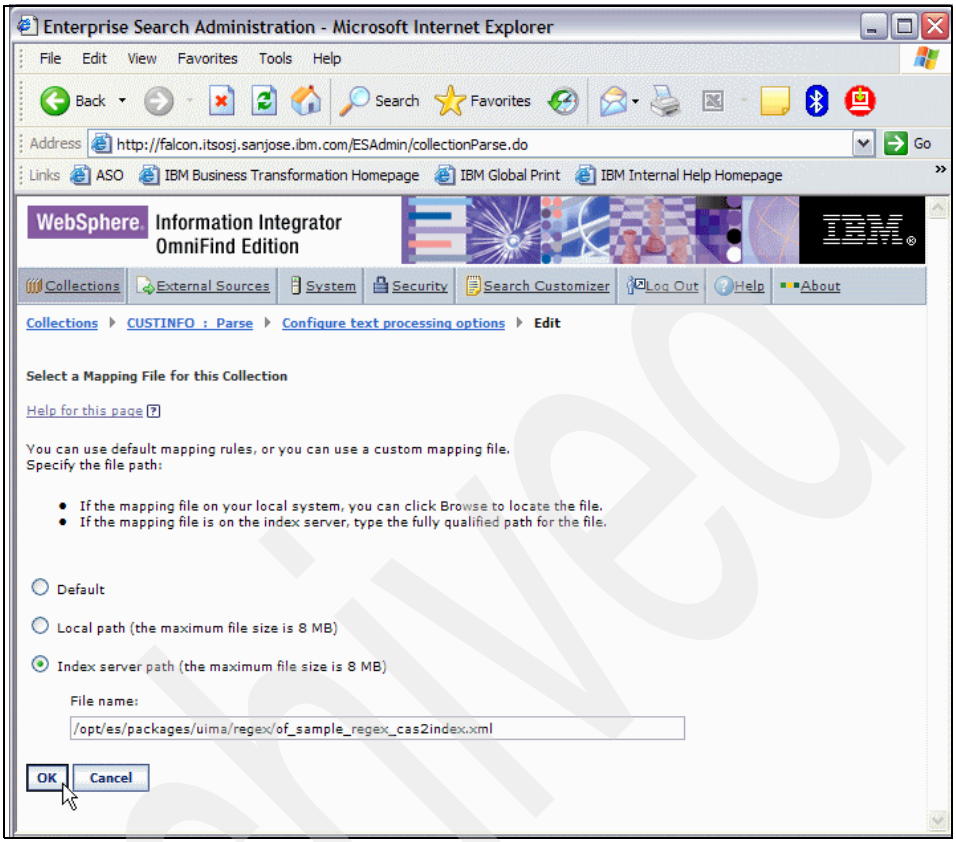

Figure 3-96 Specify the path of the mapping file

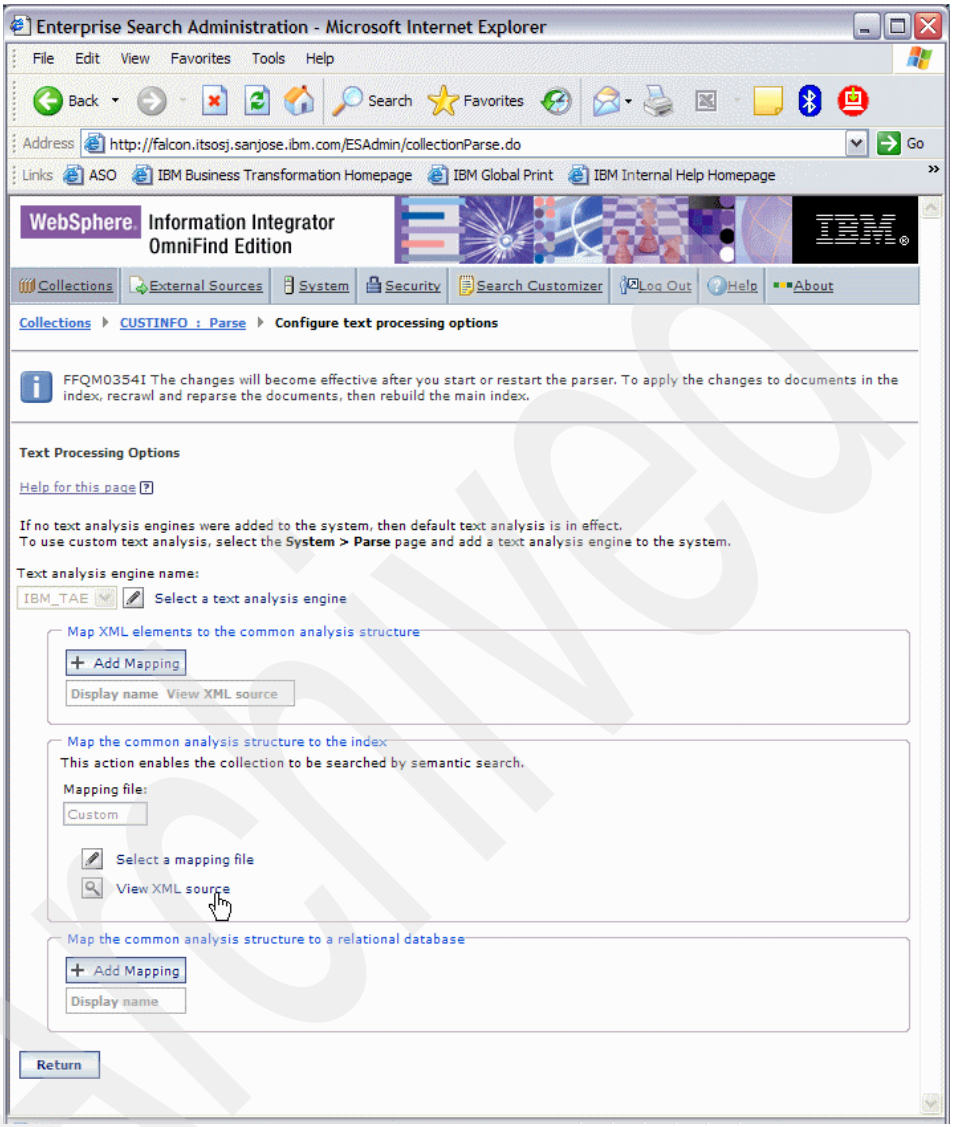

Figure 3-97 View XML source 1/2

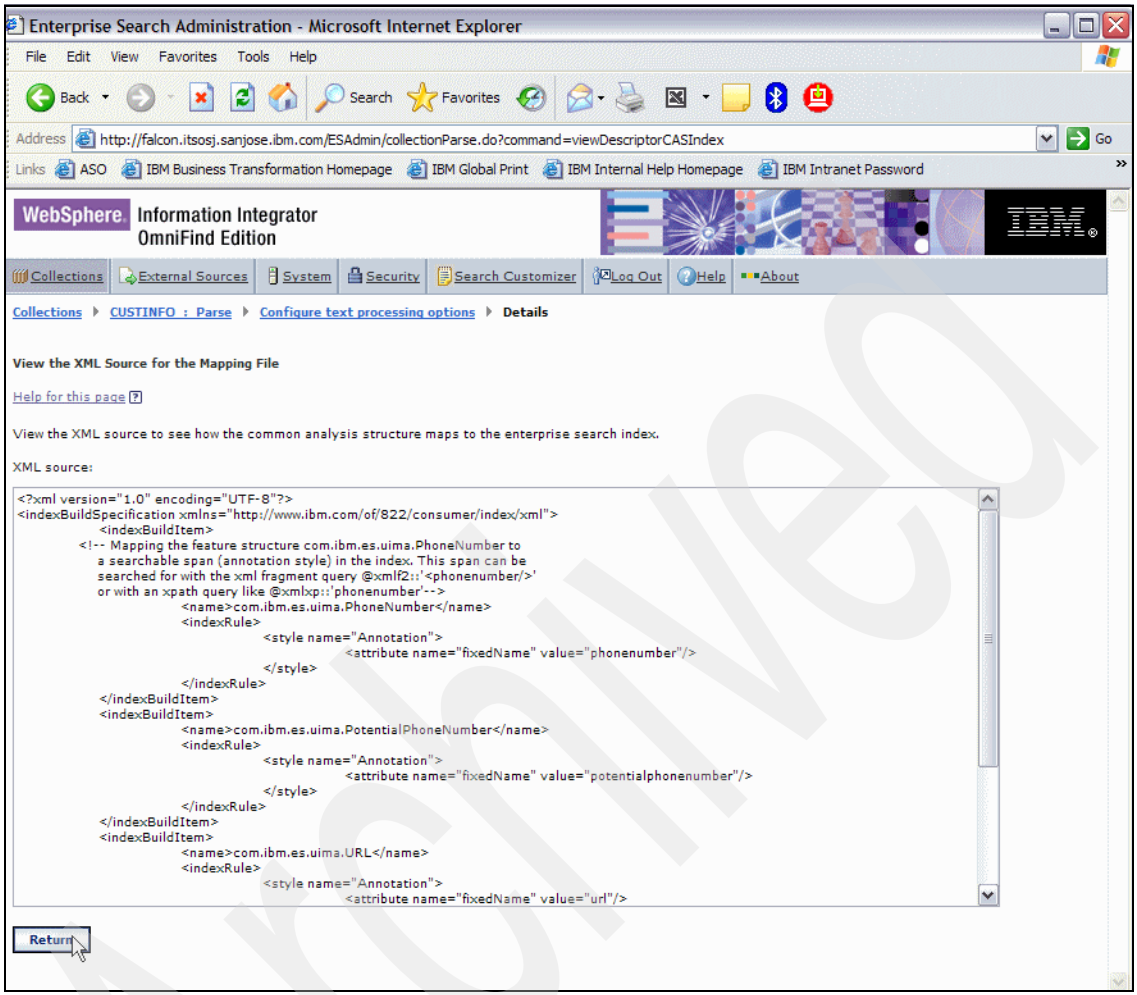

Figure 3-98 View XML source 2/2

LSTEP4i: Configure synonym dictionary for CUSTINFO collection

Since we mapped the CAS to the enterprise search index, the annotated documents can be searched with a semantic search. When users query a collection that uses a synonym dictionary, documents that contain synonyms of the query terms are included in the search results if the synonyms are present in the enterprise search index.

To use synonym dictionaries for searching the documents in a collection, you must first associate the dictionaries with the enterprise search system. You can later choose which synonym dictionary you want to use for searching a collection.

Figure 3-99 on page 249 through Figure 3-105 on page 255 describe the steps involved in defining an IBM OmniFind synonym dictionary to the system, and then associating this synonym dictionary with the CUSTINFO collection.

From the System view, in Edit mode under the Search tab, click **Configure synonym dictionaries**, as shown in Figure 3-99 on page 249. Click **Add Synonym Dictionary**, as in Figure 3-100 on page 249, to provide the synonym dictionary details. In Figure 3-101 on page 250, provide the Synonym dictionary name (REGEX_DICT) and the location and name of the dic file (/opt/es/packages/uima/regex/of_sample_synonym_dic.dic) in the Index server path File name field, and click **OK** to complete the addition of the synonym dictionary to the system.

Example 3-3 on page 251 shows the synonym dictionary contents that includes synonyms for telephone number (phone number, telephone nbr, phone nbr, and the CAS to index mapping for phone number, as shown in Figure 3-98 on page 247), URL (unified resource locator, Web address, and the CAS to index mapping for phone number, as shown in Figure 3-98 on page 247), facsimile (fax number, fax nbr, facsimile nbr, and the CAS to index mapping for phone number, as shown in Figure 3-98 on page 247) and e-mail address (e-mail address, and the CAS to index mapping for phone number, as shown in Figure 3-98 on page 247).

Figure 3-102 on page 252 shows the REGEX_DICT being in the system. Click **Return.**

Before the newly added REGEX_DICT can be associated with the CUSTINFO collection, the search servers must first be stopped. Figure 3-103 on page 253 through Figure 3-105 on page 255 show the steps in associating the REGEX_DICT synonym dictionary with the CUSTINFO collection.

From the Collections view for the CUSTINFO collection, in Monitor mode under the Search tab, click the stop icon to stop both the search servers, as shown in Figure 3-103 on page 253. Then switch to Edit mode, and under the Search tab, click **Configure search server options**, as shown in Figure 3-104 on page 254. On the Search Server options, select REGEX_DICT from the Synonym dictionary name drop-down list and click **OK** to complete this definition, as shown in Figure 3-105 on page 255.

We can now proceed to crawl the CUSTINFO collection data sources, as described in “LSTEP4j: Crawl CUSTINFO data sources” on page 256.

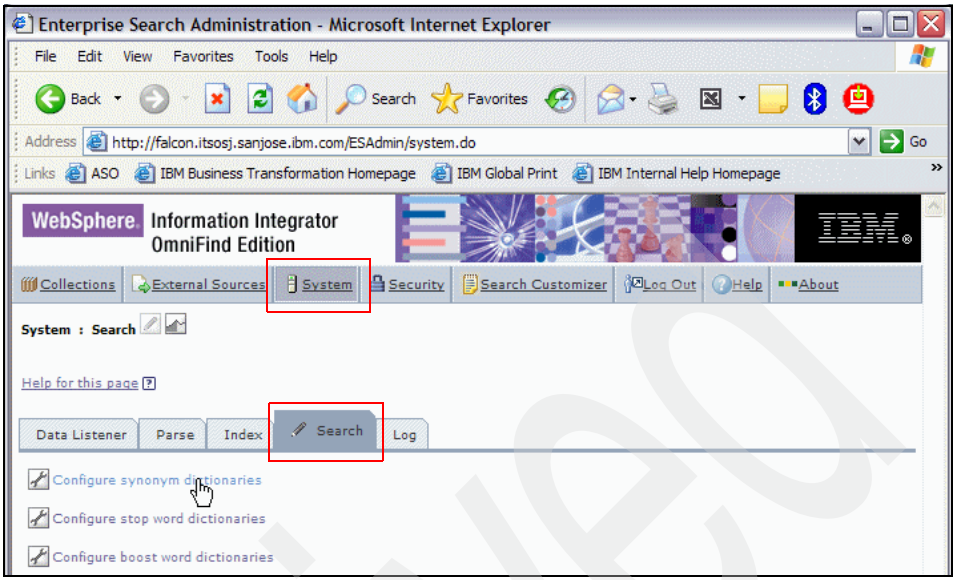

Figure 3-99 Configure the synonym dictionaries in Edit mode in the System view under Search

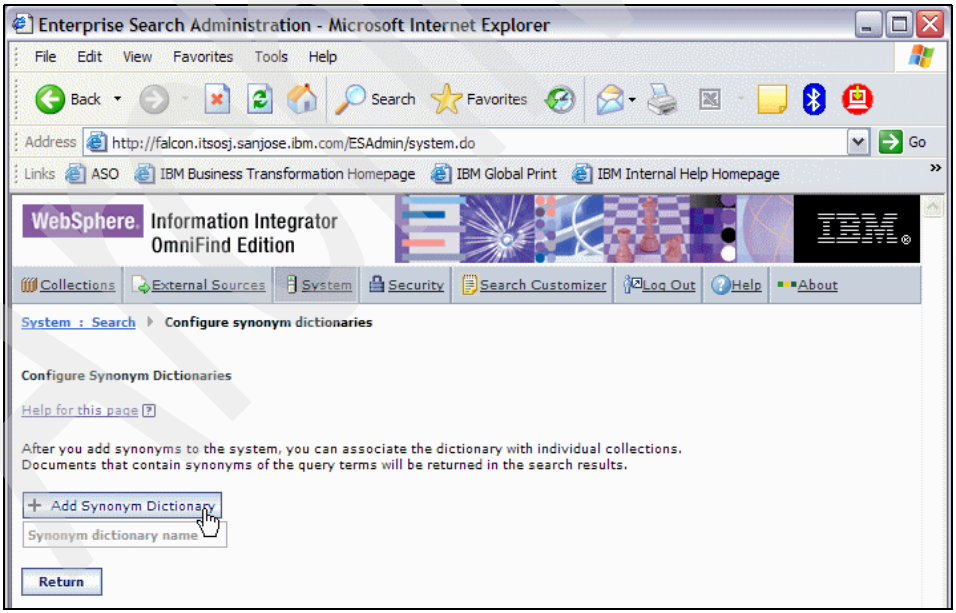

Figure 3-100 Add Synonym Dictionary 1/3

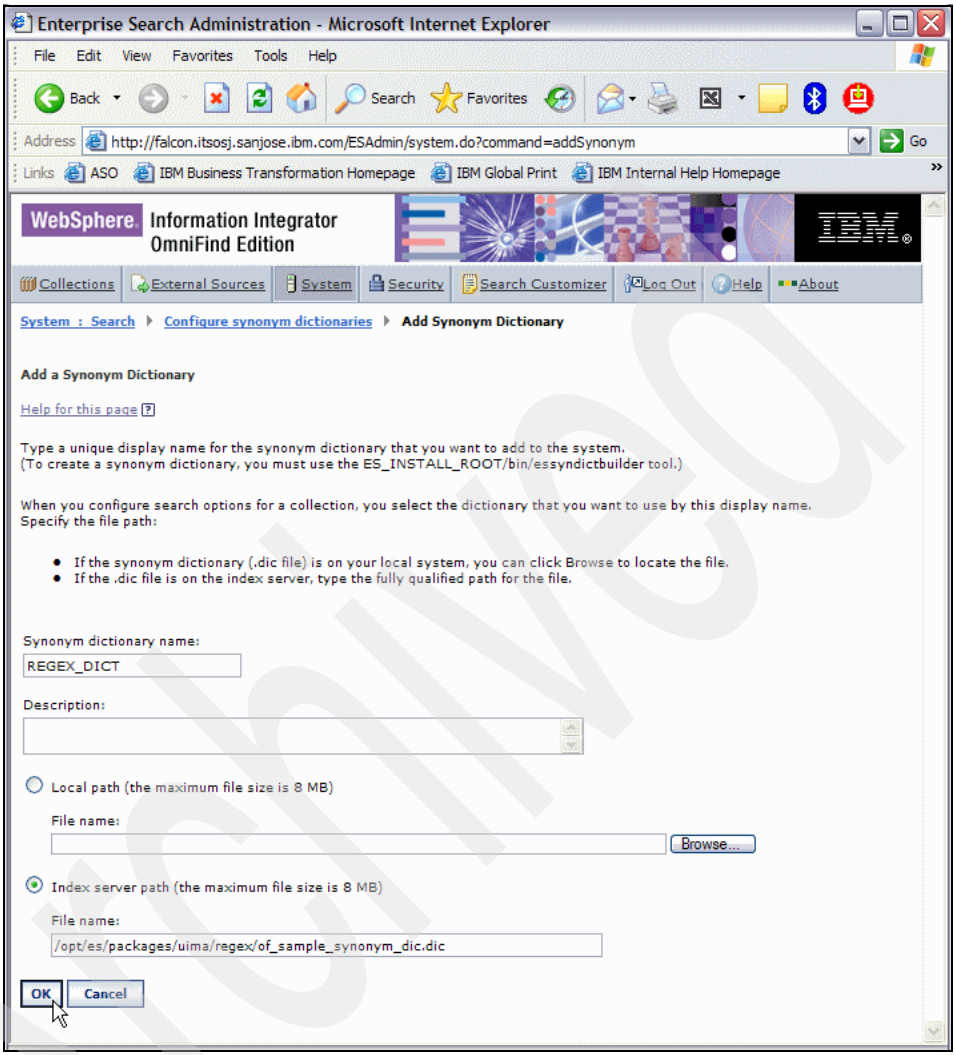

Figure 3-101 Add Synonym Dictionary 2/3

Example 3-3 Synonym dictionary of_sample_synonym_dic.dic

```
<?xml version="1.0" encoding="UTF-8"?>
<synonymgroups xmlns="http://www.ibm.com/of/822/synonym/xml">
  <synonymgroup>
    <synonym>telephone number</synonym>
    <synonym>phone number</synonym>
    <synonym>telephone nbr</synonym>
    <synonym>phone nbr</synonym>
    <synonym>@xmlf2::'&lt;#phonenumber/&gt;'</synonym>
  </synonymgroup>
  <synonymgroup>
    <synonym>facsimile number</synonym>
    <synonym>fax number</synonym>
    <synonym>facsimile nbr</synonym>
    <synonym>fax nbr</synonym>
    <synonym>@xmlf2::'&lt;#phonenumber/&gt;'</synonym>
  </synonymgroup>
  <synonymgroup>
    <synonym>email address</synonym>
    <synonym>e-mail address</synonym>
    <synonym>@xmlf2::'&lt;#email/&gt;'</synonym>
  </synonymgroup>
  <synonymgroup>
    <synonym>URL</synonym>
    <synonym>unified resource locator</synonym>
    <synonym>web address</synonym>
    <synonym>@xmlf2::'&lt;#url/&gt;'</synonym>
  </synonymgroup>
</synonymgroups>
```

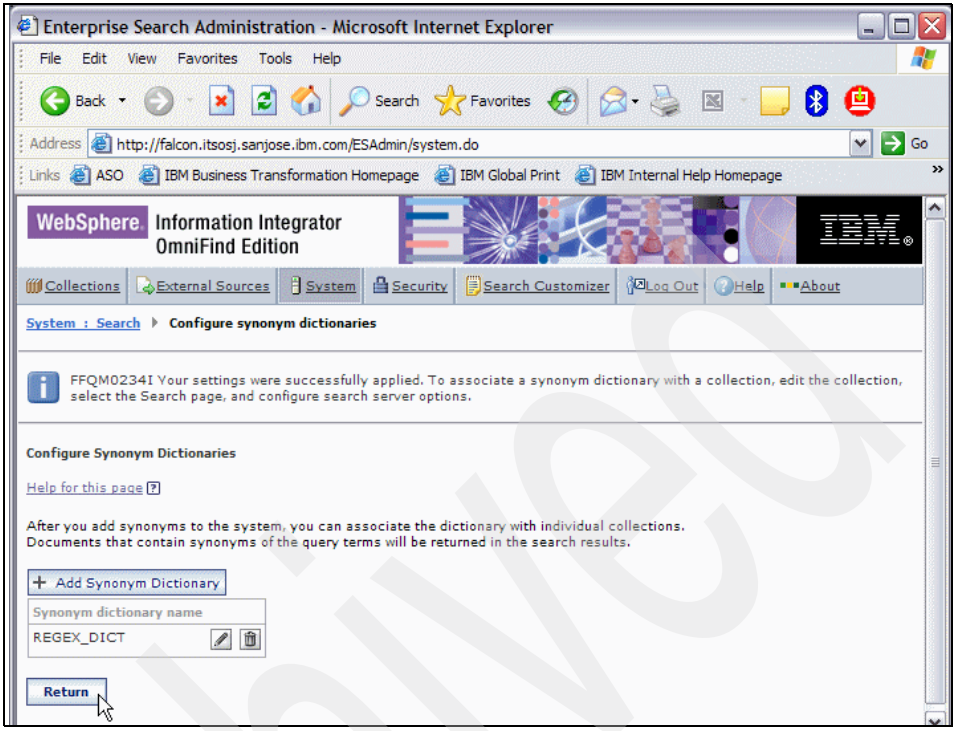

Figure 3-102 Add Synonym Dictionary 3/3

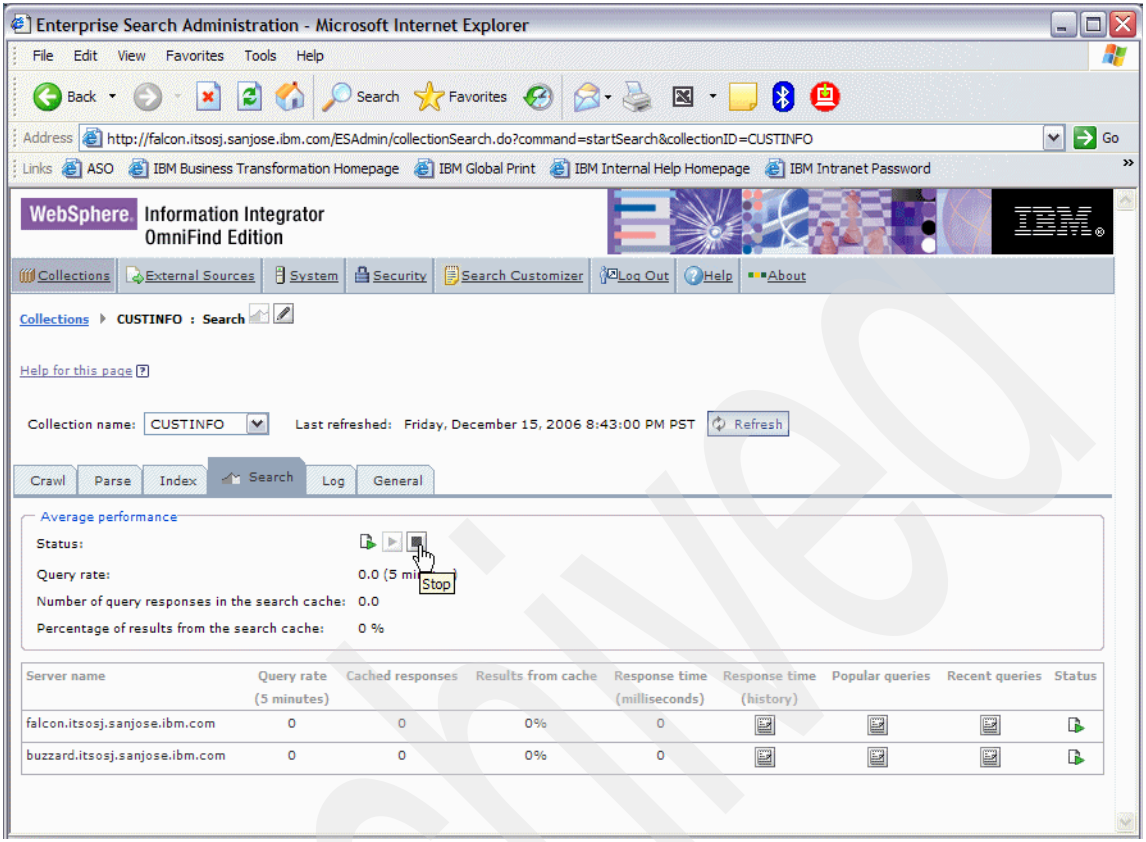

Figure 3-103 Stop (both) the Search servers

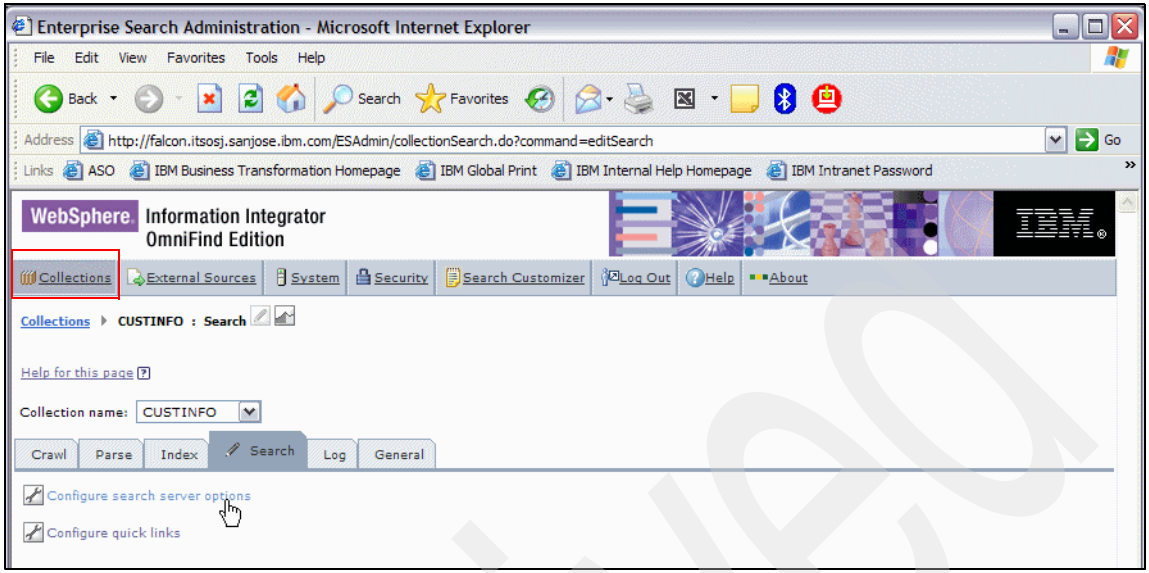

Figure 3-104 Configure search server options for CUSTINFO in Edit mode in Collections view under Search

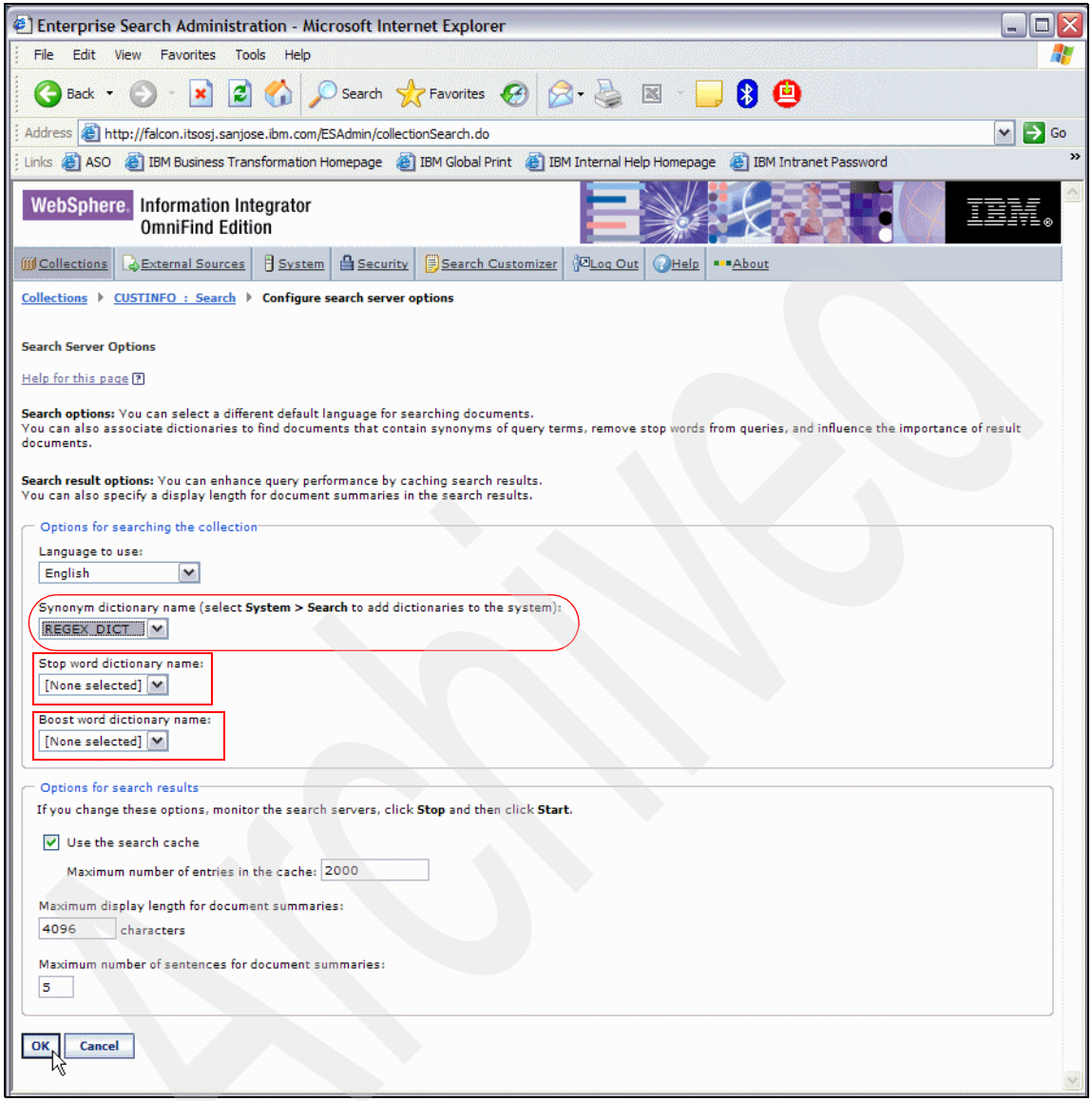

Figure 3-105 Associate the REGEX_DICT synonym dictionary with the CUSTINFO collection

LSTEP4j: Crawl CUSTINFO data sources

In this step, we initiate a full crawl of the data sources crawled by the Content Edition and WebSphere Portal crawlers. Figure 3-106 on page 257 through Figure 3-112 on page 262 describe the steps involved. Click the **Crawl** icon for the CUSTINFO collection in the Collections view in Monitor mode, as shown in Figure 3-106 on page 257, to view the crawlers defined for this collection.

Attention: When running crawlers in OmniFind V8.4, we strongly recommend that you run the parser at the same time. This is because a file queue is used to store crawled data instead of a DB2 table used in OmniFind V8.3. The file queue can fill up if the parser is not used to parse and delete the documents in the file queue while the crawler is running.

The Content Edition and WebSphere Portal crawlers can now be started as follows:

- Content Edition crawler

  Start the crawler session by clicking the start button for the pdm crawler, as shown in Figure 3-107 on page 257. Once the Status icon turns green, click **Details** to proceed to Figure 3-108 on page 258. Start a full crawl by clicking the appropriate icon, as shown in Figure 3-108 on page 258. Monitor the status of the crawl by clicking the **Refresh** button intermittently, until the completion of the crawl, as shown in Figure 3-109 on page 259. The Status icon turns green.

  Note: We stopped the crawler to conserve resources in our constrained environment.

  We can now start a full crawl of the WebSphere Portal crawler data sources, as described in “WebSphere Portal crawler” on page 259.

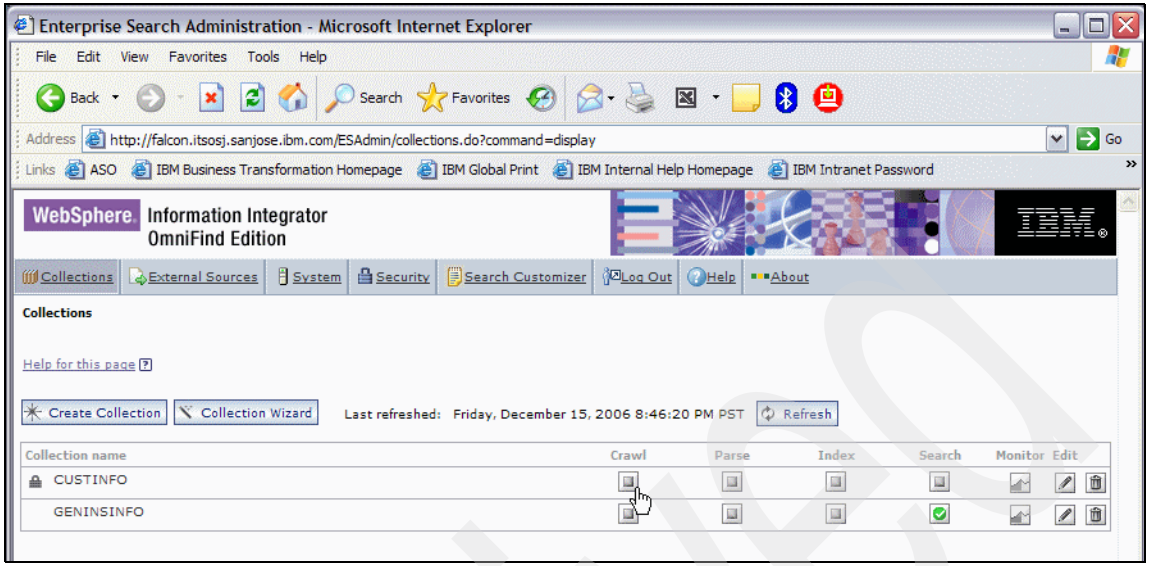

Figure 3-106 Click Crawl icon for CUSTINFO collection

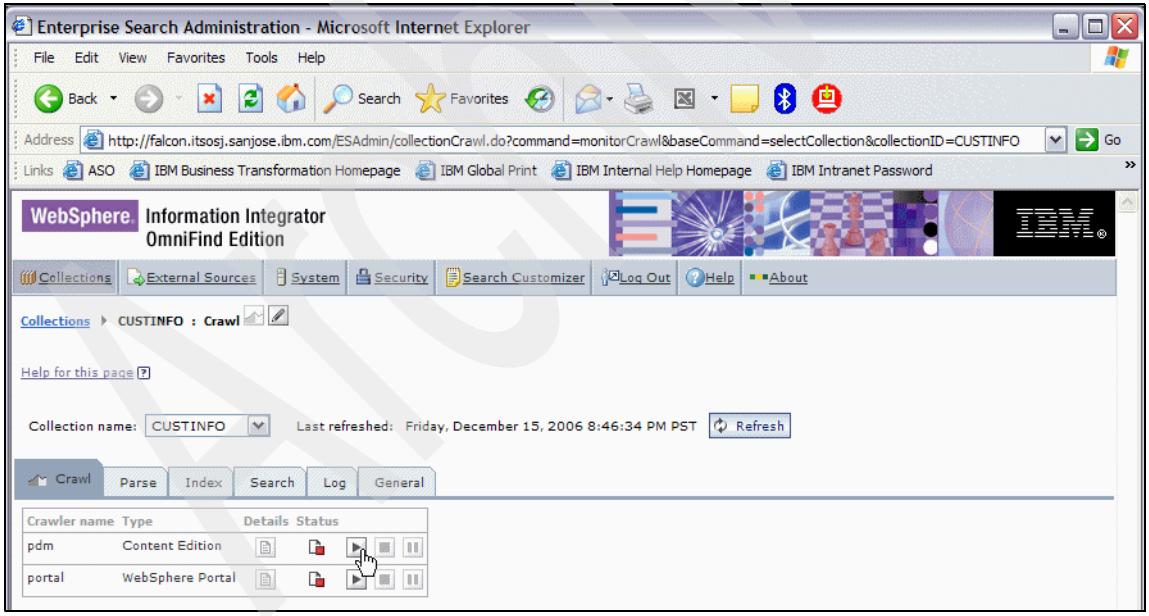

Figure 3-107 Start Content Edition (pdm) crawler session

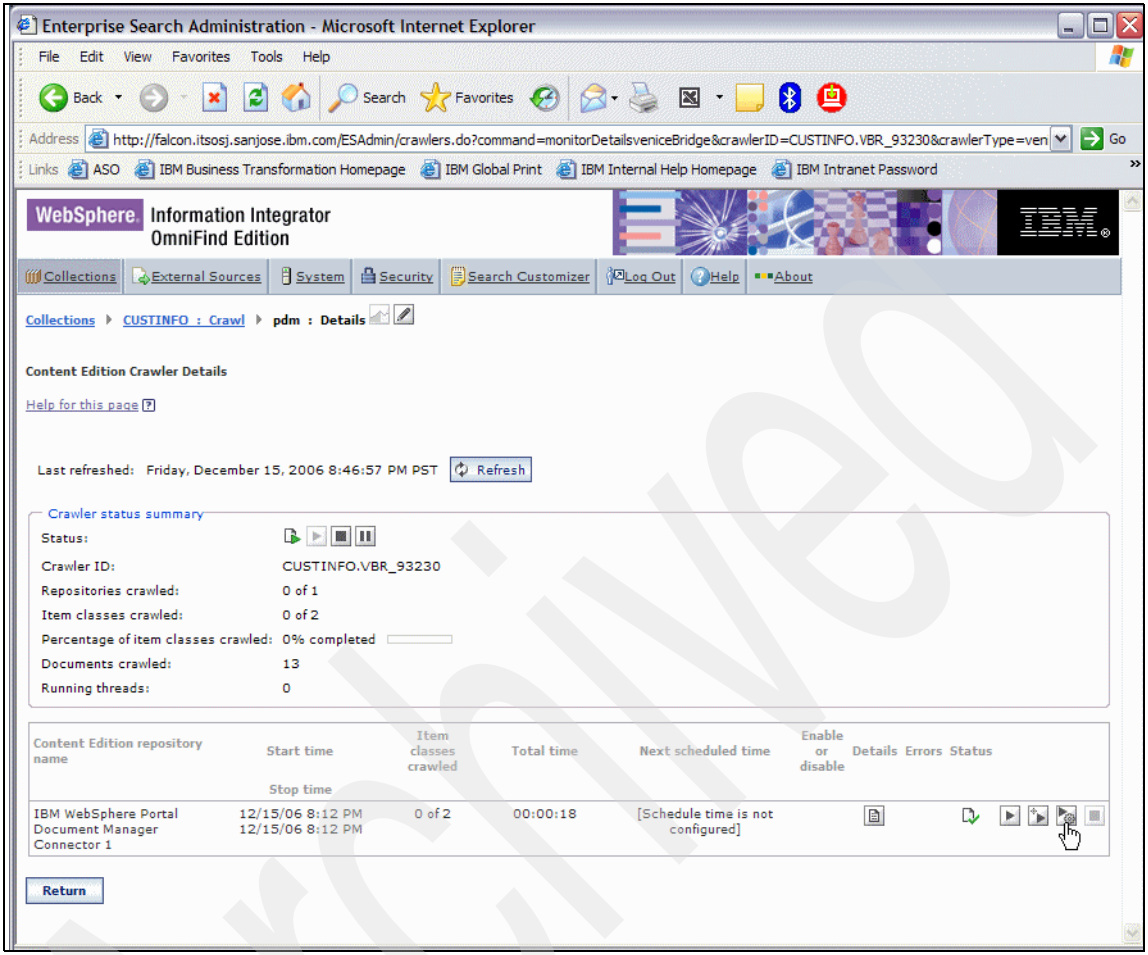

Figure 3-108 Start a full crawl

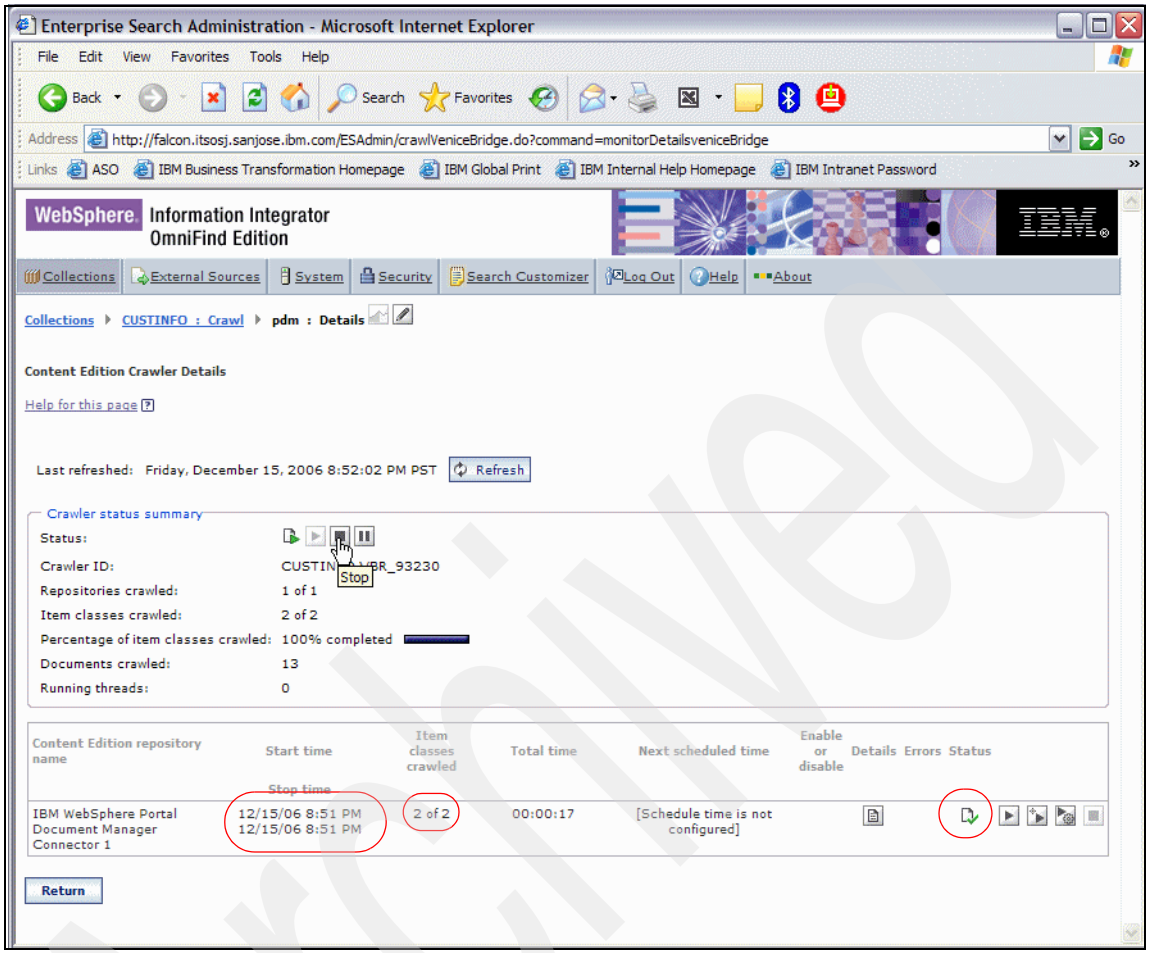

Figure 3-109 Crawler completion status

- WebSphere Portal crawler

  Start the crawler session by clicking the start button for the portal crawler, as shown in Figure 3-110 on page 260. Once the Status icon turns green, click **Details** to proceed to Figure 3-111 on page 261. Start a full crawl by clicking the appropriate icon, as shown in Figure 3-111 on page 261. Monitor the status of the crawl by clicking the **Refresh** button intermittently, until the completion of the crawl, as shown in Figure 3-112 on page 262. The Status icon turns green.

  Note: We stopped the crawler to conserve resources in our constrained environment.

We can now proceed to parse the crawled data, as described in “LSTEP4k: Parse CUSTINFO collection” on page 262.

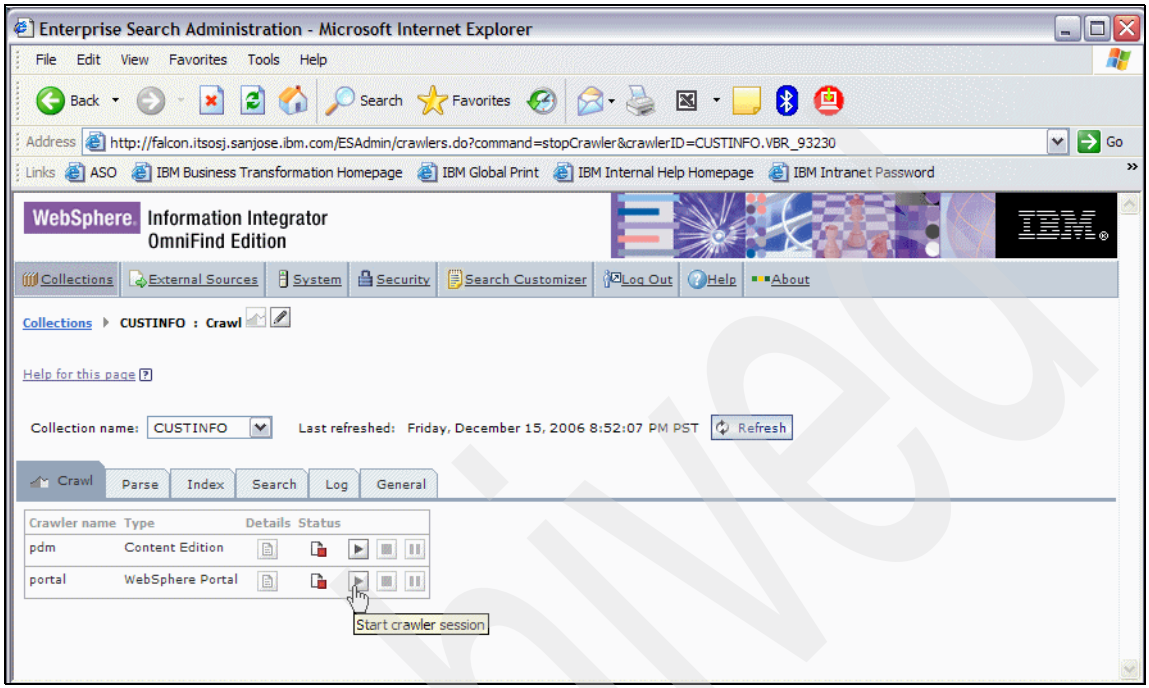

Figure 3-110 Start WebSphere Portal (portal) crawler session

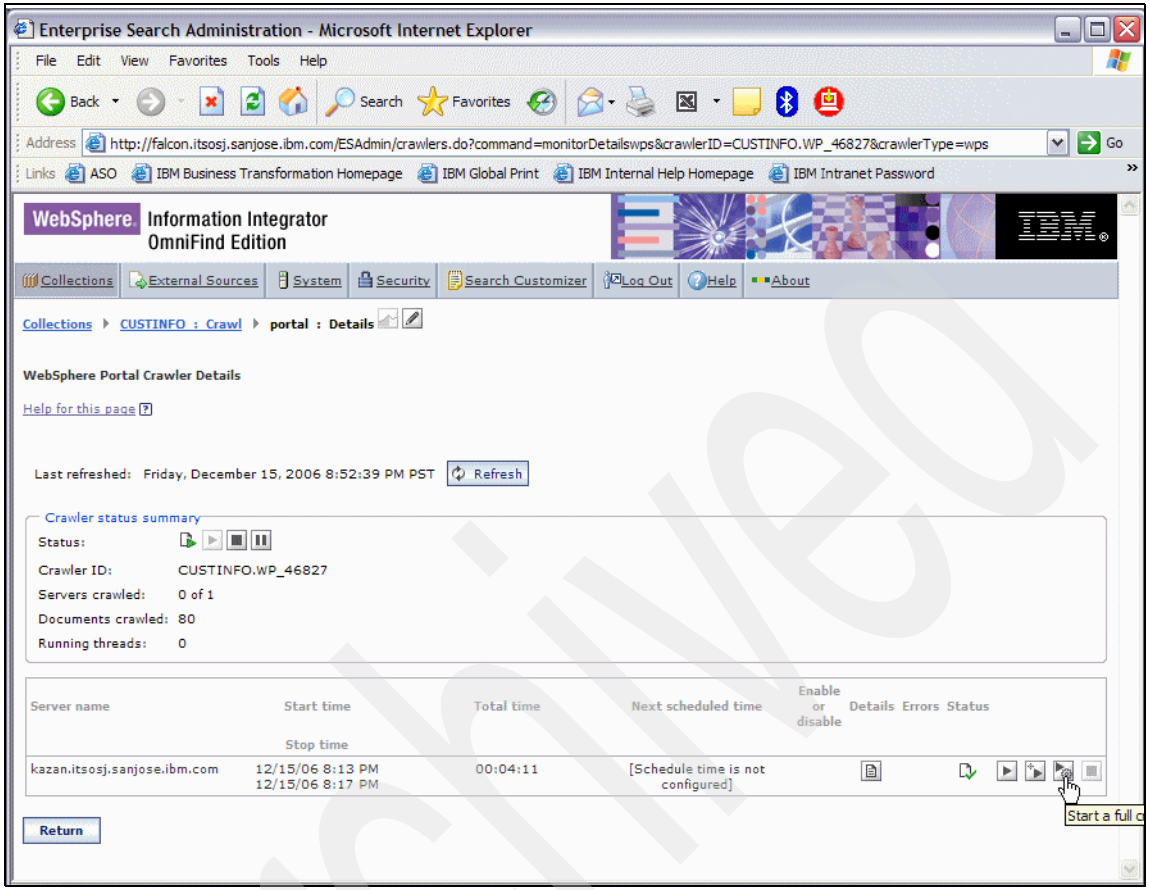

Figure 3-111 Start a full crawl

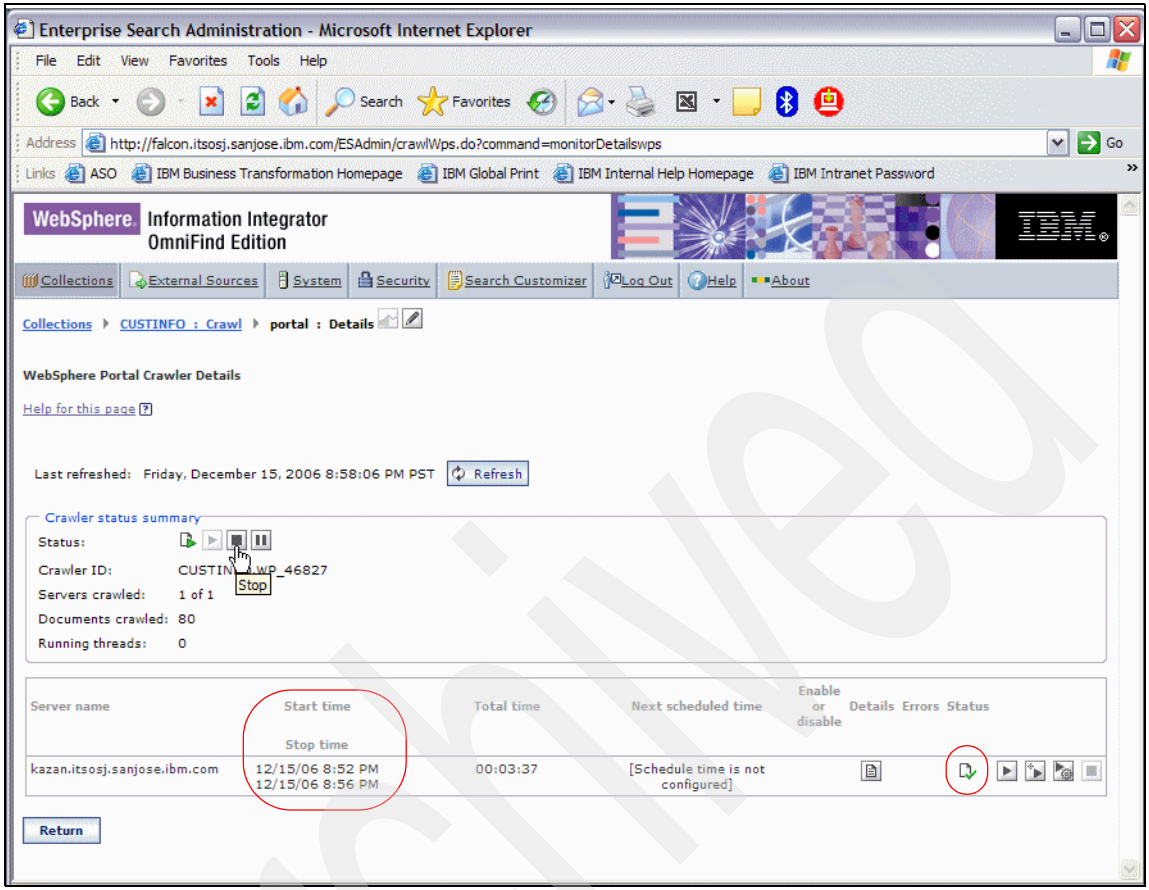

Figure 3-112 Crawler completion status

LSTEP4k: Parse CUSTINFO collection

Figure 3-113 on page 263 through Figure 3-115 on page 265 describe the steps in parsing the crawled data.

From the Parse tab in Monitor mode, start the parser by clicking the start icon, as shown in Figure 3-113 on page 263. After the Status icon turns green, you can monitor the progress of parsing by clicking **Details**, as shown in Figure 3-114 on page 264. Periodically click the **Refresh** button until the parser completes processing, as shown in Figure 3-115 on page 265. Review the parsing statistics.

Note: We stopped the parser to conserve resources in our constrained environment. In a real world environment, you would have the parser running continuously.

We can now proceed to build the main index, as described in “LSTEP4I: Build CUSTINFO collection index” on page 265.

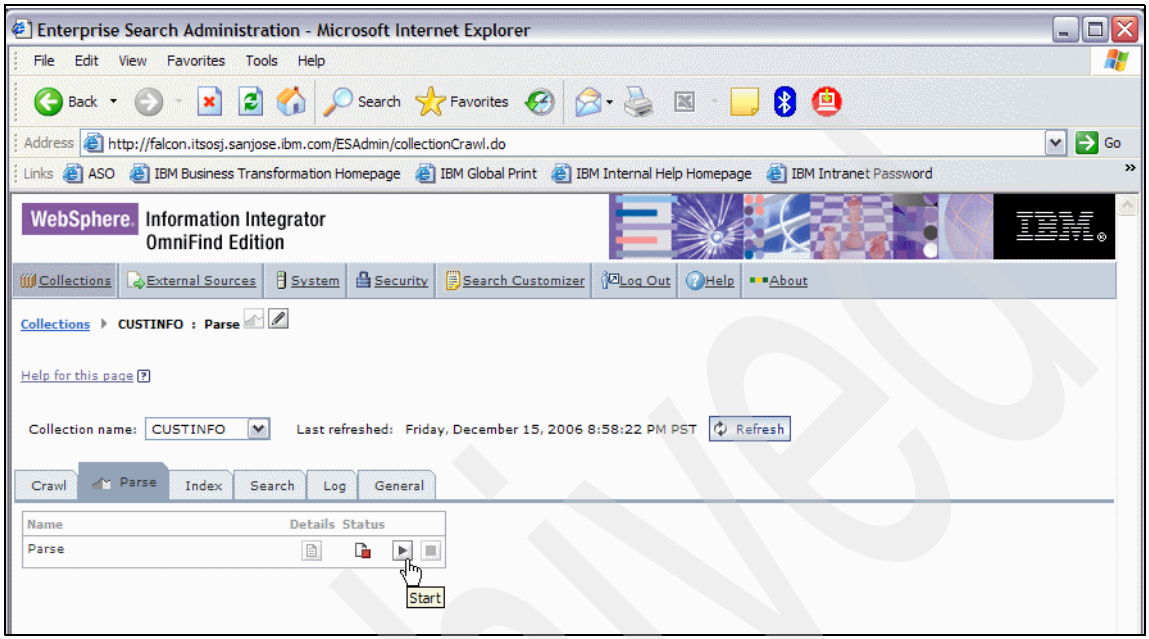

Figure 3-113 Start parser

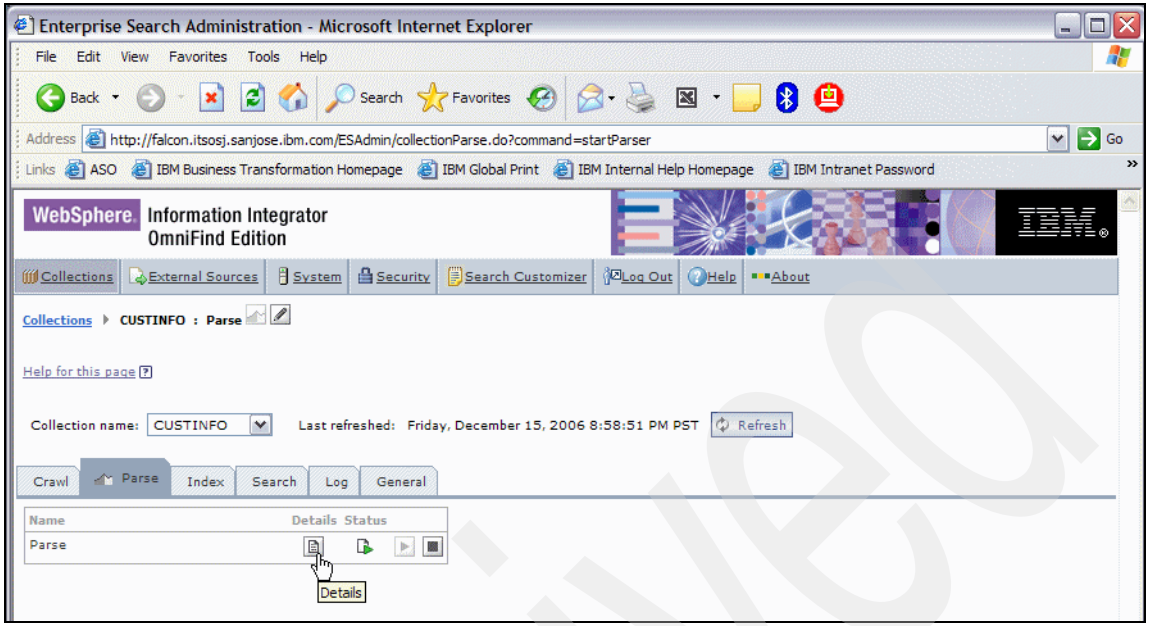

Figure 3-114 Click Details icon

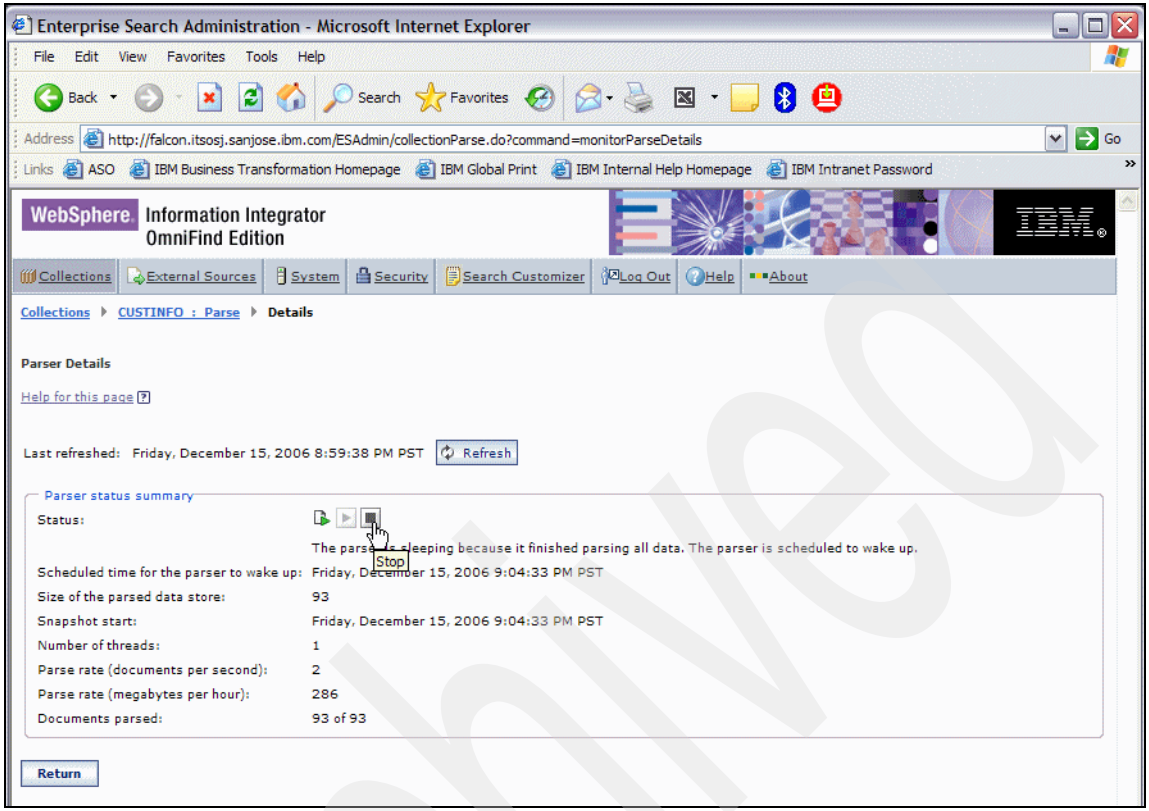

Figure 3-115 Parser completion status

LSTEP4l: Build CUSTINFO collection index

From the Index tab in Monitor mode, start the main index build by clicking the start icon, as shown in Figure 3-116 on page 266. Periodically click the **Refresh** button until the index build completes processing, as shown in Figure 3-117 on page 267. Review the index statistics.

If not started, start the search servers by clicking the start icon from the Collections view in Monitor mode under the Search tab, as shown in Figure 3-118 on page 268.

We can now proceed to configure security for the CUSTINFO and GENINSINFO collections, as described in “LSTEP4m: Define security settings” on page 268.

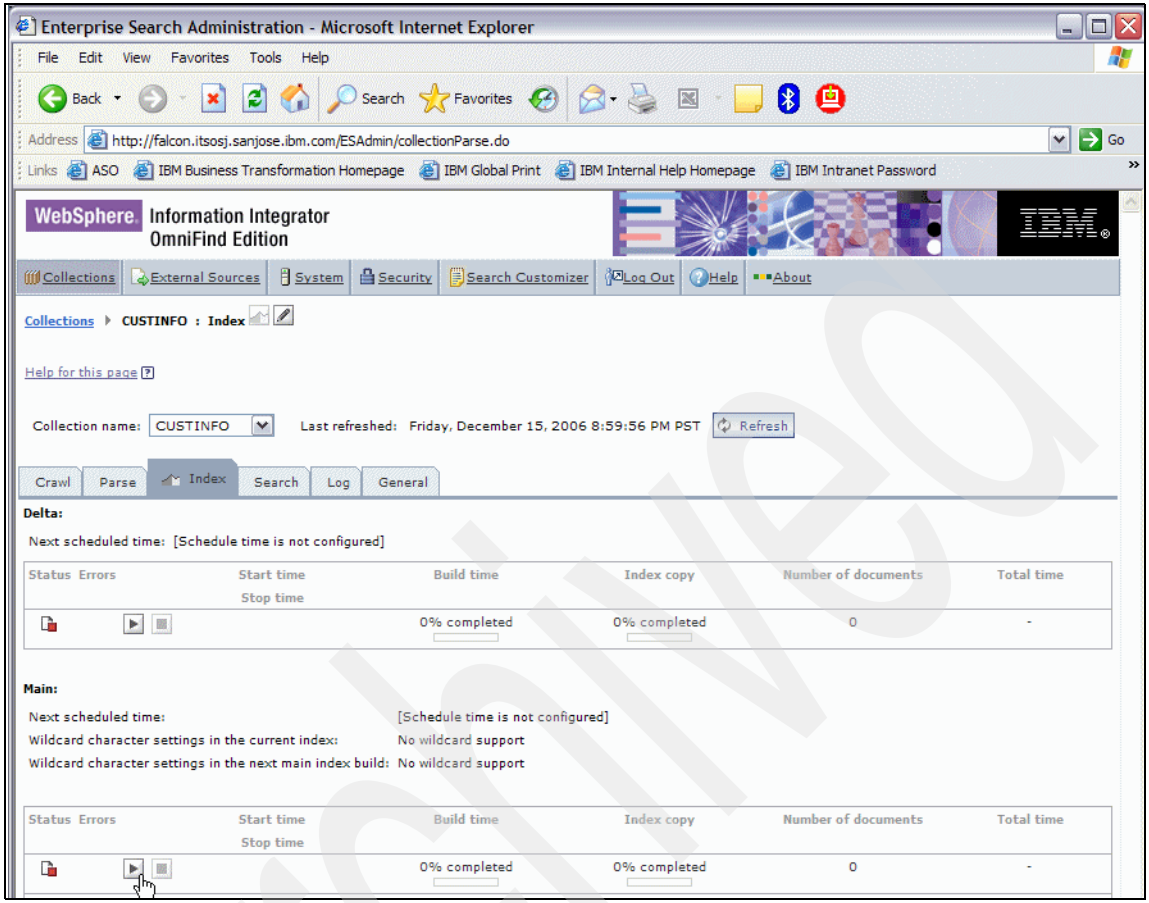

Figure 3-116 Start main index build

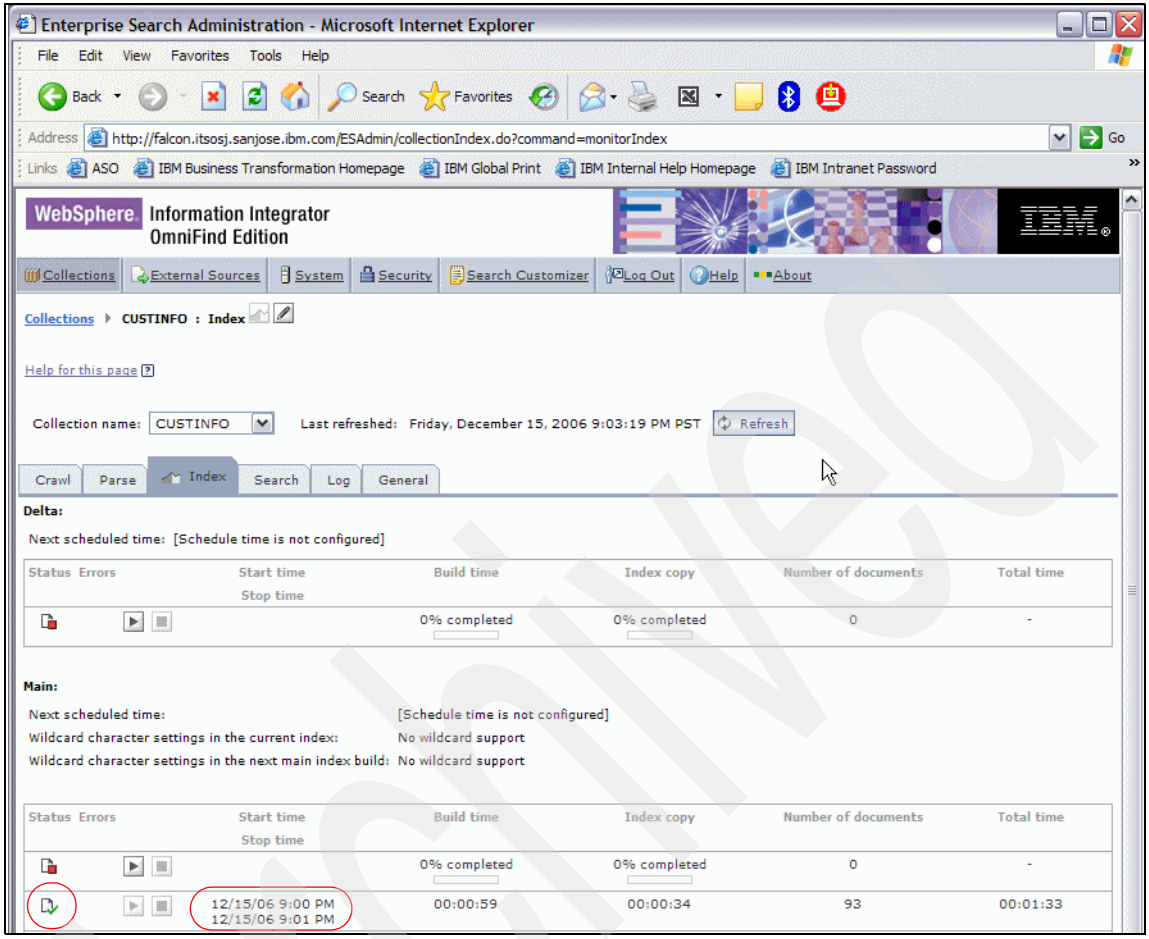

Figure 3-117 Main index build completion

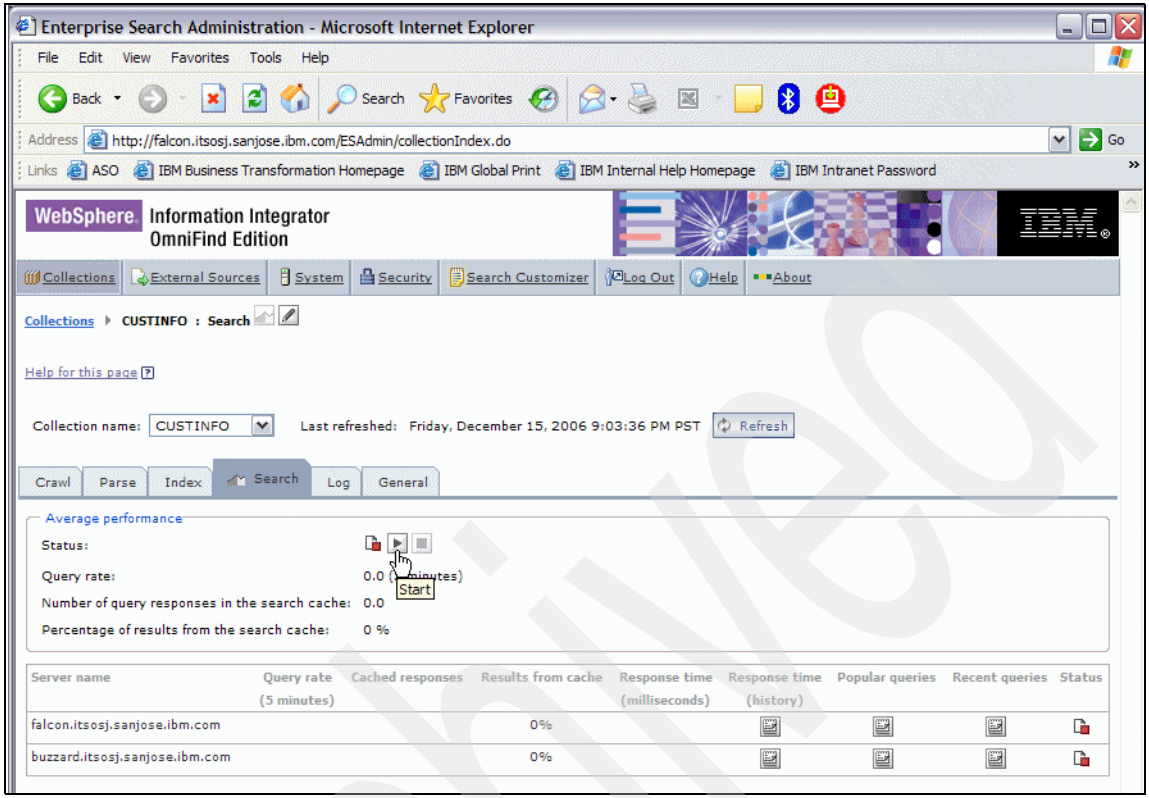

Figure 3-118 Start (both) Search servers

LSTEP4m: Define security settings

In this step, we configure search applications to access the GENINSINFO and CUSTINFO collections, as described in Figure 3-119 on page 269 through Figure 3-125 on page 275. Since the GENINSINFO collection is accessible to all employees within the organization, while the CUSTINFO collection is only accessible by authorized employees, the Default search application is modified to only access the GENINSINFO collection, and a new search application SEQUOIA_secure is defined that has access to both the CUSTINFO and GENINSINFO collections.

Note: We stayed with the defaults for identity management component (enabled) and single sign-on (enabled for all data source types) for the CUSTINFO collection, and therefore do not show the relevant screen captures here. Refer to Figure 2-61 on page 120 through Figure 2-63 on page 121 for the screen captures associated with defining IMC and single sign-on.

Unless specific action is taken, the Default search application name automatically has access to the CUSTINFO and GENINSINFO collections. This is desirable at least until one has tested the CUSTINFO and GENINSINFO collections using the sample search Web application or portlet. After successful testing, you can disable access to the CUSTINFO collection by the Default search application name (Figure 3-119 through Figure 3-121 on page 271). A new search application SEQUOIA_secure is given access to both the CUSTINFO and GENINSINFO collections (Figure 3-122 on page 272 through Figure 3-123 on page 273).

Figure 3-124 on page 274 shows the two search applications defined in the system, while Figure 3-125 on page 275 shows the two collections defined in the system.

We can now proceed to query the CUSTINFO and GENINSINFO collections, as described in 3.3.5, “LSTEP5: Query GENINSINFO and CUSTINFO collections” on page 275.

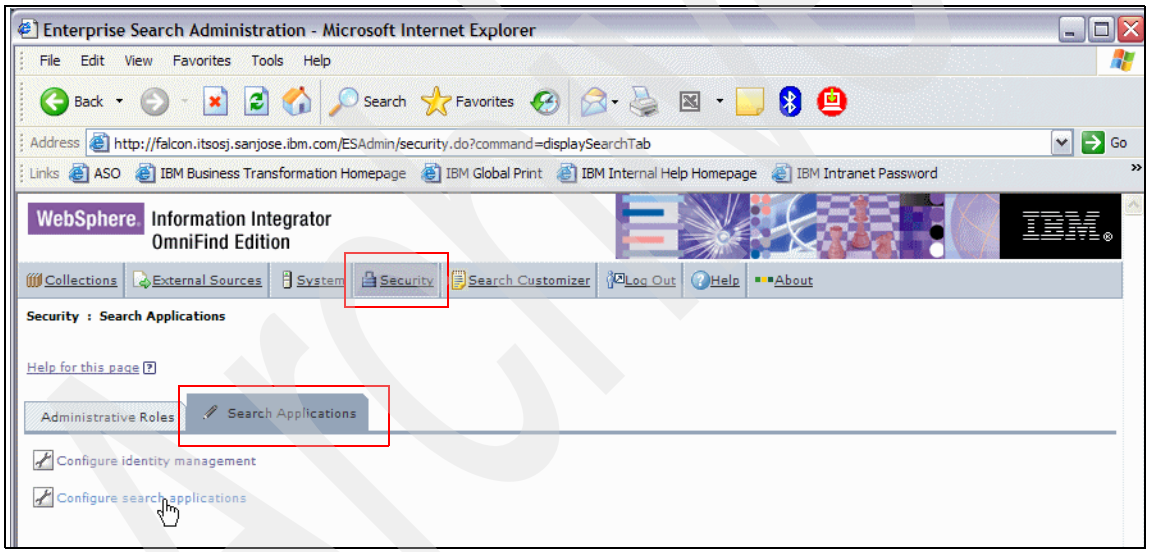

Figure 3-119 Configure search applications

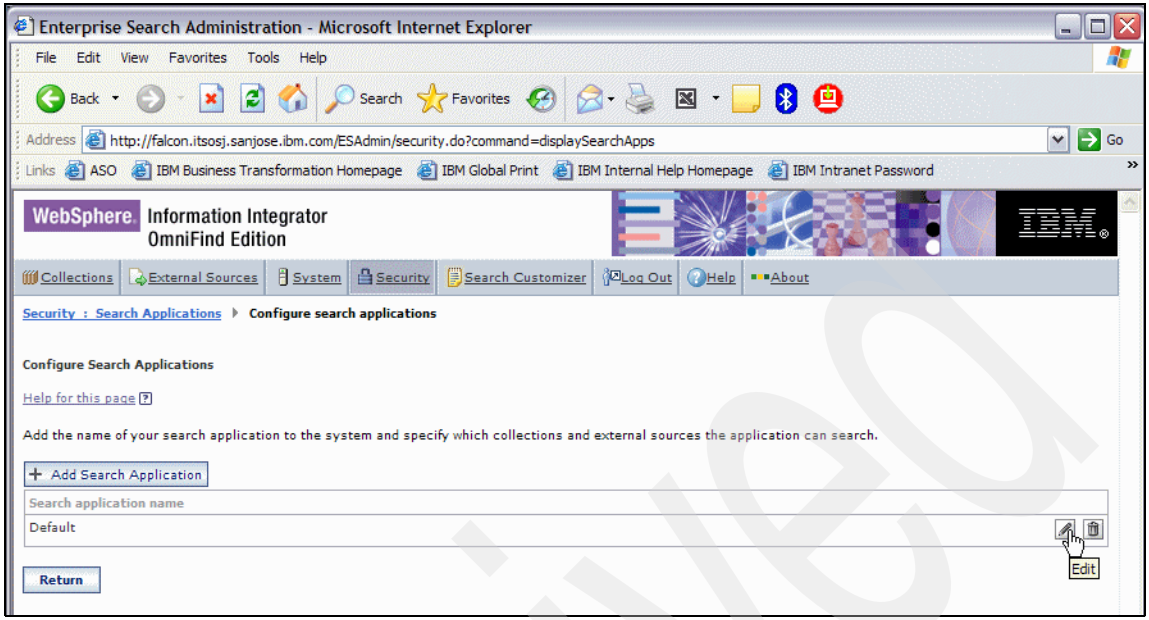

Figure 3-120 Click Edit icon for the Default Search application name

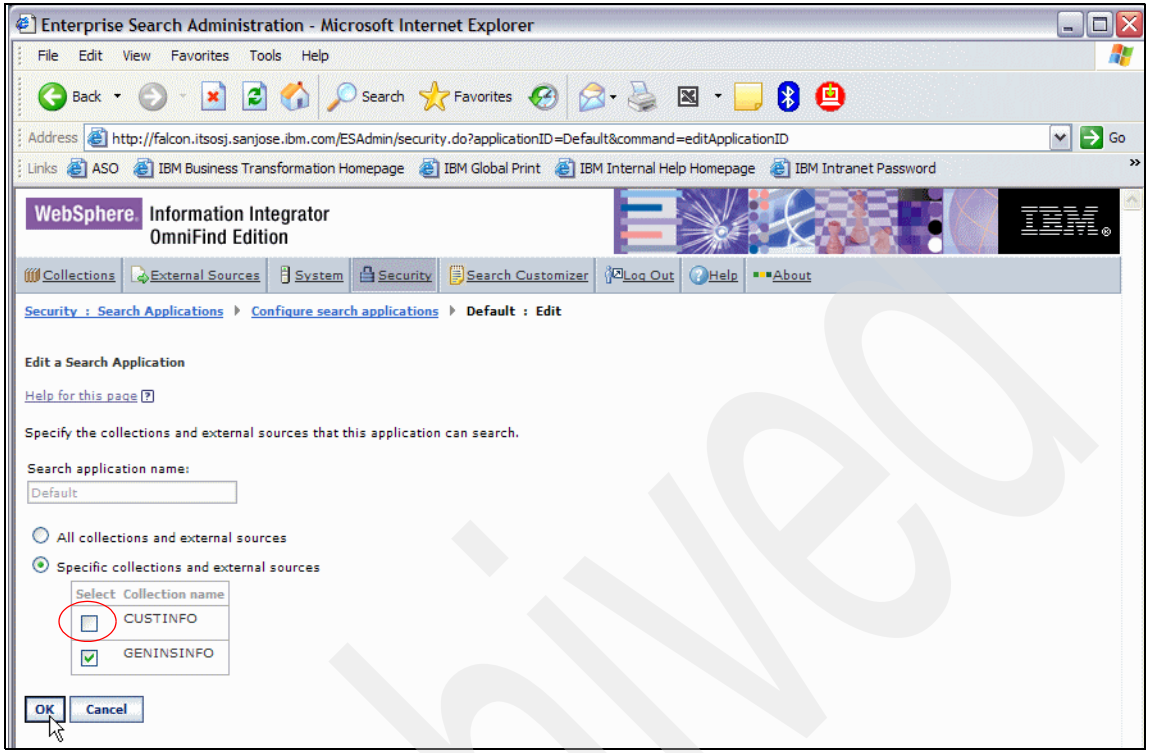

Figure 3-121 Deselect CUSTINFO collection access to Default

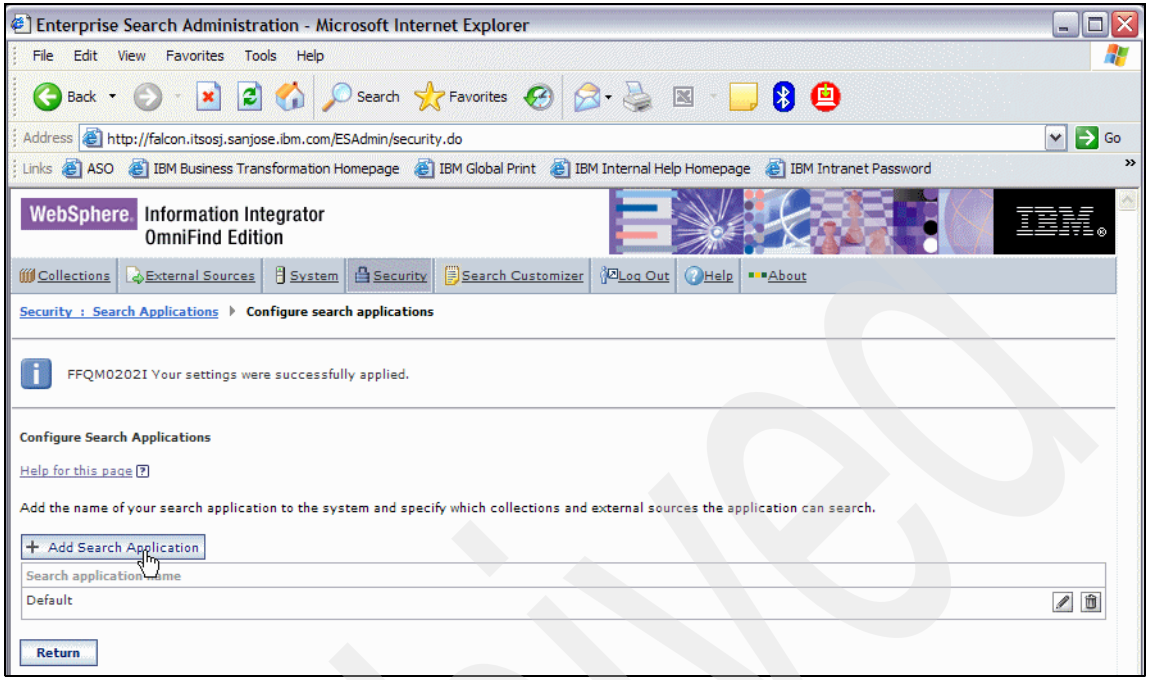

Figure 3-122 Add Search Application 1/2

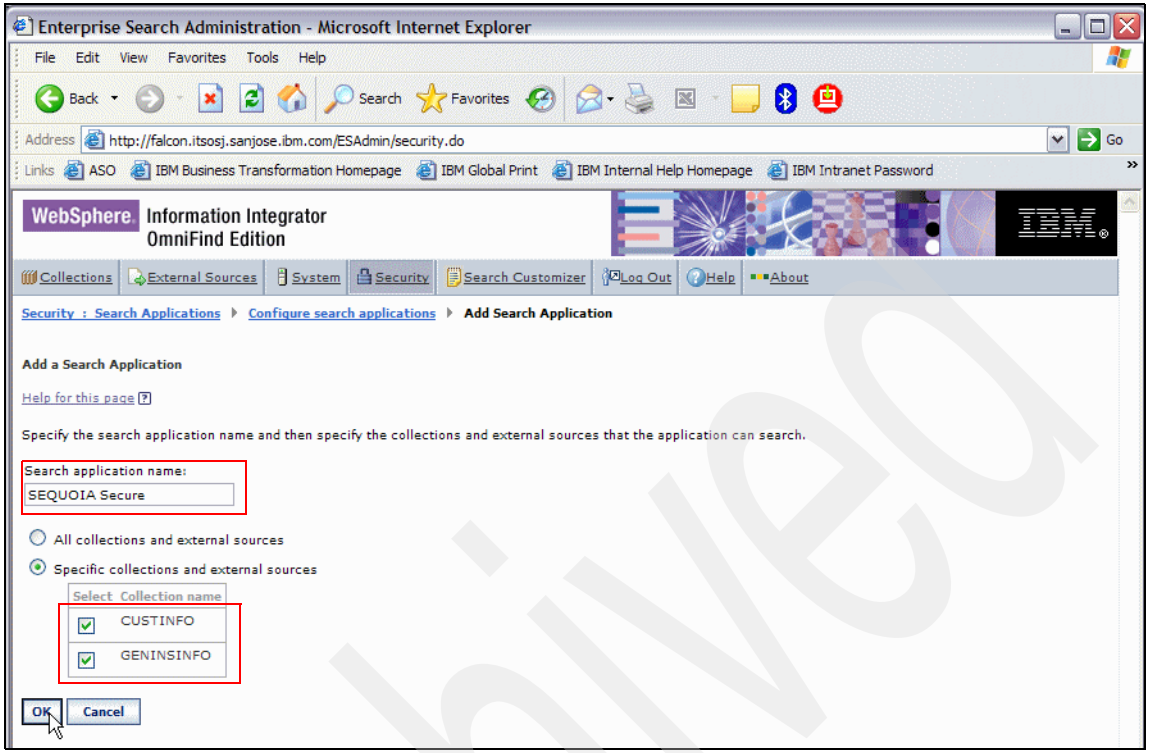

Figure 3-123 Add Search Application 2/2

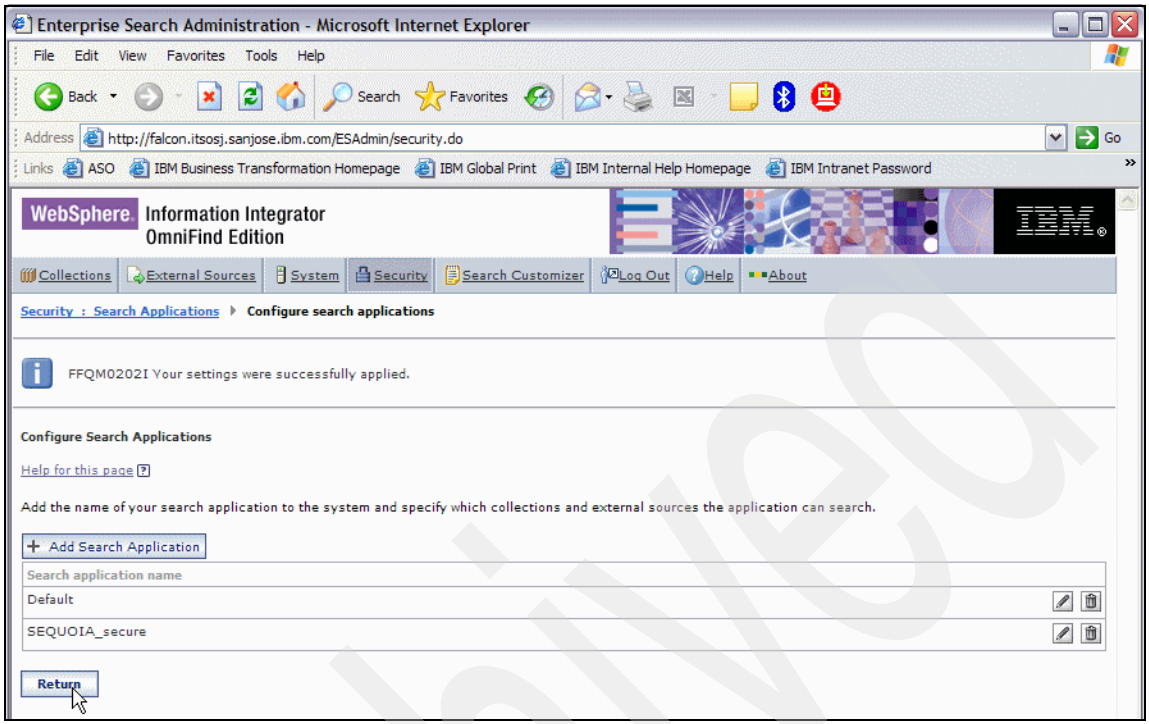

Figure 3-124 Search applications in the system

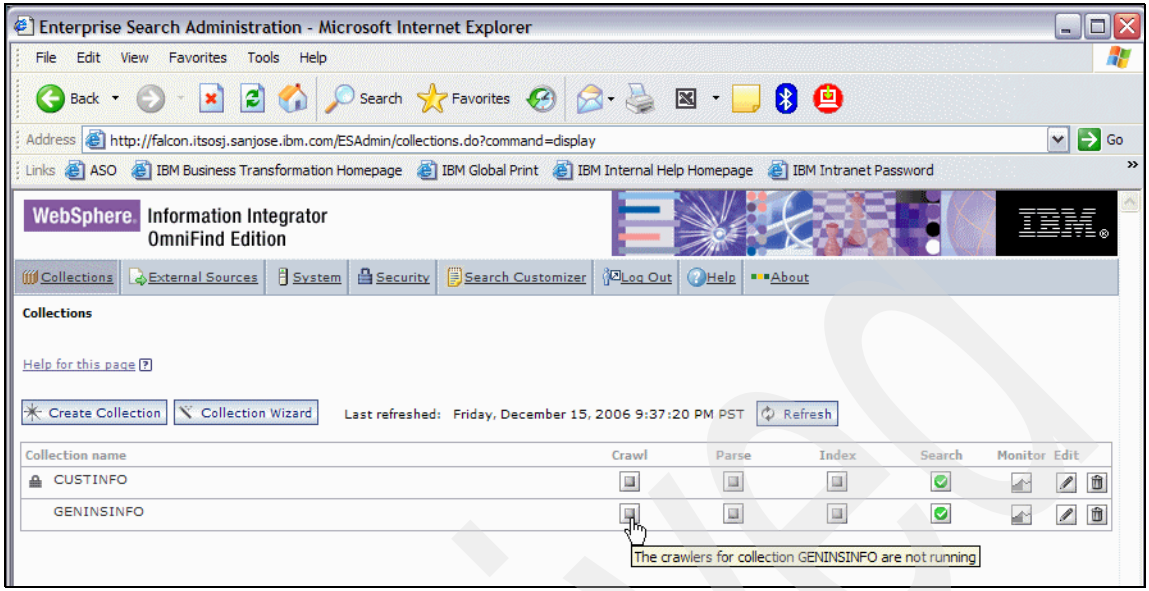

Figure 3-125 Status of the CUSTINFO and GENINSINFO collections with Search running

3.3.5 LSTEP5: Query GENINSINFO and CUSTINFO collections

In this step, we query the GENINSINFO and CUSTINFO collections using the sample search Web application, and the sample search portlet application with a modified config.properties file specifying the SEQUOIA_secure search application ID.

Using the Web sample application

OmniFind Enterprise Edition V8.4 has a new capability to customize a custom search application, type the URL for the Search Application, and append the name of the configuration file for your search application. For example:

```
http://SearchServer.com/ESSearchApplication/search.do?configFile=/WEB-I
NF/myConfig.properties
```

If the file that you specify (myConfig.properties file in the above example) does not exist, values in the config.properties file (which has the applicationName property value of Default, as shown in Example 3-4 on page 276) for the sample search application are used.

Example 3-4 config.properties file

```
# Search Application
applicationName=Default

.........
.......
```

We are showcasing this capability here since the Default search application only has access to the GENINSINFO collection, while the SEQUOIA_secure search application has access to both the GENINSINFO and CUSTINFO collections. This requires the creation of a separate config.properties file named sec_config.properties with the applicationName property changed to SEQUOIA_secure, as shown in Example 3-5.

Example 3-5 sec_config.properties file contents

```
# Search Application
applicationName=SEQUOIA_secure

.........
........
# SSO related values
# The ssoCookieName corresponds to the name of the HTTP header
# that contains the Single Sign On cookie value.
ssoCookieName=LtpaToken
.........
.......
```

Accessing the GENINSINFO collection

Figure 3-126 on page 277 through Figure 3-129 on page 280 describe the user interactions searching the GENINSINFO collection.

The Sample Search application is invoked without specifying a config.properties file (http://falcon.itsosj.sanjose.ibm.com/ESSearchApplication/), which implicitly uses the config.properties file. After logging in to the sample search Web application with the esadmin user ID (Figure 3-126 on page 277), you are presented with the Search box, as shown in Figure 3-127 on page 278. Click the **Preferences** link in Figure 3-127 on page 278 to view the default preferences and modify it as required. Figure 3-128 on page 279 shows a number of options that can be modified for the search session, including inclusion of quick links, the data sources to be searched, file types to be excluded/included, number of results per page, and whether to search for synonyms. By default, if multiple collections exist, the sample search application automatically searches across all the collections using a “remote” federator”.[8] In this case, since only the Default

application only has access to the GENINSINFO collection, only that collection appears in the list of choices in the Preferences window. Click **Apply** to save the choices made.

In Figure 3-129 on page 280, enter the string “insurance” in the search box and click **Search**. The results of this query is shown in Figure 3-129 on page 280.

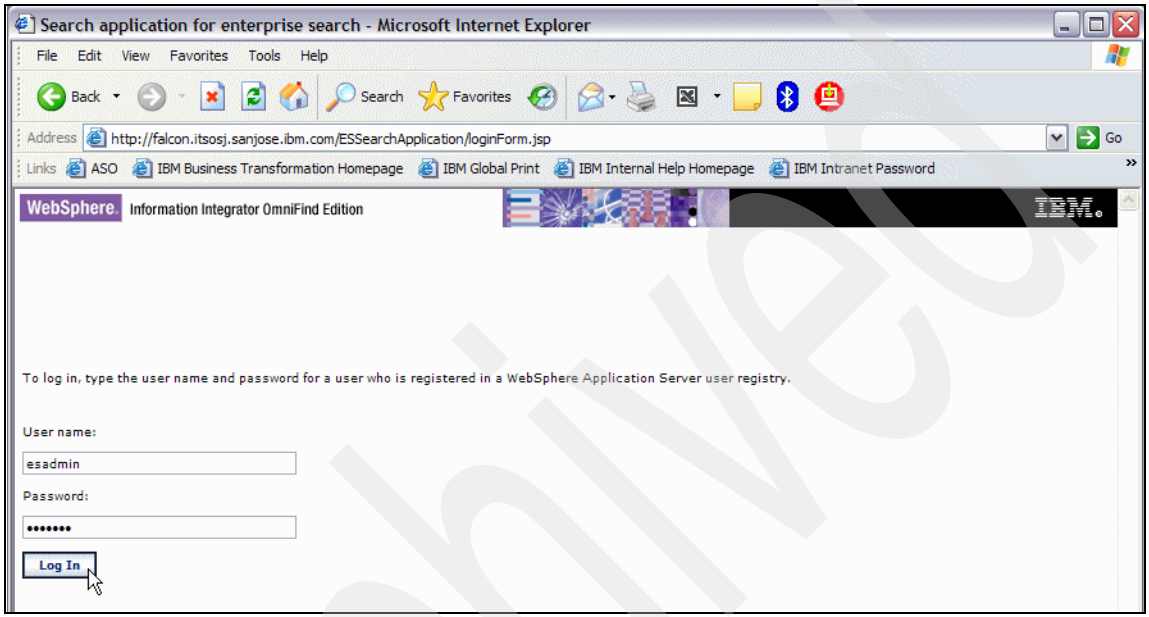

Figure 3-126 Login to Sample Search application

[8] A federator is used to issue a federatedsearch request across a set of heterogeneous searchable collections and get a unified document result set. Search federators are intermediary components that exist between the requestors of service and the agents that perform that service. They coordinate resources to manage the multitude of searches that are generated from a single request. Enterprise search provides “local” and “remote” federators. A local federator federates from the client over a set of searchable objects across collections in more than one server, while a remote federator federates from a server over a set of collections in the same server. For a brief description of federators, refer to Chapter 5, “Merger of SMB and medium-size organizations” on page 409. For more detailed information about federators, refer to the *IBM OmniFind Enterprise Edition V8.4 Programming Guide and API Reference for Enterprise Search*, SC18-9284.

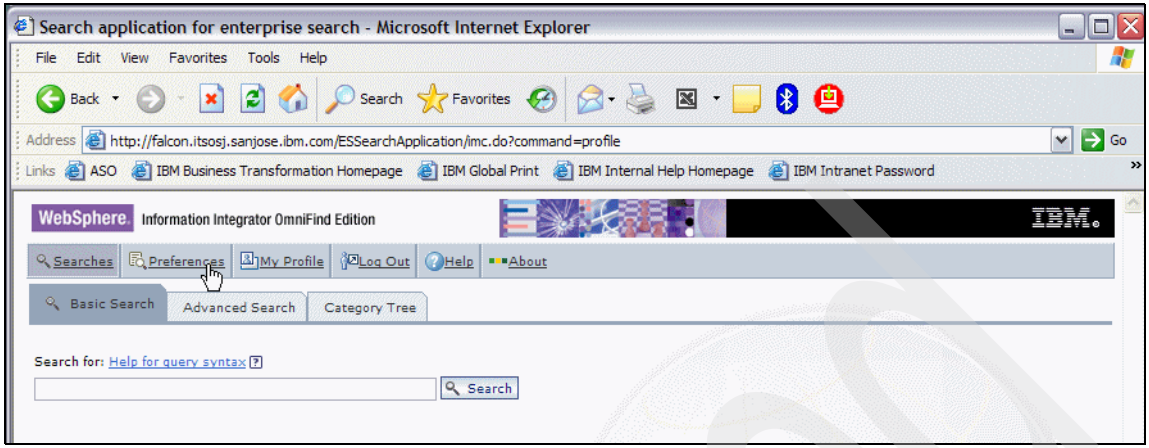

Figure 3-127 Search box

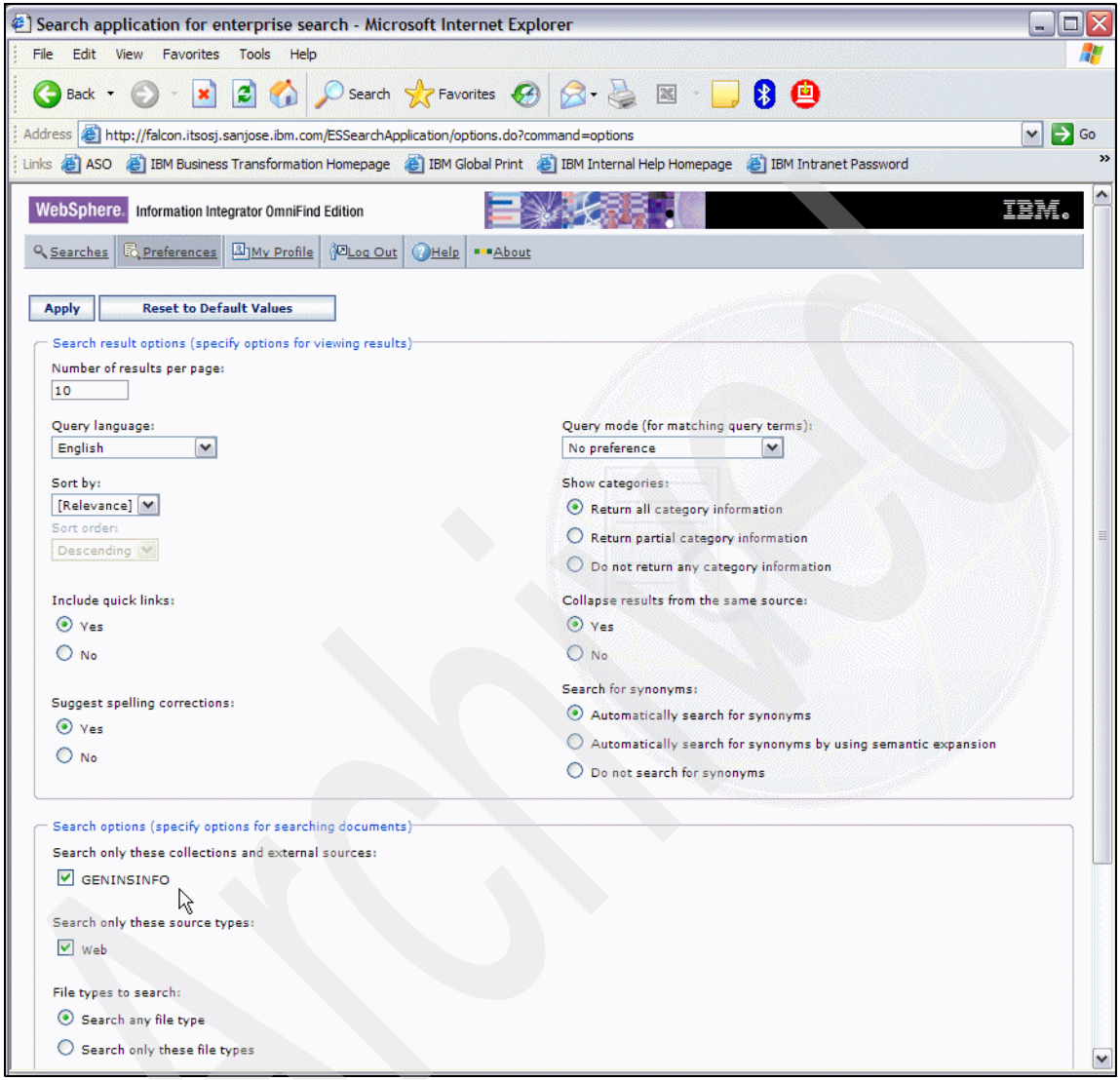

Figure 3-128 Preferences options

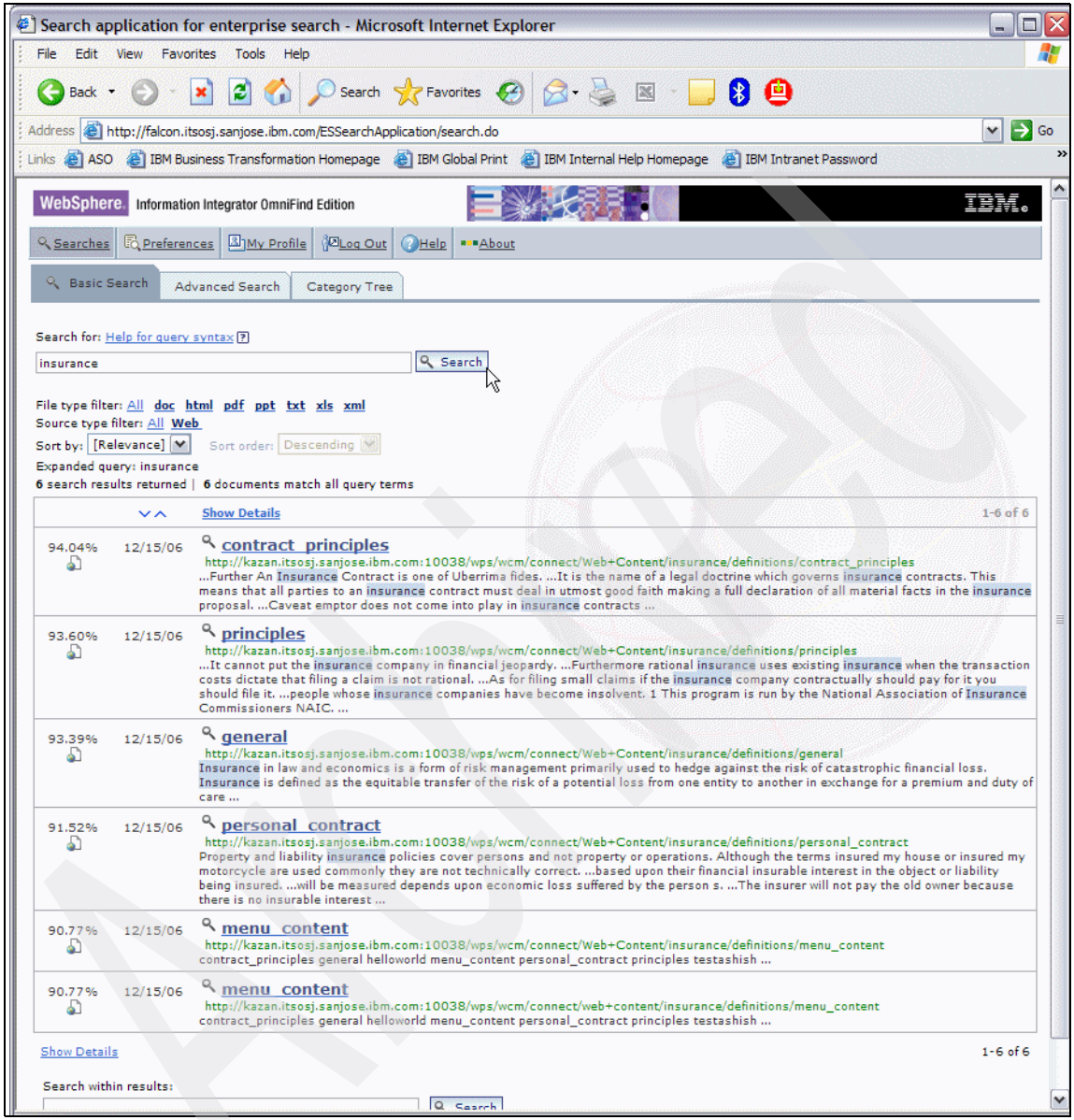

Figure 3-129 Search results for “insurance”

Accessing the CUSTINFO and GENINSINFO collections

Figure 3-130 on page 282 through Figure 3-137 on page 288 describe the user interactions while searching the CUSTINFO and GENINSINFO collections.

The Sample Search application is invoked by specifying a sec_config.properties file, as shown in Figure 3-130 on page 282.

After logging in to the sample search Web application with the esadmin user ID (Figure 3-131 on page 283), since IMC is enabled without SSO, the search runtime prompts the user for the Windows PDM credentials, as shown in Figure 3-132 on page 283. Provide the credentials and click **Apply**, which takes you to the search window (Figure 3-133 on page 284). Since IMC and SSO support for WebSphere Portal is enabled, no prompts of it are presented, as the search runtime has the required credentials in the LTPA token.

Figure 3-134 on page 285 shows the search results for the string “john smith”, which has results from the PDM and WCM data sources, but not any results from WebSphere Portal.

Note: To search secure WebSphere Portal pages, you must submit searches by using the Search portlet for enterprise search from within WebSphere Portal. Searches submitted from the sample search Web application, ESSearchApplication, do not have the proper credentials and cannot verify the user’s authority to access documents.

Click the **Preferences** link in Figure 3-134 on page 285 to view the default preferences and modify it as required. As mentioned earlier, Figure 3-135 on page 286 shows a number of options that can be modified for the search session, including inclusion of quick links, the data sources to be searched, file types to be excluded/included, number of results per page, and whether to search for synonyms. Also by default, if multiple collections exist, the sample search application automatically searches across all the collections using a “remote” federator. In this case, two collections (CUSTINFO and GENINSINFO) have been defined, and since the SEQUOIA_secure search application has access to both these collections, they both appear in the list of choices in the preferences window along with their corresponding data sources.

Note: We selected the **Automatically search for synonyms by using semantic expansion** option in the Search for synonyms section. This is required to leverage the IBM_TAE text analysis engine and REGEX_DICT synonym dictionary configurations in the search queries.

Click **Apply** to save the choices made.

Figure 3-136 on page 287 shows the search results for the string “e-mail address”. The search results highlight not only the occurrences of the string “e-mail” (document named Customer Book3.xls) and “address” (document Customer Book7.xls), but also actual e-mail addresses, such as kulakowd_zus@foo.com in document Customer Book1.xls. This is the semantic query aspect where search recognizes that kulakowd_zus@foo.com is an e-mail address and therefore qualifies to appear in the search result when the string “e-mail address” is entered in the search box.

Figure 3-137 on page 288 shows the search results for the string “URL”. The search result has a single document CATALACCESS FR, which highlights the URL “www.espresso.com” even though the string “URL” does not explicitly appear in the document. This is the semantic query aspect where search recognizes that www.espresso.com is a URL and therefore qualifies to appear in the search result when the string “URL” is entered in the search box.

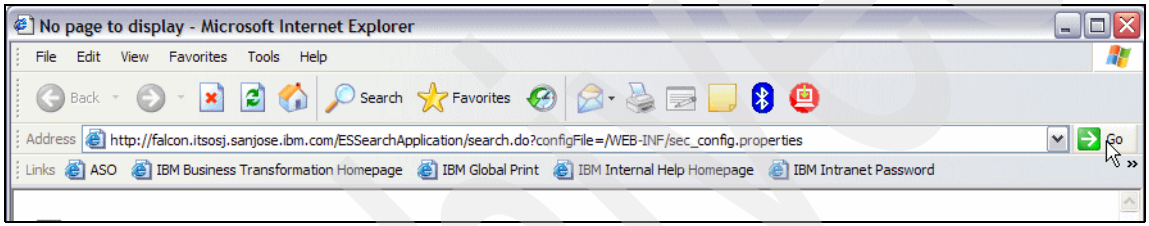

Figure 3-130 Sample Search application invocation with sec_config.properties file

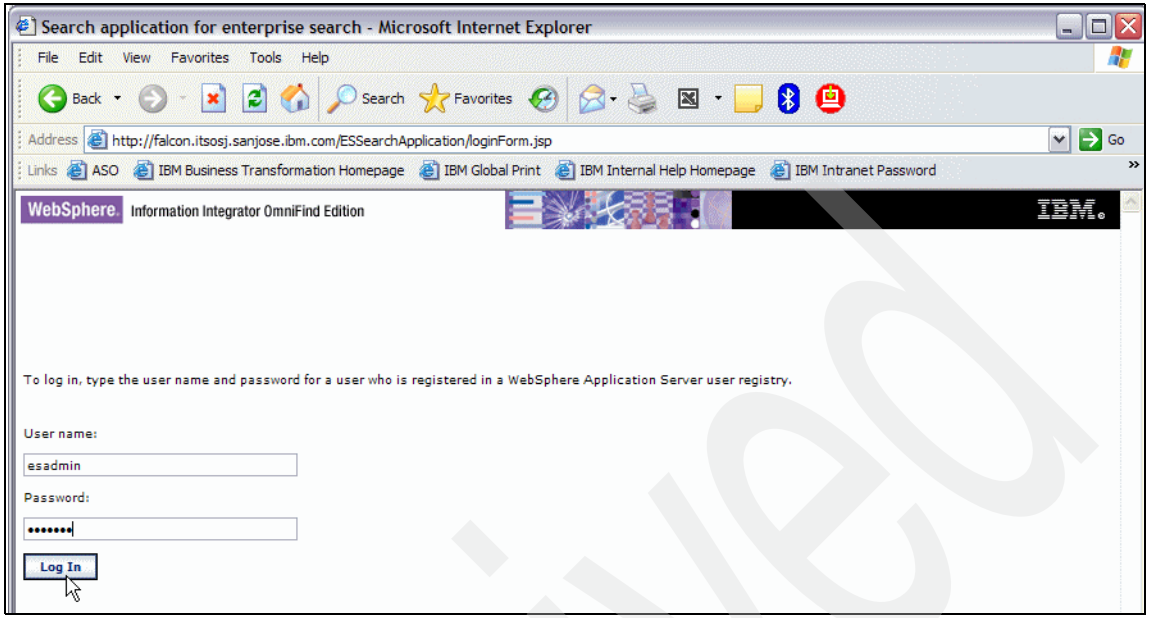

Figure 3-131 Log in to Sample Search application

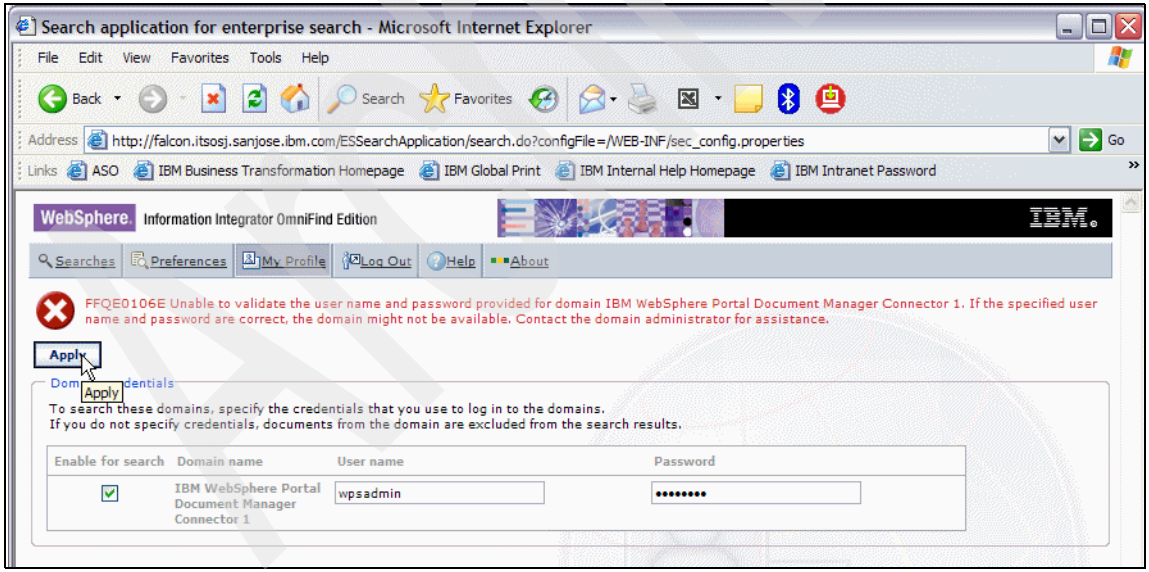

Figure 3-132 Identity management component (IMC) credentials prompt for Portal Document Manager

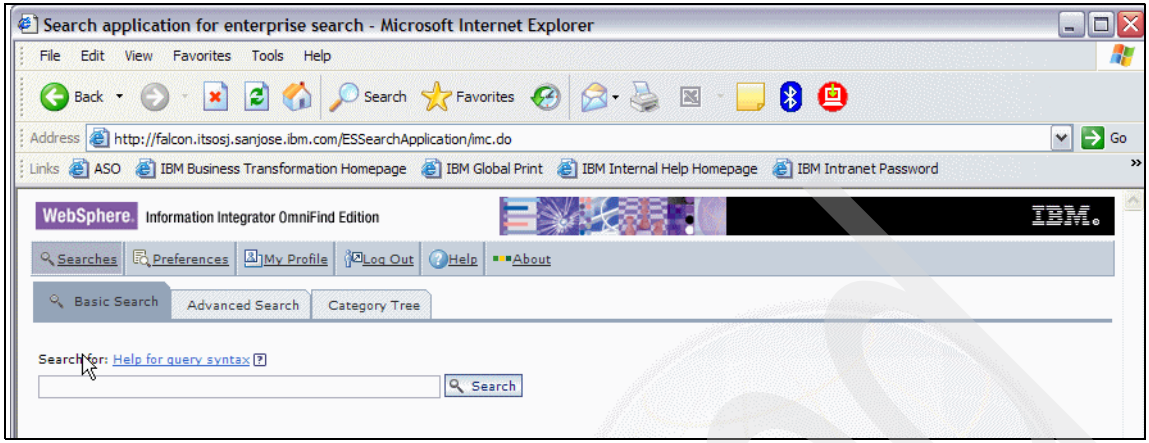

Figure 3-133 Search box

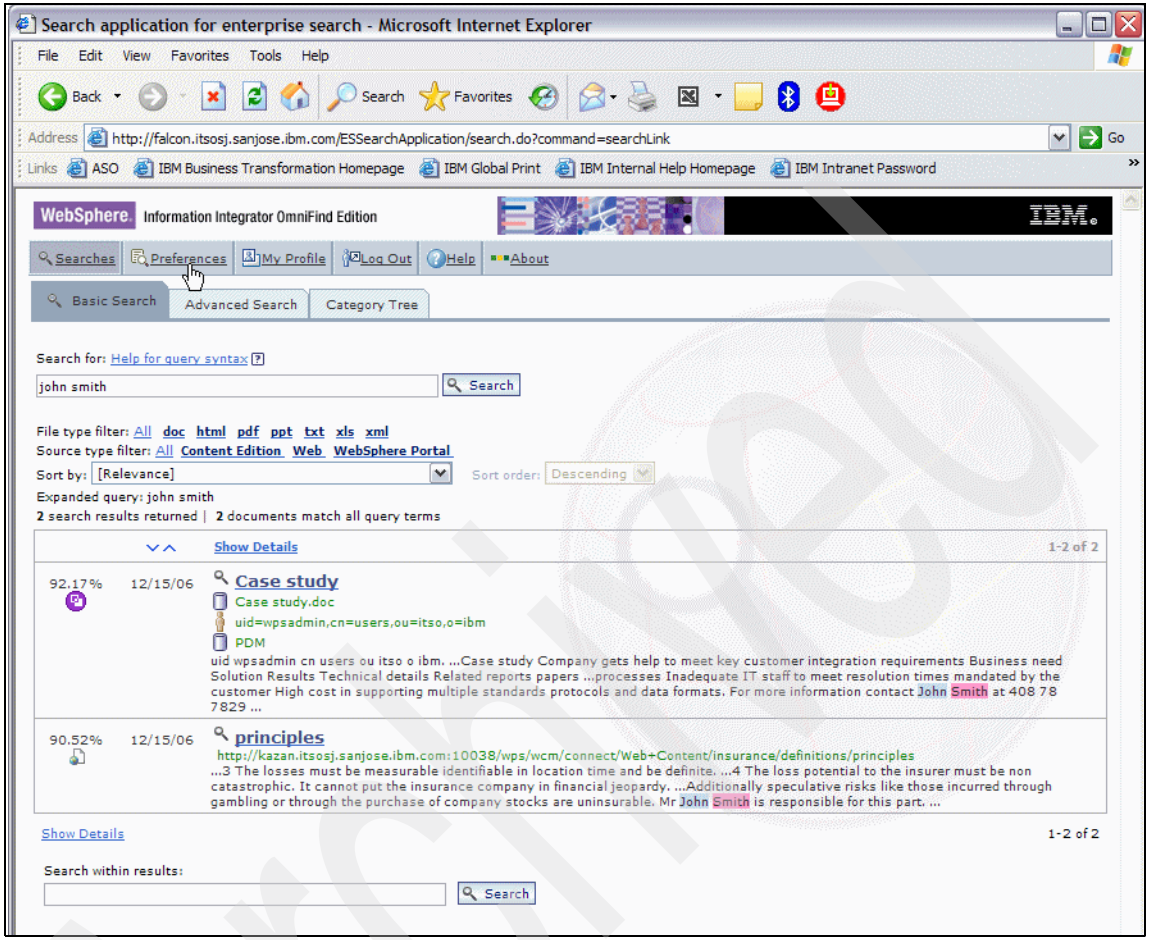

Figure 3-134 Search results for “john smith”

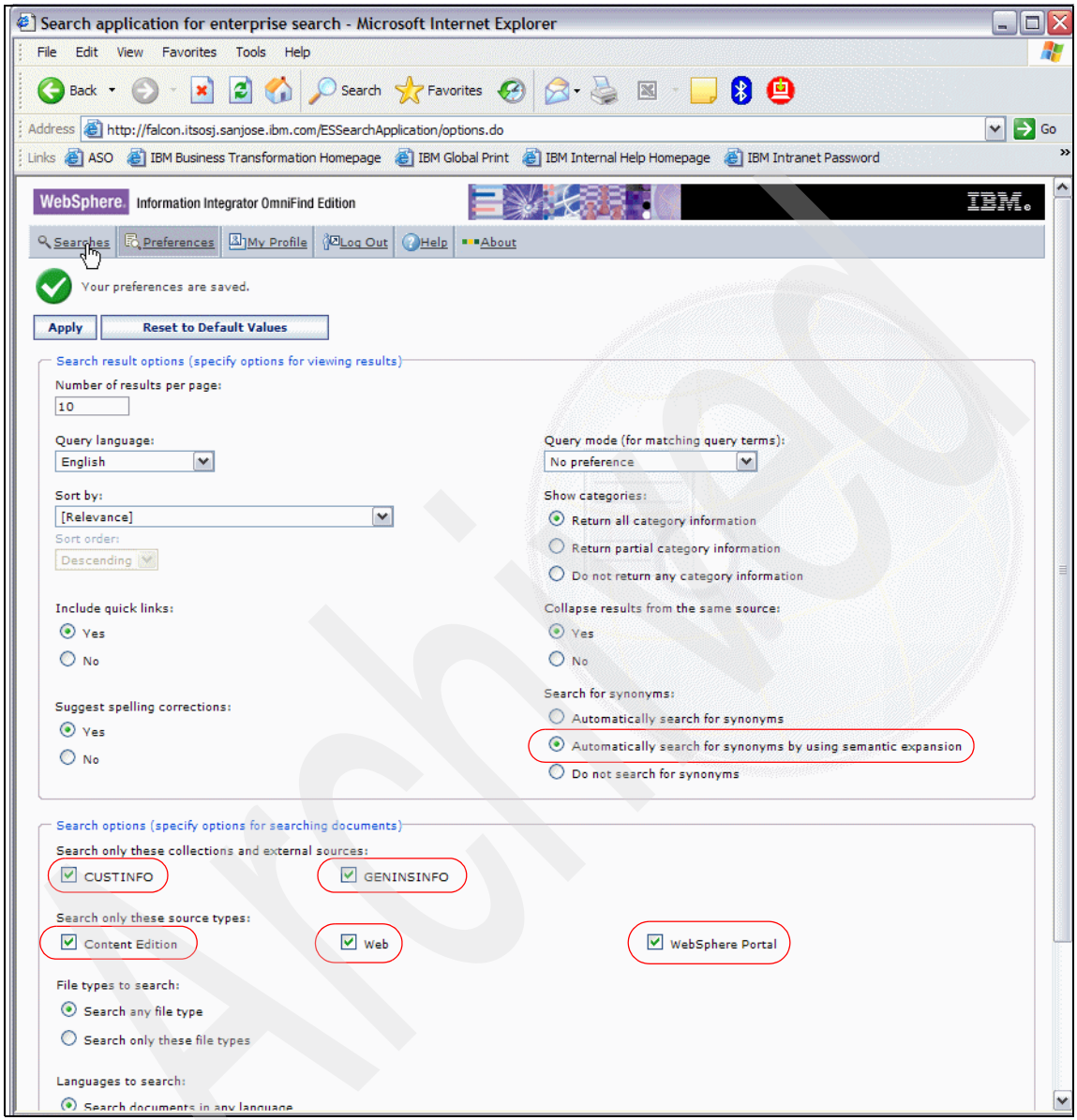

Figure 3-135 Preferences options

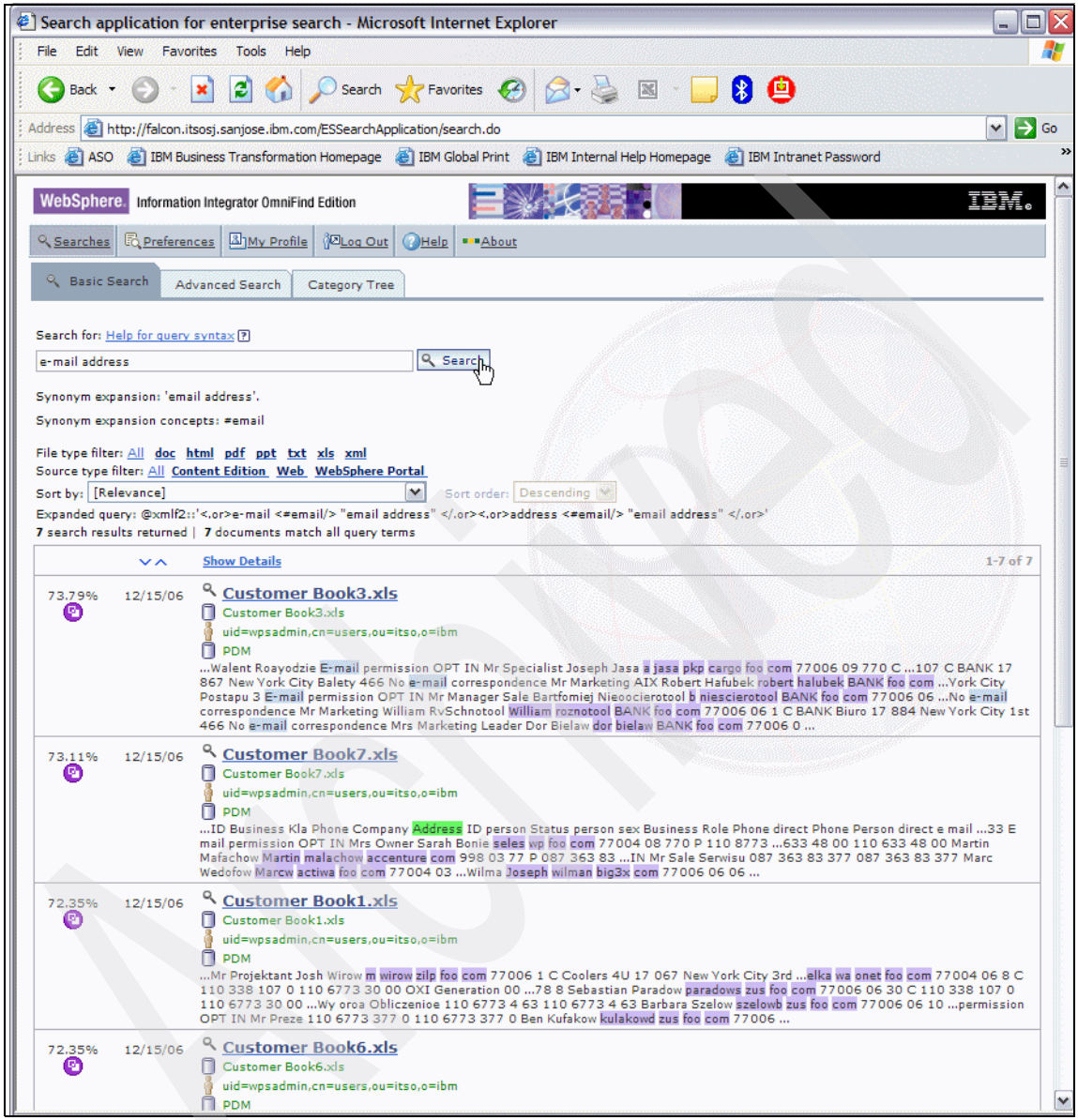

Figure 3-136 Search results for “e-mail address”

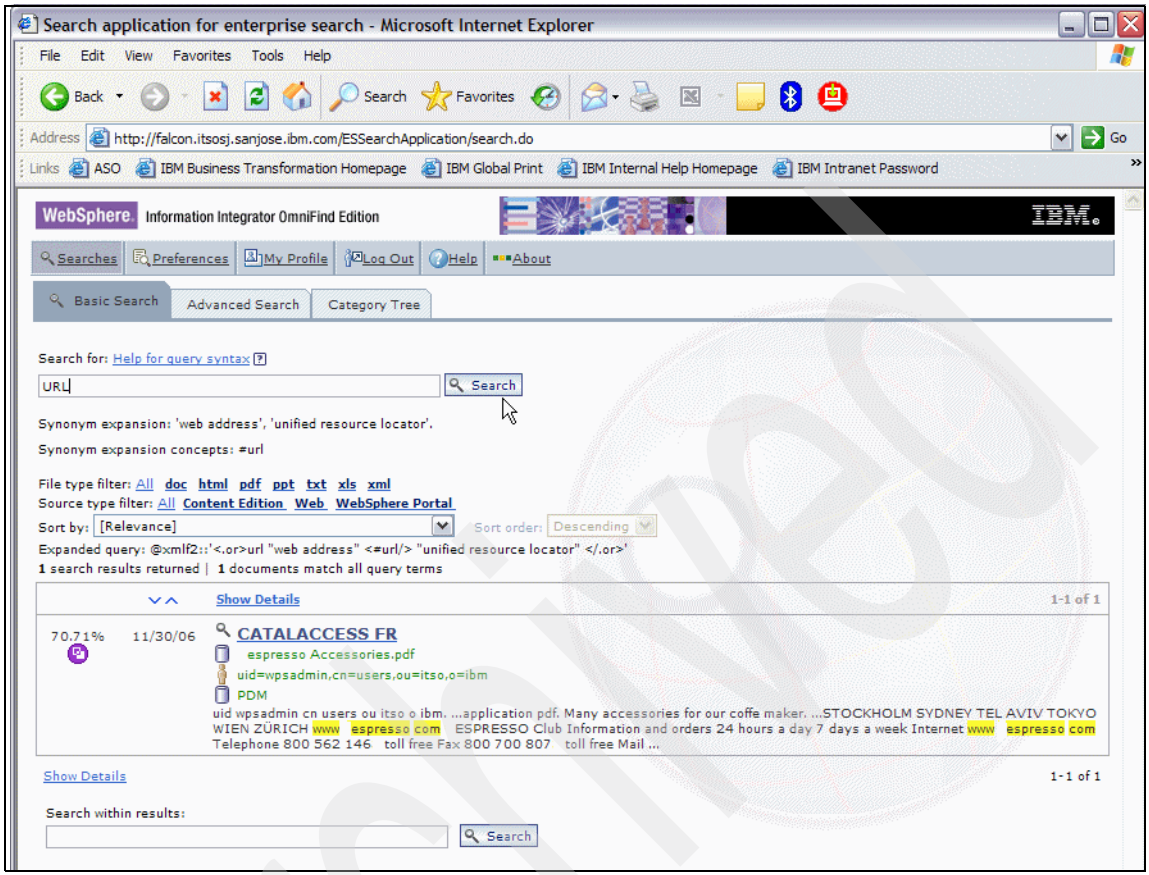

Figure 3-137 Search results for "URL"

Using the sample search portlet

In this step, we used the sample search portlet to access the GENINSINFO and CUSTINFO collections. Appendix A, "Install Sample Search application portlet" on page 431 describes the installation of the sample search portlet in WebSphere Portal Server.

Accessing the GENINSINFO collection

Figure 3-138 on page 289 through Figure 3-140 on page 291 describe some user interactions with the sample search portlet that uses the applicationName property value of Default. The user logs in to the WebSphere Portal, as shown in Figure 3-138 on page 289.

Invoke the sample search portlet from the OmniFind-Linux tab, click the **Preferences** link to view the various options, as shown in Figure 3-139 on page 290. It shows that this portlet only has access to the GENINSINFO collection. Click **Apply** to save any changes.

Figure 3-140 on page 291 shows the search results for string “insurance”; this matches the results obtained from the sample search Web application in Figure 3-129 on page 280.

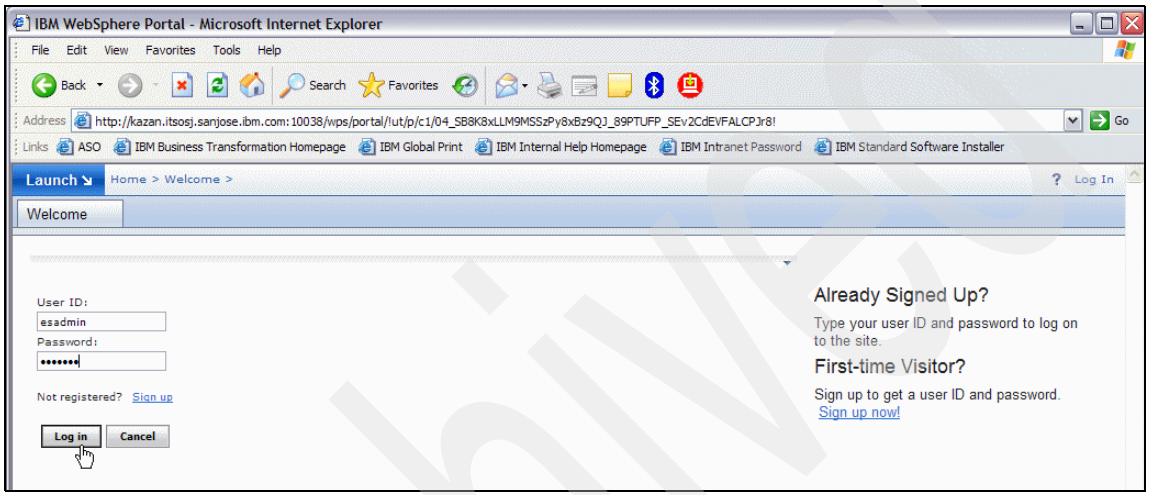

Figure 3-138 Login to WebSphere Portal Server

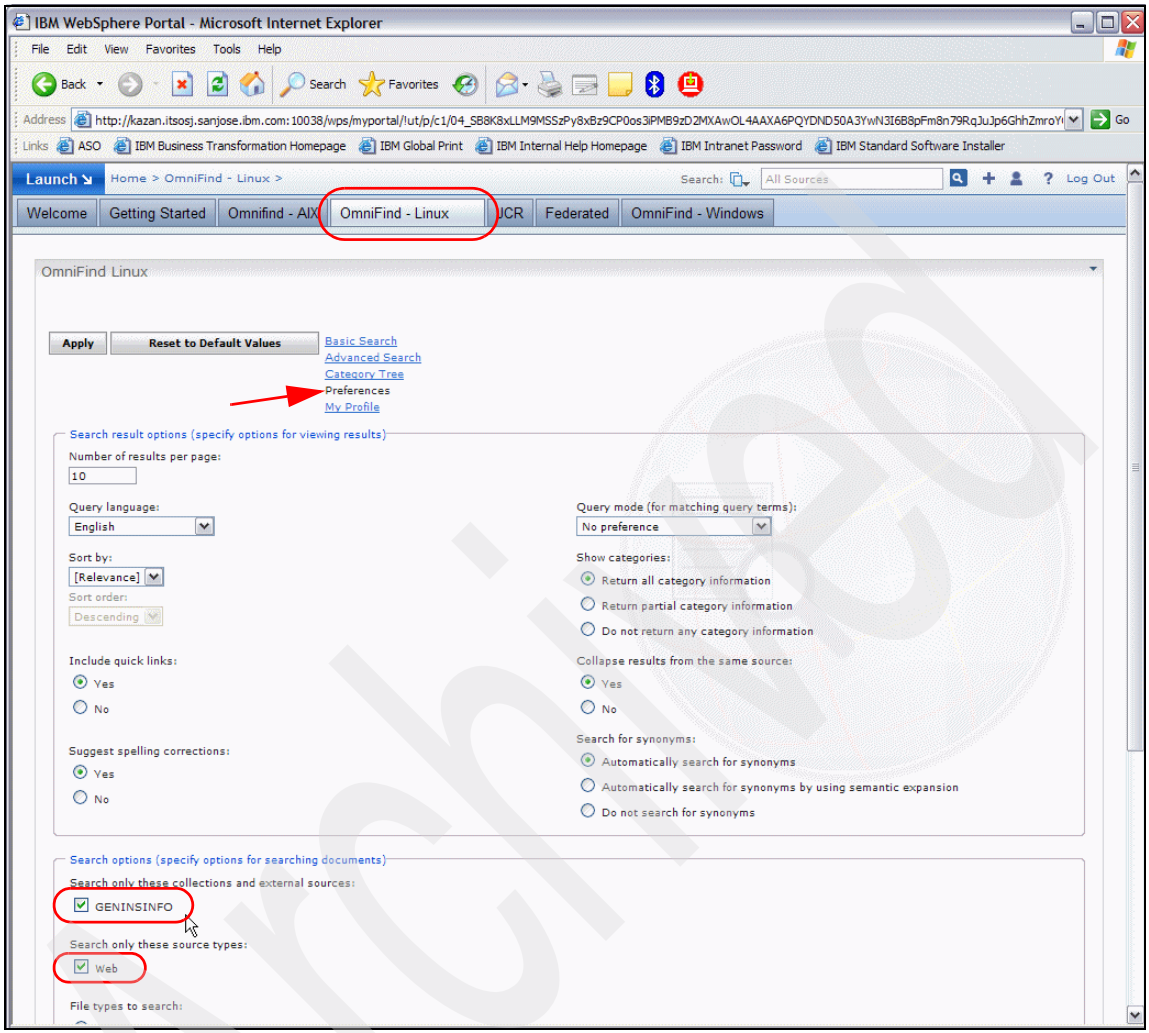

Figure 3-139 Preferences options

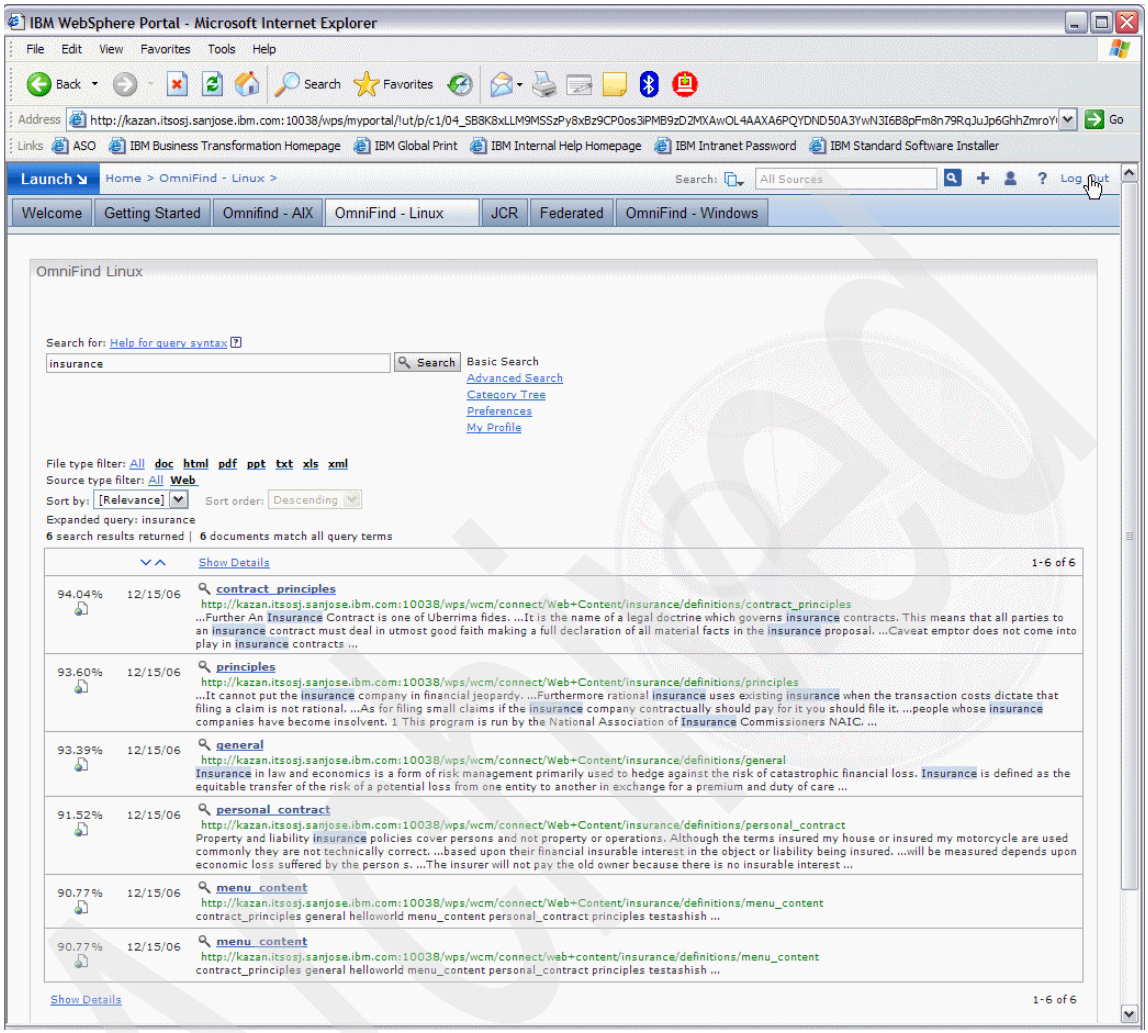

Figure 3-140 Search results for "insurance"

Accessing the CUSTINFO and GENINSINFO collections

Figure 3-141 on page 293 through Figure 3-147 on page 298 describe the user interactions searching the CUSTINFO and GENINSINFO collections.

After logging in to the WebSphere Portal with the esadmin user ID (Figure 3-141 on page 293), invoke the sample search portlet by clicking the **OmniFind-Linux** tab, as shown in Figure 3-142 on page 293. Since IMC and SSO are enabled, the search runtime prompts the user for the WebSphere PDM credentials, as shown in Figure 3-143 on page 294. Provide the credentials and click **Apply.** Since

single sign-on support for WebSphere Portal is enabled, no prompts are presented for it, as the search runtime has the required credentials in the LTPA token.

Figure 3-144 on page 295 shows the search results for the string “john smith”, which has results from the PDM, WCM, and WebSphere Portal data sources. The results show four results, as compared to two in Figure 3-134 on page 285, since a search request is submitted from the search portlet and now includes two additional qualifying documents from WebSphere Portal.

Click the **Preferences** link in Figure 3-144 on page 295 to view the default preferences and modify it as required. As mentioned earlier, Figure 3-145 on page 296 shows a number of options that can be modified for the search session, including inclusion of quick links, the data sources to be searched, file types to be excluded/included, number of results per page, and whether to search for synonyms. Also by default, if multiple collections exist, the sample search application automatically searches across all the collections using a “remote” federator. In this case, two collections (CUSTINFO and GENINSINFO) have been defined, and since the SEQUOIA_secure search application has access to both these collections, they both appear in the list of choices in the preferences window along with their corresponding data sources.

Note: Here again, we selected the Automatically search for synonyms by using the semantic expansion option in the Search for synonyms section. This is required to leverage the IBM_TAE text analysis engine and REGEX_DICT synonym dictionary configurations in the search queries.

Click **Apply** to save the choices made.

Figure 3-146 on page 297 shows the search results for the string “url club”. The search result has a single document CATALACCESS FR, which highlights the URL “www.espresso.com”, even though the string “url” does not explicitly appear in the document. This is the semantic query aspect where search recognizes that www.espresso.com is a URL and therefore qualifies to appear in the search result when the string “url” is entered in the search box. The string “club” is highlighted in the search result.

Figure 3-147 on page 298 shows the search results for the string “john smith phone number”. The search results highlight not only the occurrences of the string “john smith” (document named Case study) and an actual phone number 877-123-4567, even though the string “phone number” does not explicitly appear in the Case study document. This is the semantic query aspect where search recognizes that 877-123-4567 is a phone number, and therefore qualifies to appear in the search result when the string contains “phone number” is entered in the search box.

Note: At this point, we have the functional requirements met for the enterprise search solution for Sequoia General Inc. Further tests, measurements, and tuning need to be performed to ensure that the solution fully addresses the capacity and workload requirements as well.

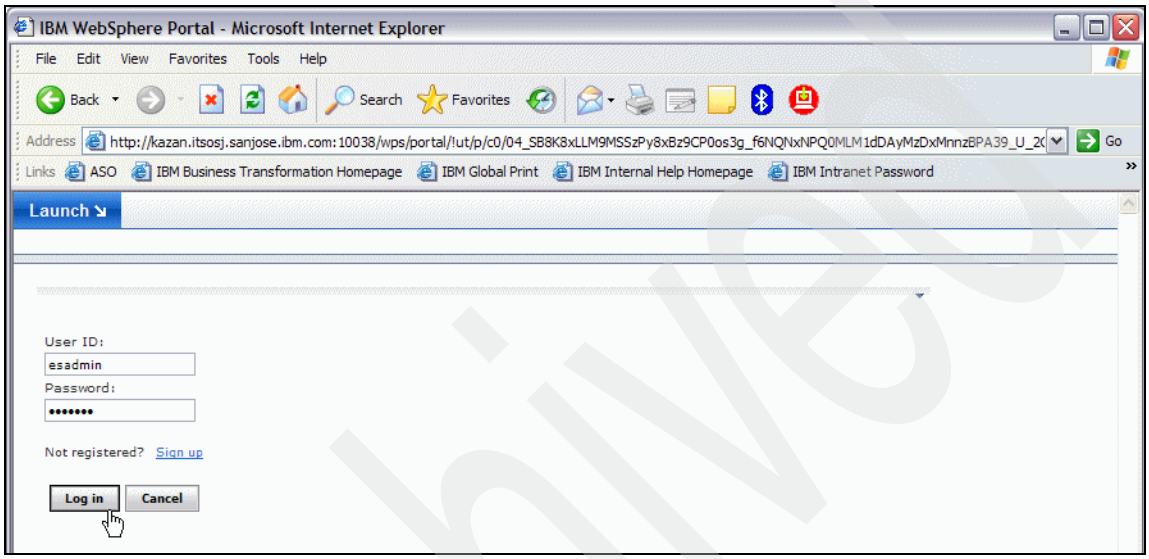

Figure 3-141 Log in to WebSphere Portal

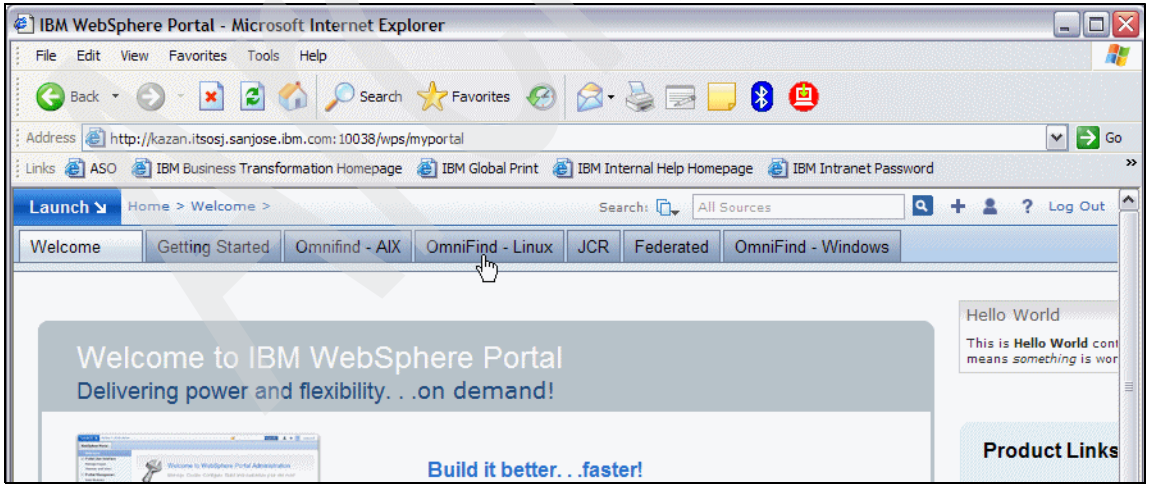

Figure 3-142 Invoke the OmniFind-Linux search portlet

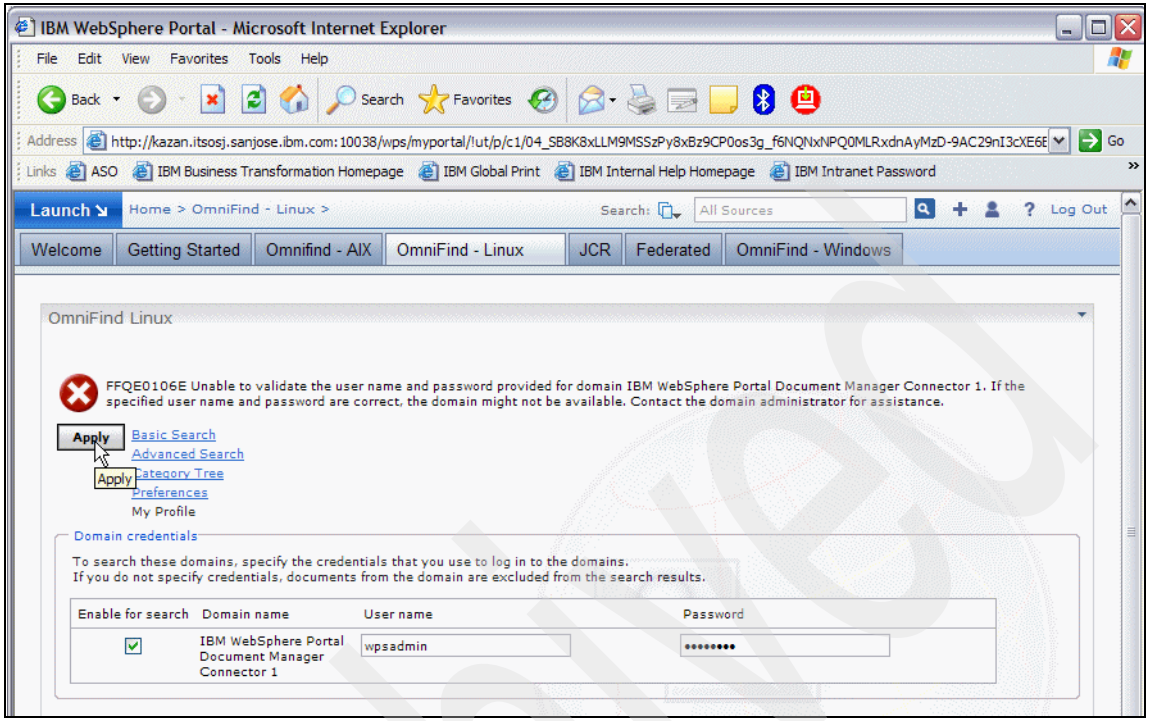

Figure 3-143 Prompt for IMC credentials for Portal Document Manager

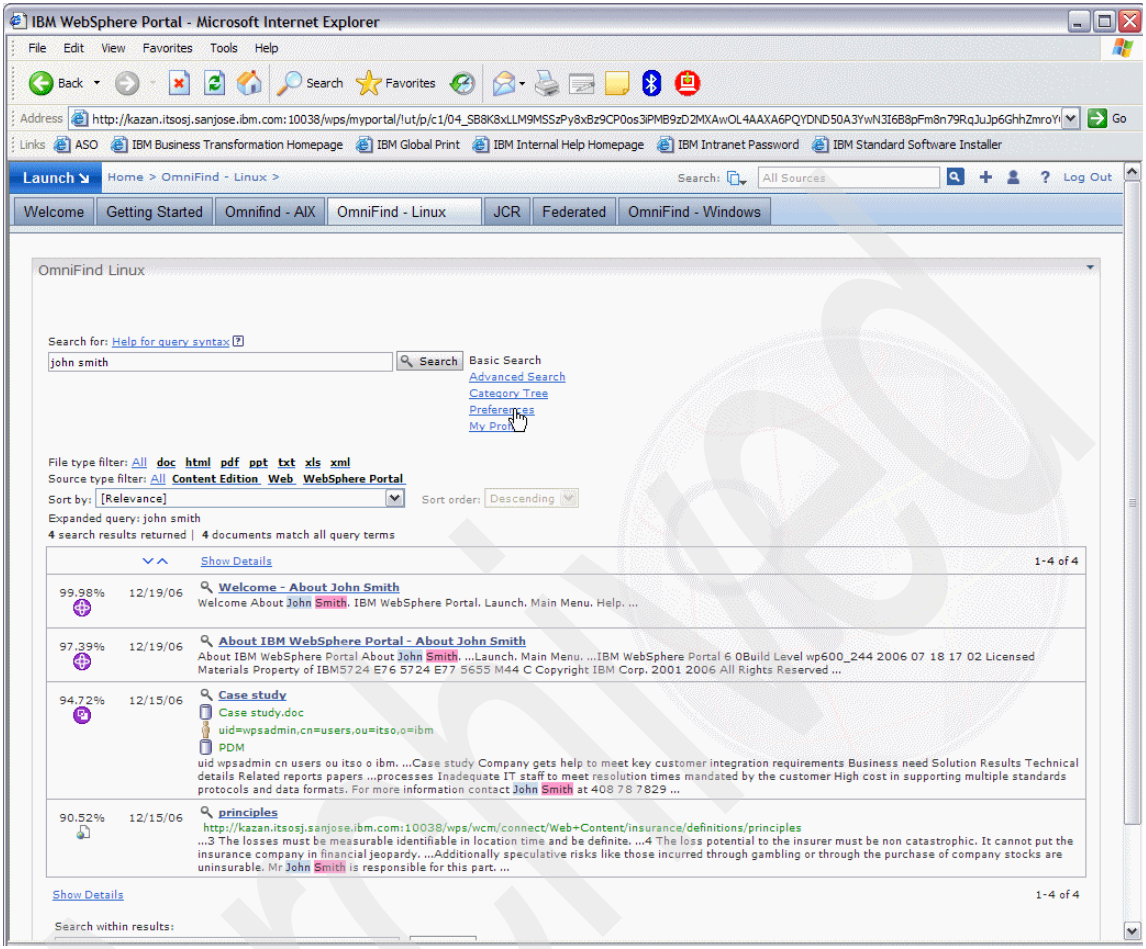

Figure 3-144 Search results for “john smith”

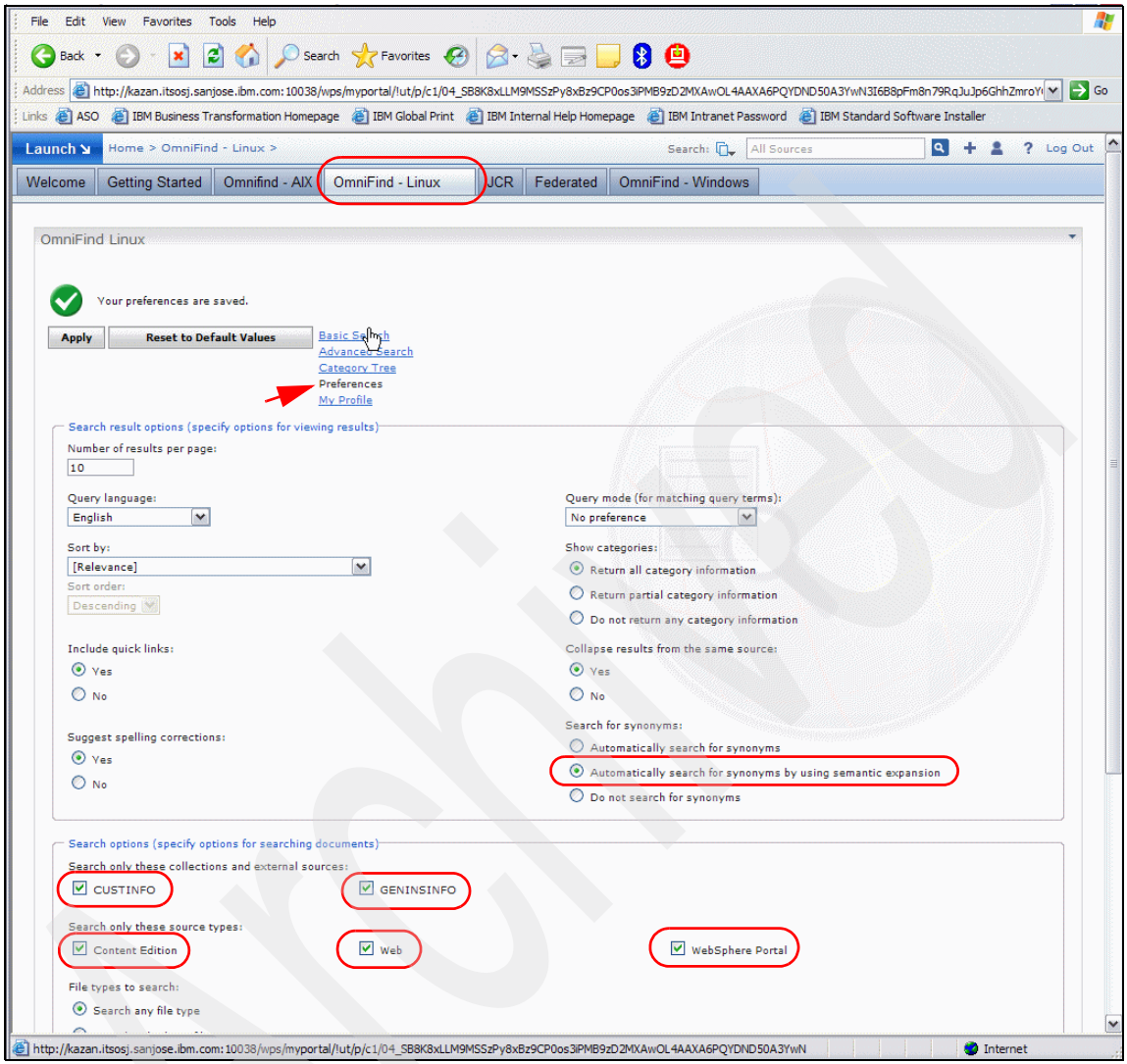

Figure 3-145 Preferences options

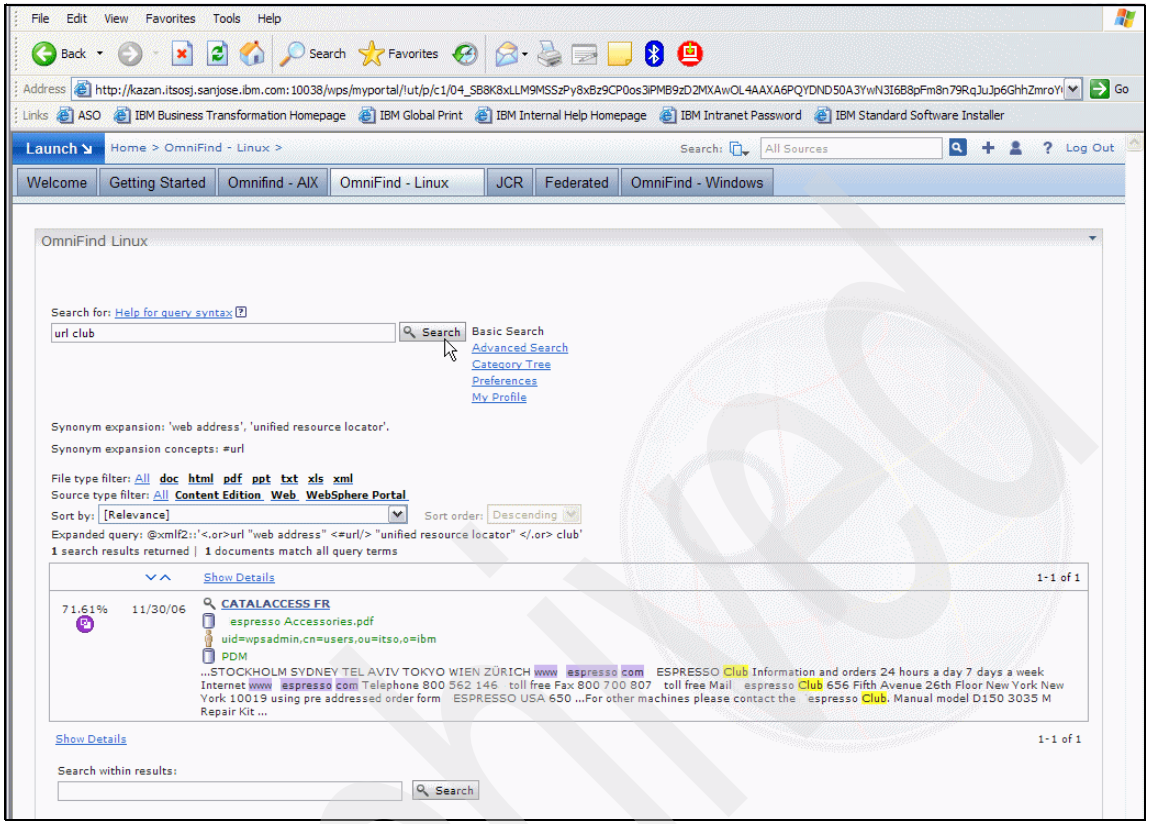

Figure 3-146 Search results for "url club"

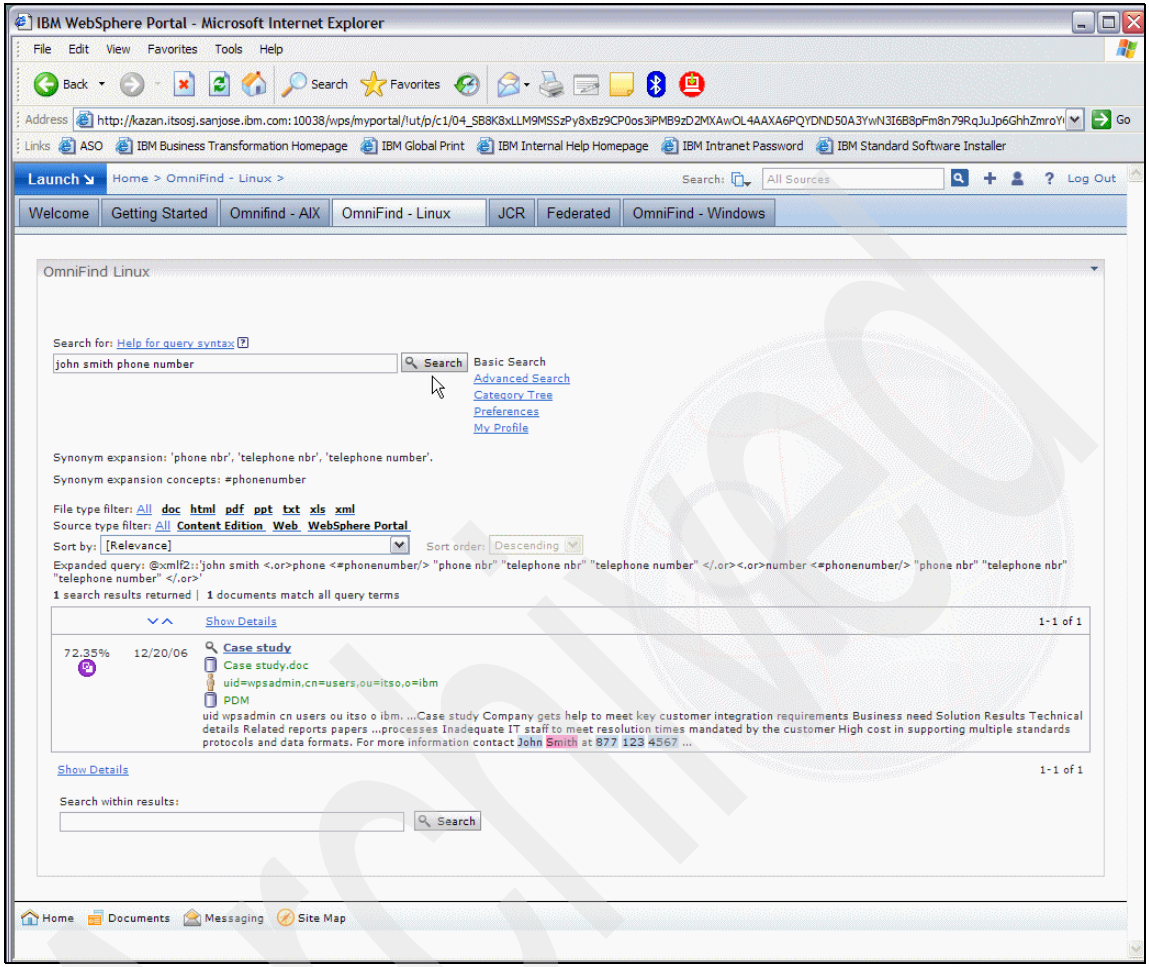

Figure 3-147 Search results for “john smith phone number”

4

Large organization OmniFind scenario on an AIX platform

In this chapter, we describe a step-by-step approach to implementing IBM OmniFind Enterprise Edition in a four server IBM AIX environment for a hypothetical computer vendor organization named IBM.

The topics are:

- Business requirement
- Environment configuration
- Configure the environment

Note: The fictional company named IBM in this chapter is NOT the IBM Corporation, and should not be equated with the IBM Corporation.

4.1 Business requirement

Our fictitious company named IBM is a computer vendor supporting customers in the United States. It has over 50,000 employees supporting over 1 million customers. In a competitive marketplace, IBM needed to provide superior customer service by improving the productivity of its sales, marketing, and technical personnel by making customer information spread across multiple data sources available on demand in response to customer and prospect inquiries. A high availability architecture is a critical requirement of any proposed solution.

One of the solutions aimed at achieving this objective was to implement an enterprise search system for the 10 million customer-related documents located in its Notes Domino, DB2 Content Manager, and DB2 UDB for LUW systems. Given the sensitive nature of customer information stored in Notes, Domino, DB2 Content Manager and DB2 UDB for LUW systems, IBM requires the enterprise search solution to support and leverage the native security capabilities of these underlying data sources; only authorized employees need to have access to secure documents. Additionally, to ease searchability of information, employees need to have the ability to restrict the scope of search by business Corporate News, Products, Services, and so on. The solution also needs to provide a simple GUI interface available as a portlet in WebSphere Portal Server.

From an IBM OmniFind Enterprise Edition implementation perspective, this translates to having:

- A four server IBM AIX platform implementation to address the high availability requirement.
- Enabled WebSphere global security with an LDAP repository.
- A single collection with document-level security and single sign-on enabled.
  - The Notes Domino system contains sales and marketing support information.
  - The DB2 Content Manager systems contains general corporate news and sales and technical support information.
  - The DB2 UDB for LUW system contains skills inventory information about employees within the organization.
- Enabled Identity Management Component (IMC) and single sign-on.
- A Notes crawler for Notes Domino system, DB2 Content Manager crawler for the DB2 Content Manager system, and a DB2 crawler for the DB2 UDB for LUW system.
- Enabled rules-based categorization when defining the collection.
- The sample search portlet installed on WebSphere Portal Server.

4.2 Environment configuration

The IBM workload demands (large employee community), large number of documents (10 million) to be indexed, and the need for a high availability environment permits the adoption of a sufficiently configured four server IBM OmniFind Enterprise Edition environment.

The IBM AIX platform is considered necessary to address the IBM enterprise search solution needs.

Note: In the real world, the high availability environment for IBM OmniFind Enterprise Edition requires a Network Dispatcher to provide the load balancing capability across the two search servers. In our contrived environment, however, we did not implement this functionality due to time constraints.

Figure 4-1 on page 302 shows the configuration used in the IBM' four server IBM AIX configuration, including:

- A Windows 2003 server (kazan.itsosj.sanjose.ibm.com) provides the IBM enterprise portal through which authorized users will access the enterprise search solution. This server uses LDAP (Tivoli Directory Server) to address the security requirements of the enterprise search solution.
- Tivoli Directory Server (boron.itsosj.sanjose.ibm.com) is installed on a separate server that is physically well secured given the sensitive nature of the information it contains.
- Four server IBM AIX servers (bonnie.itsosj.sanjose.ibm.com, denmark.itsosj.sanjose.ibm.com, clyde.itsosj.sanjose.ibm.com, and jamaica.itsosj.sanjose.ibm.com) for the IBM OmniFind Enterprise Edition search, indexer, parser and crawler components.
- The Notes Domino data source is located on server kazan.itsosj.sanjose.ibm.com, while the DB2 Content Manager and DB2 UDB for LUW data sources are located on server nile.itsosj.sanjose.ibm.com.
- The various crawler types defined in this configuration.

  Note: The Windows 2003 server has direct connectivity and authorization to the data sources in order to render the document that a user selects in the search result.

- IBM OmniFind Enterprise Edition administrators administer and manage the environment through the administration console GUI.

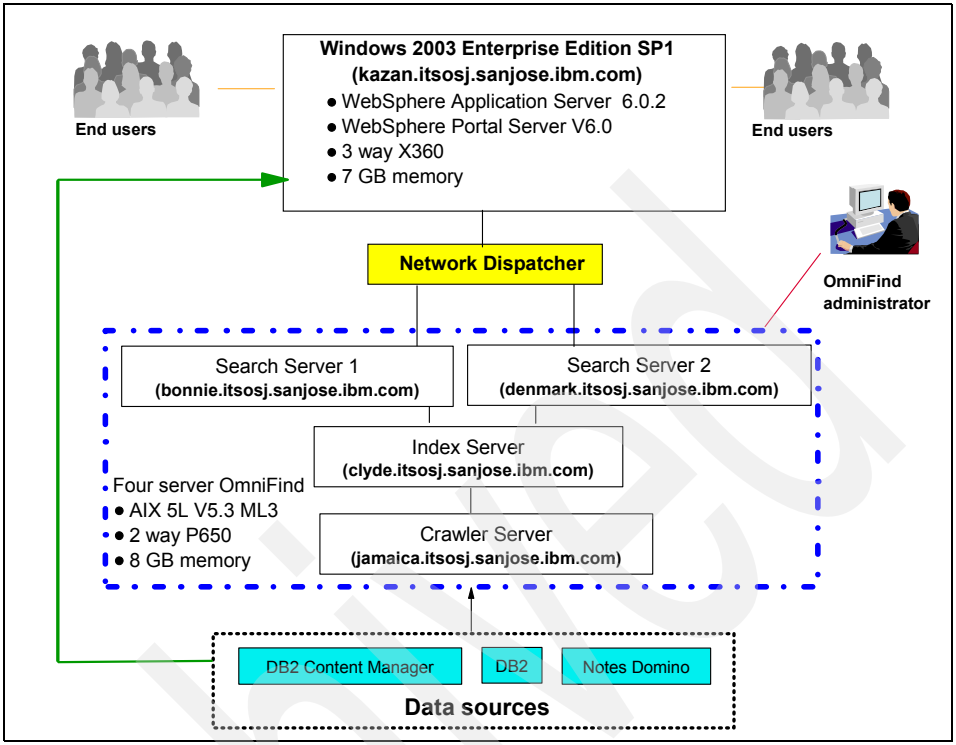

Figure 4-1 The IBM four-server IBM AIX enterprise search solution

4.3 Configure the environment

In this section, we document the step-by-step configuration of the four server AIX configuration in our fictitious company IBM. Figure 4-2 on page 303 lists the main steps involved in configuring this environment. First, the administrator and users that need to access the IBM OmniFind Enterprise Edition environment must be defined in the Tivoli Directory Server LDAP repository. Next, global security must be enabled on each of the two WebSphere Application Servers of the IBM OmniFind Enterprise Edition search servers. Once global security is enabled, the es.cfg configuration file must be updated with the WebSphere Application Server user ID and password. IBM's single collection can now be created (with collection security enabled to enforce document-level security), populated, and queried using a modified sample search Web application, and modified sample search portlet.

Note: The sample search Web application and sample search portlet need to be modified because the DB2 security groups must be added to the USC string before the search runtime is invoked, as described in 4.3.5, “ASTEP5: Query IBMCIF collection” on page 382.

Each of these steps is described in detail in the following subsections.

Note: We assume that the IBM OmniFind Enterprise Edition has been verified to have been correctly installed on the four AIX servers with the proper prerequisites.

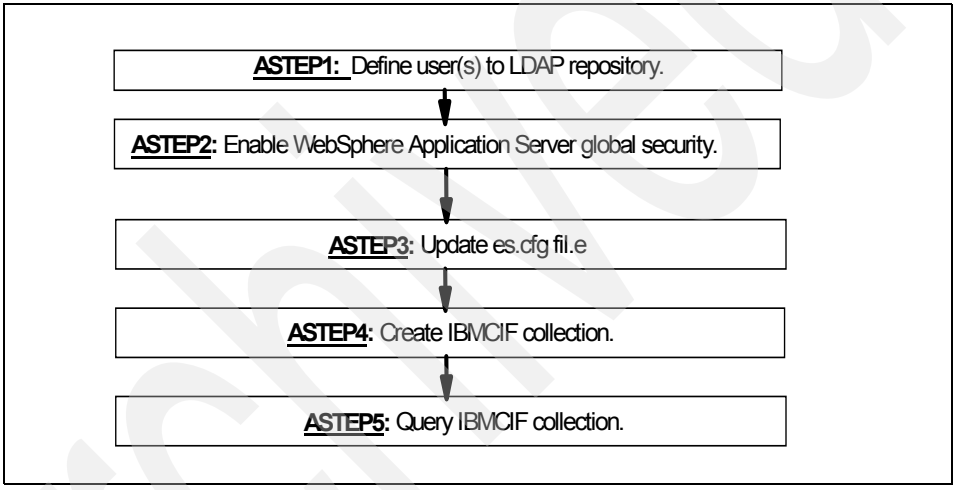

Figure 4-2 Steps to configure the IBM four-server configuration

Attention: In all the following sections, for the purposes of avoiding screen capture overload, we have *not* included all the windows that you would typically navigate through in order to perform the desired function. Instead, we have focused on including select screen captures (and in some cases portions of selected screen captures) that highlight the key items of interest, thereby skipping both initial as well as intervening screen captures in the process.

4.3.1 ASTEP1: Define user(s) to LDAP repository

In this step, we add all the users authorized to access IBM OmniFind Enterprise Edition to the Tivoli Directory Server LDAP repository using the Tivoli Directory Server Web Administration Tool.

Since the process is identical to that described in 2.3.1, “WSTEP1: Define users in LDAP repository” on page 57, it is not repeated here.

4.3.2 ASTEP2: Enable WebSphere Application Server global security

In this step, we enable global security on both the WebSphere Application Servers with the Search Runtime, and specify that they use the Tivoli Directory Server as the LDAP repository as its user registry. LTPA keys are generated and then exported to other servers participating in the single sign-on domain.

Since this process is identical to that described in 2.3.2, “WSTEP2: Enable WebSphere Application Server global security” on page 66, it is not repeated here.

Note: The same user ID / password should be used on both WebSphere Application Servers.

4.3.3 ASTEP3: Update es.cfg file

Once WebSphere global security is enabled in the IBM OmniFind Enterprise Edition Search Runtime servers, you must update the es.cfg file with the WebSphere Application Server user ID and password.

This action must be performed on each server, followed by **`esadmin stop`** and **`esadmin start`** on each server.

Note: When WebSphere global security is enabled, the Common Communications Layer (CCL) component of IBM OmniFind Enterprise Edition needs to authenticate with WebSphere Application Server in order to start the search runtime. It obtains the required user ID / password from the es.cfg file; it has the key to decrypt the WASPassword.

Since this process is identical to that described in 2.3.3, “WSTEP3: Update es.cfg file” on page 77, it is not repeated here.

4.3.4 ASTEP4: Create IBMCIF collection

In this step, we create the IBMCIF collection with the appropriate crawlers, then parse and index the crawled data. The individual steps involved are shown in Figure 4-3 and described in more detail in the following subsections.

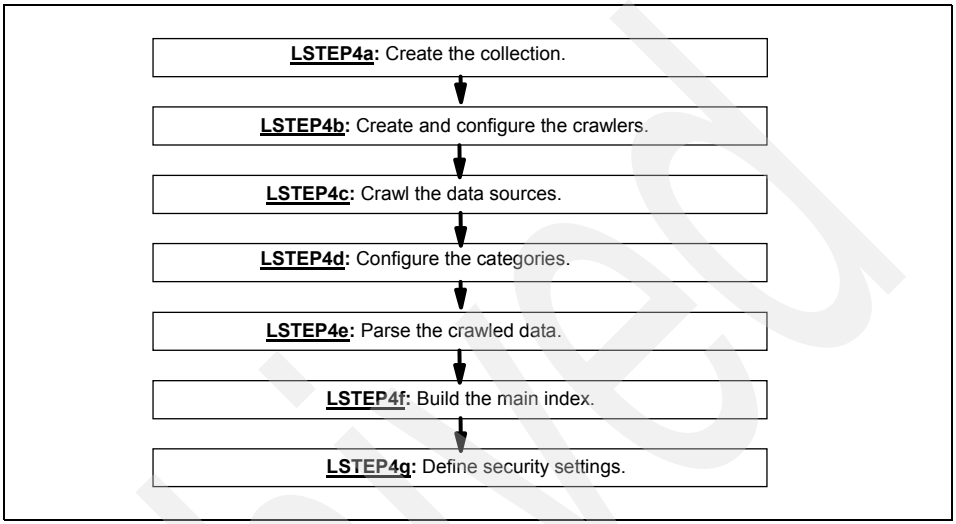

Figure 4-3 Steps to create and configure the IBMCIF collection

Note: A number of parameters can be specified during the creation and configuration of a collection and the associated crawlers. A description of these parameters is beyond the scope of this book. You are encouraged to read the product documentation or invoke **Help** on the GUI for detailed information about these parameters.

ASTEP4a: Create the collection

In this step, we create the IBMCIF collection with the Notes Domino, DB2 Content Manager, and DB2 crawlers.

Note: A number of parameters can be specified during the creation and configuration of a collection and the associated crawlers. A description of these parameters is beyond the scope of this book. You are encouraged to read the product documentation or invoke **Help** on the GUI for detailed information about these parameters.

Create the collection

After logging in to the GUI administration console as the enterprise search administrator, click the **Collections** view and click **Create Collection**, as shown in Figure 4-4. Provide the details shown in Figure 4-5 on page 307 about the collection, such as the Collection name (IBMCIF), Collection security[1] (Enable security for the collection), Document importance (Do not apply any static ranking), and Categorization type (Rule-based (category rules that you configure for this collection)) during parsing.

Note: We also chose to explicitly name the Collection ID to be the same as the collection name IBMCIF. We recommend explicitly specifying this ID rather than let it default to the format col-nnnn, which is difficult to memorize when used in custom applications and command-line tools.

Click **OK** to complete the creation of the collection.

Note: The key point here is to enable security for the collection so that document-level security can be enforced, since this option cannot be changed once the collection is created.

Once the collection is created, we can proceed to the creation of the Notes Domino, DB2 Content Manager, and DB2 crawlers, as described in “ASTEP4b: Create and configure the crawlers” on page 308.

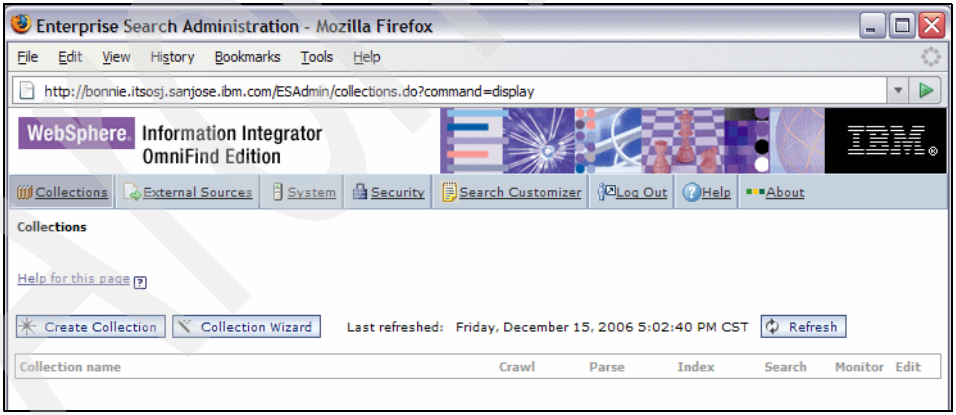

Figure 4-4 Create Collection

[1] Required for enforcing document-level security

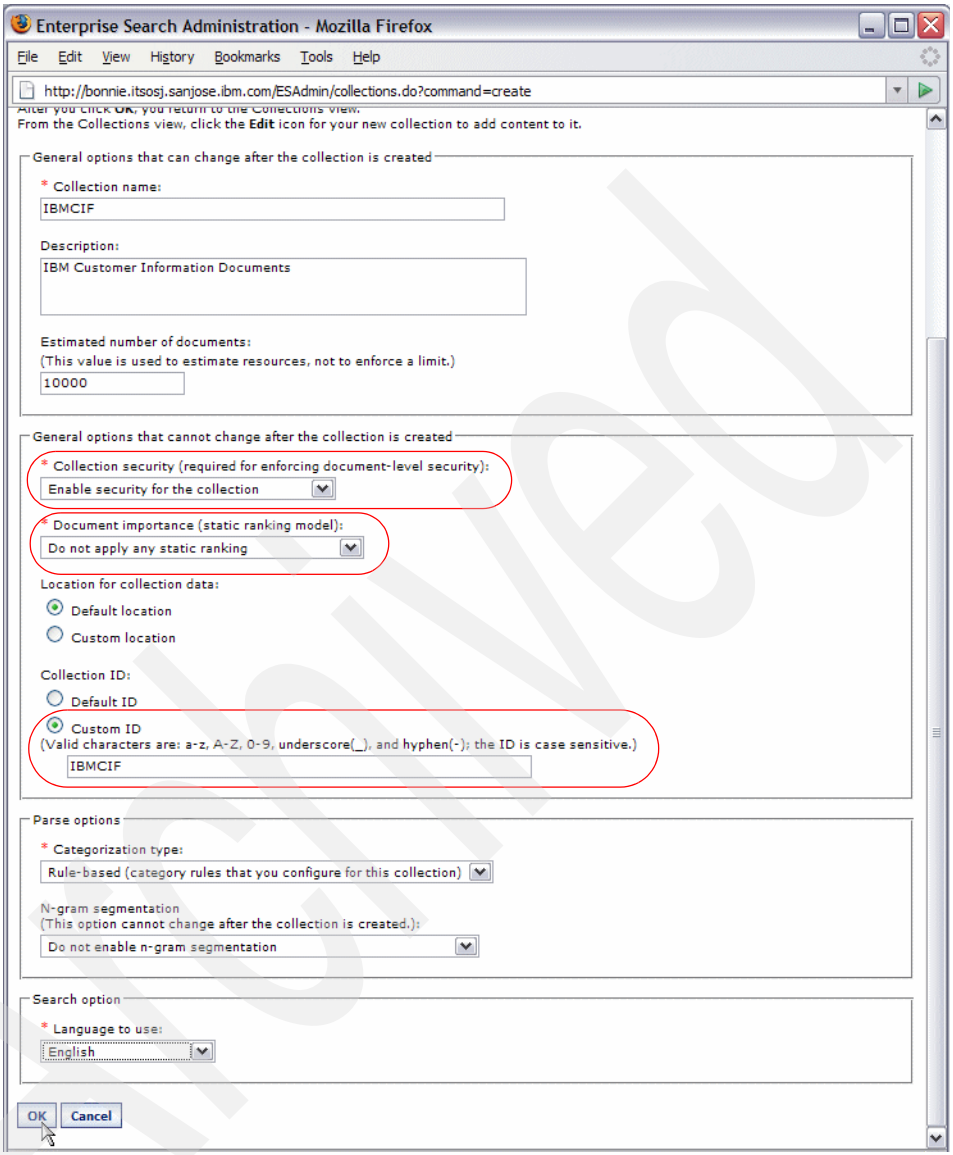

Figure 4-5 IBMCIF collection details

ASTEP4b: Create and configure the crawlers

In this step, the Notes Domino, DB2 Content Manager, and DB2 crawlers are defined with the appropriate security configurations.

- Notes crawler

  Figure 4-6 on page 309 through Figure 4-21 on page 322 describe the creation and configuration of the WebSphere Portal crawler.

  After logging in to the administration console, select the **Collections** view and click the **Crawl** icon, as shown in Figure 4-6 on page 309. In the following window (Figure 4-7 on page 310), switch to Edit mode by clicking the **Edit** icon. From the Crawl tab in Figure 4-8 on page 310, click **Create Crawler**. Select **Notes for Crawler type** and click **Next** in Figure 4-9 on page 311.

  Provide the details of the Notes crawler in Figure 4-10 on page 312, such as the Crawler name (IBM Offerings) and Maximum number of documents to crawl (20000). Click **Next** to provide details of the Notes server to crawl.

  Specify the Notes Server to crawl in the Lotus Notes server name (kazan.itsosj.sanjose.ibm.com), the protocol to use DIIOP, and the Lotus Notes user ID (esadmin[2]) and password to access the server, as shown in Figure 4-11 on page 313. Since we want to leverage single sign-on with Notes Domino, select **Enabled for SSO** from the Single sign-on (SSO) drop-down list, and then click **Edit advanced DIIOP options**, as shown in Figure 4-11 on page 313. Specify the method to obtain the IOR (HTTP and the user ID esadmin and password), and select **Do not use SSL over DIIOP** from the DIIOP over SSL drop-down list, as shown in Figure 4-12 on page 314.

  Note: In the real world, we recommend that you use **SSL** for the DIIOP over SSL drop-down list. We chose not to do so due to time constraints.

  Click **OK** in Figure 4-12 on page 314 and **Next** in Figure 4-13 on page 315, and proceed to choose the Notes databases to crawl. In Figure 4-14 on page 315, select **Search for databases**, and click **Next**.

  Figure 4-15 on page 316 shows the selected databases to crawl (TrainingOfferings, Marketing Documents and Corporate News) obtained by first discovering available databases ('*' in the Database name or pattern followed by a click of **Search for databases**, which lists all those found with the matching criteria in the Available databases box and then copying those of interest to the Databases to crawl box). Click **Next** in Figure 4-15 on page 316 to specify the crawl schedule.

[2] We chose esadmin as the QuickPlace administrator as well.

Select the **Enable when system starts** box, click **Apply**, and then click **Next** in Figure 4-16 on page 317 to select the documents to crawl. Select **Crawl all documents**, click **Apply**, and then click **Next** in Figure 4-17 on page 318 to select the individual Notes data sources to configure. Click **Finish** in Figure 4-18 on page 319 to assume the default options for the entire crawl space. To modify the default crawl space options, click **Edit Crawl Space options** in Figure 4-19 on page 320.

Figure 4-20 on page 321 and Figure 4-21 on page 322 show the options for the entire Notes crawl space, and it includes the default options for Document-Level security, which is to Validate current credentials during query processing, and Index database and server access control lists. Click **OK** to save any changes made.

Important: Every check box selected in Figure 4-20 on page 321 can significantly increase the size of the store and index. You should therefore only select those check boxes that you know will be used, rather than select them thinking that they *might be* used in future.

We can now proceed to create and configure the DB2 Content Manager Content Edition crawler in "DB2 Content Manager crawler" on page 322.

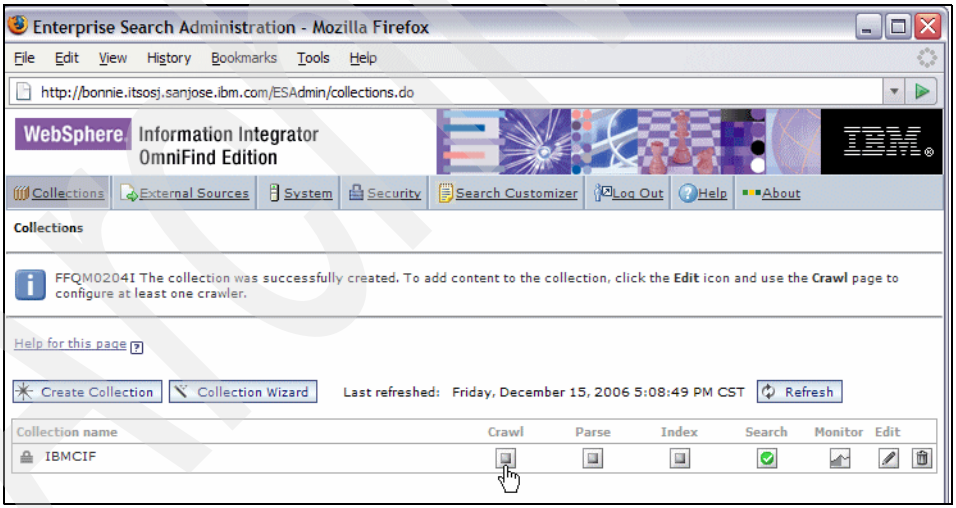

Figure 4-6 Click Crawl icon

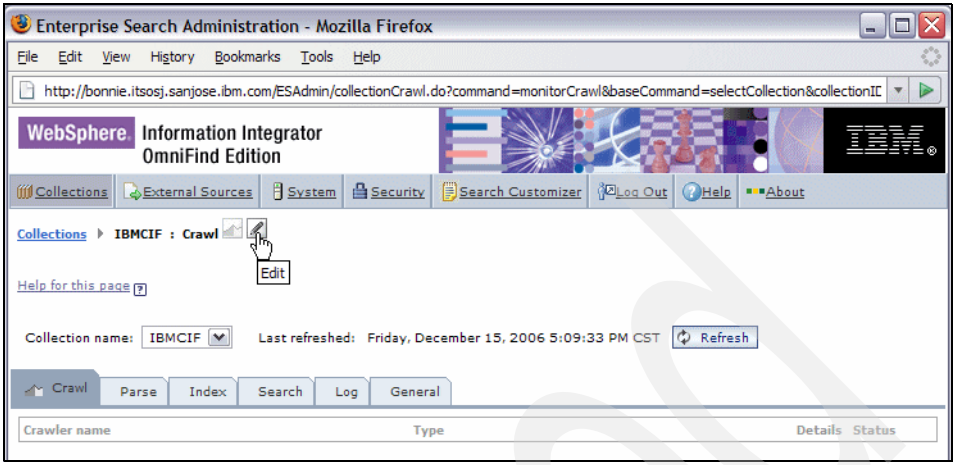

Figure 4-7 Click Edit icon

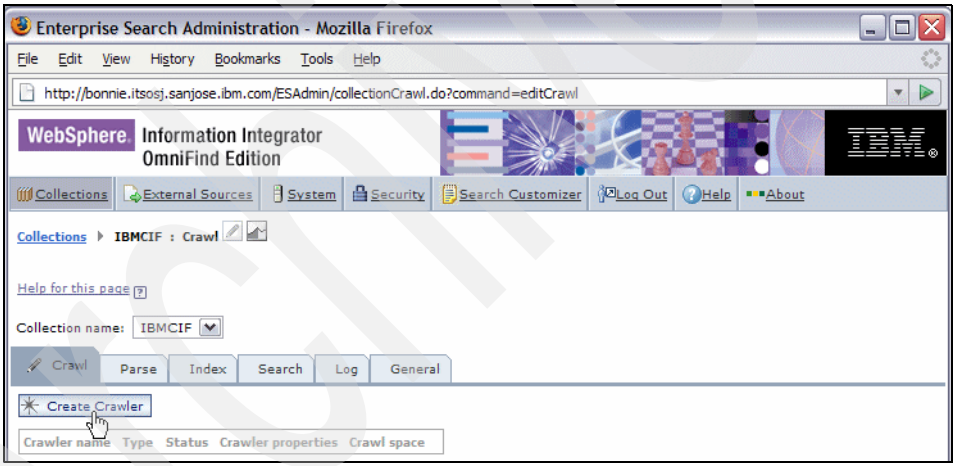

Figure 4-8 Create Crawler

Figure 4-9 Notes crawler type

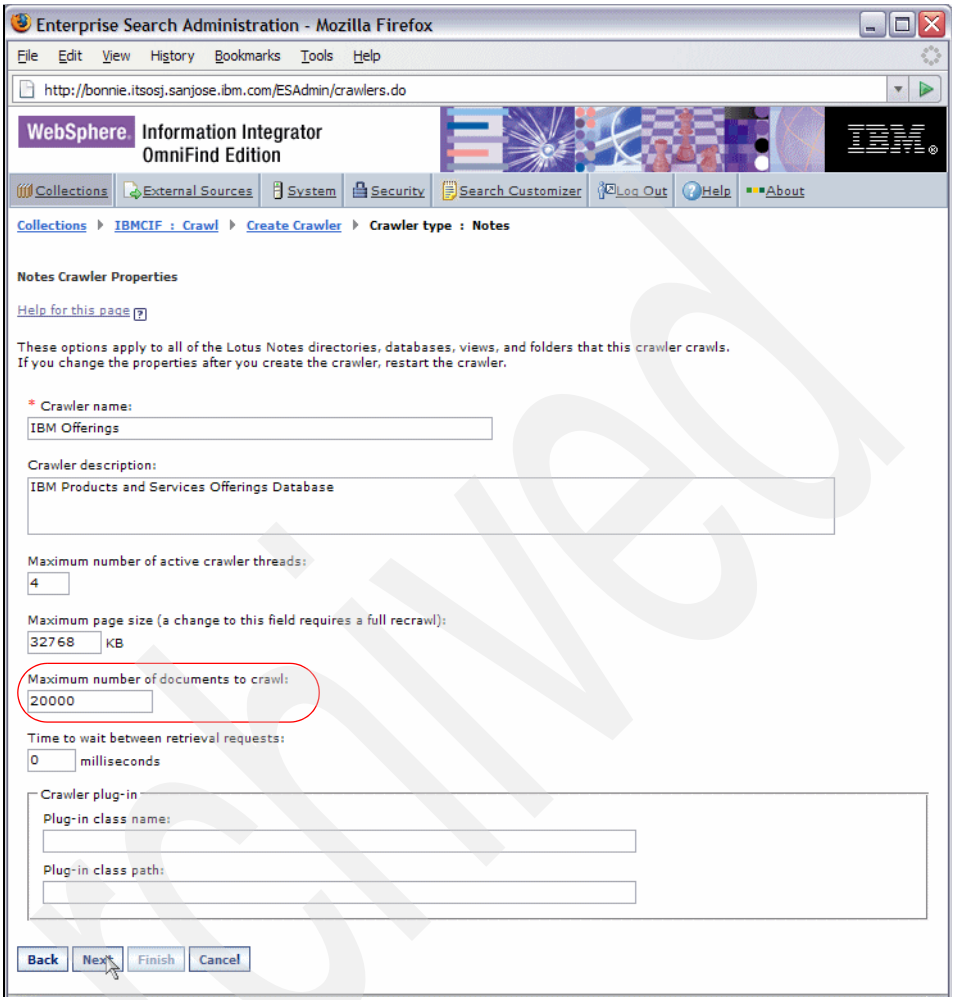

Figure 4-10 Crawler details

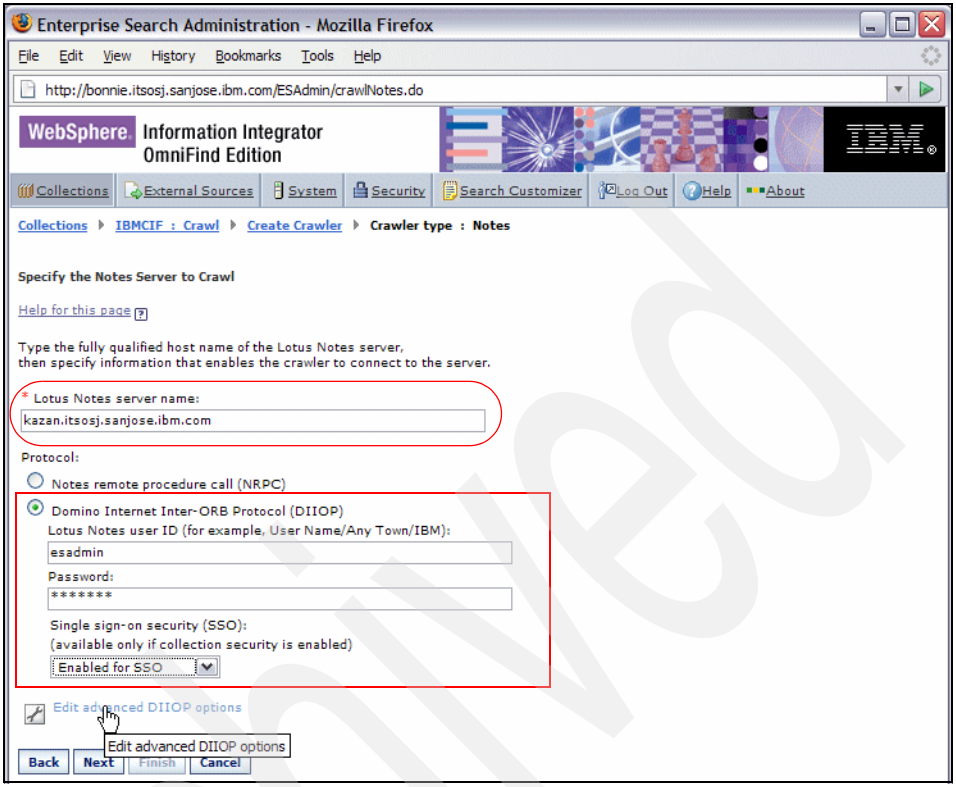

Figure 4-11 Notes Server to Crawl

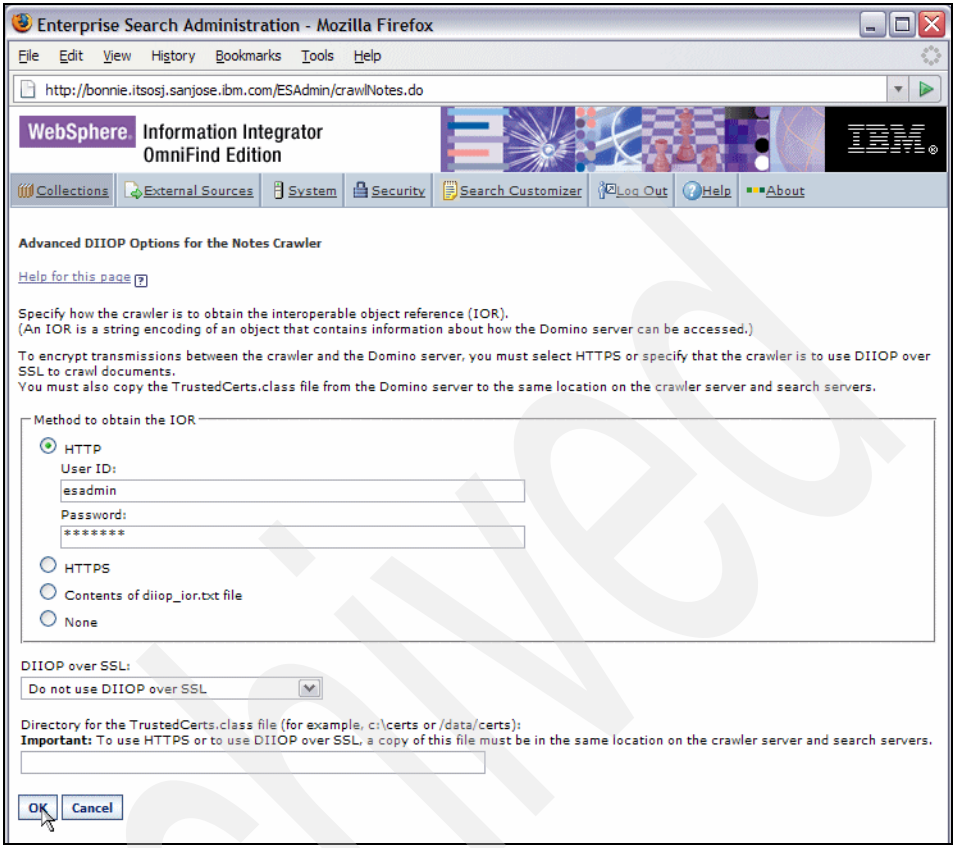

Figure 4-12 Advanced DIOP Options for the Notes Crawler

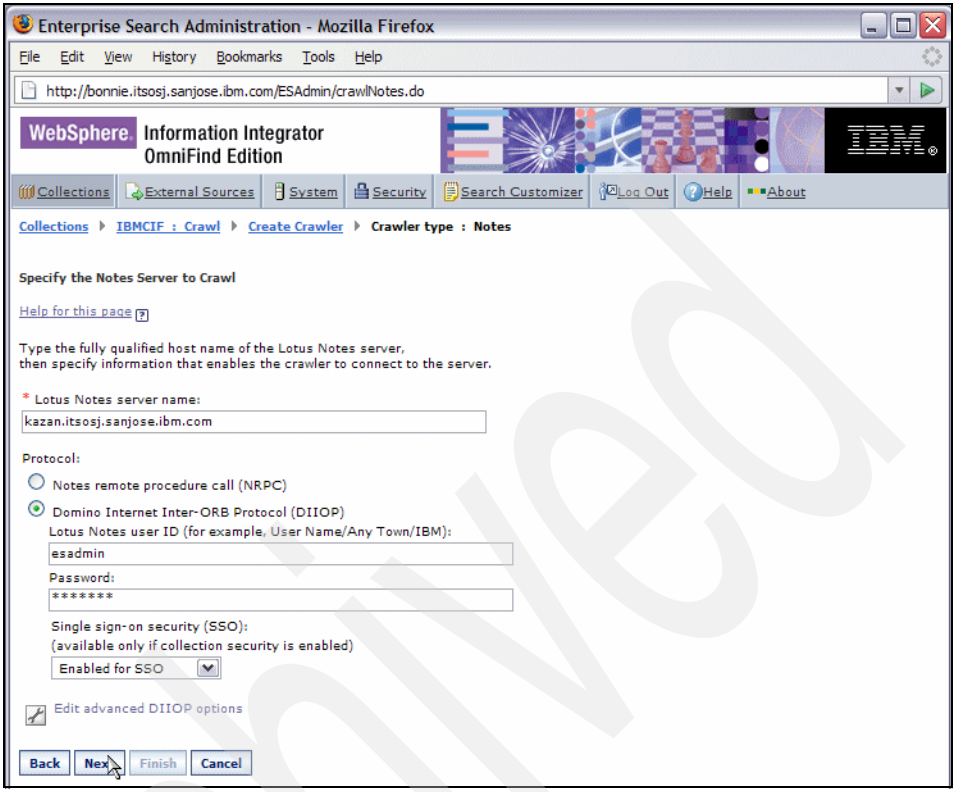

Figure 4-13 Specify the Notes Server to Crawl

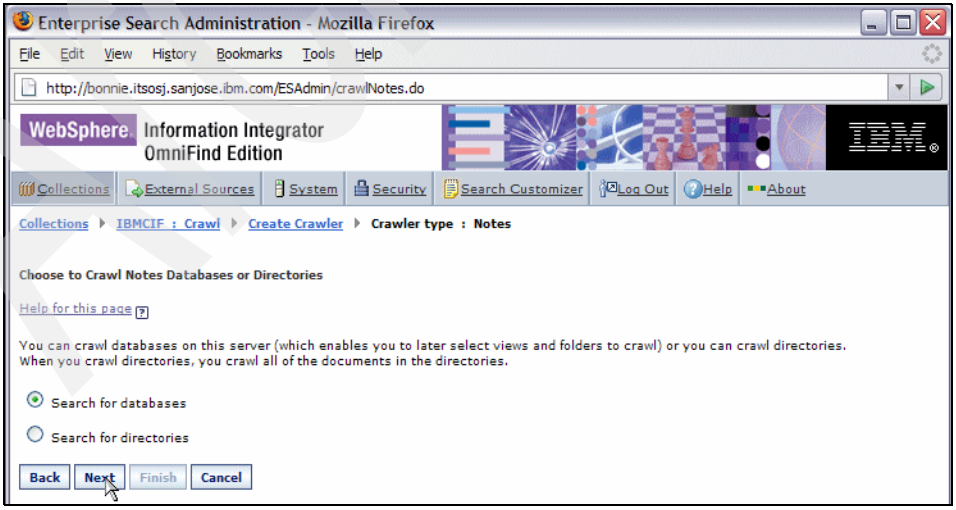

Figure 4-14 Choose to Crawl Note Databases or Directories

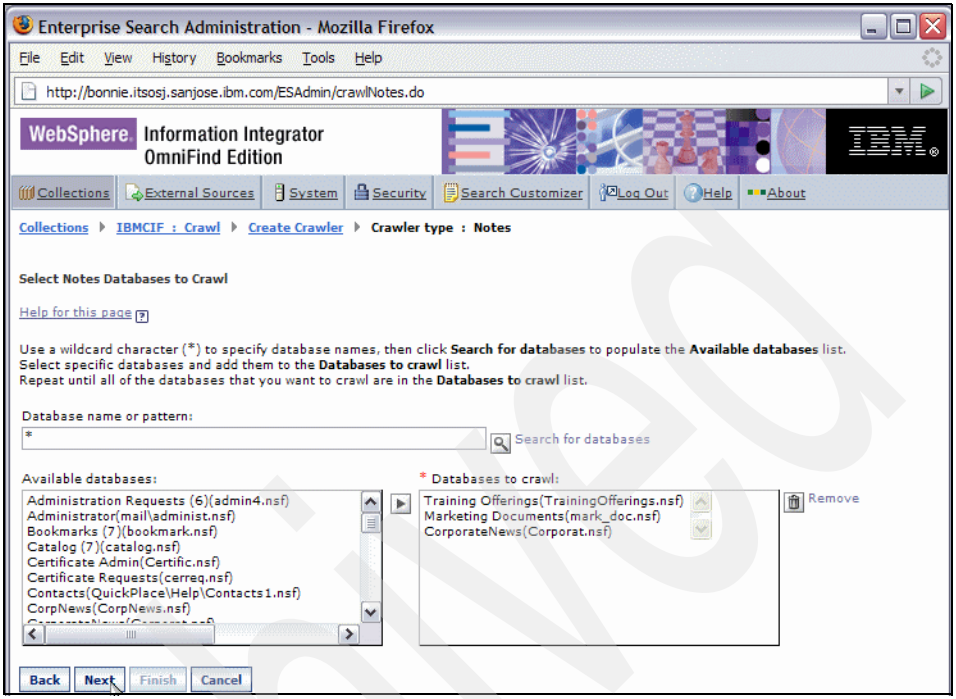

Figure 4-15 Select Notes Databases to Crawl

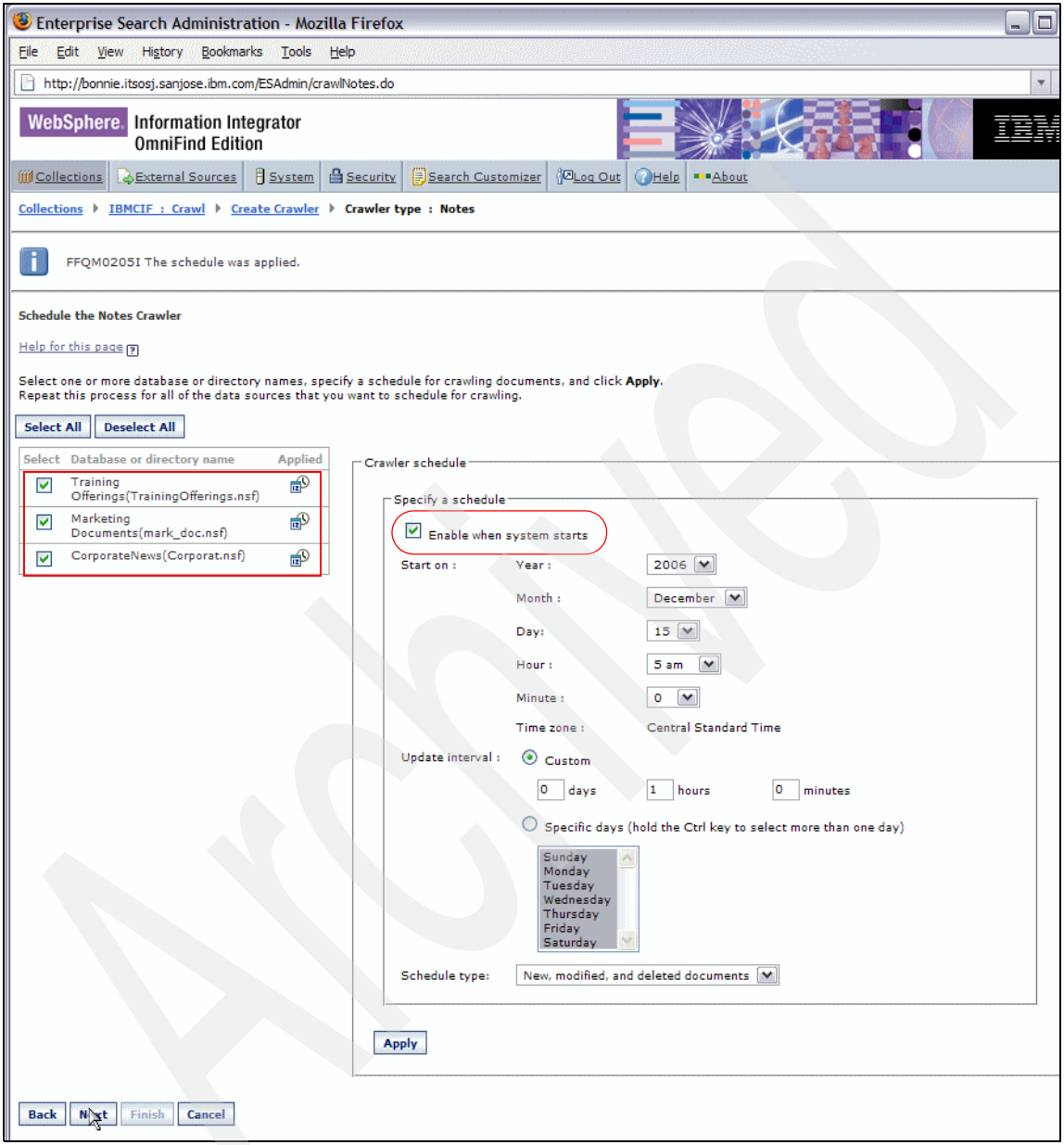

Figure 4-16 Crawl schedule

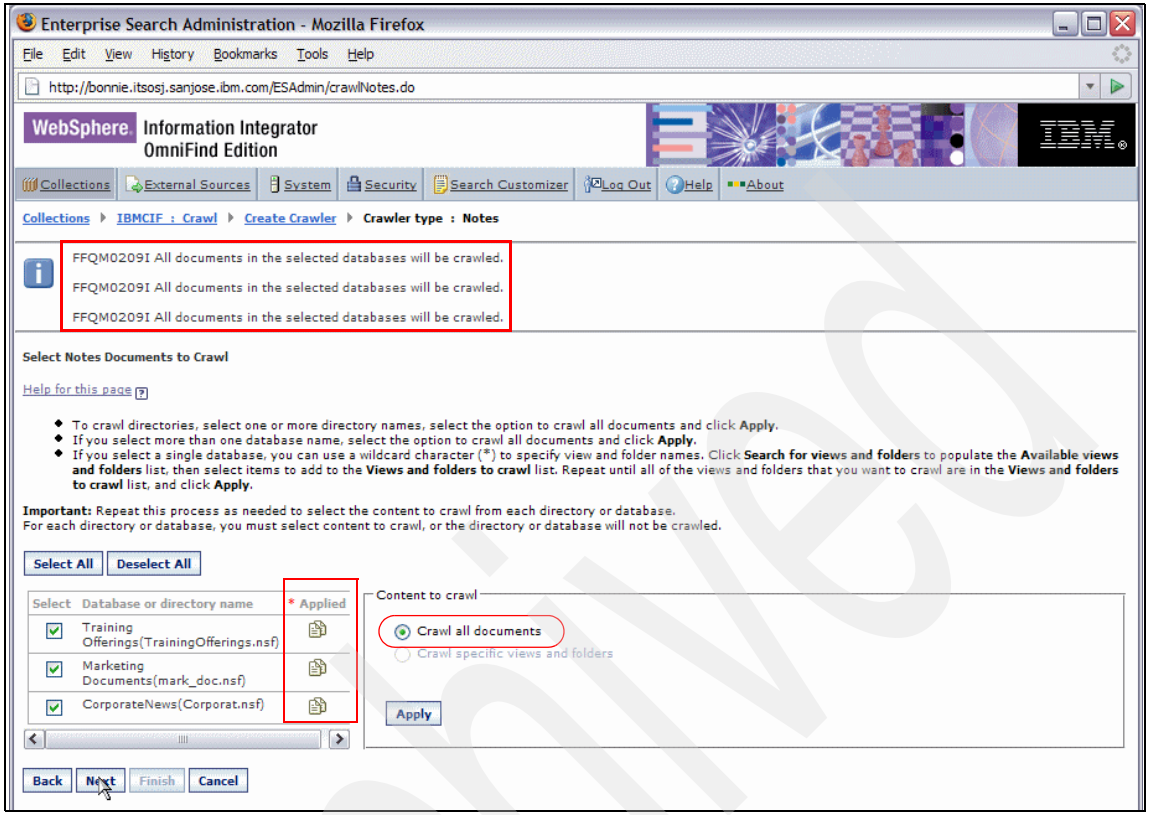

Figure 4-17 Select Notes Documents to Crawl

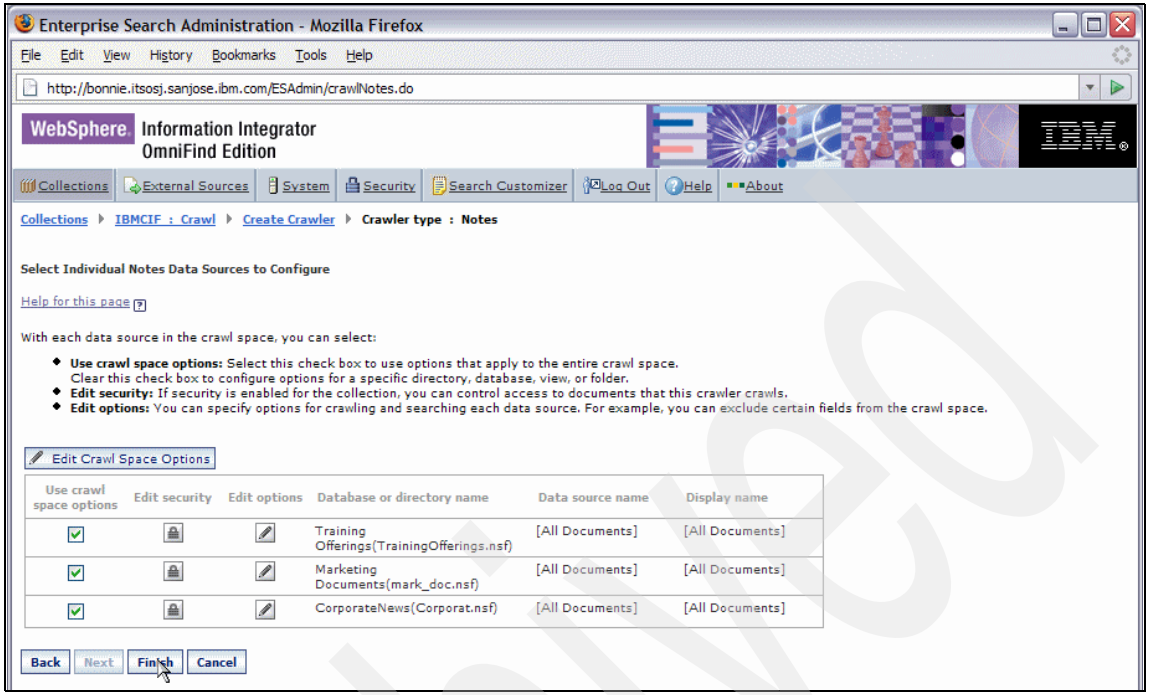

Figure 4-18 Select Individual Notes Data Sources to Configure

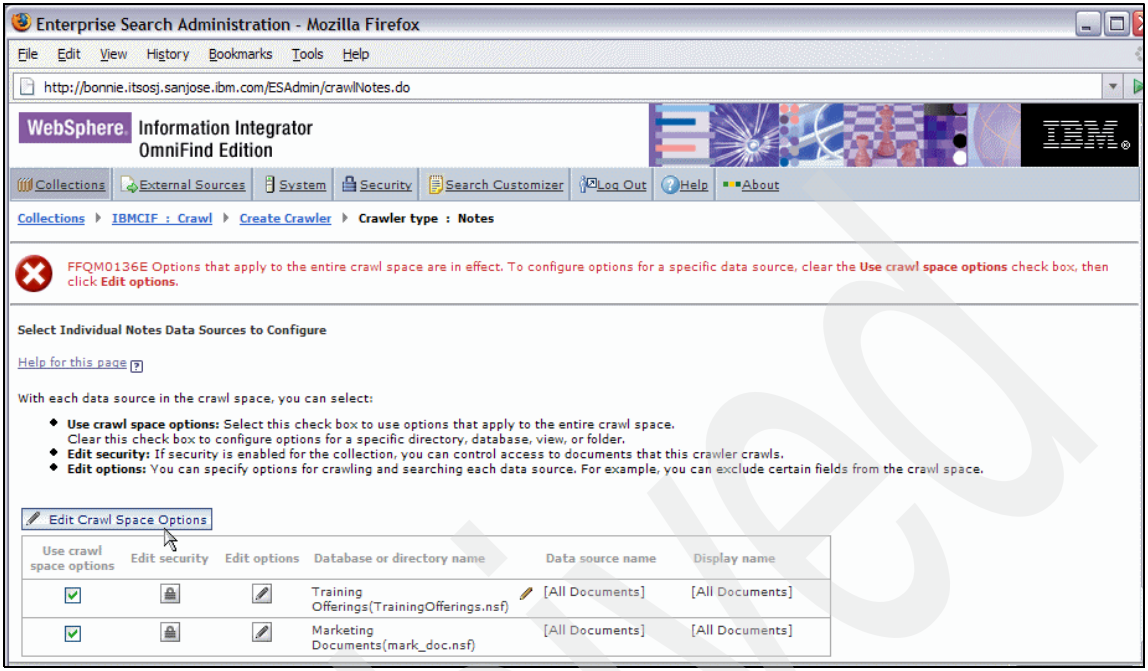

Figure 4-19 Edit Crawl Space Options

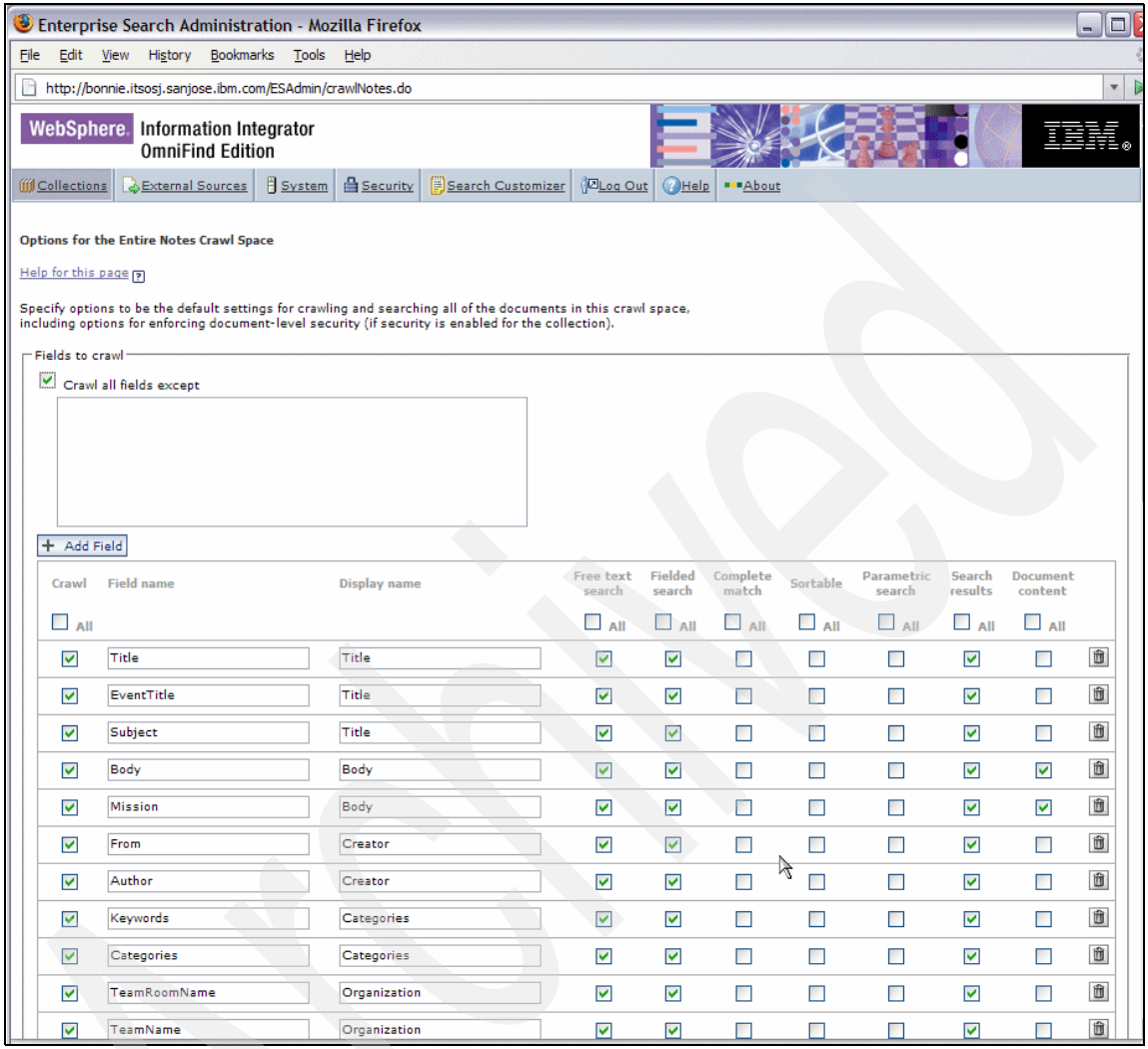

Figure 4-20 Options for the Entire Notes Crawl Space 1/2

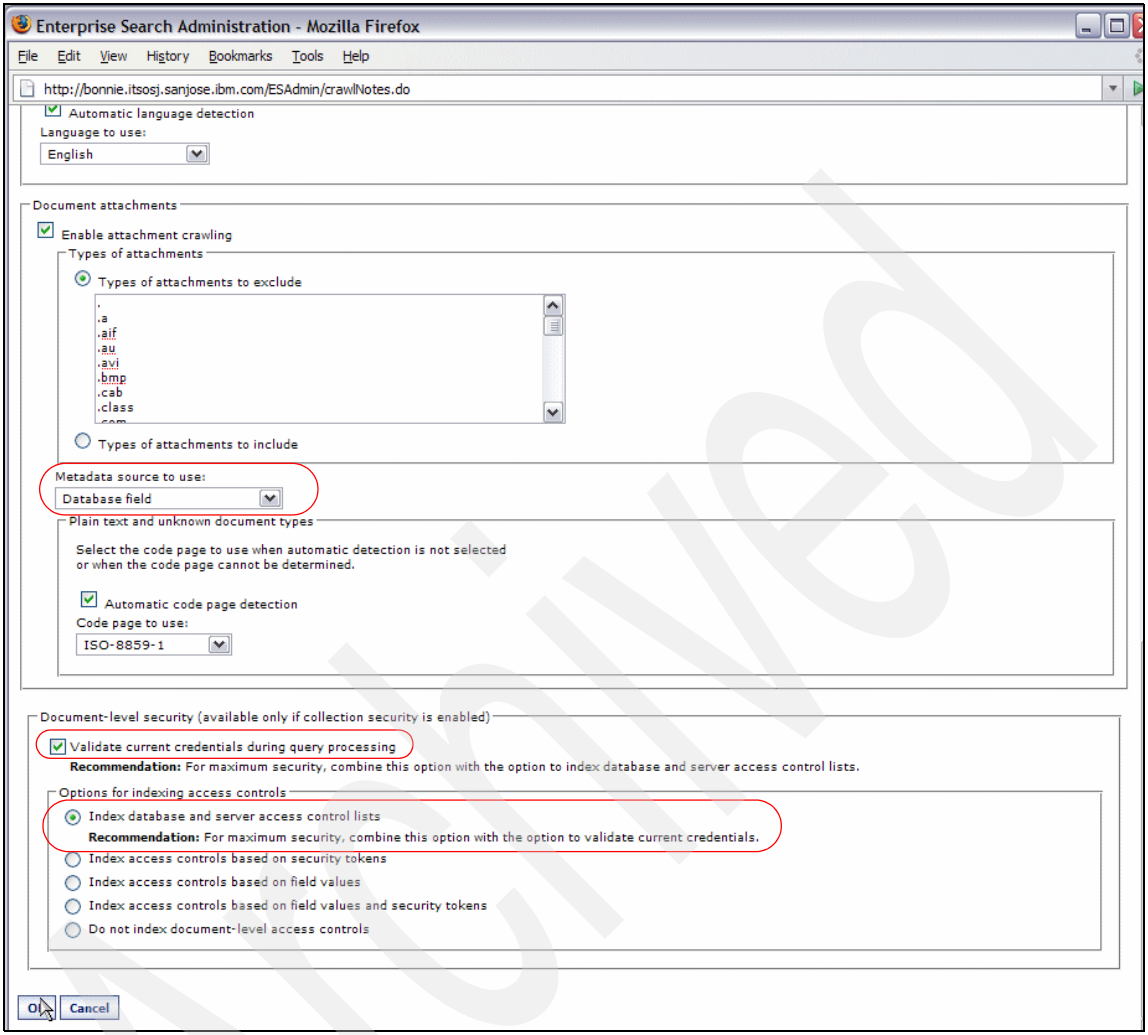

Figure 4-21 Options for the Entire Notes Crawl Space 2/2

- DB2 Content Manager crawler

  Figure 4-22 on page 324 through Figure 4-34 on page 335 describe the creation and configuration of the DB2 Content Manager crawler.

  After logging in to the administration console, select the **Collections** view, and under the Crawl tab, click **Create Crawler**, as shown in Figure 4-22 on page 324. Select **DB2 Content Manager for Crawler** type and click **Next** in Figure 4-23 on page 324.

Provide details of the DB2 Content Manager crawler in Figure 4-24 on page 325, such as the Crawler name (IBM Support) and Maximum number of documents to crawl (200000). Click **Next** to provide details of the DB2 Content Manager servers to crawl.

Figure 4-25 on page 326 shows the selected servers to crawl (icmnlsdb), which are obtained by first discovering available servers ('*' in the Server name or pattern followed by a click of **Search for servers**, which lists all those found with the matching criteria in the Available servers box and then copying those of interest to the Servers to crawl box). Click **Next** in Figure 4-25 on page 326 to specify the DB2 Content Manager Server User IDs and password to access the server.

Specify the user ID (icmadmin) and Password in the Server User ID box and click **Apply**, and then **Next**, as shown in Figure 4-26 on page 327, to specify the crawl schedule.

Select the **Enable when system starts** box, click **Apply**, and then click **Next** in Figure 4-27 on page 328 to select the DB2 Content Manager Item Types to crawl.

Figure 4-28 on page 329 shows the selected item types to crawl (Support_Pubs and Support_TS) obtained by first discovering available item types ('*' in the Item type or pattern followed by a click of **Search for item types**, which lists all those found with the matching criteria in the Available item types box and then copying those of interest to the Item types to crawl box). Click **Next** in Figure 4-28 on page 329 to configure the individual item types.

Click **Edit security** for the Support_Pubs item type, as shown in Figure 4-29 on page 330, to view the default options for the Document-Level security, which is to Validate current credentials during query processing, and Index database and server access control lists, as shown in Figure 4-30 on page 331. Click **OK** to save any changes made.

Click **Edit options** for the Support_Pubs item type, as shown in Figure 4-31 on page 332, to view the default options for the Support_Pubs item type, as shown in Figure 4-32 on page 333 and Figure 4-33 on page 334. Click **OK** to save any changes made. Edit the other item type Support_TS as required (not shown here).

Click **Finish** in Figure 4-34 on page 335 to complete the creation and configuration of the DB2 Content Manager crawler.

We can now proceed to create and configure the DB2 crawler in "DB2 crawler" on page 335.

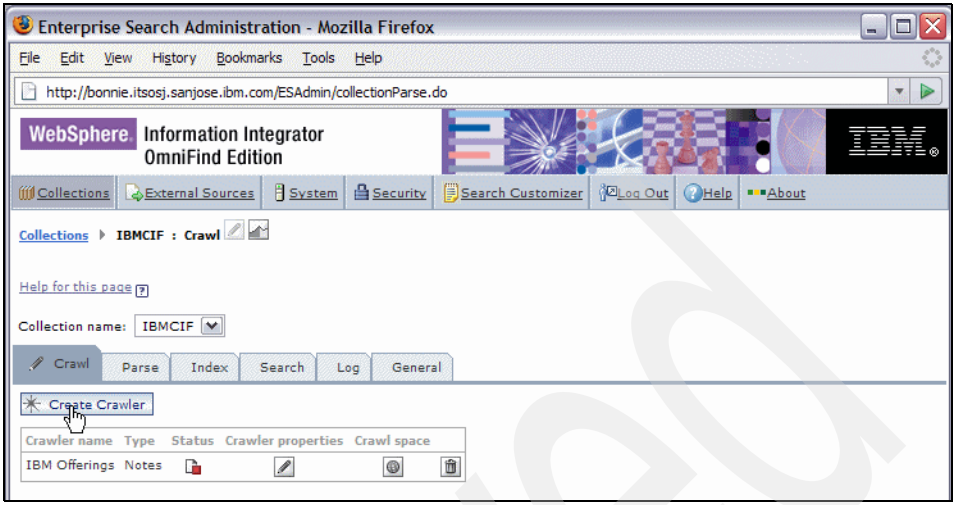

Figure 4-22 Create Crawler

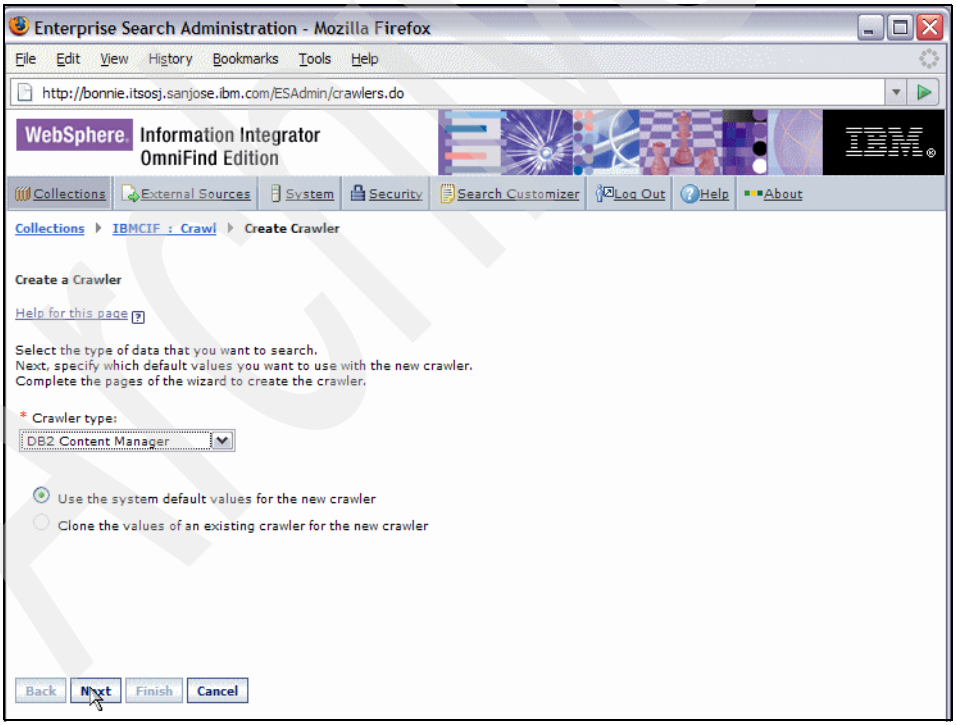

Figure 4-23 DB2 Content Manager crawler type

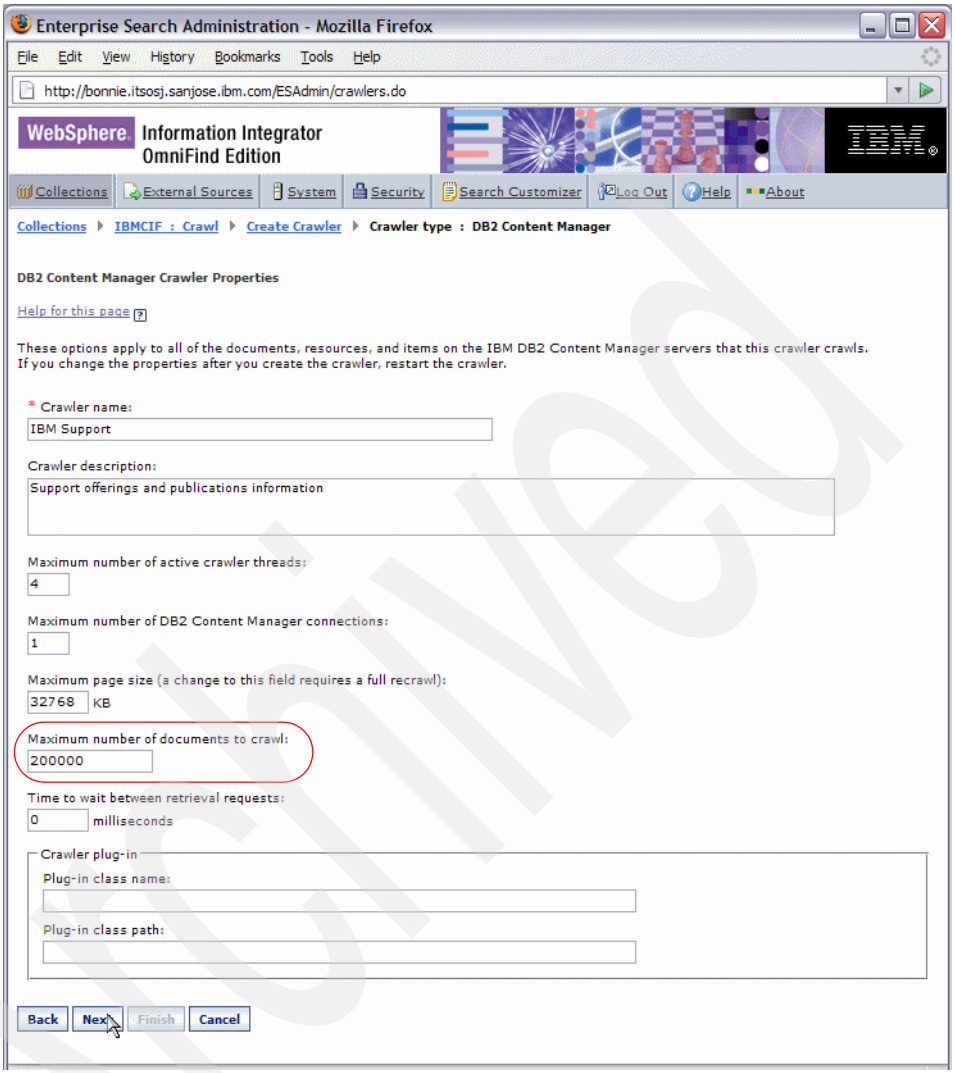

Figure 4-24 Crawler details

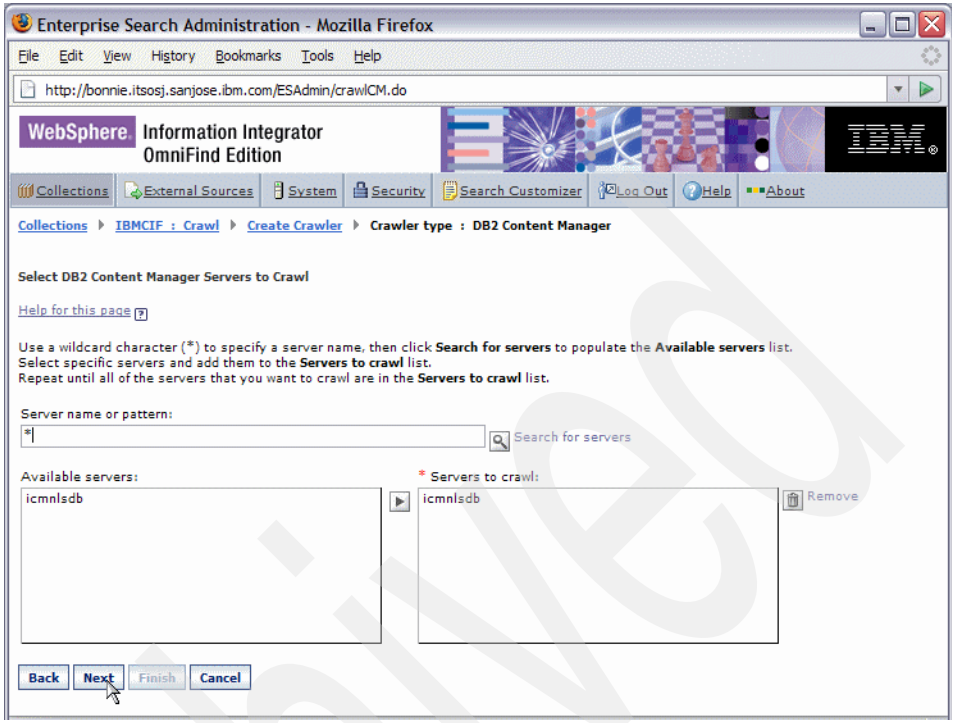

Figure 4-25 Select DB2 Content Manager Servers to Crawl

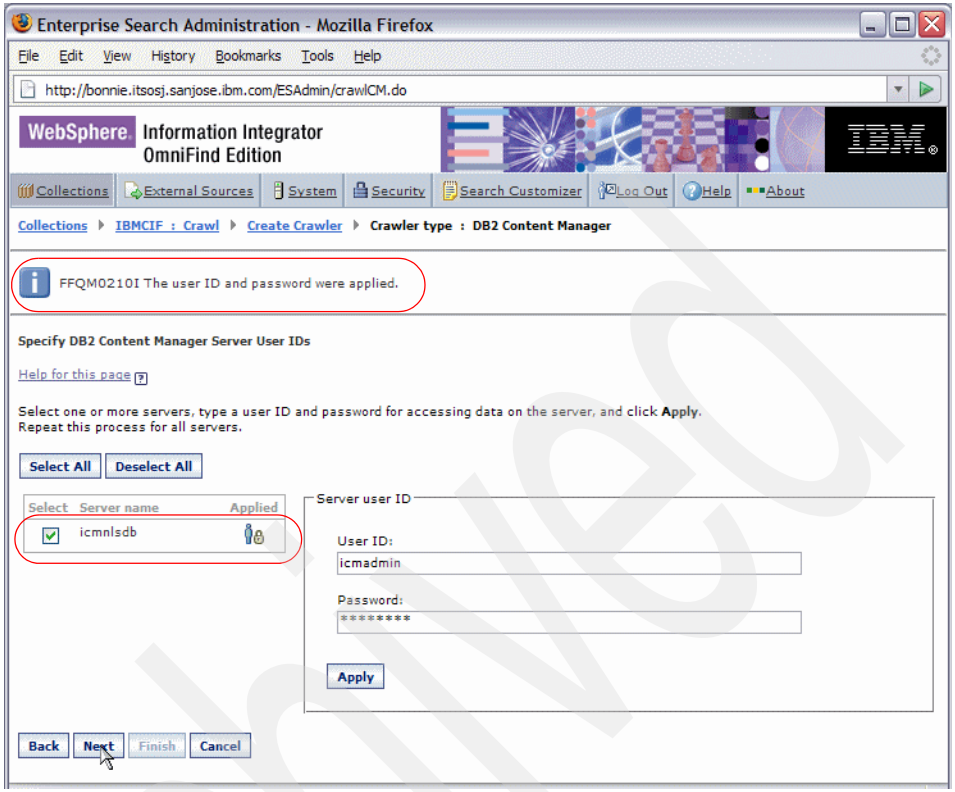

Figure 4-26 Specify DB2 Content Manager Server User IDs

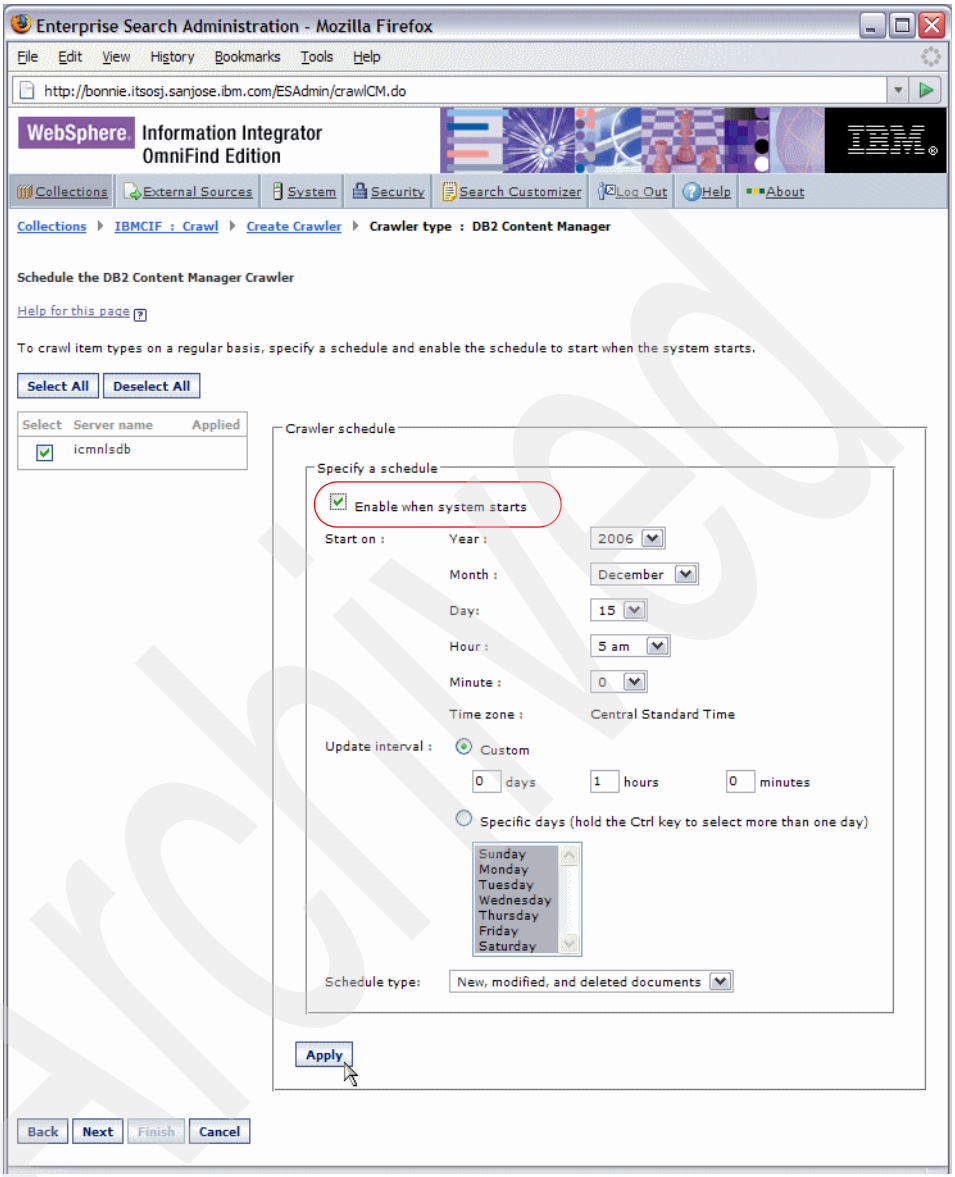

Figure 4-27 Crawl schedule

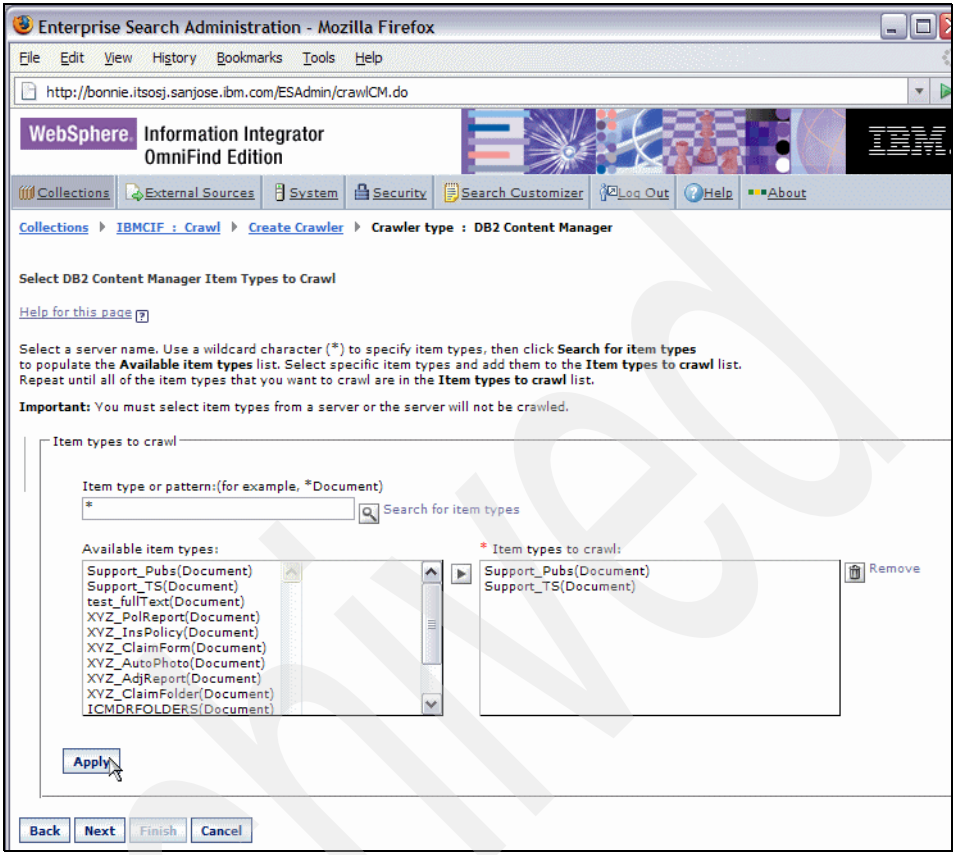

Figure 4-28 Select DB2 Content Manager Item Types to Crawl

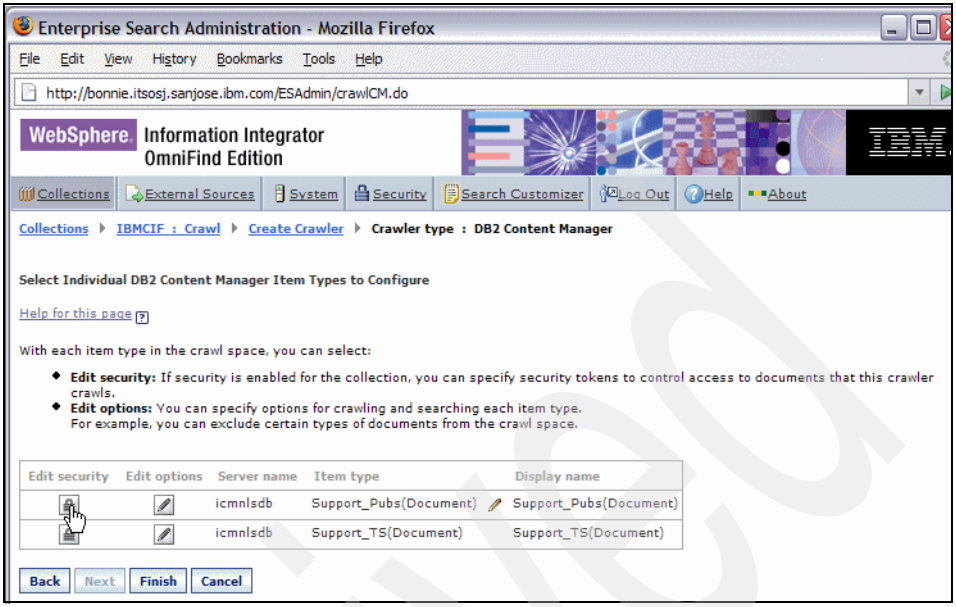

Figure 4-29 Select Individual DB2 Content Manager Item Types to Configure

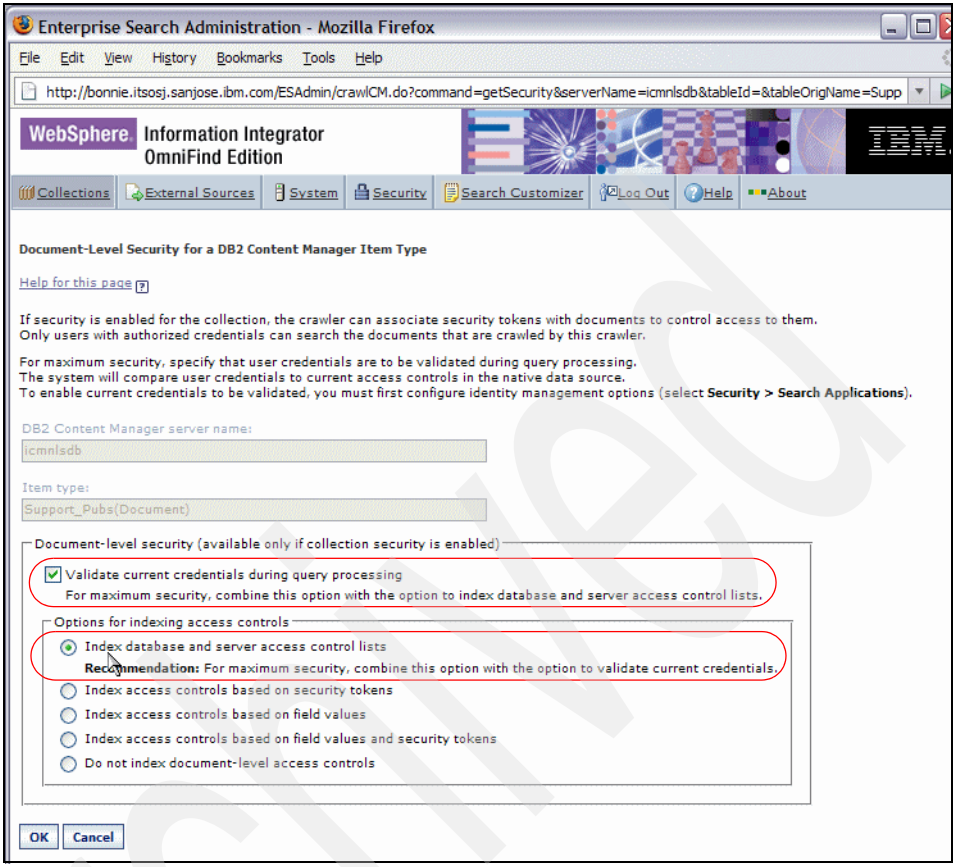

Figure 4-30 Document-Level Security for a DB2 Content Manager Item Type

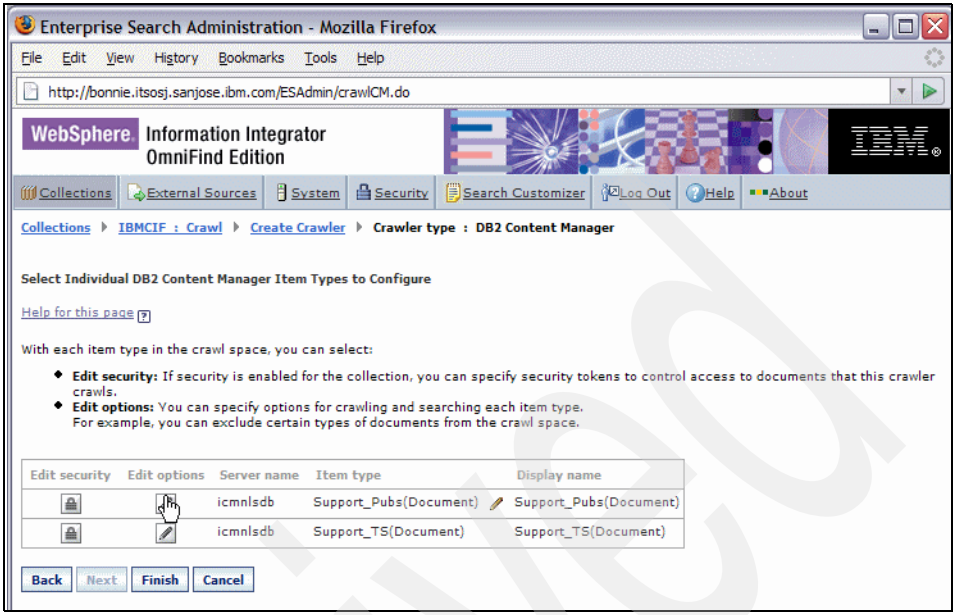

Figure 4-31 Select Individual DB2 Content Manager Item Types to Configure

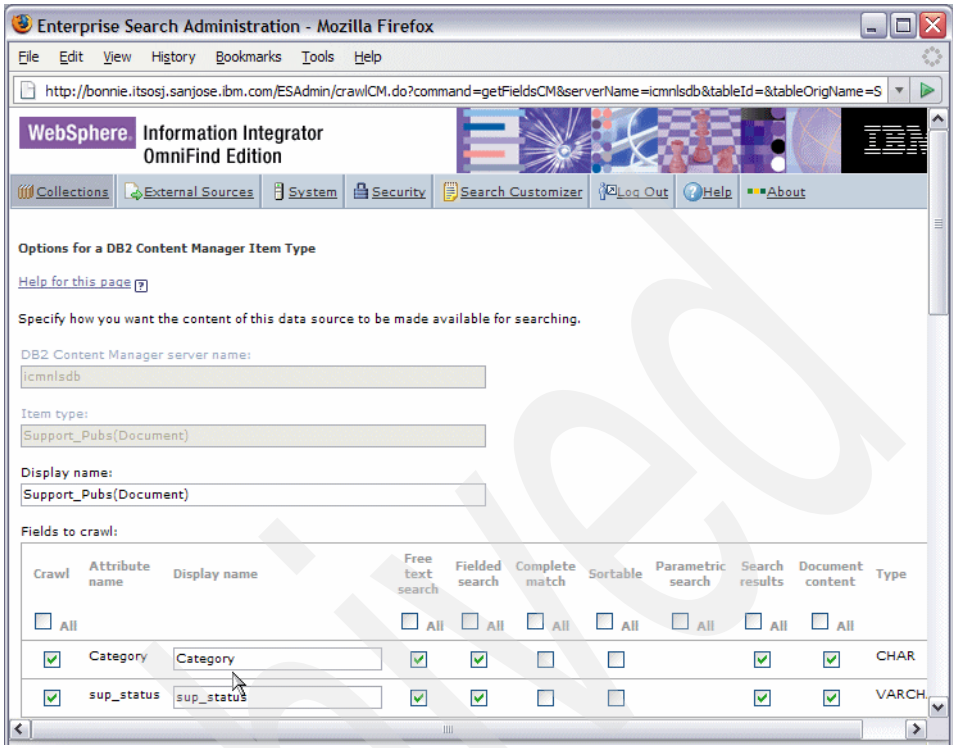

Figure 4-32 Options for a DB2 Content Manager Item Type 1/2

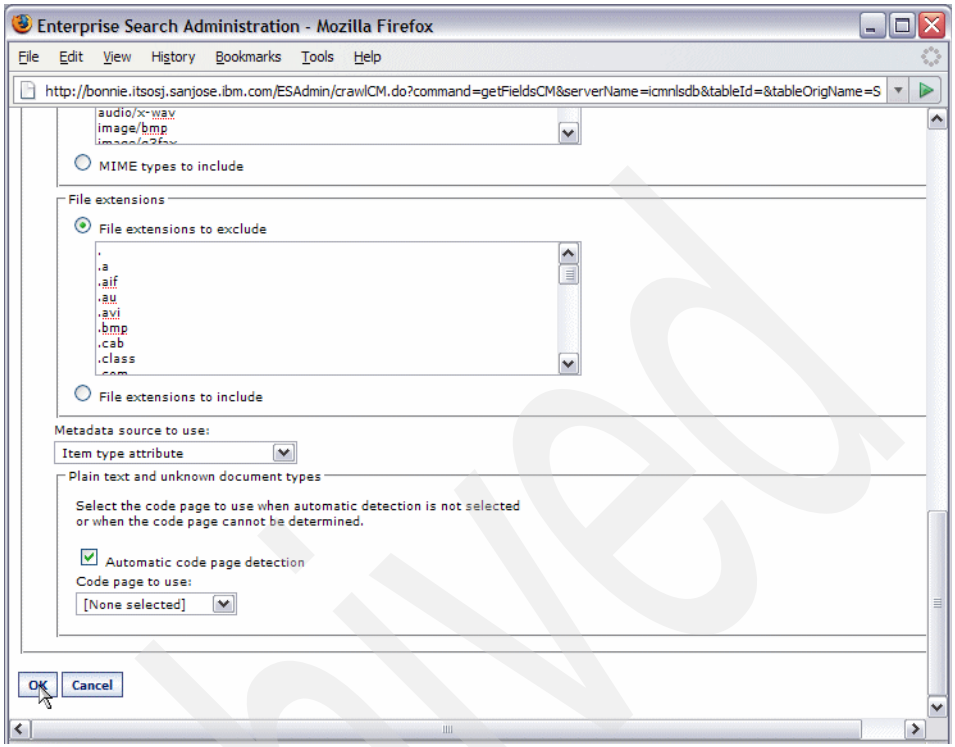

Figure 4-33 Options for a DB2 Content Manager Item Type 2/2

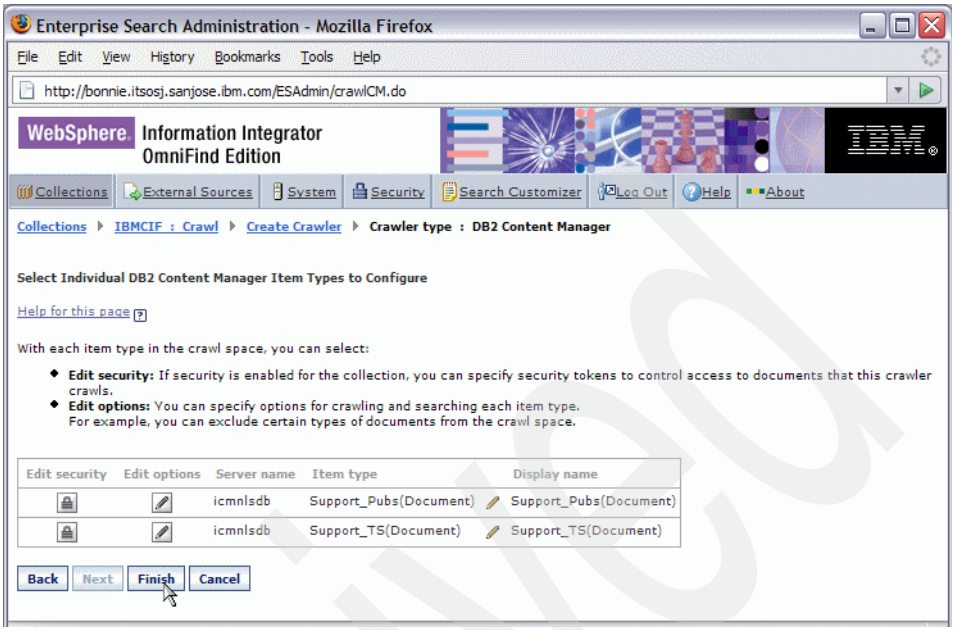

Figure 4-34 Select Individual DB2 Content Manager Item Types to Configure

- DB2 crawler

  Figure 4-35 on page 337 through Figure 4-48 on page 347 describe the creation and configuration of the DB2 crawler.

  After logging in to the administration console, select the **Collections** view, and under the Crawl tab, click **Create Crawler**, as shown in Figure 4-35 on page 337. Select **DB2 for Crawler** type and click **Next** in Figure 4-36 on page 337.

  Provide details of the DB2 crawler in Figure 4-37 on page 338, such as the Crawler name (Support Locations) and Maximum number of documents to crawl (200000). Click **Next** to select the DB2 database type. Select **Local or cataloged databases** and click **Next** in Figure 4-38 on page 339 to select the DB2 databases to crawl.

  Figure 4-39 on page 339 shows the selected databases to crawl (SAMPLE) obtained by first discovering available databases ('*' in the Database name or pattern followed by a click of **Search for databases**, which lists all those found with the matching criteria in the Available databases box and then copying those of interest to the Databases to crawl box). Click **Next** in Figure 4-39 on page 339 to specify the DB2 Database User IDs and password to access the database.

Specify the User ID (icmadmin) and Password in the Database User ID box and click **Apply**, and then **Next**, as shown in Figure 4-40 on page 340 to proceed to specify the crawl schedule.

Select the **Enable when system starts** box, click **Apply**, and then click **Next** in Figure 4-41 on page 341 to select the DB2 tables to crawl.

Figure 4-42 on page 342 shows the selected tables to crawl (ADMINISTRATOR.PROJECT and Support Locations) obtained by first discovering available tables ('*' in the Schema name or pattern and Table name or pattern, followed by a click of **Search for tables**, which lists all those found with the matching criteria in the Available tables box and then copying those of interest to the Tables to crawl box). Click **Next** in Figure 4-42 on page 342 to configure the individual tables.

Since we are not going to use event publishing, click **Next** in Figure 4-43 on page 343.

Click **Edit options** for the ADMINISTRATOR.PROJECT table (not shown here) to view the default options for this table. Modify the options as desired, as shown in Figure 4-45 on page 345, and click **OK** to save any changes made.

Click **Edit security** for the ADMINISTRATOR.PROJECT table (not shown here) to view and modify the default options for the Document-Level security, as shown in Figure 4-46 on page 346. In particular, select **MAJPROJ** from the Field to use for access control drop-down list and click **OK** to save the changes.

Click **Finish** in Figure 4-47 on page 347 to complete the creation and configuration of the DB2 crawler.

Figure 4-48 on page 347 shows the status of all the crawlers defined in the IBMCIF collection.

We can now proceed to crawl the data sources in the IBMCIF collection, as described in "ASTEP4c: Crawl the data sources" on page 348.

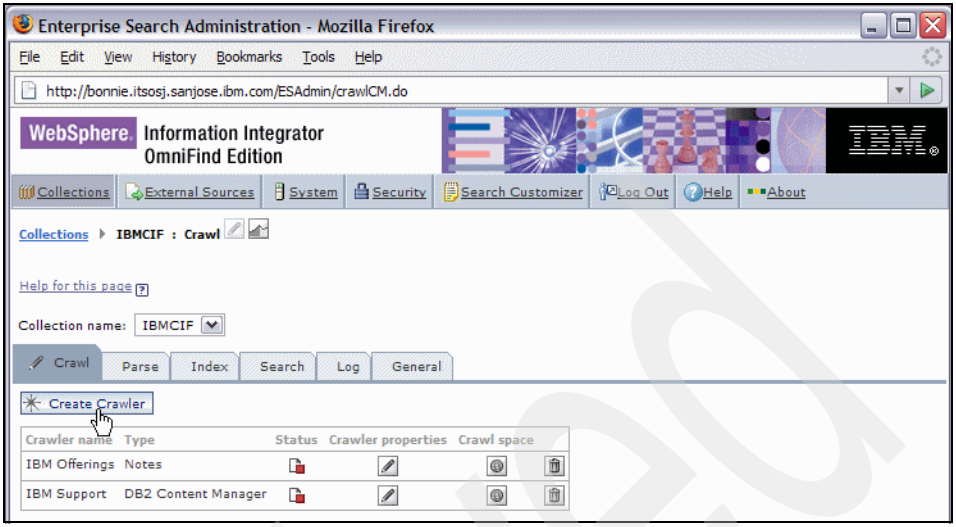

Figure 4-35 Create Crawler

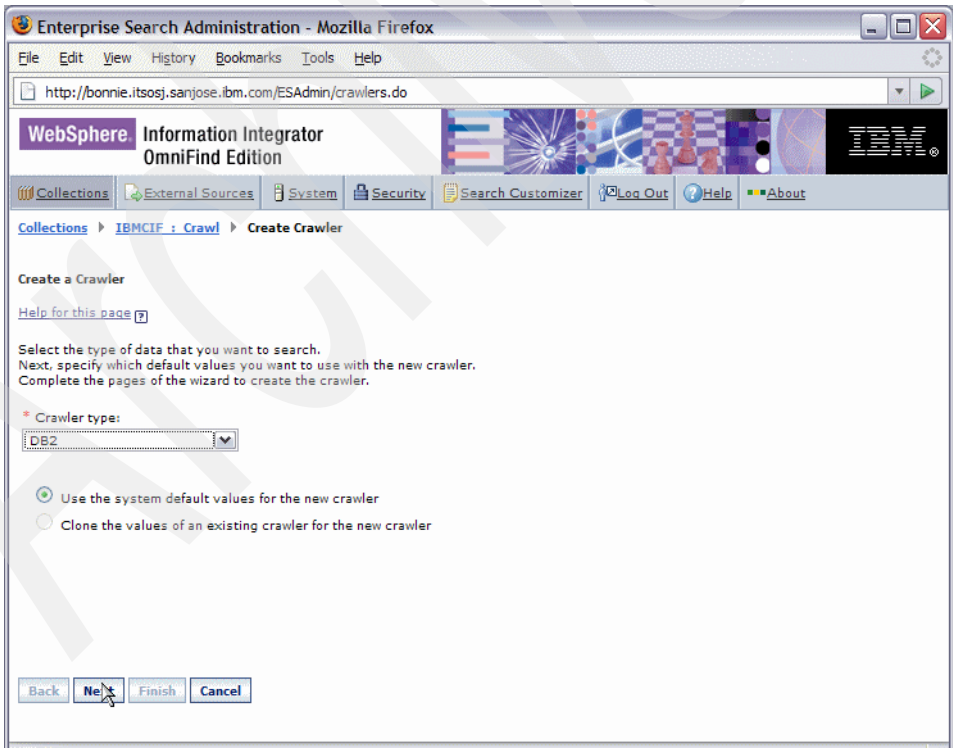

Figure 4-36 DB2 crawler type

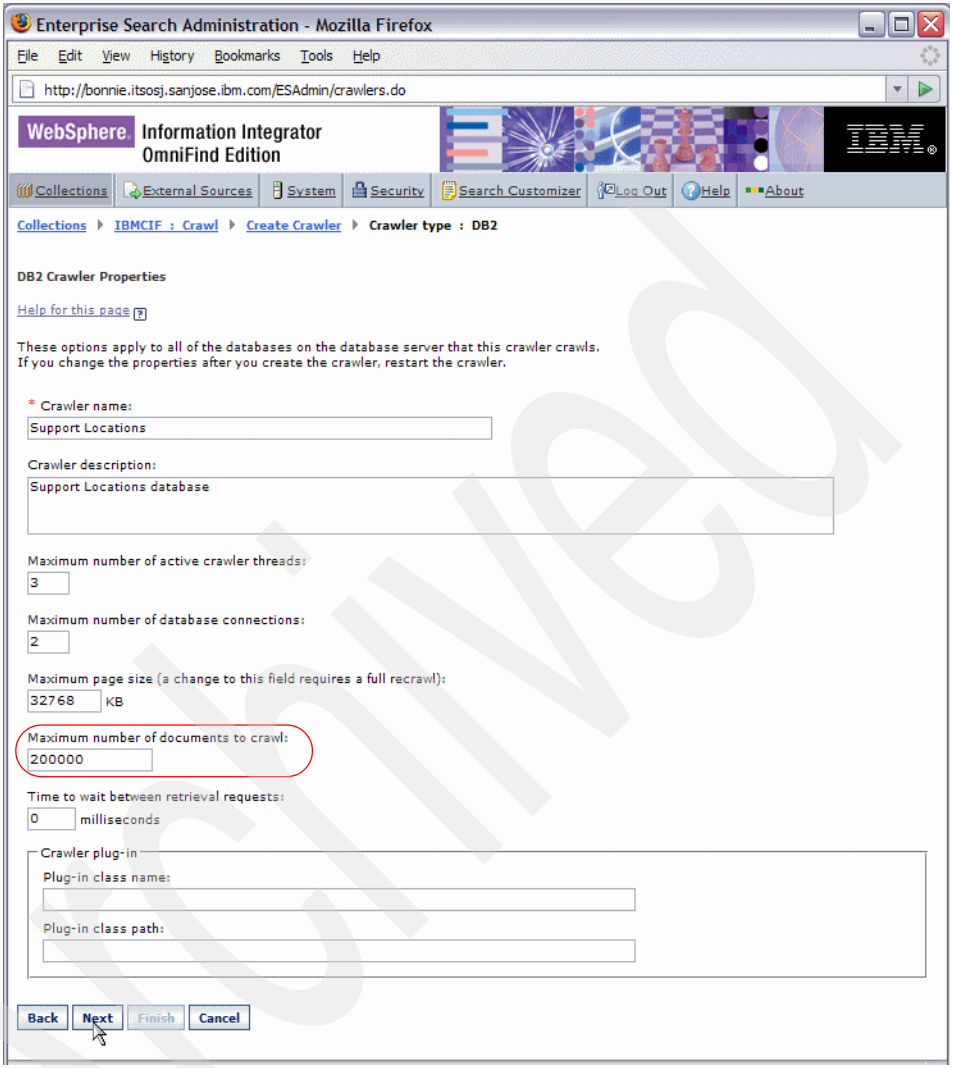

Figure 4-37 Crawler details

Figure 4-38 Select the DB2 Database Type

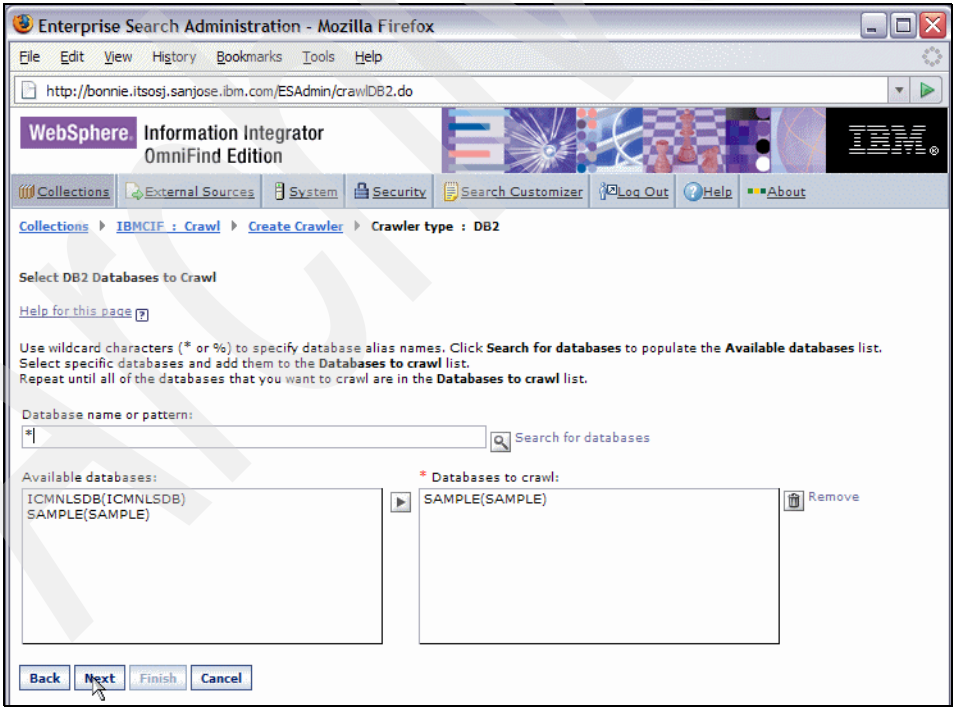

Figure 4-39 Select DB2 Databases to Crawl

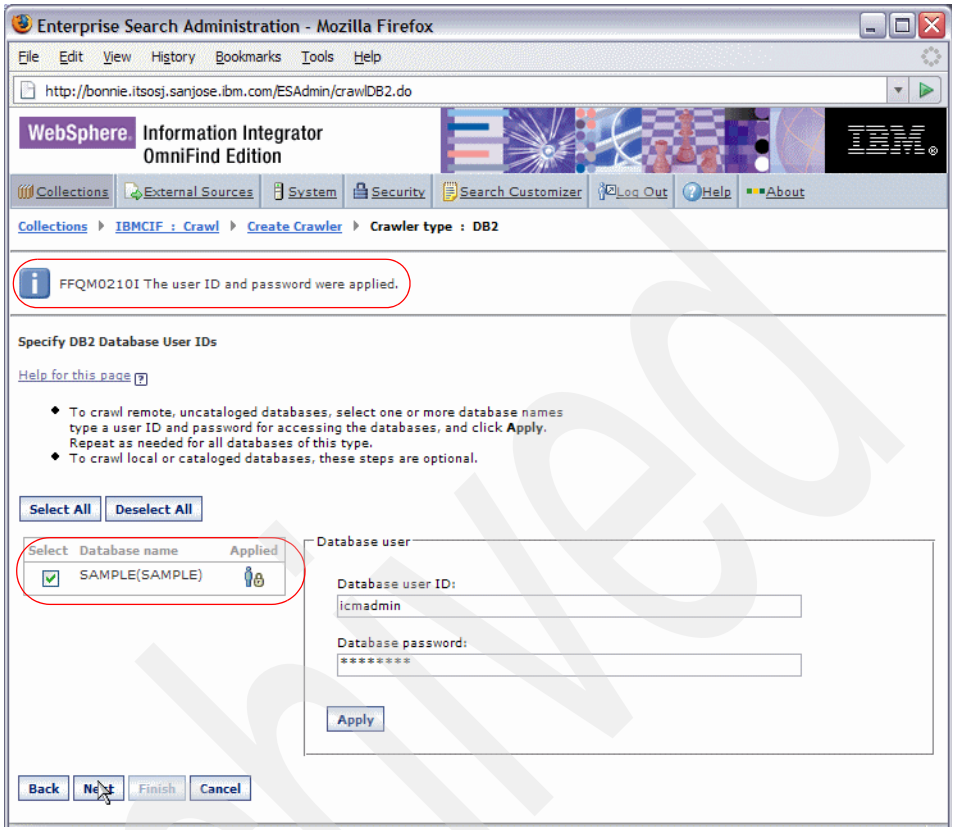

Figure 4-40 Select DB2 Database User IDs

Figure 4-41 Crawler schedule

Figure 4-42 Select DB2 Tables to Crawl

Figure 4-43 Select DB2 Tables that Use Event Publishing

Figure 4-44 Select Individual DB2 Tables to Configure

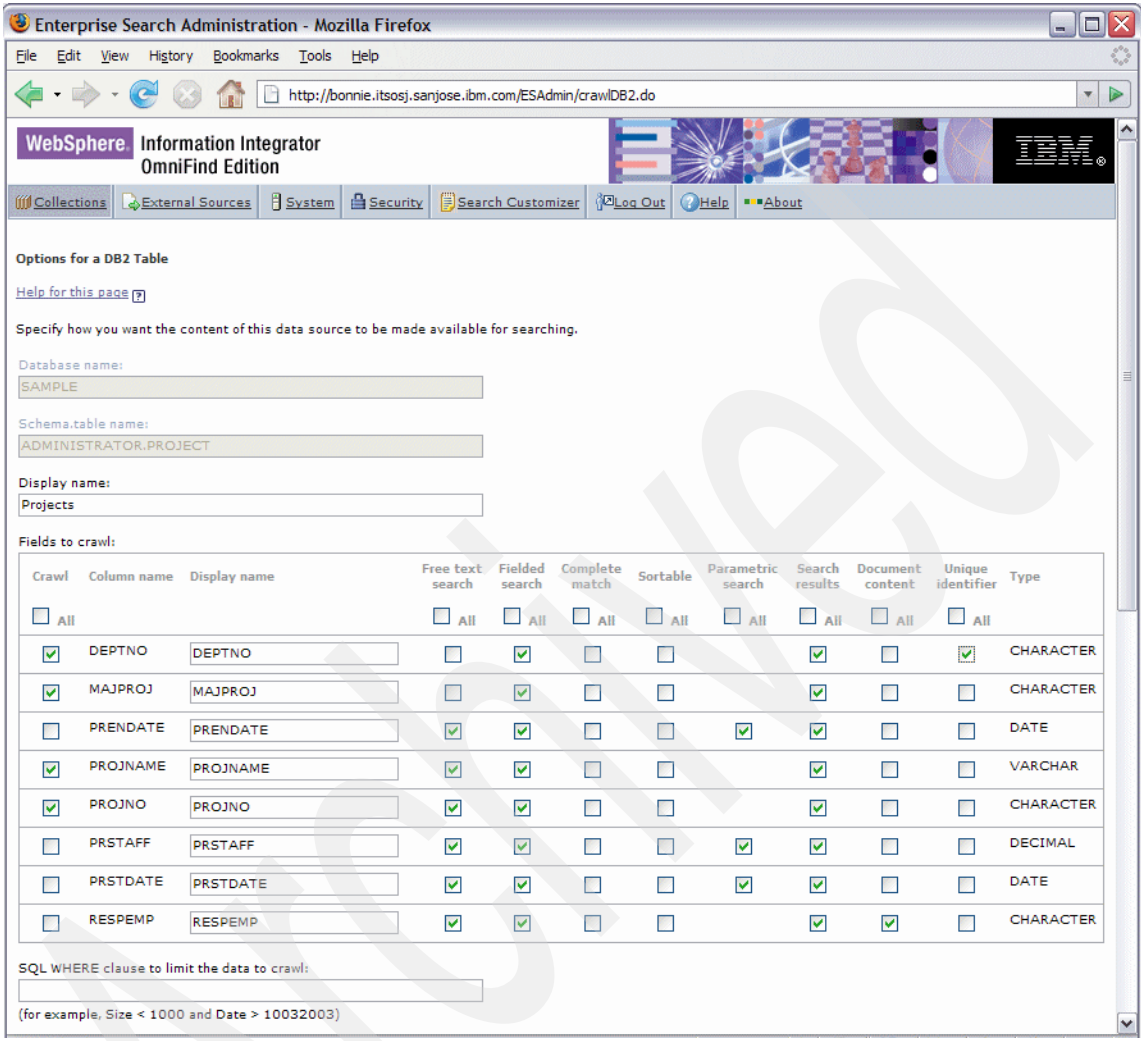

Figure 4-45 Options for a DB2 Table

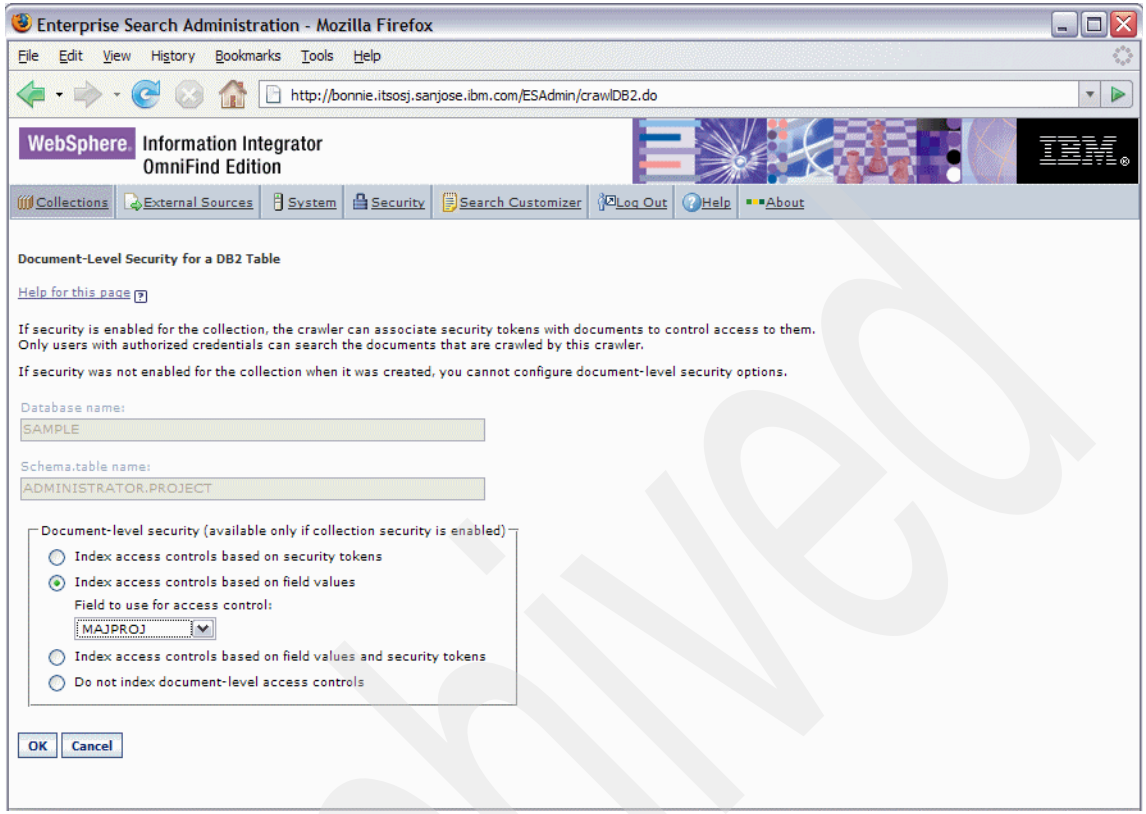

Figure 4-46 Document-Level Security for a DB2 table

Figure 4-47 Select Individual DB2 Tables to Configure

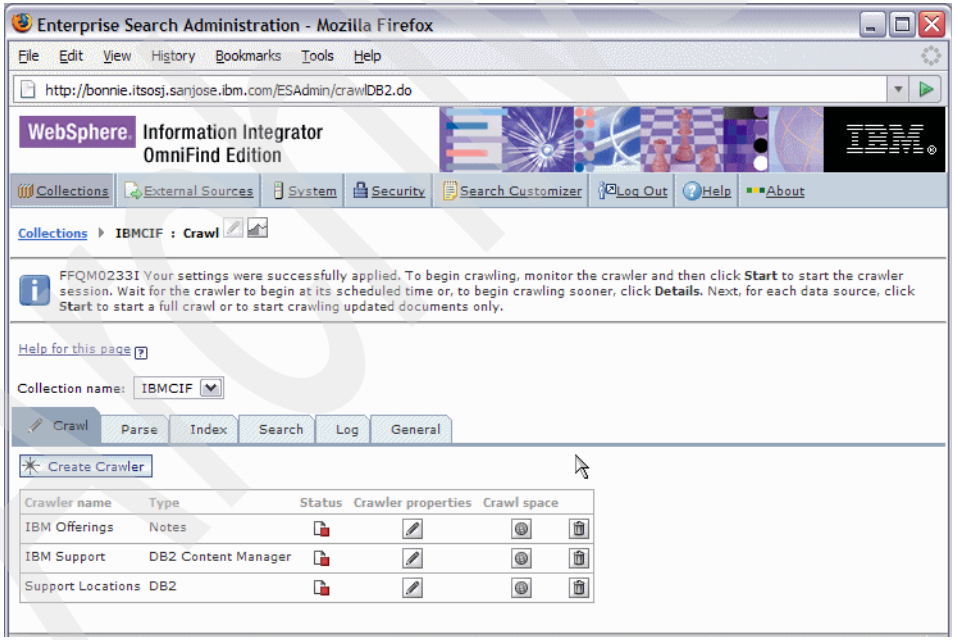

Figure 4-48 Status of all crawlers defined in the IBMCIF collection

ASTEP4c: Crawl the data sources

In this step, the we initiate a full crawl of the data sources crawled by the Notes, DB2 Content Manager, and DB2 crawlers. Figure 4-49 on page 349 through Figure 4-66 on page 363 describe the steps involved. Click the **Crawl** icon for the IBMCIF collection in the Collections view in Monitor mode, as shown in Figure 4-49 on page 349, to view the crawlers defined for this collection in Figure 4-50 on page 349.

Attention: When running crawlers in OmniFind V8.4, we strongly recommend that you run the parser at the same time. This is because a file queue is used to store crawled data instead of a DB2 table used in OmniFind V8.3. The file queue can fill up if the parser is not used to parse and delete the documents in the file queue while the crawler is running.

The Notes, DB2 Content Manager, and DB2 crawlers can now be started as follows:

- Notes data sources

  Start the crawler session by clicking the start icon for the IBM Offerings crawler, as shown in Figure 4-50 on page 349. Once the Status icon turns green, click **Details**, as shown in Figure 4-51 on page 350. Start a full crawl of the TrainingOffering.nsf database by clicking the appropriate icon, as shown in Figure 4-52 on page 351. Monitor the status of the crawl by clicking the **Refresh** button intermittently, until the completion of the crawl, as shown in Figure 4-53 on page 352. The Status icon turns green.

  Repeat this process for the mark_doc.nsf and Corporat.nsf databases, as shown in Figure 4-53 on page 352 through Figure 4-56 on page 355

  Note: We stopped the crawler to conserve resources in our constrained environment.

  We can now start a full crawl of the DB2 Content Manager data sources as described in “DB2 Content Manager data sources” on page 355.

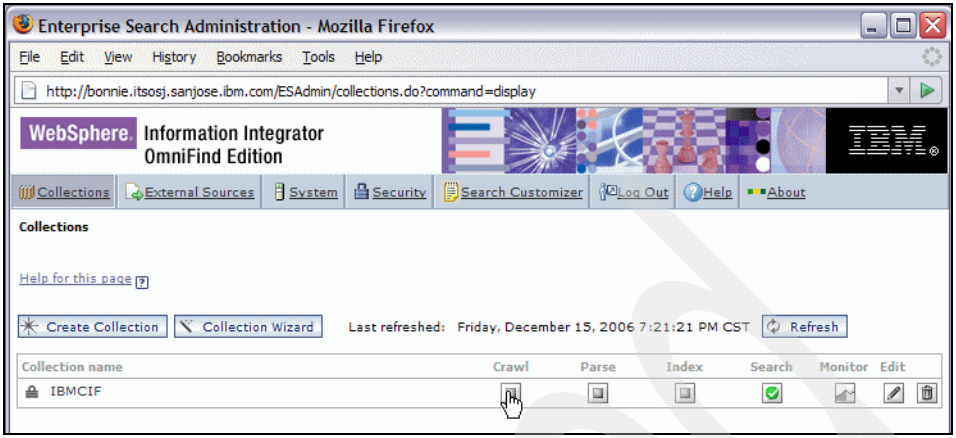

Figure 4-49 Click Crawl icon for IBMCIF collection

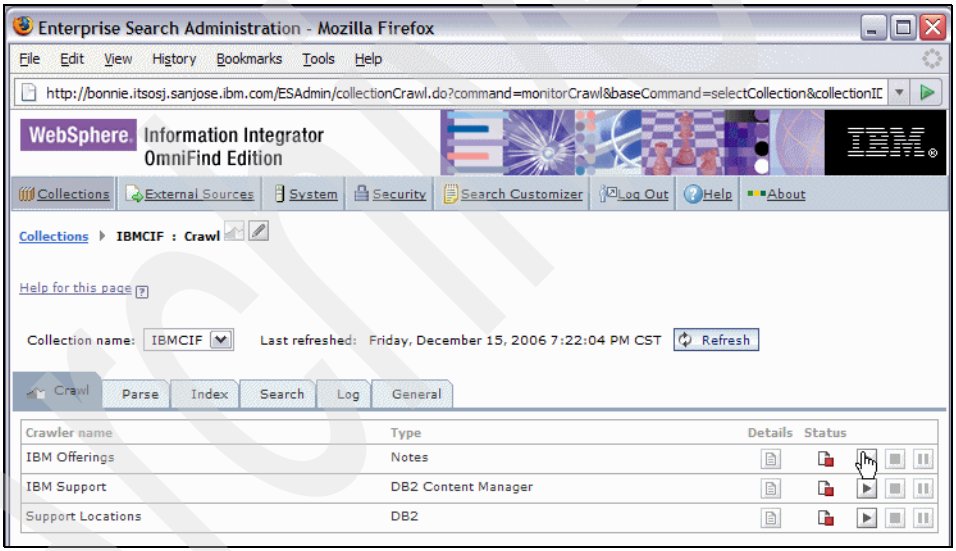

Figure 4-50 Start crawler session for Notes crawler

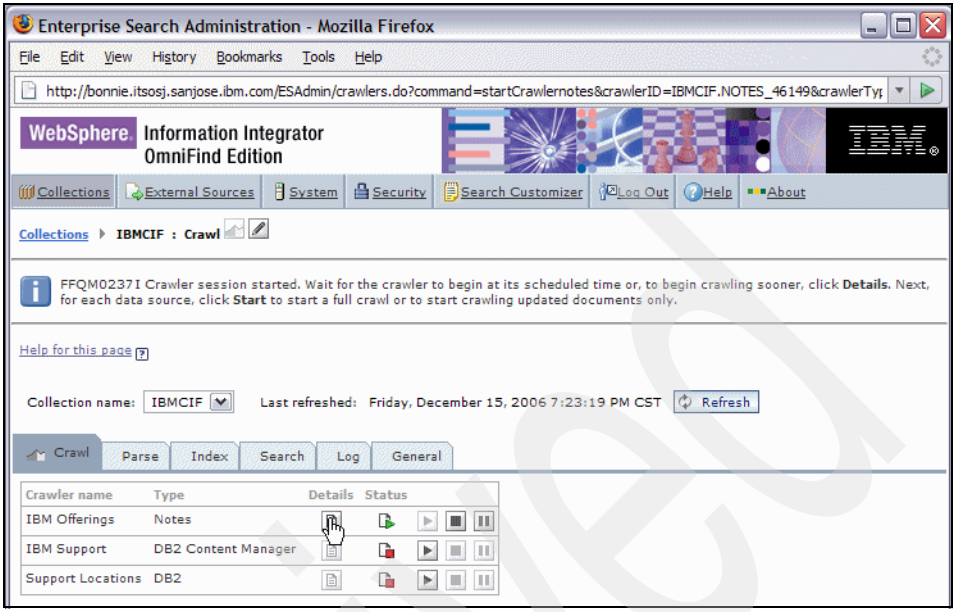

Figure 4-51 Click Details icon

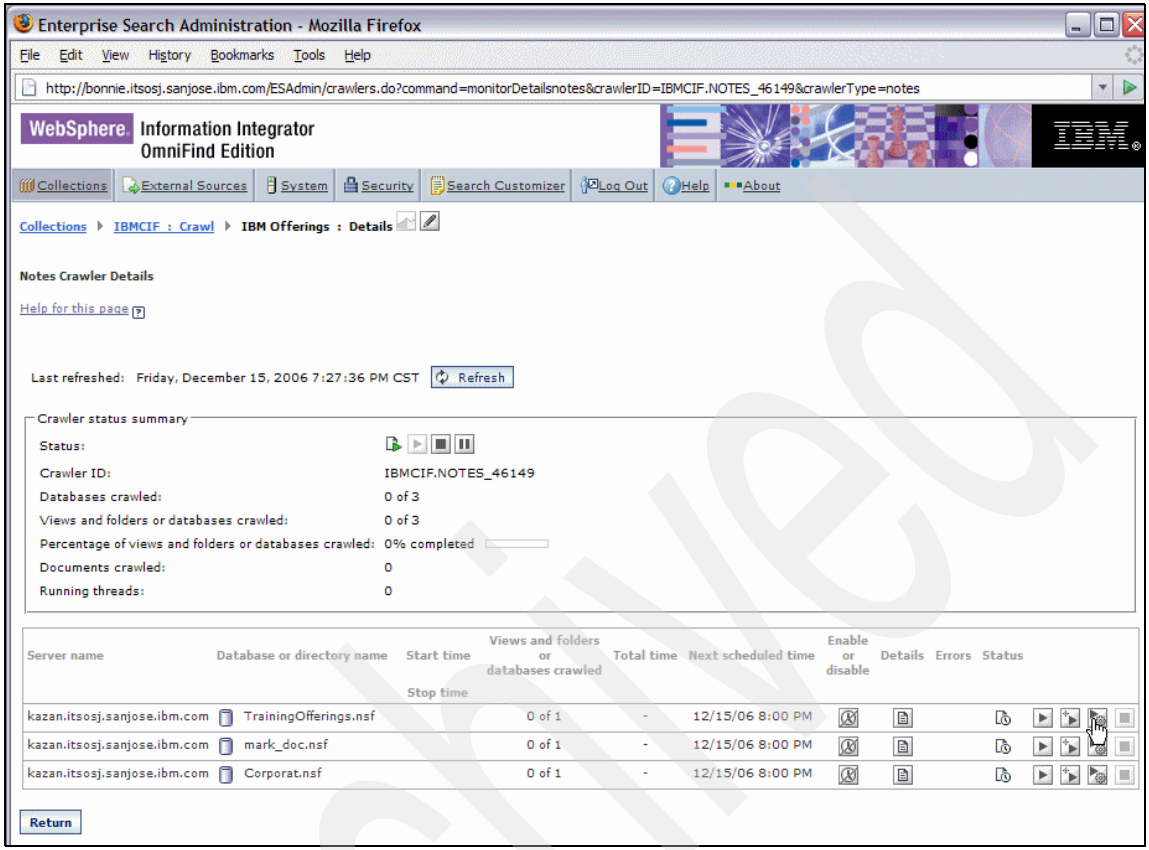

Figure 4-52 Start a full crawl of the TrainingOfferings.nsf database

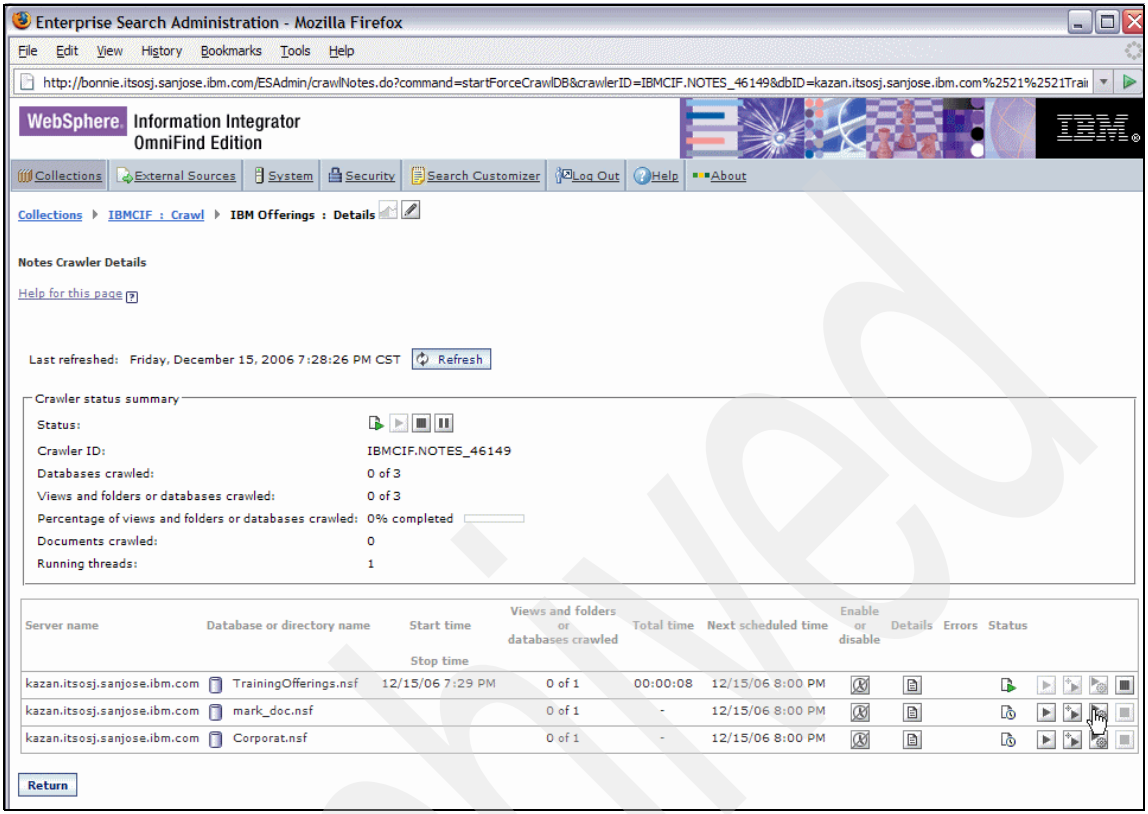

Figure 4-53 Start a full crawl of the mark_doc.nsf database

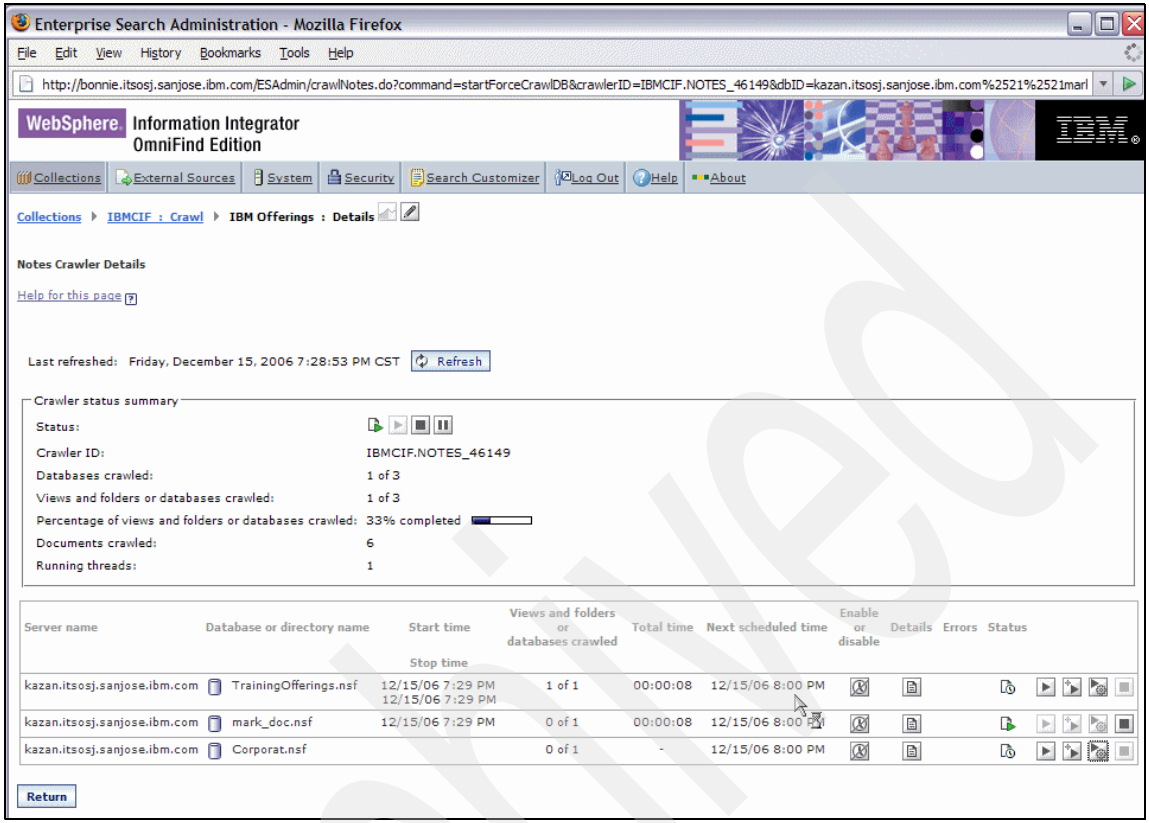

Figure 4-54 Start a full crawl of the Corporat.nsf database

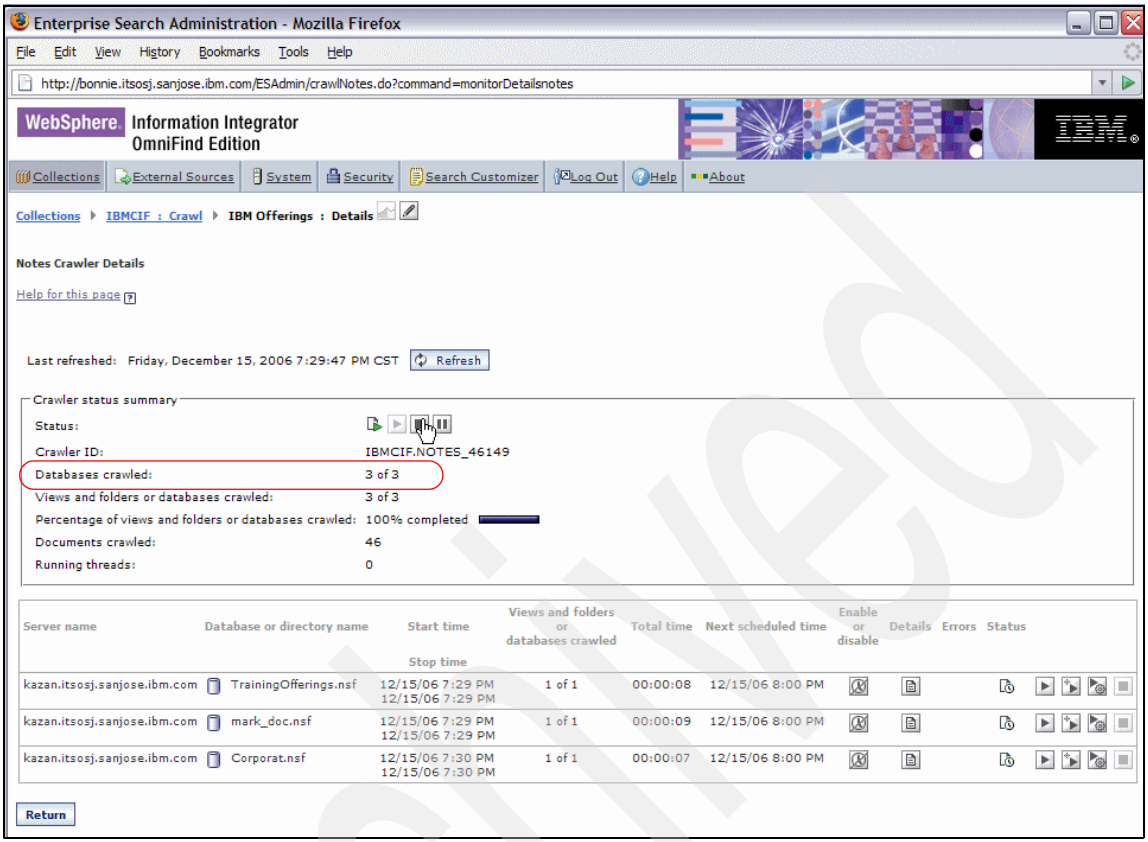

Figure 4-55 Successful crawl of all three databases

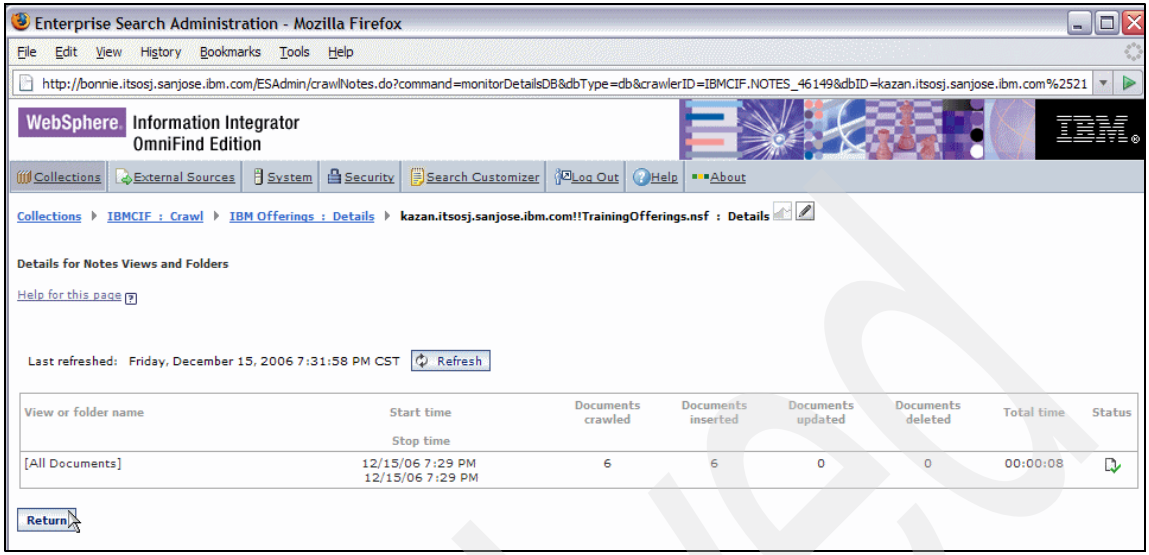

Figure 4-56 Details for Notes Views and Folders

- DB2 Content Manager data sources

  Start the crawler session by clicking the start icon for the IBM Support crawler, as shown in Figure 4-57 on page 356. Once the Status icon turns green, click **Details**, as shown in Figure 4-58 on page 356. Start a full crawl of the icmnlsdb database by clicking the appropriate icon, as shown in Figure 4-59 on page 357. Monitor the status of the crawl by clicking the **Refresh** button intermittently, until the completion of the crawl as shown in Figure 4-60 on page 358. The Status icon turns green. Click **Details** to view statistics about the crawled item types Support_Pubs and Support_TS in Figure 4-61 on page 359.

  Note: We stopped the crawler in Figure 4-62 on page 360 to conserve resources in our constrained environment.

  We can now start a full crawl of the DB2 data sources, as described in “DB2 data sources” on page 360.

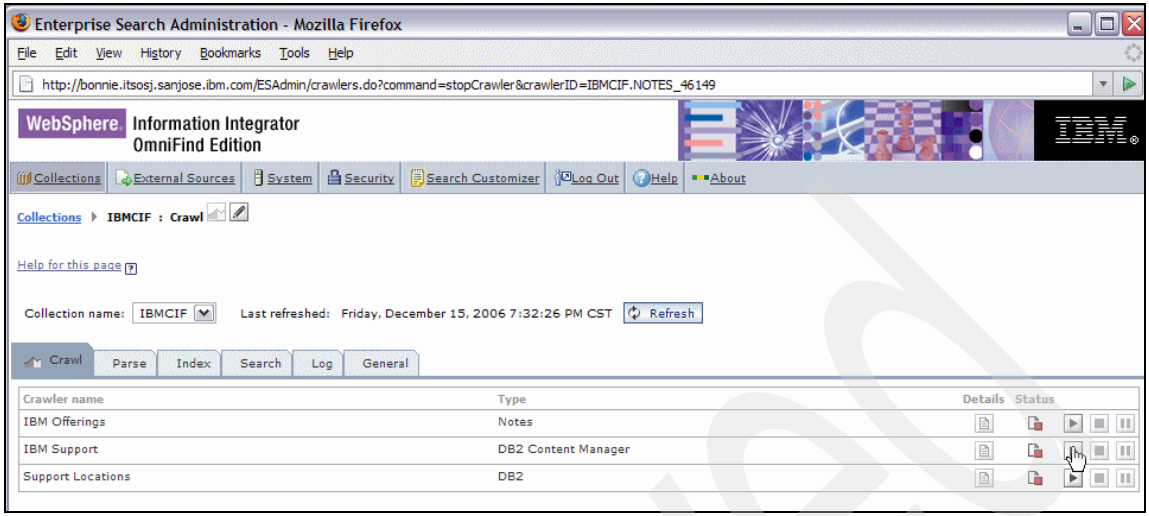

Figure 4-57 Start crawler session for DB2 Content Manager crawler

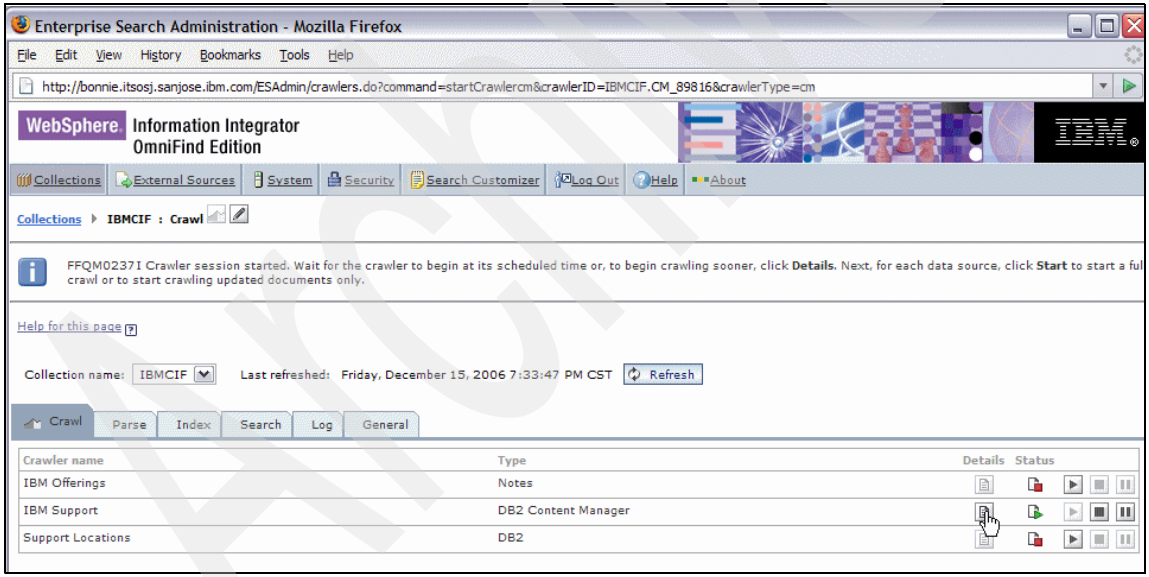

Figure 4-58 Click Details icon

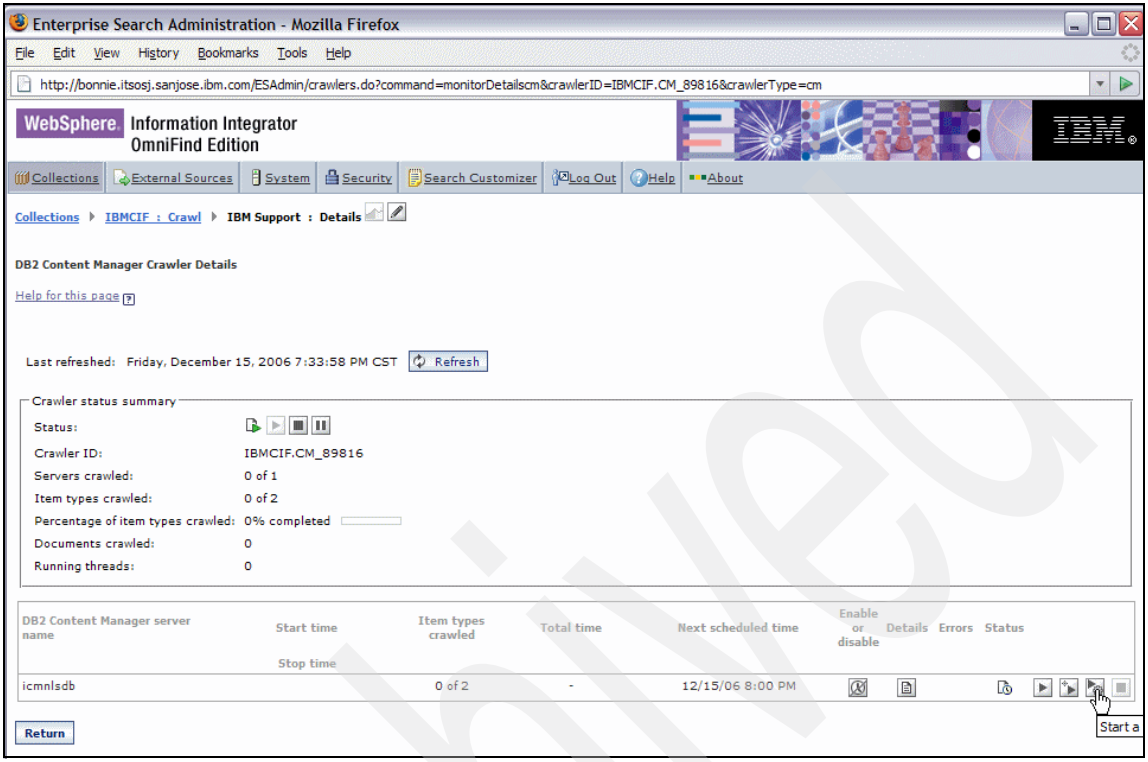

Figure 4-59 Start a full crawl

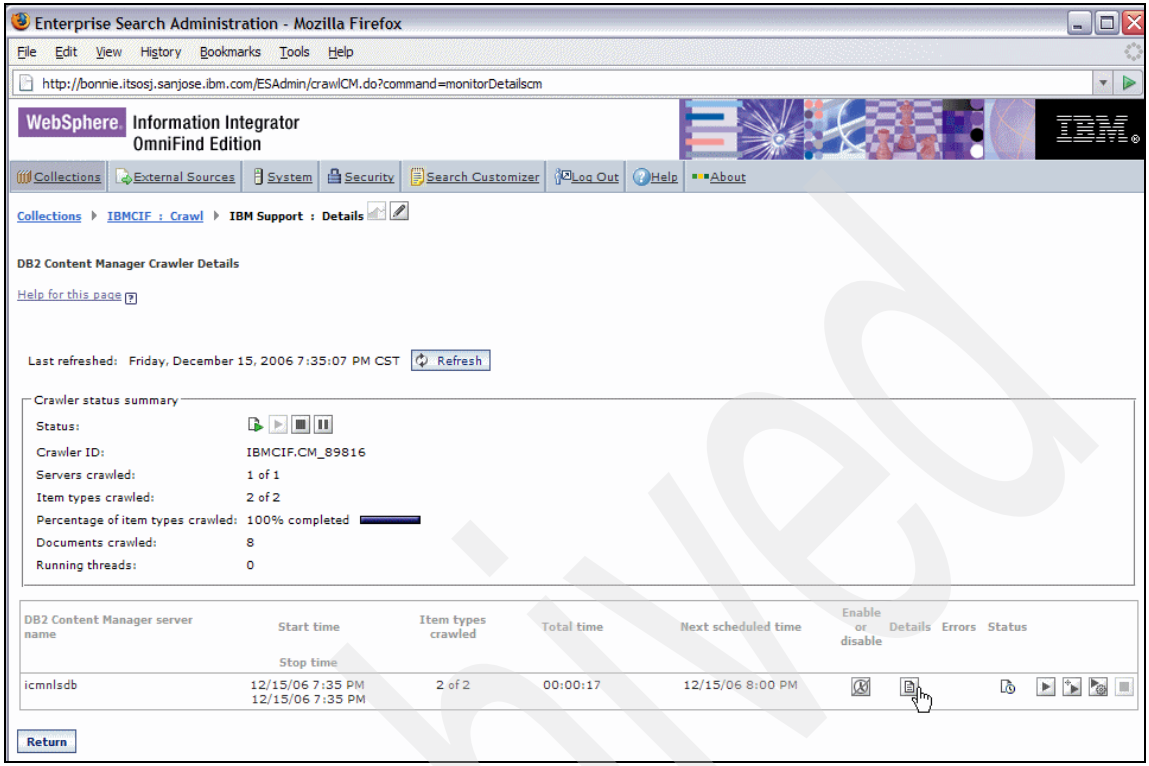

Figure 4-60 Click Details icon

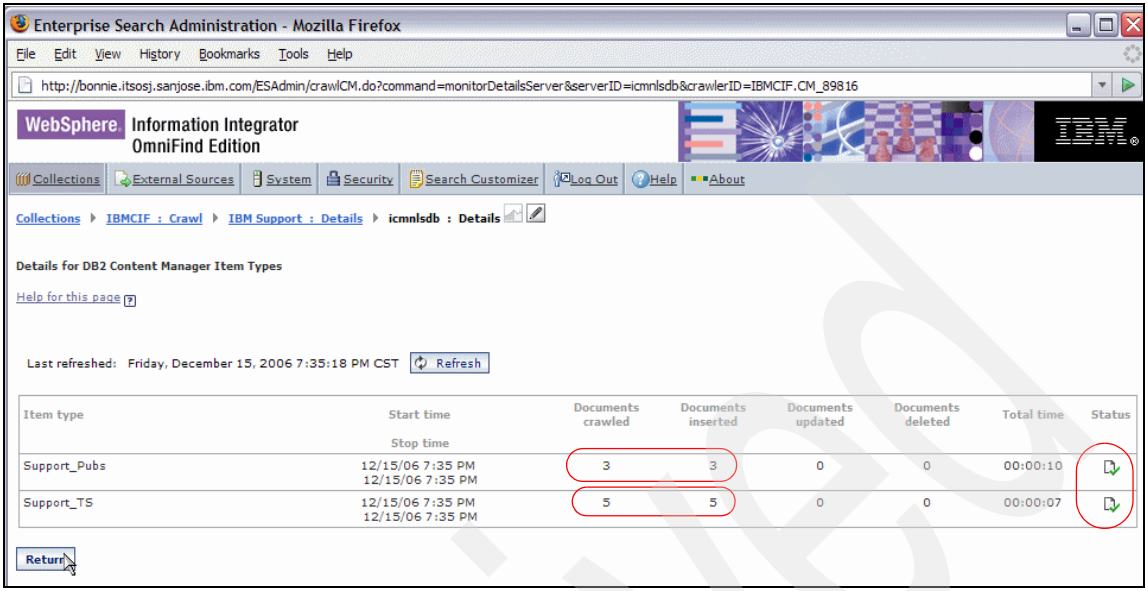

Figure 4-61 Successful completion of the crawl of DB2 Content Manager data sources 1/2

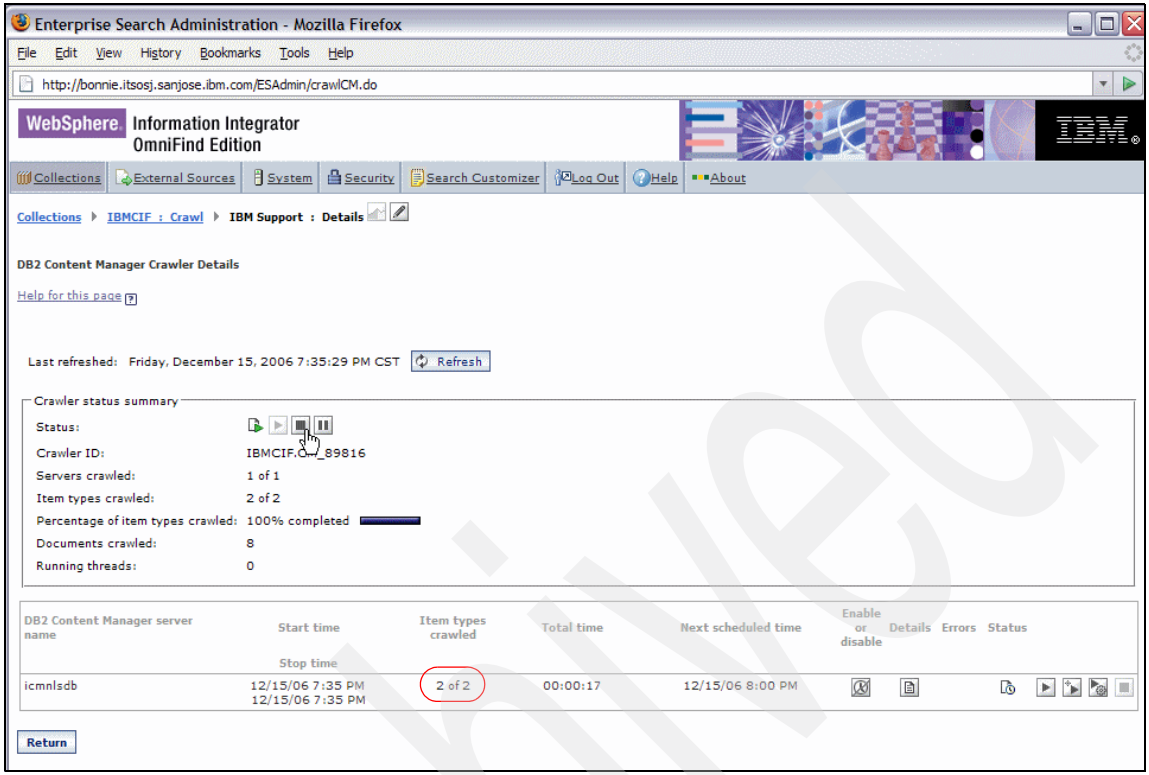

Figure 4-62 Successful completion of the crawl of DB2 Content Manager data sources 2/2

- DB2 data sources

  Start the crawler session by clicking the start icon for the IBM Support crawler, as shown in Figure 4-63 on page 361. Once the Status icon turns green, start a full crawl of the SAMPLE database by clicking the appropriate icon (not shown here). Monitor the status of the crawl by clicking the **Refresh** button intermittently, until the completion of the crawl, as shown in Figure 4-64 on page 361. The Status icon turns green. Click **Details** to view statistics about the crawled ADMINISTRATOR.DEPARTMENT, as shown in Figure 4-65 on page 362.

  Note: We stopped the crawler in Figure 4-66 on page 363 to conserve resources in our constrained environment.

  We can now proceed to configure the categories for the IBMCIF collection, as described in “ASTEP4d: Configure the categories” on page 364.

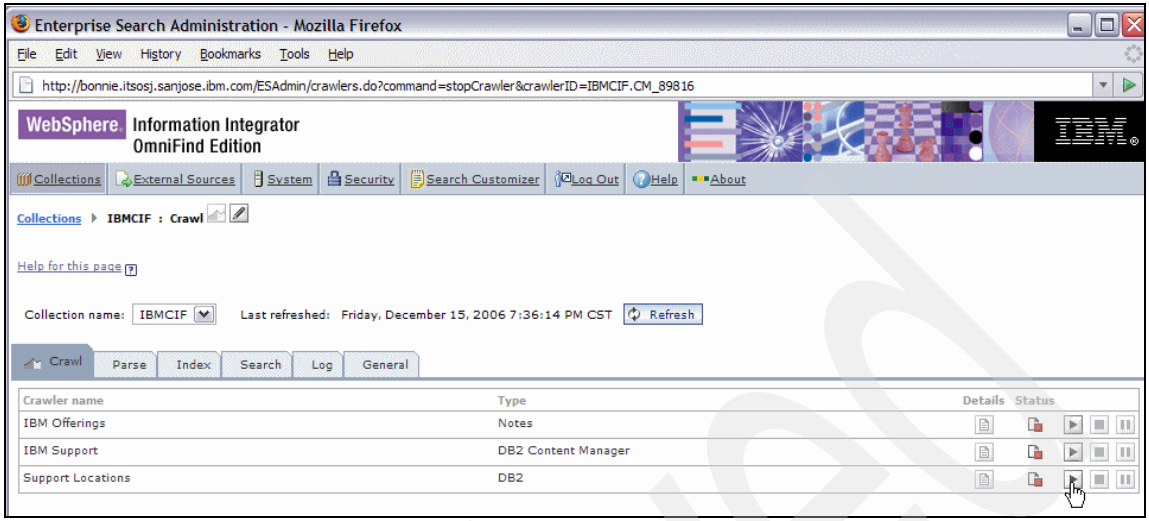

Figure 4-63 Start crawler session for the DB2 crawler

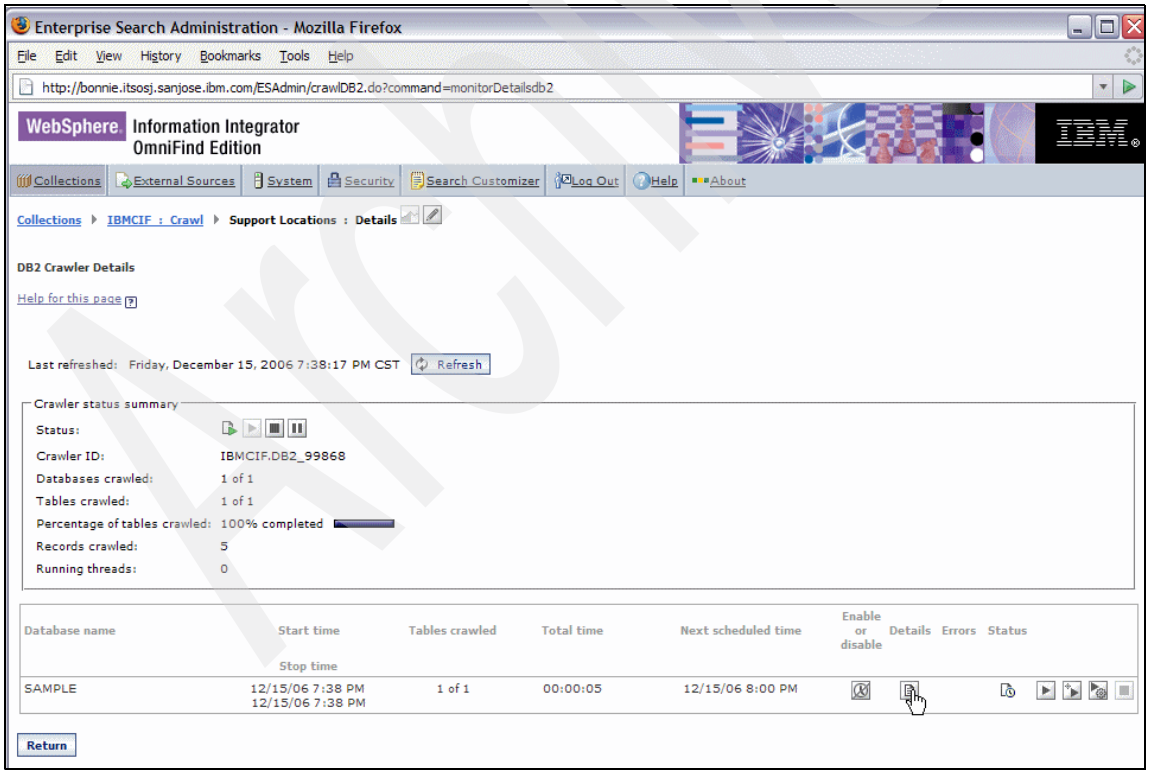

Figure 4-64 Click Details icon

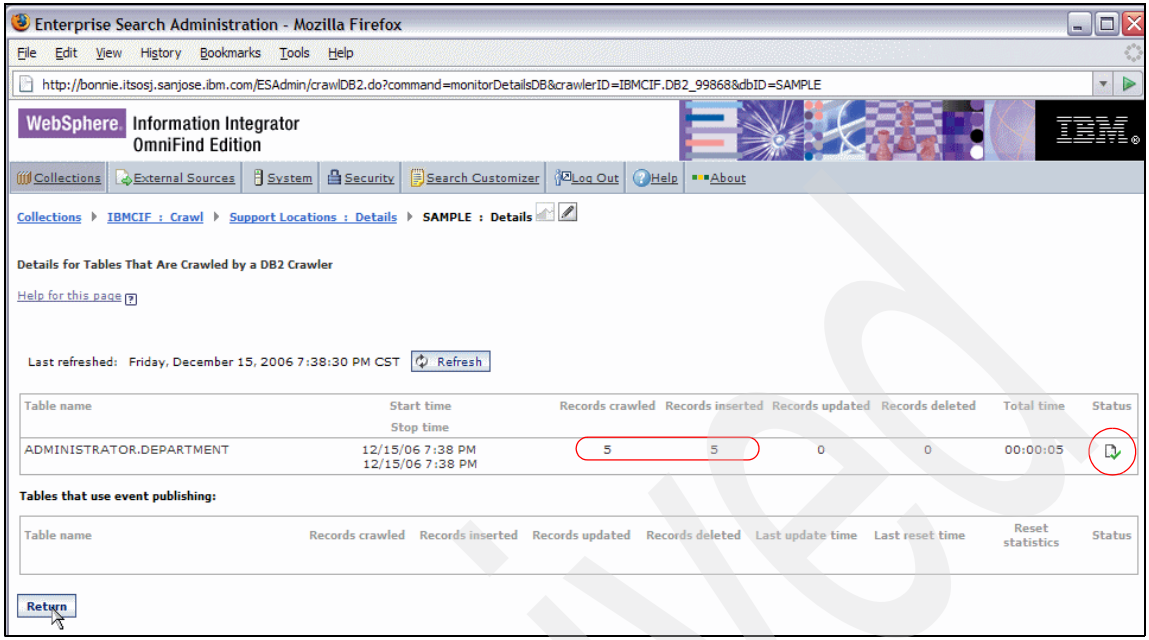

Figure 4-65 Successful crawl of DB2 data sources 1/2

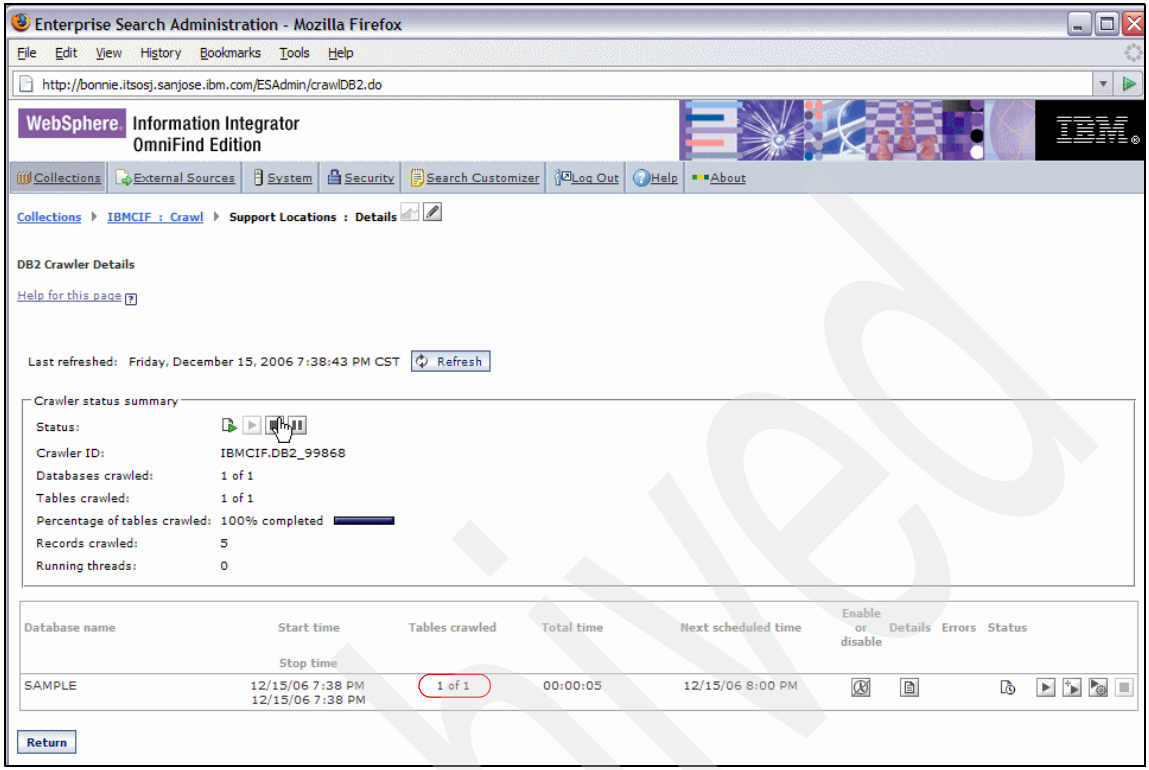

Figure 4-66 Successful crawl of DB2 data sources 2/2

ASTEP4d: Configure the categories

In this step, we define the category tree[3] designed for the IBMCIF collection, as shown in Figure 4-67, in order to enable users to search a subset of the collection by specifying the category name.

Figure 4-67 Category tree for the IBMCIF collection

Users will be able to select a category in the search results and browse only the documents that belong to the selected category. When users search a category, or browse documents that belong to a category when they browse search results — the fact that a document belongs to multiple categories enhances the likelihood that users will find it. Briefly:

- If a user searches a high-level category, that category and all of its subcategories are searched for documents that match the search criteria. If a user searches a category that has no additional subcategories, only that category is searched.
- If a user is browsing search results and selects an option to browse documents that belong to a specific category, only the documents in that category are displayed. The names of any subcategories are also displayed in

[3] A category tree, which is also called a taxonomy, is arranged in a hierarchy. The tree starts with the root category, and all other categories stem from the root category. You can nest any number of categories and subcategories to provide users with different choices for browsing and retrieving documents. For example, if a document passes the rules in several categories, it is associated with all of those categories.

the search results, so that the user can navigate between categories and view subsets of documents at a time.

Note: You also use the category tree to delete categories from the collection and to change the rules for associating documents with categories. When you edit a category, you can rename the category, add or delete categorization rules, or modify the content of individual rules. If you change categories or category rules after you crawl data and create an index for a collection, the index becomes inconsistent. To ensure the accuracy of search results, re-crawl all documents in the collection and rebuild the main index.

Figure 4-68 on page 366 through Figure 4-85 on page 376 describe the steps in creating some of the categories shown in Figure 4-67 on page 364, with the intention that the steps could easily be extrapolated by the reader to include all the categories designed.

From the Collections view for the IBMCIF collection in Edit mode, under the Parse tab, click **Configure the category tree**, as shown in Figure 4-68 on page 366. On the Category Tree page, select the location[4] in the tree (root) where you want to add a category and click **Create a category**, as shown in Figure 4-69 on page 367.

A wizard opens to help you specify rules for associating documents with the new category. Specify the Category name (Corporate News) and click **Next**, as shown in Figure 4-70 on page 367. On the Create Category Rules page, click **Add Rule**, as shown in Figure 4-71 on page 368, and then type a unique name for the rule in the Rule name (**News**) field, as shown in Figure 4-72 on page 368. This name must be unique across all categories in the collection. Specify the rule that you want to use for associating documents with this category. To use the URI of a document to determine whether the document belongs to the category, click **URI pattern** and then specify the URI pattern (*Corporat.nsf*) in Figure 4-72 on page 368; if the text that you specify exists in the URI, the document is associated with the category. Click **OK** in Figure 4-72 on page 368, and then **Finish** in Figure 4-73 on page 369 to complete the definition of the Corporate News category. Repeat the process to create the Products category

Figure 4-74 on page 369 through Figure 4-76 on page 370 show the creation of the Products category without the creation of a rule. This is because the Products category will have subcategories defined with the appropriate rules. The Product category then inherits the rules of all its subcategories.

[4] If you select the root, the new category is created at the root level. If you select a category name, the new category is nested below the selected category in the category tree.

To create subcategories for the Products category, select the Products category in the navigation pane and click **Create a category**, as shown in Figure 4-77 on page 371. On the Create a Category page, specify the Category name (Hardware) and click **Next**, as shown in Figure 4-78 on page 371. In Create a Category Rule in Figure 4-79 on page 372, to determine whether a document belongs to the category by querying searchable content, select **Document content**, select the language (**English**) of the documents from the Default query language drop-down list, and then specify the words and phrases (**Products > Hardware >**) that must or must not appear in the document content. You express the rule in the same format as a query, but only the include (+), exclude (-), phrase (• •), and field name (field_name:) query operators are allowed. N-gram segmentation is not supported with content rules. If a document includes or excludes the words that you specify, the document is associated with the category. Click **OK** in Figure 4-79 on page 372, and then **Finish** in Figure 4-80 on page 372 to complete the addition of the Hardware category to the Category Tree page, as shown in Figure 4-81 on page 373.

This process is repeated until all the categories and subcategories are defined. Figure 4-82 on page 374 shows the partial category tree defined for the IBMCIF collection.

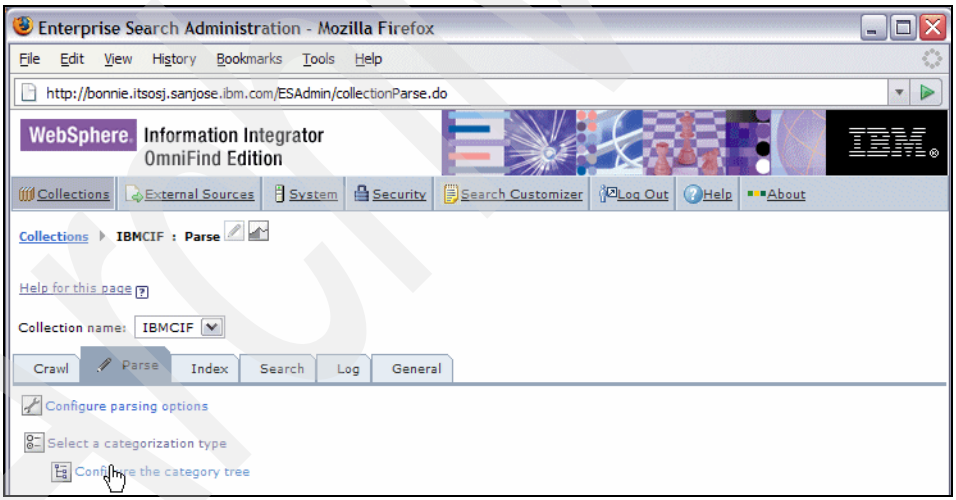

Figure 4-68 Configure the category tree

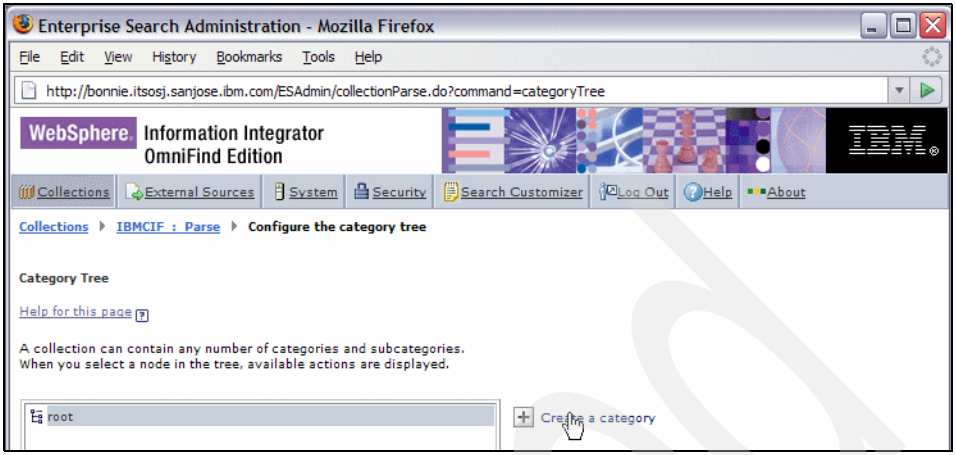

Figure 4-69 Create a category

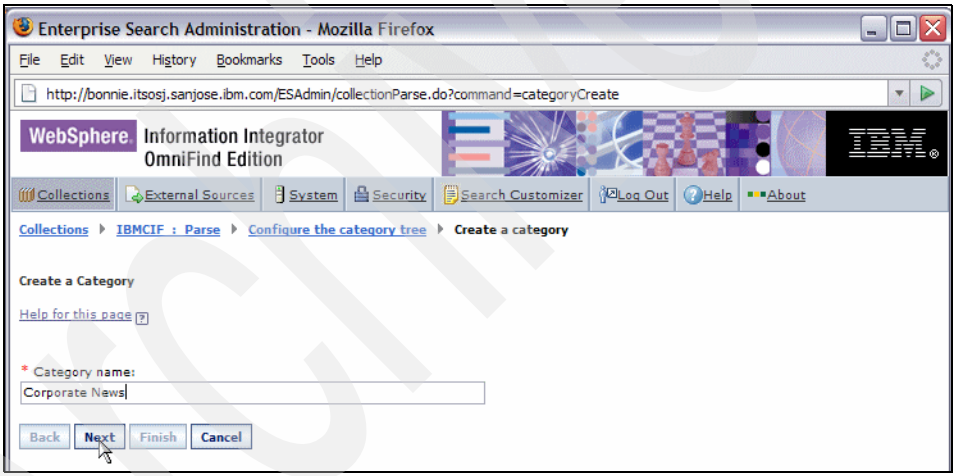

Figure 4-70 Corporate News category

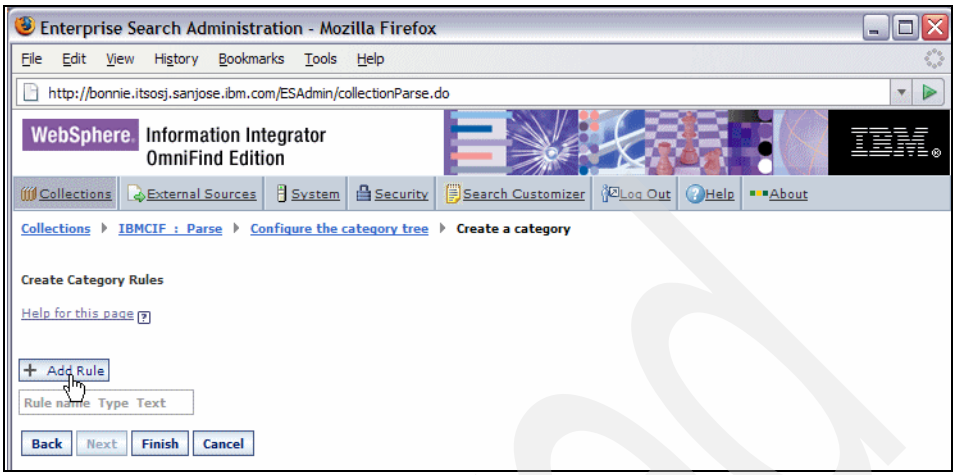

Figure 4-71 Add Rule for Corporate News category 1/3

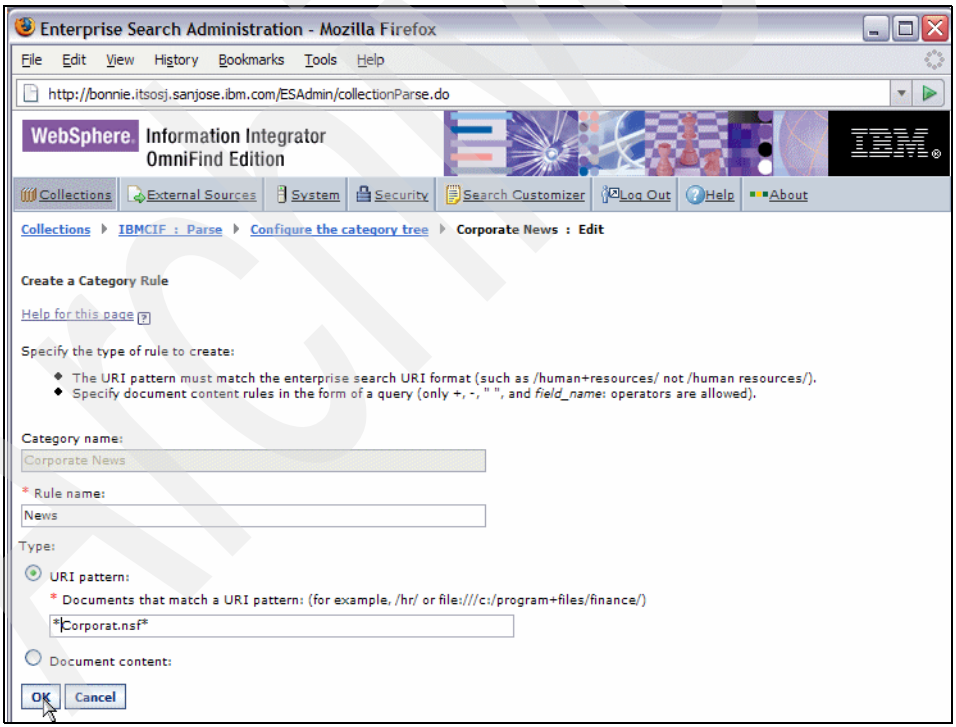

Figure 4-72 Add Rule for Corporate News category 2/3

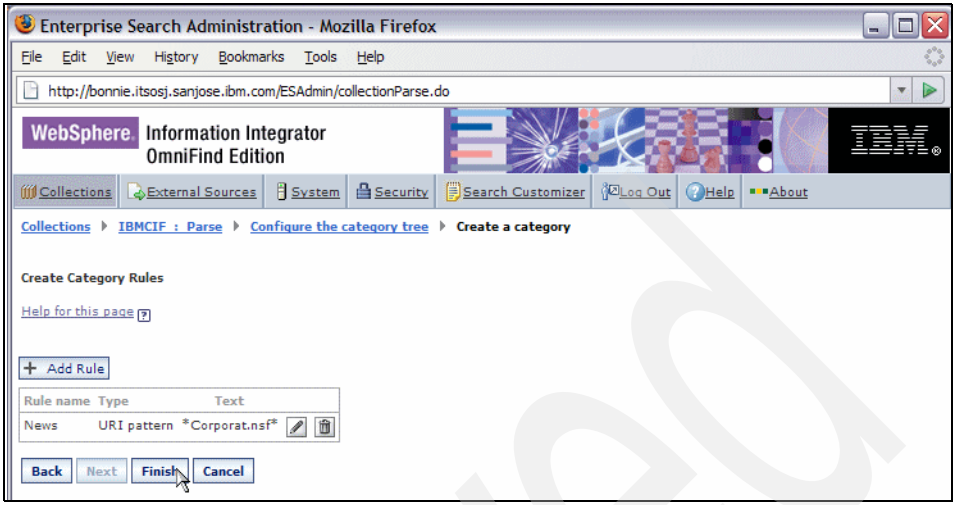

Figure 4-73 Add Rule for Corporate News category 3/3

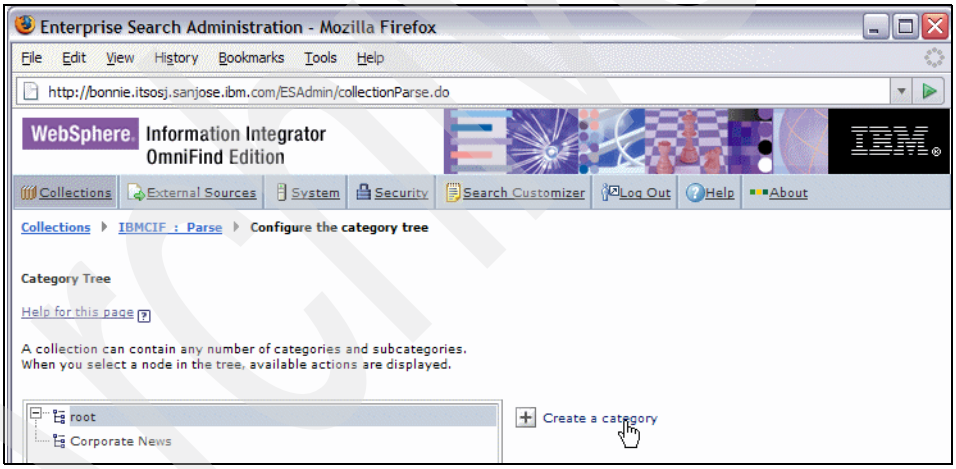

Figure 4-74 Create another category under root

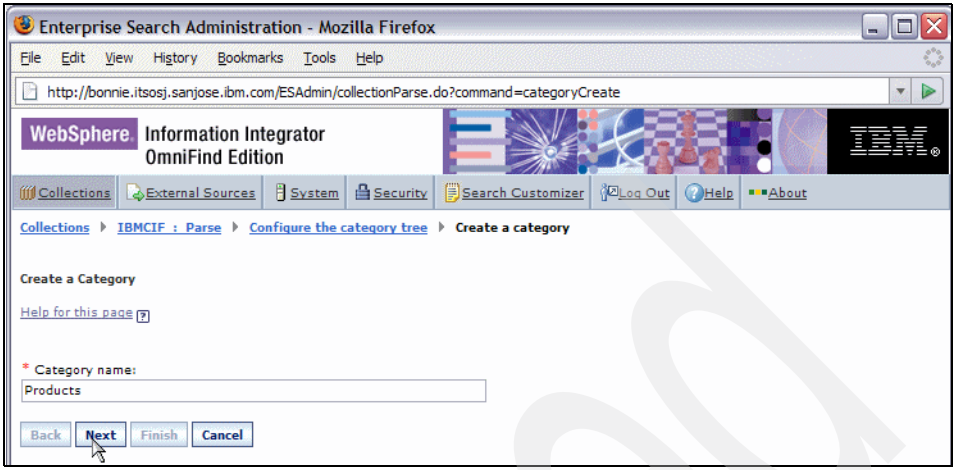

Figure 4-75 Product category

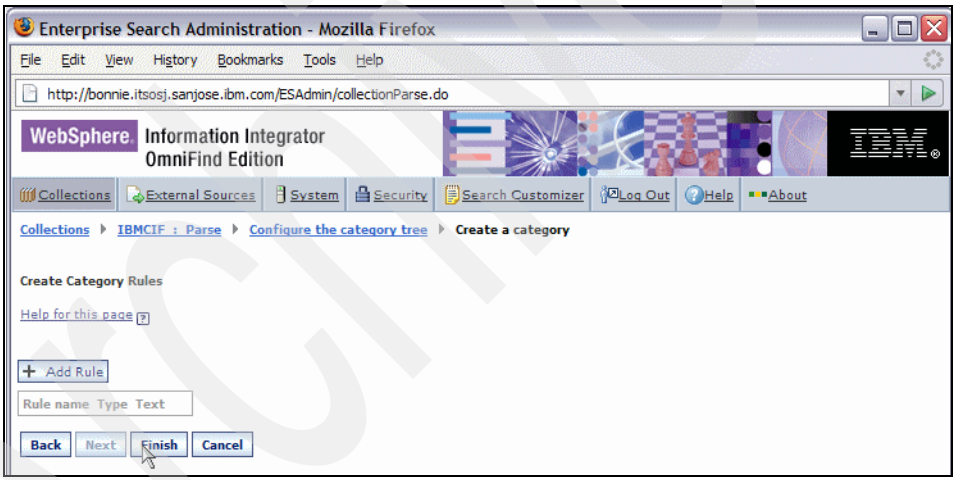

Figure 4-76 No rule for Products category

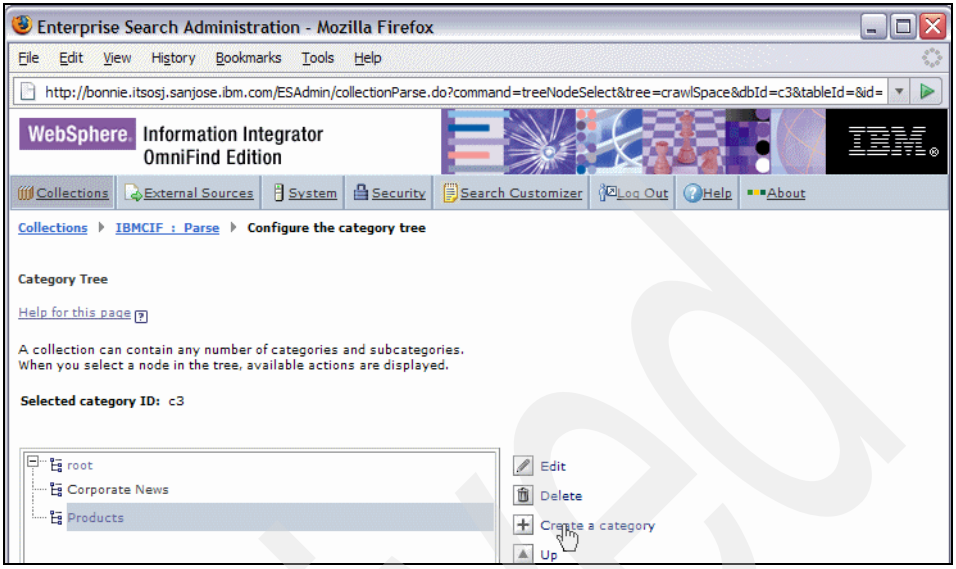

Figure 4-77 Create a category under the Products category

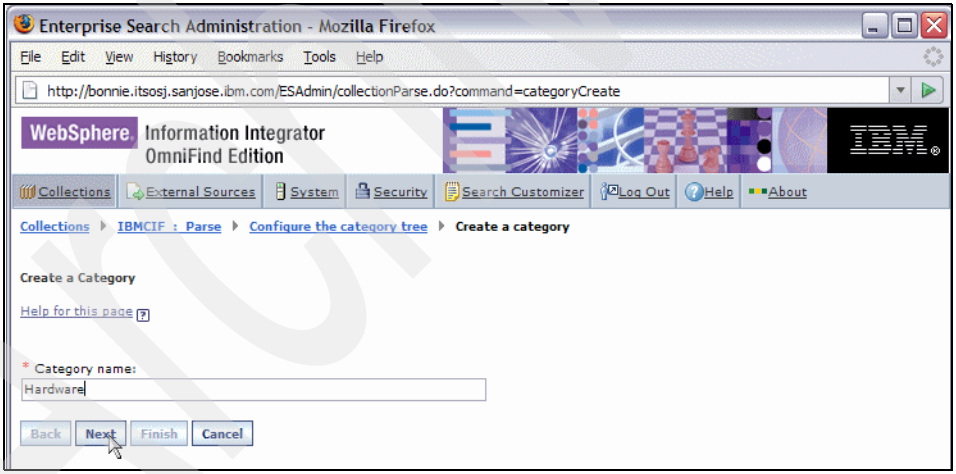

Figure 4-78 Hardware category under Products

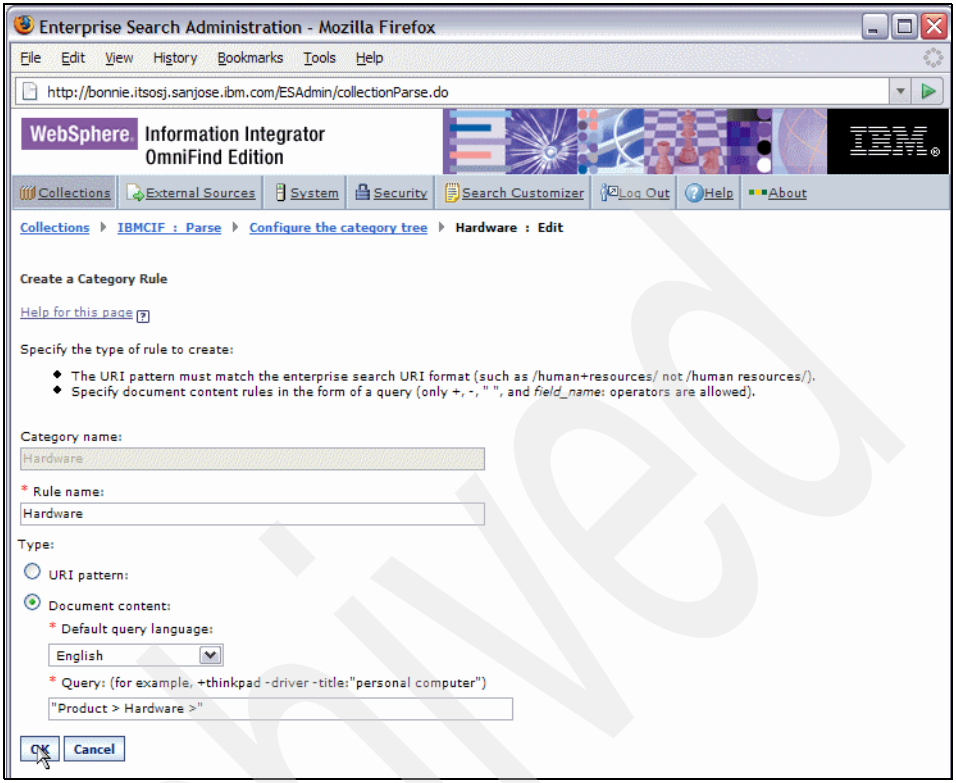

Figure 4-79 Add Rule for Hardware category under the Products category 1/2

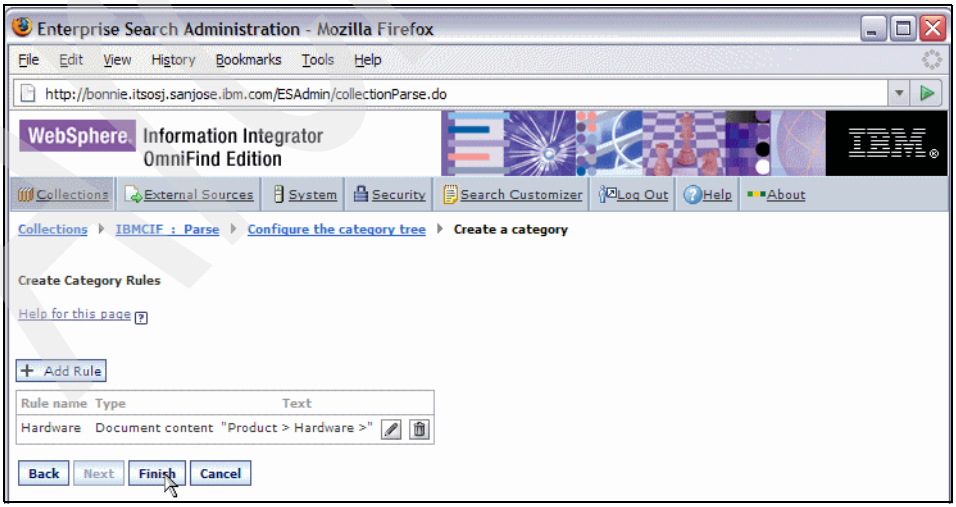

Figure 4-80 Add Rule for Hardware category under the Products category 2/2

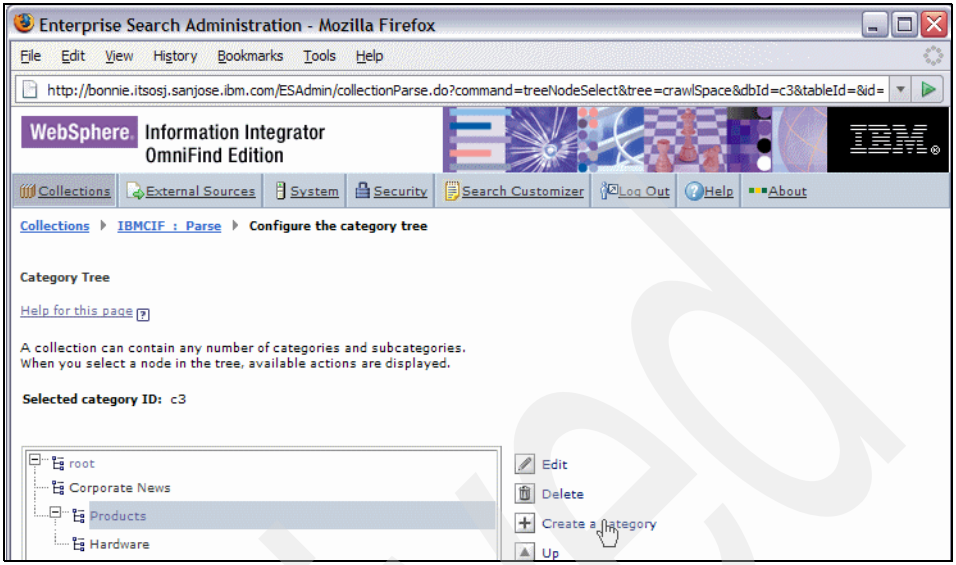

Figure 4-81 Create a category under the Products category

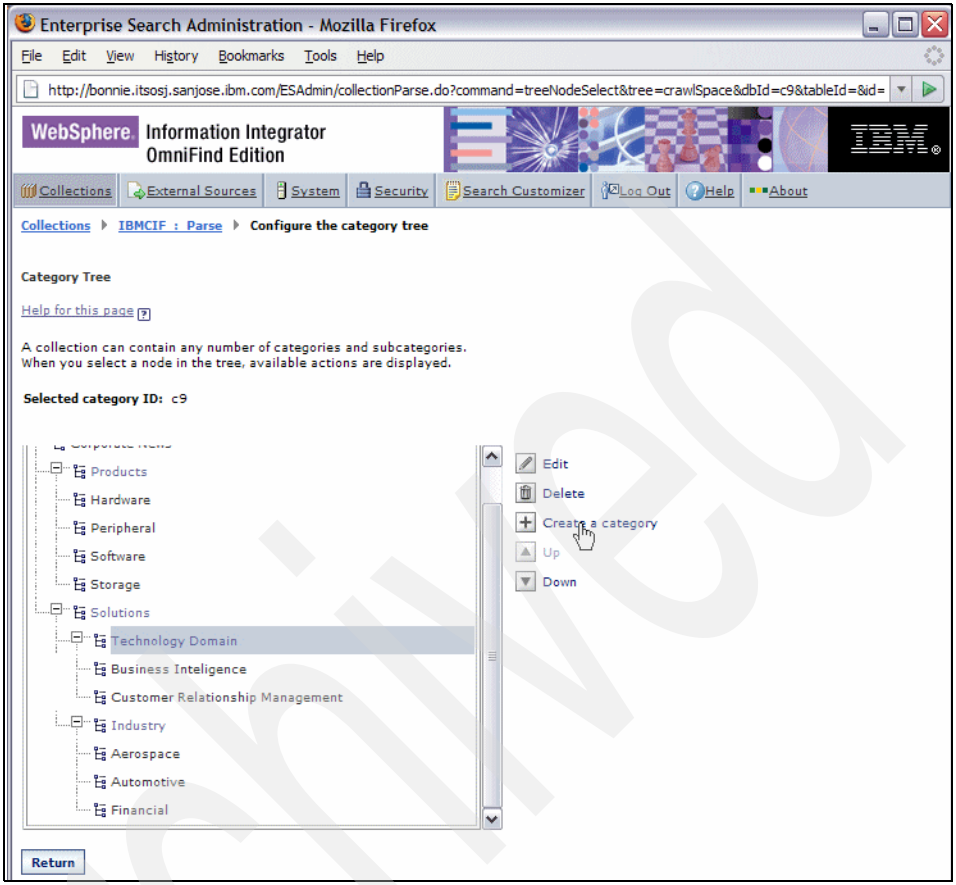

Figure 4-82 Partial category tree for the IBMCIF collection

ASTEP4e: Parse the crawled data

Figure 4-83 on page 375 through Figure 4-85 on page 376 describe the steps in parsing the crawled data of the IBMCIF collection.

From the Parse tab in Monitor mode, start the parser by clicking the start icon, as shown in Figure 4-83 on page 375. After the Status icon turns green, you can monitor the progress of parsing by clicking **Details**, as shown in Figure 4-84 on page 375. Periodically click the **Refresh** button until the parser completes processing, as shown in Figure 4-85 on page 376. Review the parsing statistics.

We can now proceed to build the main index, as described in “ASTEP4f: Build the main index” on page 376.

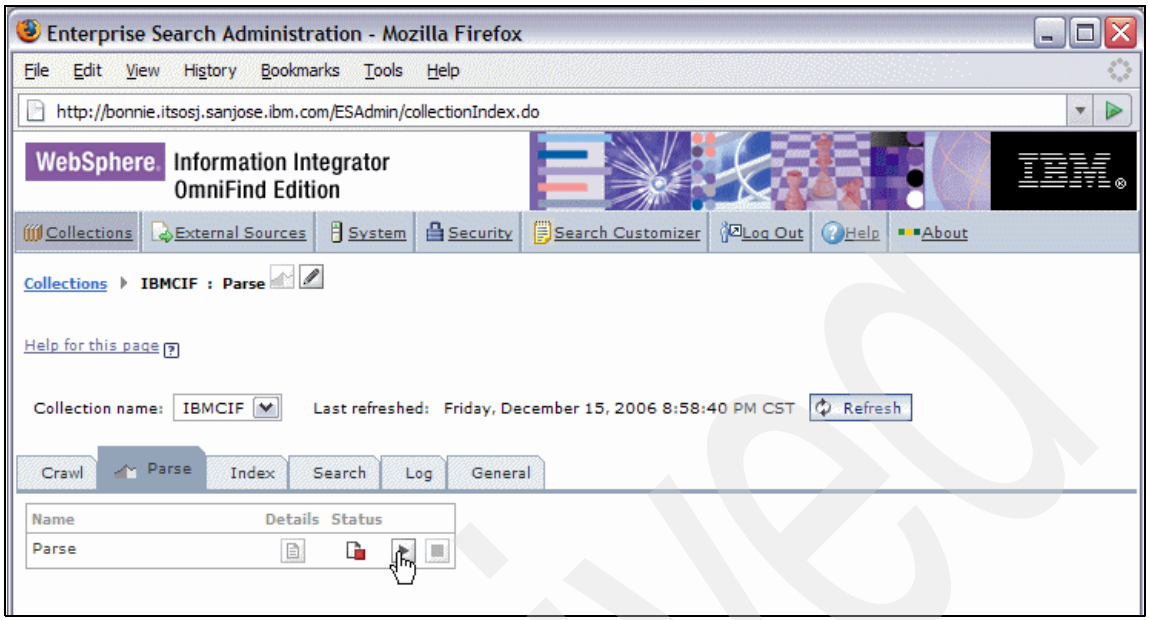

Figure 4-83 Start parser

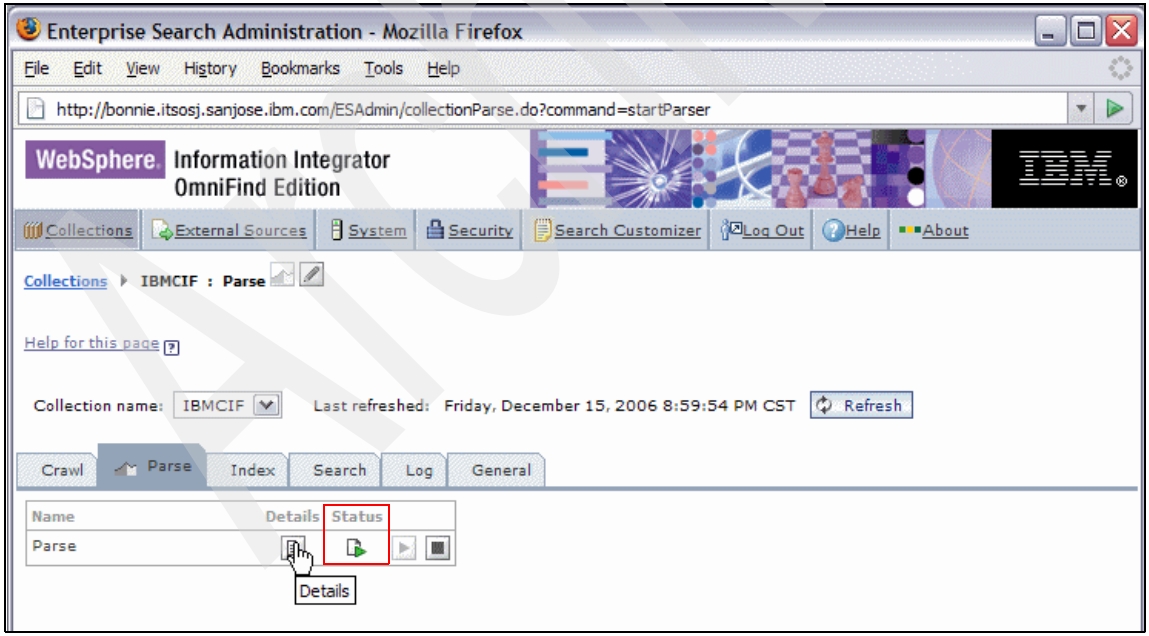

Figure 4-84 Click Details icon

Figure 4-85 Parser execution completion status

ASTEP4f: Build the main index

From the Index tab in Monitor mode, start the main index build by clicking the start icon, as shown in Figure 4-86 on page 377. Periodically click the **Refresh** button until the index build completes processing, as shown in Figure 4-87 on page 378. Review the index statistics.

We can now proceed to configure the security settings for the IBMCIF collection, as described in “ASTEP4g: Define security settings” on page 378.

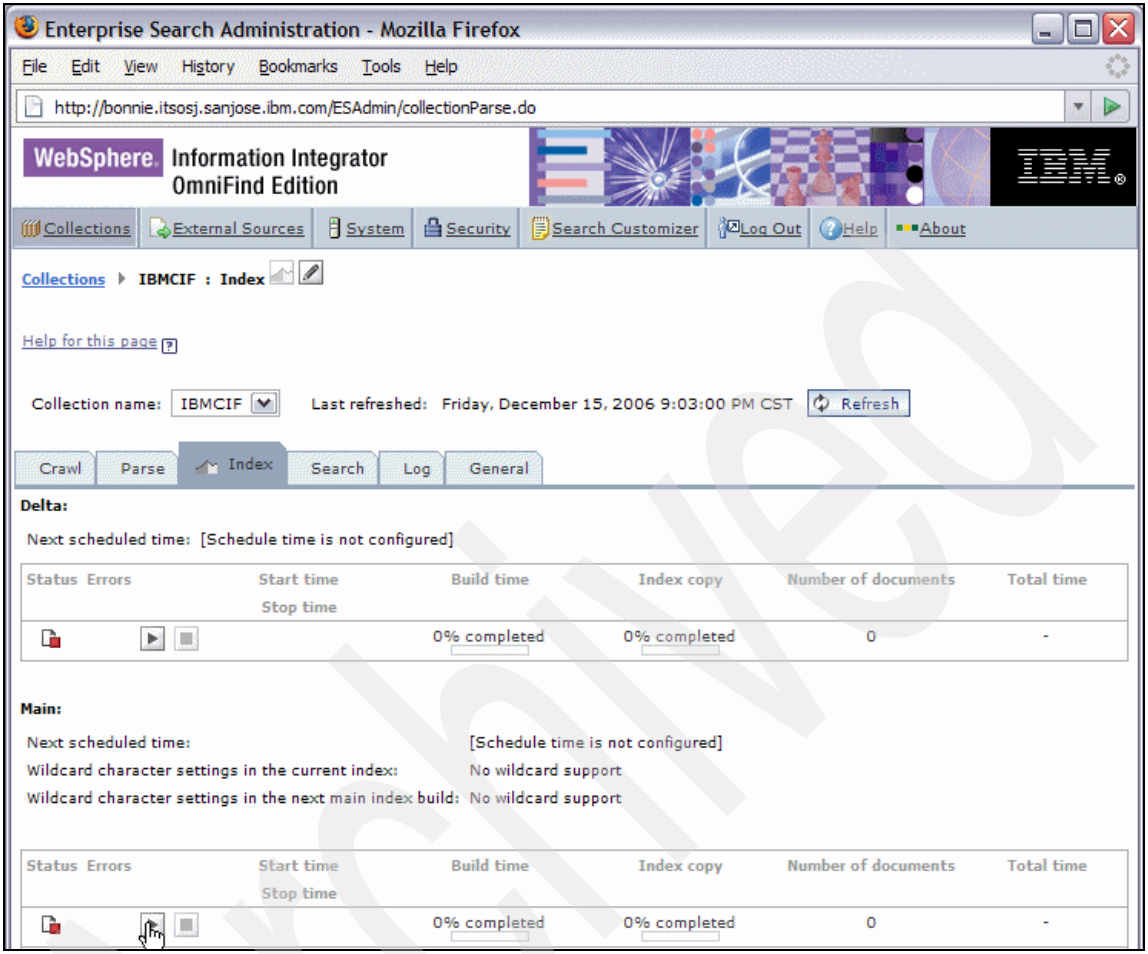

Figure 4-86 Start main index build

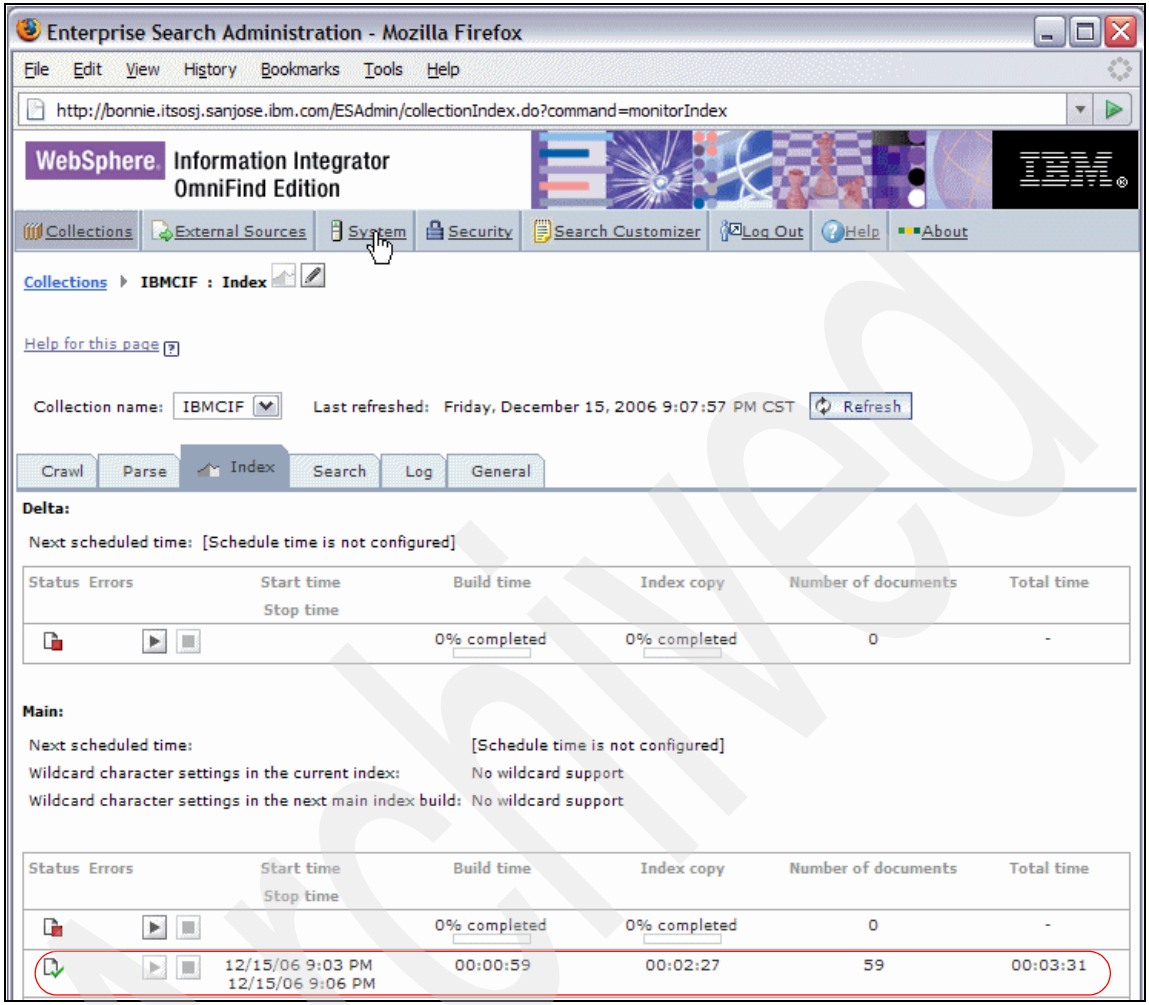

Figure 4-87 Main index build completion status

ASTEP4g: Define security settings

In this step, we configure the search application to access the IBMCIF collection, as described in Figure 4-88 on page 379 through Figure 4-92 on page 381. Since the IBMCIF collection is only accessible by authorized employees, the Default search application is restricted from accessing it (we chose to delete it), and a new search application IBMCIF is defined that has access to only the IBMCIF collection.

Note: We stayed with the defaults for identity management component (enabled) and single sign-on (enabled for all data source types) for the IBMCIF collection, and therefore do not show the relevant screen captures here. Refer to Figure 2-61 on page 120 through Figure 2-63 on page 121 for the screen captures associated with defining IMC and single sign-on.

Unless specific action is taken, the Default search application name automatically has access to the IBMCIF collection. This is desirable at least until one has tested the IBMCIF collection using the sample search Web application or portlet. After successful testing, you can disable access to the IBMCIF collection by the Default search application name (we chose to delete it, as shown in Figure 4-91 on page 381). A new search application IBMCIF is given access to the IBMCIF collection (Figure 4-89 on page 380 through Figure 4-90 on page 380).

Figure 4-92 on page 381 shows the single search application IBMCIF defined in the system.

We can now proceed to query the IBMCIF collection, as described in 4.3.5, “ASTEP5: Query IBMCIF collection” on page 382.

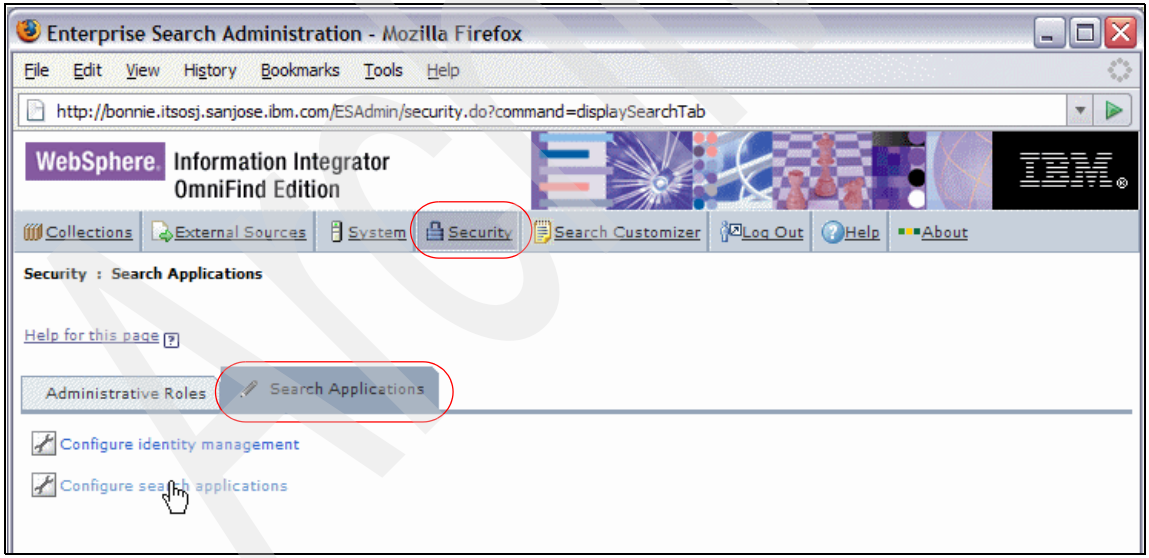

Figure 4-88 Configure search applications

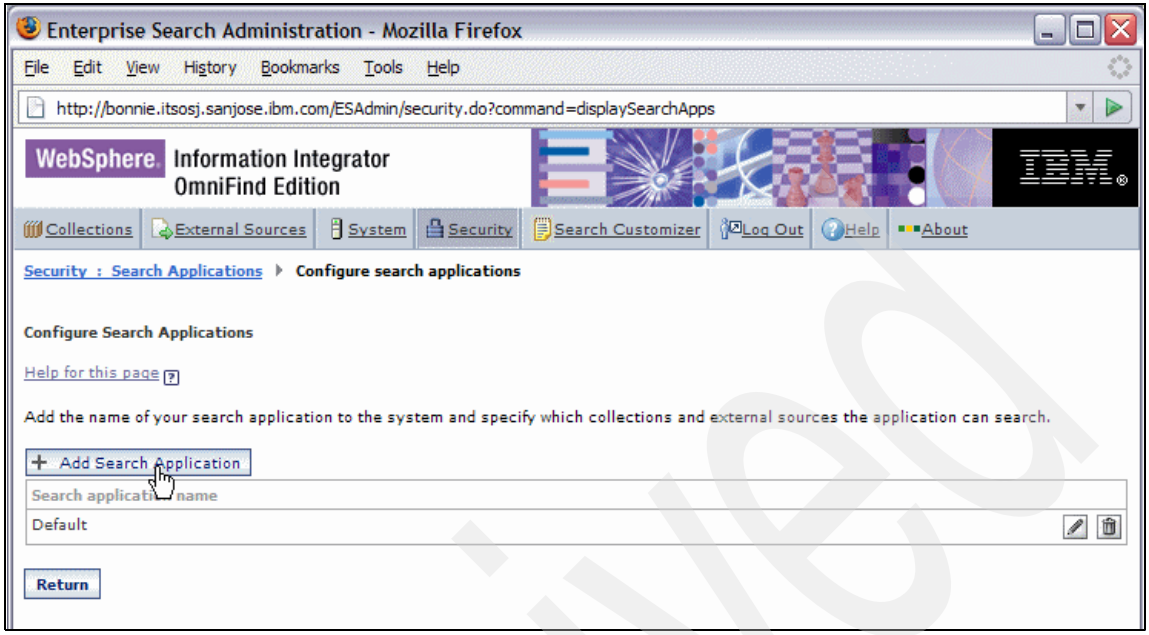

Figure 4-89 Add Search Application

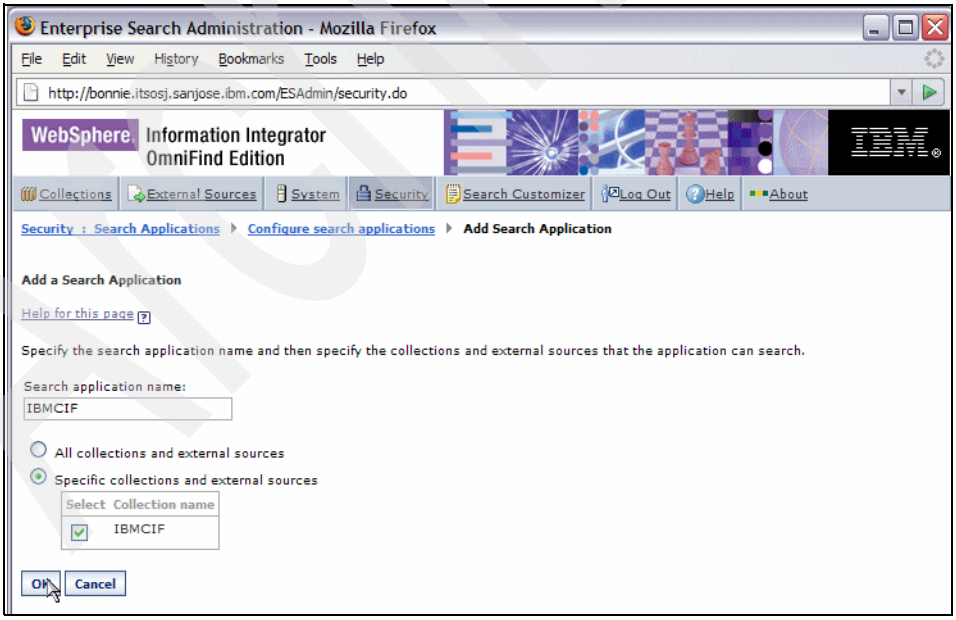

Figure 4-90 Add a Search Application IBMCIF with access only to the IBMCIF collection

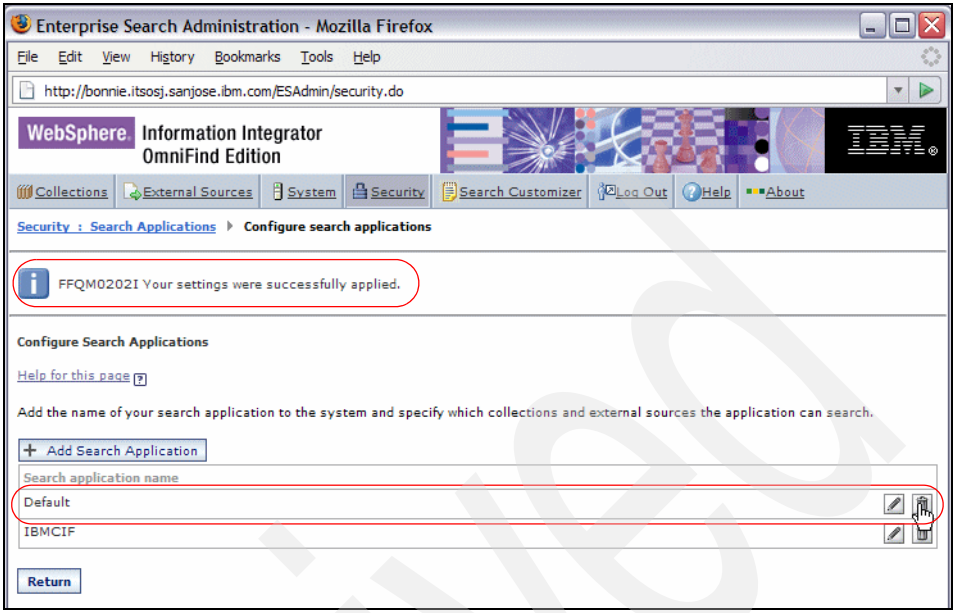

Figure 4-91 Delete the Default search application

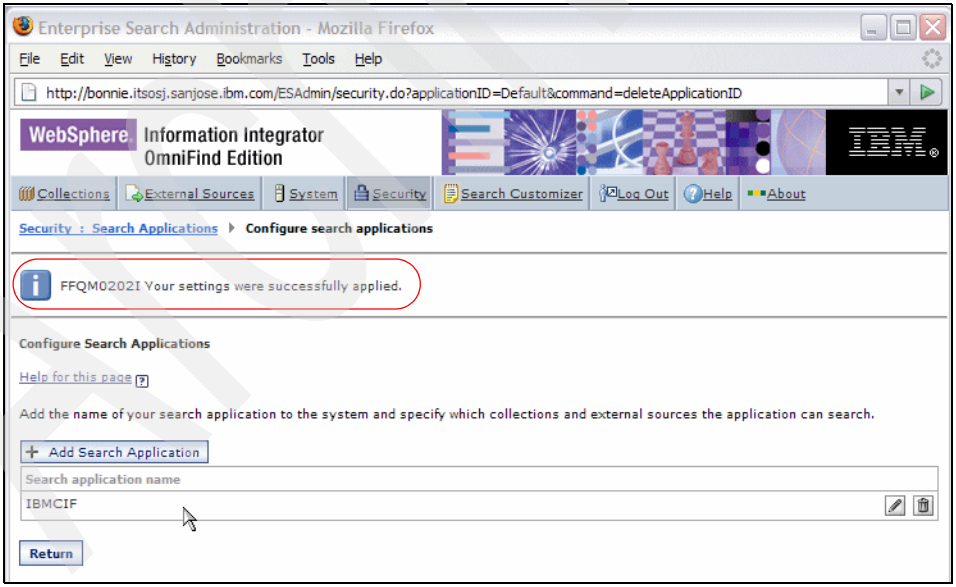

Figure 4-92 List of search applications in the system

4.3.5 ASTEP5: Query IBMCIF collection

In this step, we query the IBMCIF collection using the (unmodified) sample search Web application, the (unmodified) Search Application Customizer, and a modified sample search portlet application specifying the IBMCIF search application ID.

This section describes the following:

- Why there is a need for a modified sample search application/portlet
- Using the (unmodified) sample search Web application
- Using the modified sample search portlet

Why there is a need for a modified sample search application/portlet

The reasons for requiring a search application/portlet other than the sample search application/portlet provided are as follows:

- The IBMCIF collection has three data sources (Notes Domino, DB2 Content Manager, and DB2) with document-level security, IMC (DB2 Content Manager), and single sign-on (Notes Domino) enabled. Single sign-on and IMC are not supported for a DB2 crawler.
- Individual documents in DB2 need to be accessible only to authorized employees. This requires:
  - Each document crawled in DB2 to be associated with one or more specific security tokens during the crawl.

    Typically, these security tokens would be strings that represent user groups, rather than individual user IDs.

    For certain data sources, such as Notes Domino and DB2 Content Manager (but not DB2), OmniFind can capture the access control lists during the crawl. For DB2, however, it is up to the OmniFind administrator to capture this information, most probably with a Crawler plug-in.

    Note: For the DB2 data source in the IBMCIF collection, we chose to extract the security token from the MAJPROJ field in the ADMINISTRATOR.PROJECT table rather than write a crawler plug-in, as shown in Figure 4-46 on page 346.

  - The search application to present one or more security tokens to the search runtime in the USC string for it to be matched with tokens stored in the enterprise search index.

For certain data sources, such as Notes Domino and DB2 Content Manager (but not DB2), IMC can extract the groups and build the USC string for you. For DB2, however, it is up to the search application developer to write their own code to extract the appropriate groups for the user and build the USC string with that information; most probably by looking up a user registry that holds information about users and associated groups for that data source.

Note: For the DB2 data source in the IBMCIF collection, we defined the groups associated with the user in the LDAP repository (Tivoli Directory Server), and then had the search application look up the LDAP and build the USC string.

- The sample search Web application and sample search portlet do *not* provide a mechanism to build the USC string for data sources with no single sign-on or IMC support.

Attention: We did *not* modify the sample search application to address these issues due to time constraints.

We did modify the sample search portlet with user group lookups for DB2 to enable us to access the IBMCIF collection in a completely secure manner. A brief description of the implementation in the sample search portlet follows.

DB2 data source security token implementation

As mentioned earlier, to implement the DB2 data source security requirements, we need to capture security tokens during the crawl, and have the search application build the USC string with the security tokens.

- Capture security tokens during crawl

  As mentioned earlier, we configured the DB2 crawler to extract the security token from the MAJPROJ field in the ADMINISTRATOR.PROJECT table in the Index access controls based on field values field, as shown in Figure 4-46 on page 346. Other crawled tables did not have the security tokens extracted.

- Build USC string by search application

  The USC string structure is shown in Figure 4-93 on page 384 with the three broad elements as follows:

  – Security tokens

    The tokens stored here are common to all the data sources. These tokens are matched against all the entries in the enterprise search index regardless of the data sources from which they originated.

– SSOtoken

  This only applies to data sources that support single sign-on.

– Identity

  A multiple of these (one per domain). The “type” attribute specifies the domain, such as Notes and Windows. There can be multiple groups associated with the user name associated with each domain. This is the section of the USC populated by IMC.

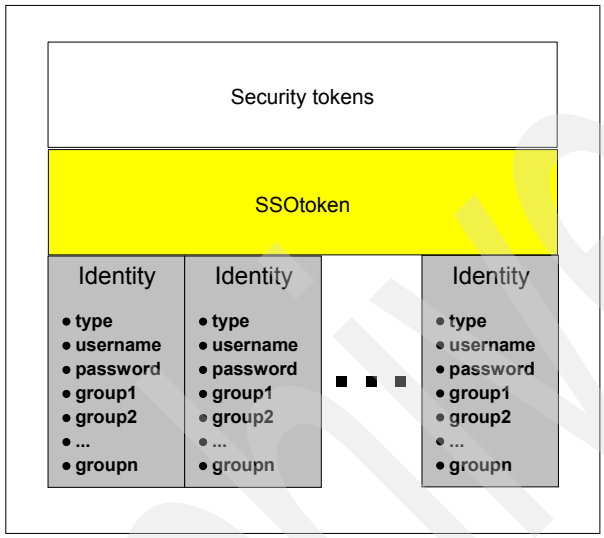

Figure 4-93 USC string structure

The sample search portlet must populate the “Security tokens” portion of the USC with the DB2 groups for the sample search portlet user. The groups from which DB2 tokens are derived, such as token-personal and token-automotive, are stored in the Tivoli Directory Server, as shown in Figure 4-94. Users added to those groups will have a token corresponding to the group name.

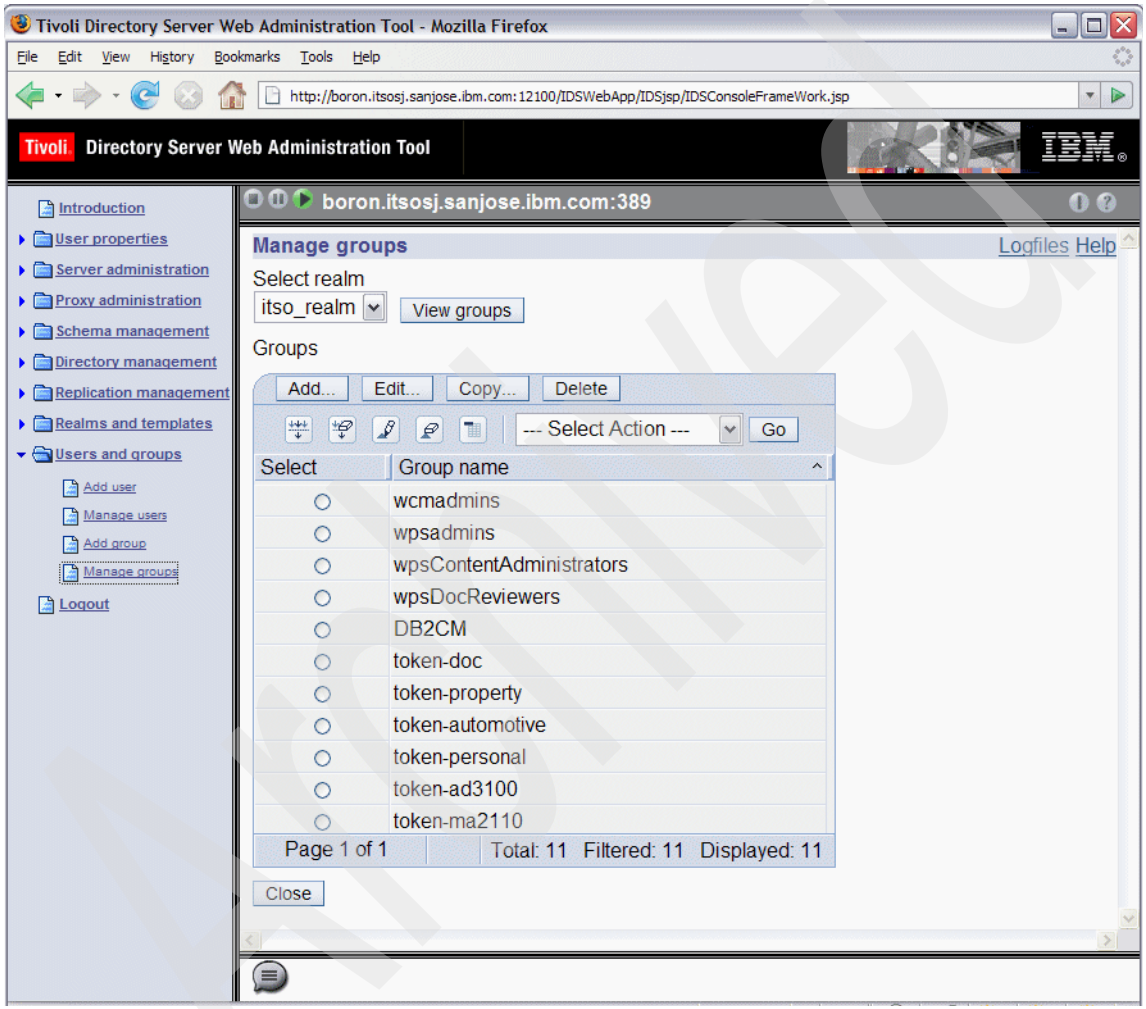

Figure 4-94 DB2 user groups in Tivoli Directory Server

The sample search portlet was modified to retrieve the logged in user’s associated DB2 groups from the Tivoli Directory Server (TDS) and append them to the Security tokens section of the USC. Specifically, the IdentityManagementHelper class of the ESSearchPortlet must be modified to obtain the DB2 groups from the TDS (done by WebSphere Portal) and

populate the USC with the security tokens. The (unmodified) sample search portlet provides the full DN of groups as security tokens, which cannot be used to match the security tokens stored in the document during the crawl operation. The modified code highlighted in Example 4-1 on page 387 calls the Portal User Management API (PUMA), which in turn retrieves the list of all DB2 groups from the TDS, extracts the relevant portion of it as a security token, and then appends them to the Security tokens section of the USC. For example, with a group with DN of "cn=token-test,cn=groups,ou=itso,o=ibm", the modified code extracts the token called "TEST". The initialization of the USC string is performed in the IdentityManagementHelper class by method createPortalIdentity. Example 4-1 on page 387 shows the entire createPortalIdentity method in the IdentityManagementHelper class with the highlighted modified code.The modified code filters the list of current user groups that the PUMA call returns to only those starting with “token-”. It then extracts the string following “token-” and folds it to uppercase. This then becomes the security token, which it appends to the “Security tokens” portion of the USC.

Note: By using PUMA in the IdentityManagementHelper class, we leveraged WebSphere Portal’s LDAP access without having to directly access the LDAP and provide configuration and credentials to do so.

The IdentityManagementHelper class is used in both the sample search application as well as the sample search portlet. However, when it is called from the sample search portlet, it verifies SSO groups in the portal, which is where the code was modified.

This implicit LDAP access and SSO group verification does not occur in the sample search application. Therefore, you have to replace PUMA calls with JNDI/LDAP calls that directly access the LDAP, and pass the necessary configuration and credential information. This is a non-trivial task, especially when nested groups need to be supported.

Example 4-1 Modified code to populate Security tokens portion of the USC string from TDS groups

```
.............
.............
/**
 * The IdentityManagementHelper class provides all the functions required to support
 * the OmniFind Identity Management components to the Struts IdentityManagementAction.
 */
...............
...........
    private void createPortalIdentity(
        HttpServletRequest request,
        SecuredDataSource datasource)
        throws Exception {
        if (PortletApiUtils.getUtilsInstance() != null) {
            if (logger.isLoggable(Level.INFO)) {
                logger.info("createPortalIdentity - entered");
            }
            Identity identity = new Identity();
            identity.setDomain(datasource.getId());
            identity.setType(datasource.getType());
            identity.setProperties(datasource.getProperties());
            identity.setProperty(IDENTITY_ENABLED, "true");

            RunData runData = RunData.from((PortletRequest) request);
            User portalUser = runData.getUser();
            if (logger.isLoggable(Level.INFO)) {
                logger.info("createPortalIdentity - portal user: " + portalUser.getId());
            }
            identity.setUsername(portalUser.getId());

            List portalGroups = portalUser.getGroups();
            String[] groups = new String[portalGroups.size() + 2];
            // let's add the portal username as well
            groups[0] = portalUser.getId();
            // default group since we know this user has been authenticated
            groups[1] = "all authenticated portal users";
            for (int i = 0; i < portalGroups.size(); i++) {
                Group group = (Group) portalGroups.get(i);
                groups[i + 2] = group.getId();

                if (logger.isLoggable(Level.INFO)) {
                    logger.info("createPortalIdentity - group: " + groups[i]);
                }
            }

            // add the generated group information to the identity
            identity.setGroups(groups);
            // add the Identity to the Hashtable
            String key = datasource.getId() + "_" + datasource.getType();
            this.identities.put(key, identity);

            // NEW CODE START

            Vector tokens = new Vector();
            // Pattern probably should be configurable in portlet properties.
            Pattern p = Pattern.compile("cn=token-([^,]+),.*");
            // now let's add them as native tokens, if they match
            for(Iterator i = portalGroups.iterator(); i.hasNext();) {
                Group g = (Group) i.next();
                String groupdn = g.getId();
                Matcher m = p.matcher(groupdn);
                if(m.matches()) {
                    String token = m.group(1).toUpperCase();
                    tokens.add(token);
                    if (logger.isLoggable(Level.INFO)) {
                        logger.info("createPortalIdentity - token: " + token);
```

```
                }
            }
        }

        if(tokens.size() > 0) {
            this.context.setNativeTokens((String[])tokens.toArray(new String[tokens.size()]));
        }

        // NEW CODE END

        if (logger.isLoggable(Level.INFO)) {
            logger.info("createPortalIdentity - returning");
        }
    }
}

.............
.........
```

Using the (unmodified) sample search Web application

Since the IBMCIF collection can only be accessed by the IBMCIF search application, we need to modify the config.properties file applicationName property to IBMCIF, as shown in Example 4-2.

Example 4-2 config.properties file contents

```
# Search Application
applicationName=IBMCIF

.........
........
# SSO related values
# The ssoCookieName corresponds to the name of the HTTP header
# that contains the Single Sign On cookie value.
ssoCookieName=LtpaToken
.........
.......
```

Note: Since the sample search application was not modified to support the DB2 security tokens, the search results from the sample search application will *not* include documents that have security tokens associated with them.

The objective of this section is to show the categorization functions implemented in this scenario, with the full knowledge that certain DB2 documents with security tokens *may* not appear in the search results.

Figure 4-95 on page 390 through Figure 4-103 on page 397 describe the user interactions searching the IBMCIF collection.

Log in to the modified sample search Web application with the esadmin user ID (Figure 4-95 on page 390). Since IMC and SSO are enabled, the search runtime prompts the user for the DB2 Content Manager icmnlsdb domain credentials, as shown in Figure 4-96 on page 390. Provide the credentials and click **Apply**, which takes you to the search window (Figure 4-97 on page 391). Since single sign-on support for Notes Domino is enabled, no prompts are presented for it, as the search runtime has the required credentials in the LTPA token.

Note: Since DB2 is not supported by IMC and single sign-on, there are no prompts for it either.

Click the **Preferences** link in Figure 4-97 on page 391 to view and modify the default preferences as required. Figure 4-98 on page 392 shows a number of options that can be modified for the search session, including inclusion of quick links, the data sources to be searched, file types to be excluded/included, number of results per page, and whether synonyms should be searched for. Also by default, if multiple collections exist, the (modified) sample search application automatically searches across all the collections using a “remote” federator. In this case, only a single collection IBMCIF has been defined, and therefore only the IBMCIF collection appears in the list of choices in the Preferences window along with its corresponding data sources Notes, DB2 Content Manager, and DB2. Click **Apply** to save the choices made.

Figure 4-99 on page 393 shows the search results (56 documents) for the string “ibm”, which has results from all the three data sources with the possible exclusion of DB2 documents that have security tokens associated with it. Click the **Category Tree** tab in Figure 4-99 on page 393 to view the category tree associated with the IBMCIF collection, as shown in Figure 4-100 on page 394. Expand the **Products** category and select the **Software** subcategory in the navigation pane, and click **Search** for “ibm”, which shows the results in Figure 4-101 on page 395.

Figure 4-102 on page 396 shows the search results for the string “ibm” in the Storage subcategory.

Figure 4-103 on page 397 shows the results of clicking **Show Details** in the search results, which also highlights the categories to which a particular document in the search results belongs.

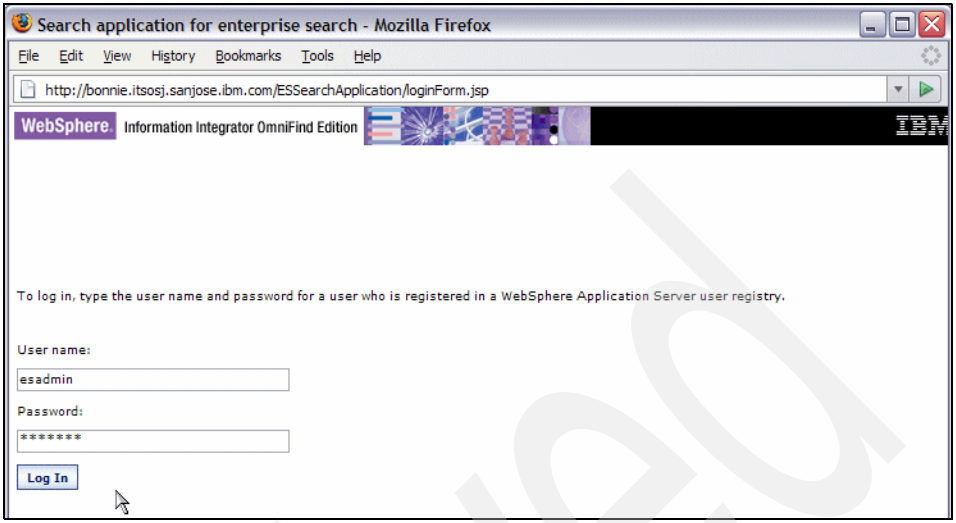

Figure 4-95 Log in to the modified sample search application

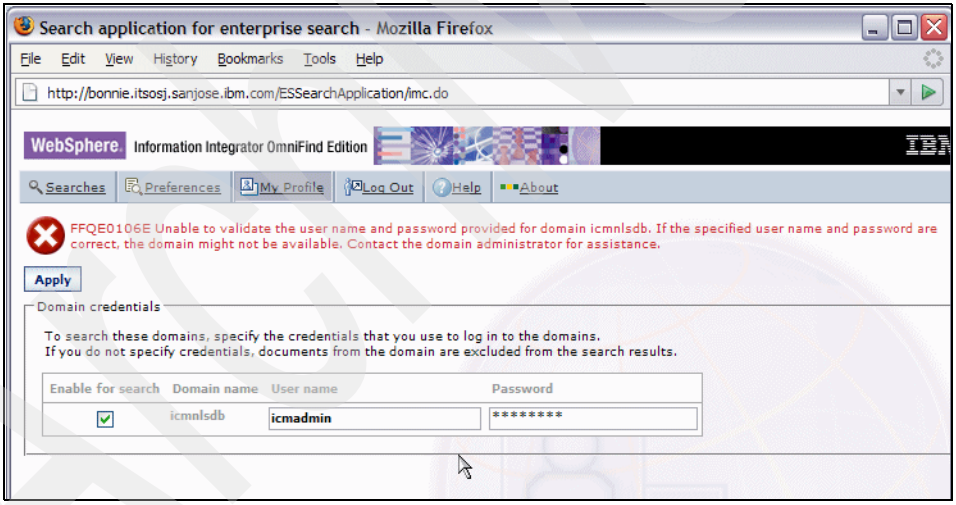

Figure 4-96 Prompt for IMC credentials: DB2 Content Manager icmnlsdb domain

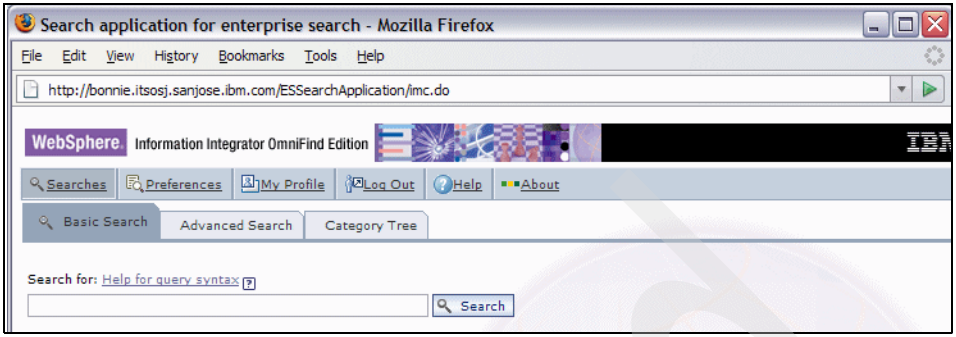

Figure 4-97 Search box

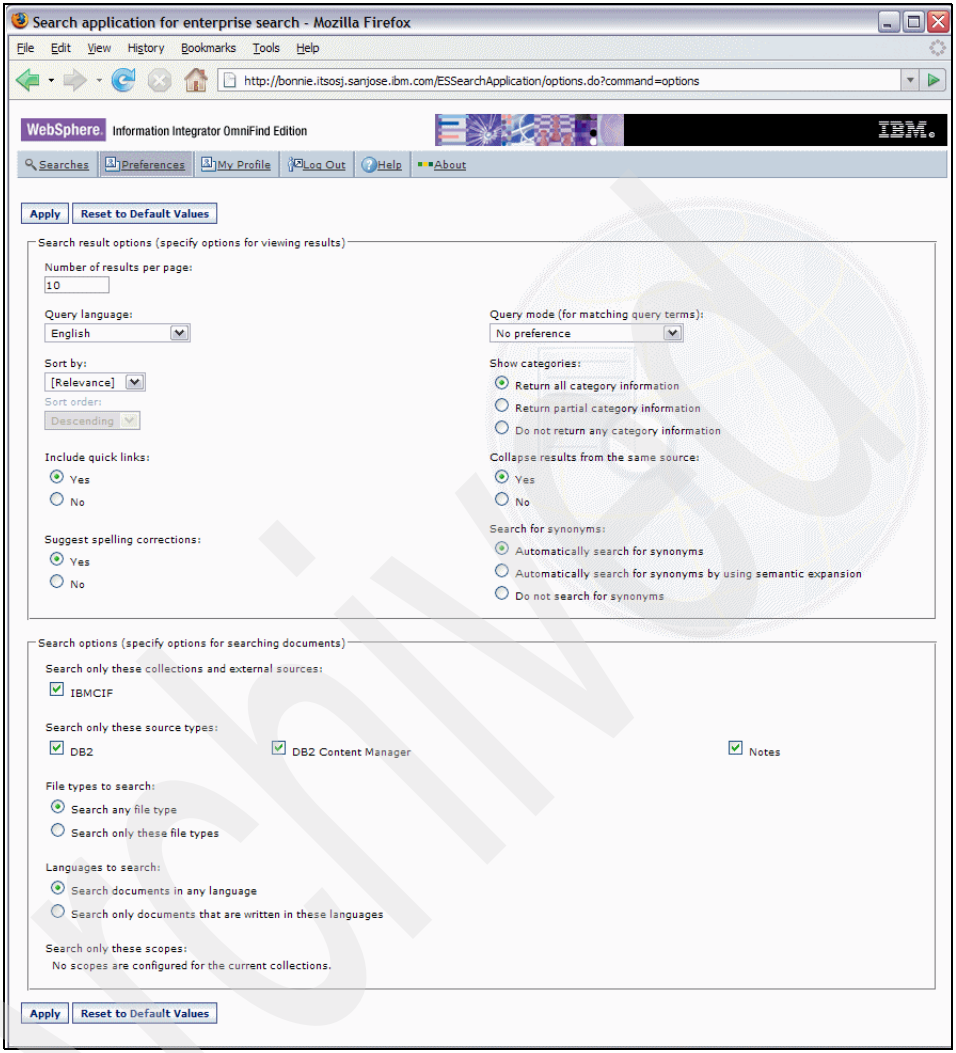

Figure 4-98 Preferences options

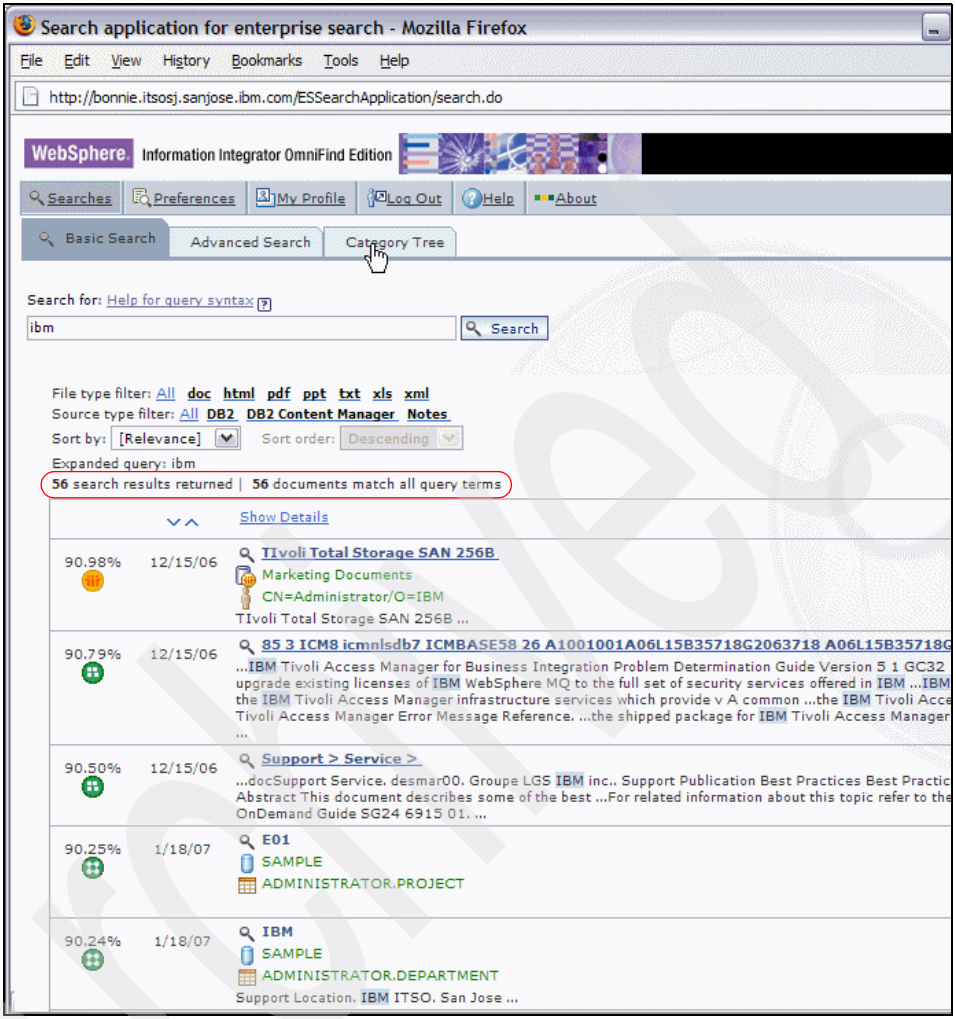

Figure 4-99 Search results for “ibm”

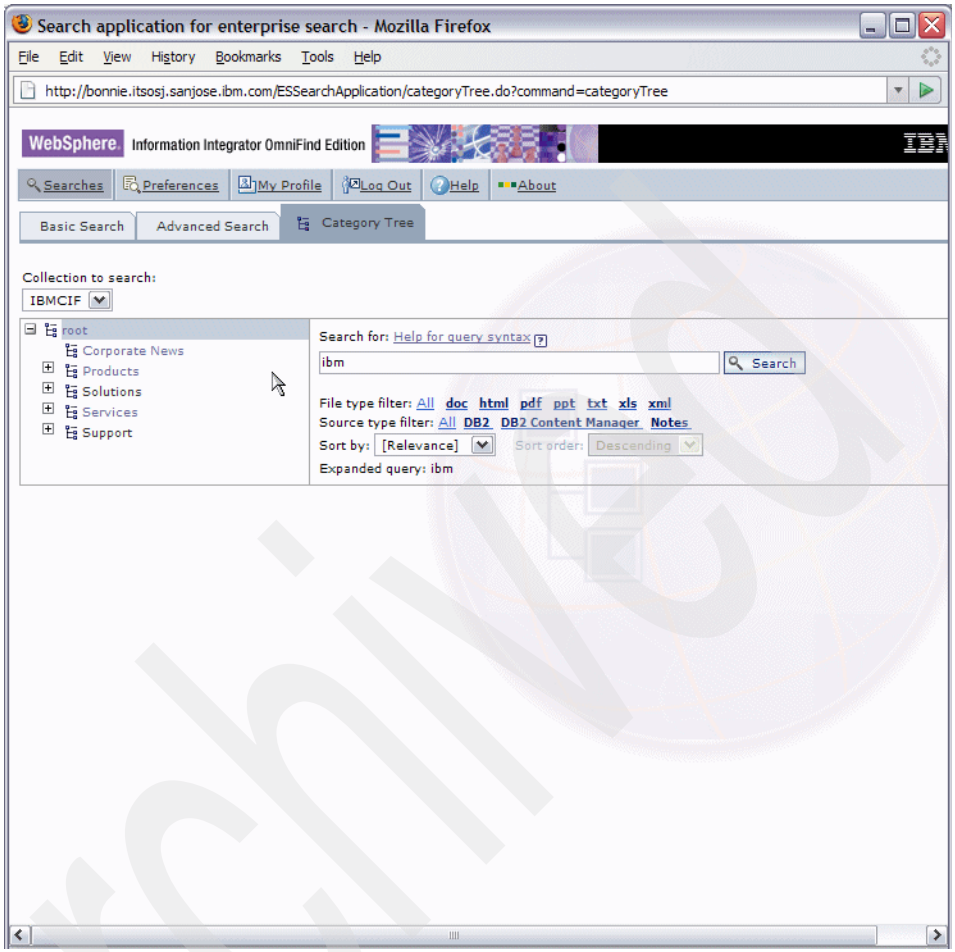

Figure 4-100 Category tree hierarchy under root

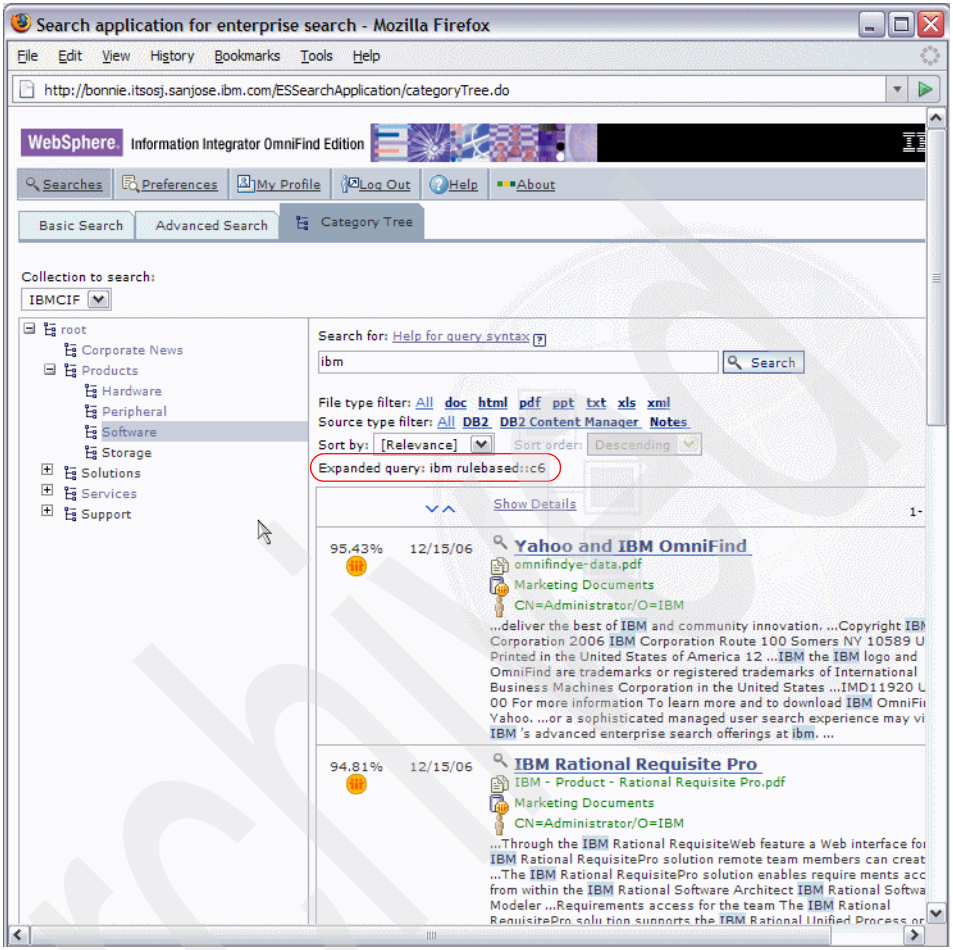

Figure 4-101 Search results for "ibm" in the Software category

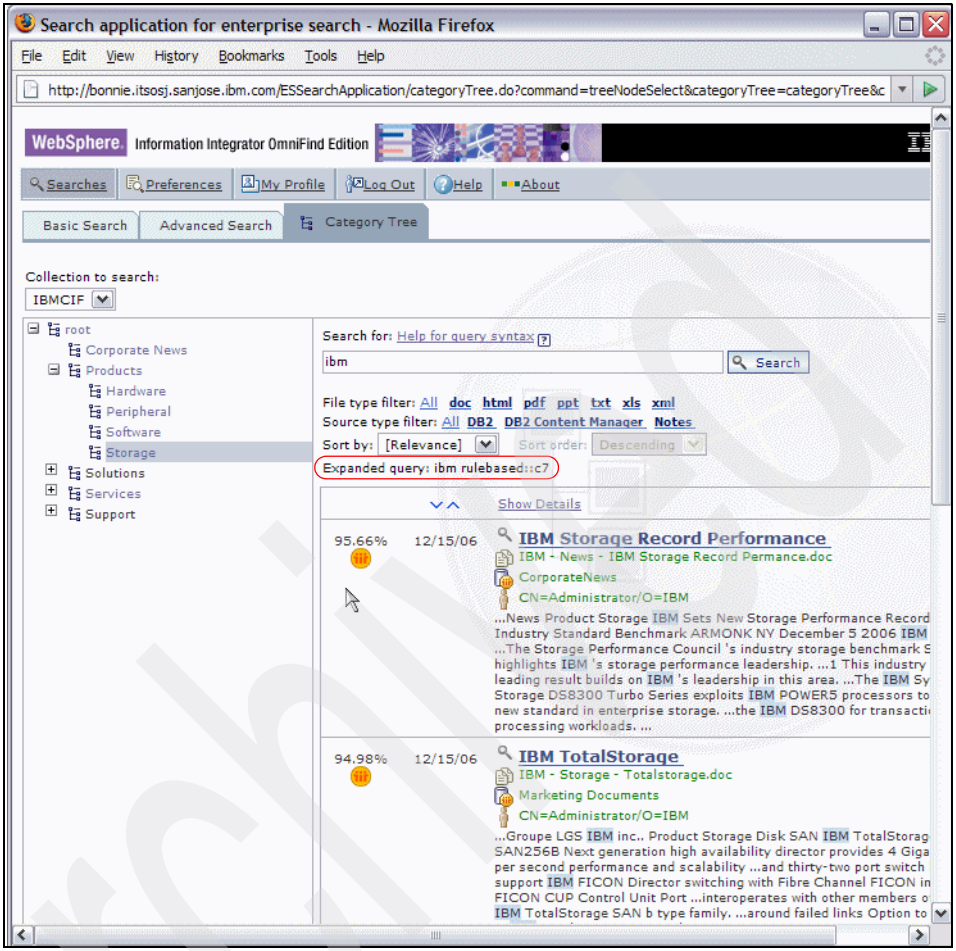

Figure 4-102 Search results for “ibm” in the Storage category

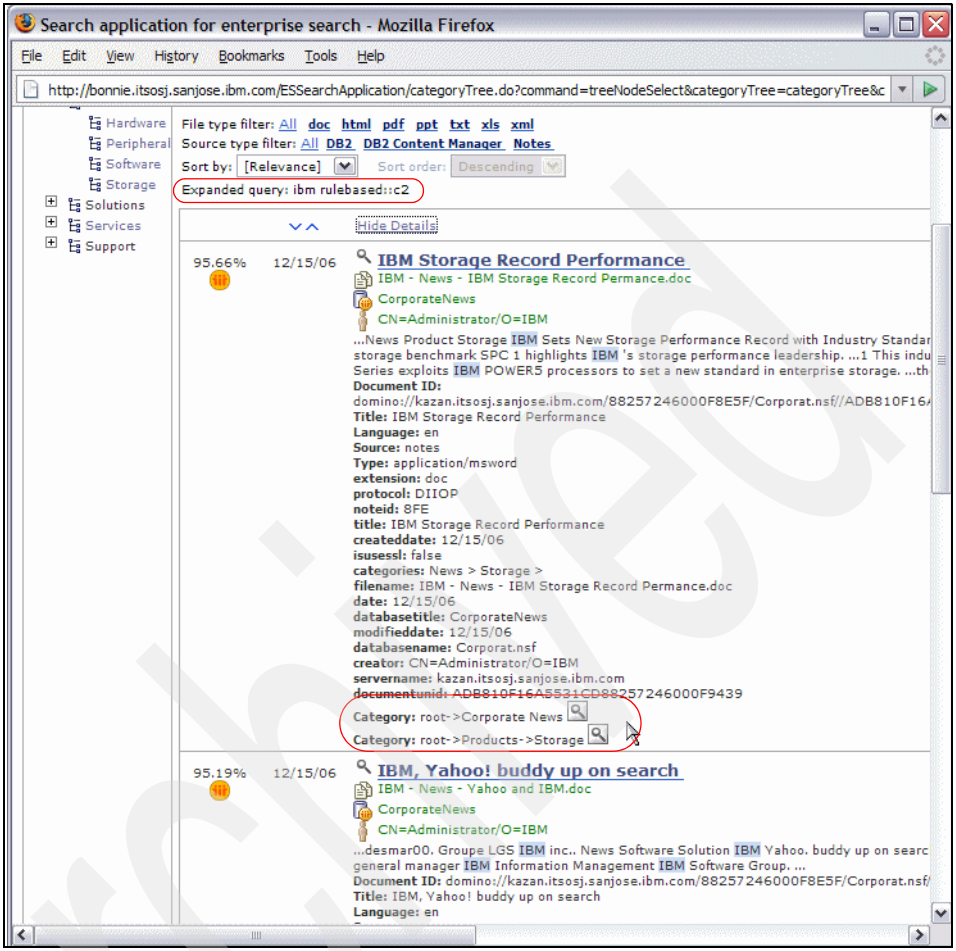

Figure 4-103 Detailed search results for “ibm” in the Corporate News and Storage categories

Using Search Application Customizer

In this step, we describe the use of the Search Application Customizer to search the IBMCIF collection.

Note: Since the sample search application was not modified to support the DB2 security tokens, the Search Application Customizer search results will *not* include documents that have security tokens associated with them.

The objective of this section is to show the Search Application Customizer's ability to modify the configuration properties during testing, with the full knowledge that certain DB2 documents with security tokens *may* not appear in the search results.

Figure 4-104 on page 399 through Figure 4-106 on page 400 describe the interactions involved.

When the Search Application Customizer is invoked from a Web browser (`http://bonnie.itsosj.sanjose.ibm.com/ESSearchApplication/palette.do`), it displays Figure 4-104 on page 399 with an error message "You must select at least one collection or external source to search." This is because the Search Application Customizer is configured by default to use the Default search application, which does not have access to the IBMCIF collection.

Expand the **Server** settings in the navigation pane to confirm that this is the case, as shown in Figure 4-104 on page 399.

Modify the Search application name to IBMCIF, as shown in Figure 4-105 on page 399. Click **Apply** and re-issue the search request to view the search results, as shown in Figure 4-106 on page 400.

You must click **Save** to save the changes made to the configuration properties.

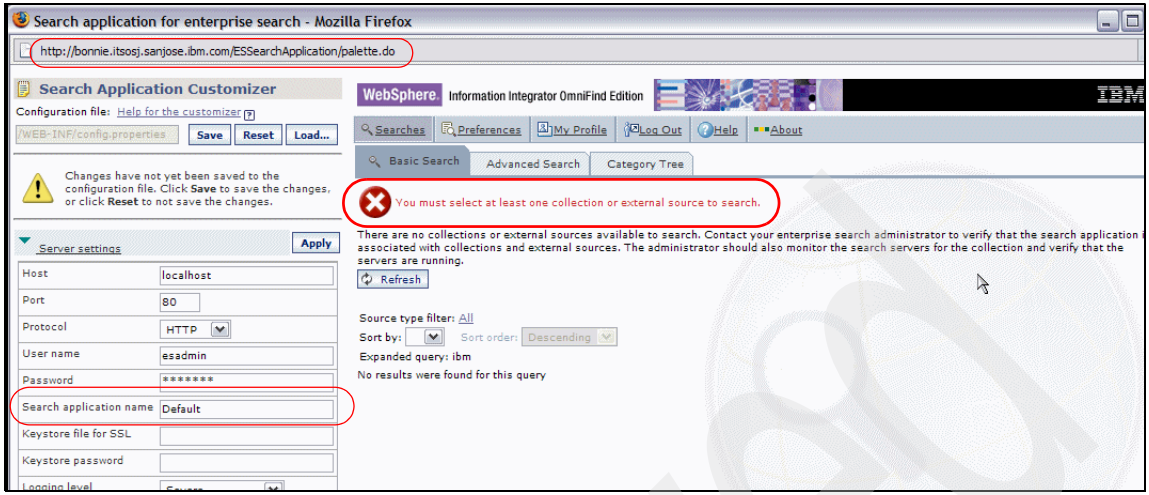

Figure 4-104 Search Application Customizer

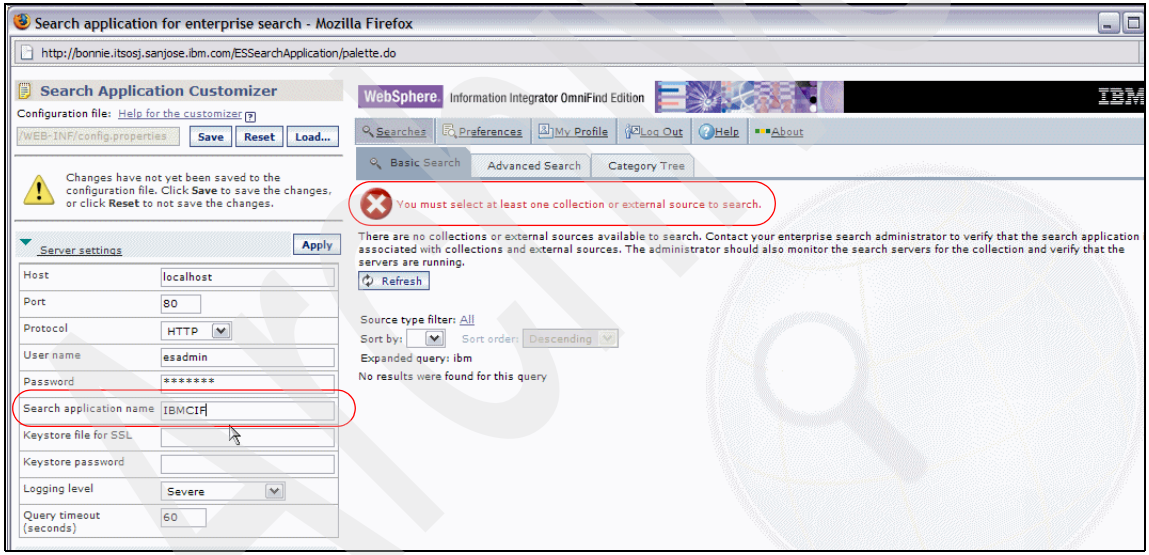

Figure 4-105 Modify Search application name to IBMCIF

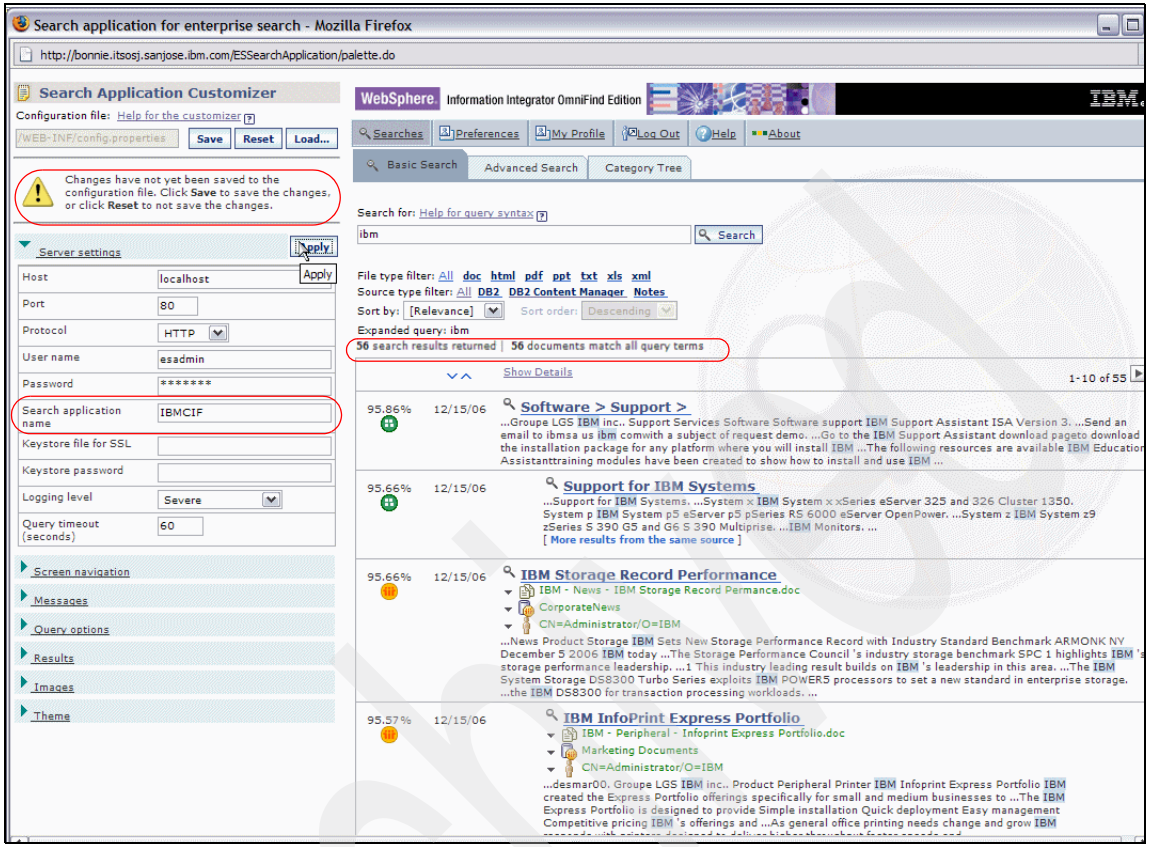

Figure 4-106 Search results for "ibm"

Using the modified sample search portlet

In this step, we used the modified sample search portlet to access the IBMCIF collection. Appendix A, "Install Sample Search application portlet" on page 431 describes the installation of the sample search portlet in WebSphere Portal Server.

Figure 4-107 on page 402 through Figure 4-114 on page 407 describe the user interactions searching the IBMCIF collection.

After logging in to the WebSphere Portal with the wpsadmin user ID (Figure 4-107 on page 402), invoke the modified sample search portlet by clicking the **OmniFind-AIX** tab, as shown in Figure 4-108 on page 402. Since IMC and SSO are enabled, the search runtime prompts the user for the DB2 Content Manager icmnlsdb domain credentials, as shown in Figure 4-109 on page 403. Provide the credentials and click **Apply**. Since single sign-on support

for Notes Domino is enabled, no prompts are presented for it, as the search runtime has the required credentials in the LTPA token.

Note: Since DB2 is not supported by IMC and single sign-on, there are no prompts for it either.

Click the **Preferences** link in Figure 4-110 on page 403 to view and modify the default preferences as required. As mentioned earlier, Figure 4-111 on page 404 shows a number of options that can be modified for the search session, including inclusion of quick links, the data sources to be searched, file types to be excluded/included, number of results per page, and whether to search for synonyms. Also by default, if multiple collections exist, the (modified) sample search application automatically searches across all the collections using a “remote” federator. In this case, only a single collection IBMCIF has been defined, and therefore only the IBMCIF collection appears in the list of choices in the Preferences window along with its corresponding data sources Notes, DB2 Content Manager, and DB2. Click **Apply** to save the choices made.

Figure 4-112 on page 405 shows the search results for the string “ibm”, which has results from all three data sources.

Click **Category Tree** link in Figure 4-112 on page 405 to view the category tree associated with the IBMCIF collection in Figure 4-113 on page 406. You can expand the appropriate categories in the navigation pane and click **Search** to view the search results for the chosen categories (this is not shown here).

Figure 4-114 on page 407 shows the results of clicking **Show Details** in the search results, which also highlights the categories to which a particular document in the search results belongs.

Note: At this point, we have the functional requirements met for the enterprise search solution for IBM. Further tests, measurements, and tuning need to be performed to ensure that the solution fully addresses the capacity and workload requirements as well.

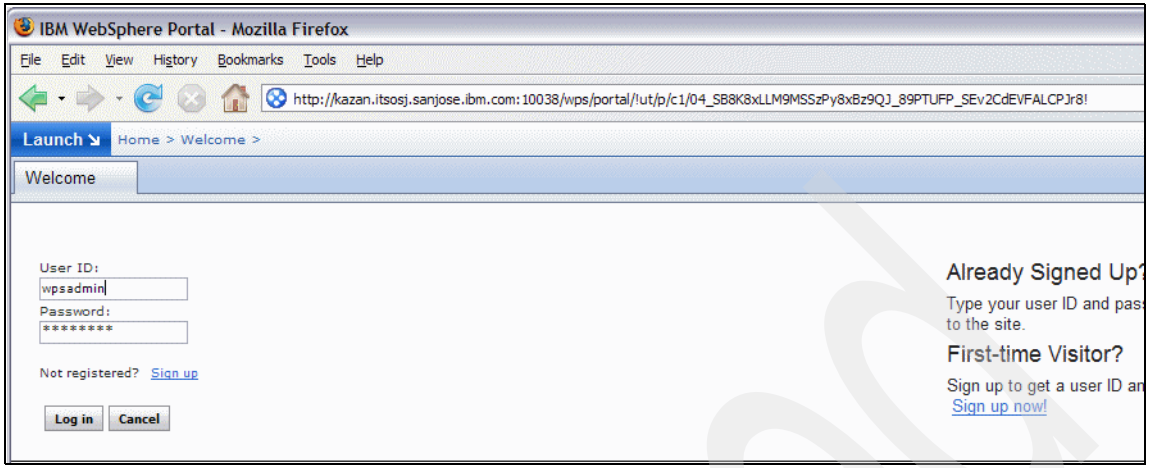

Figure 4-107 Log in to WebSphere Portal

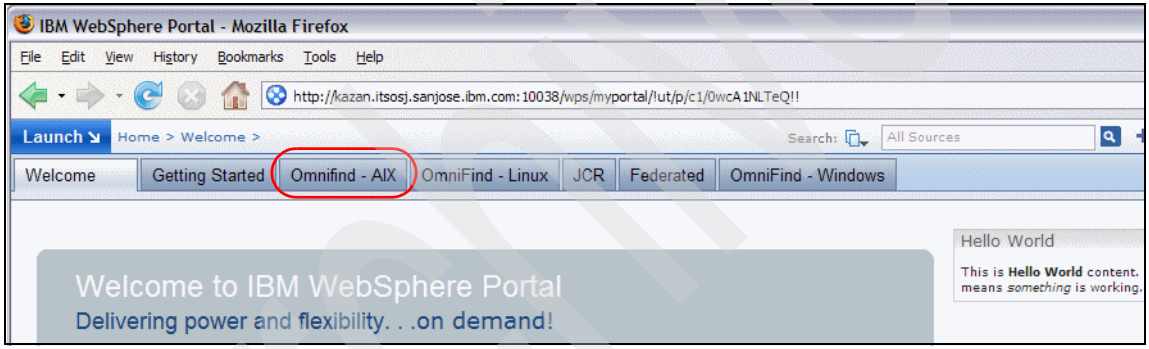

Figure 4-108 WebSphere Portal

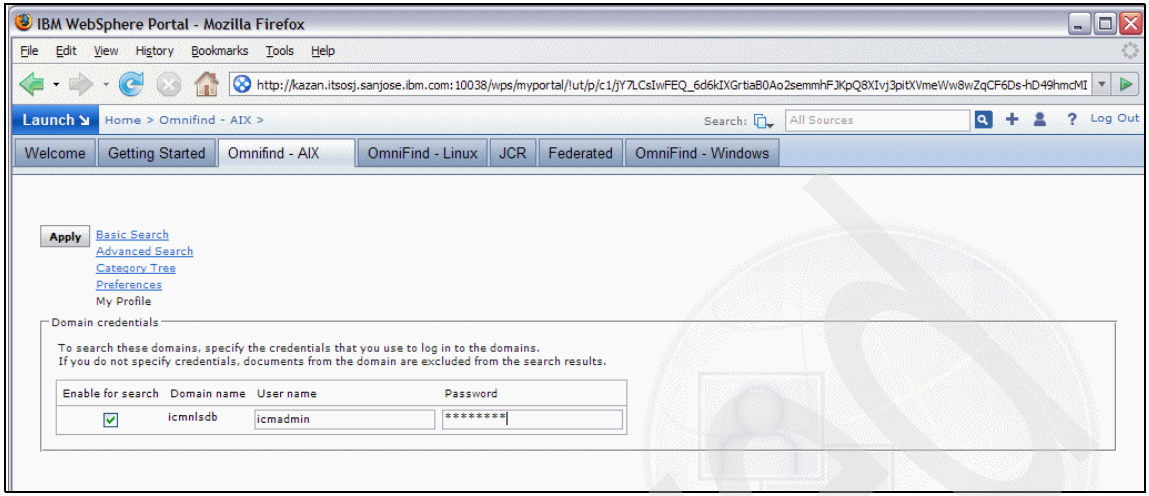

Figure 4-109 IMC credentials prompt for DB2 Content Manager icmnlsdb domain

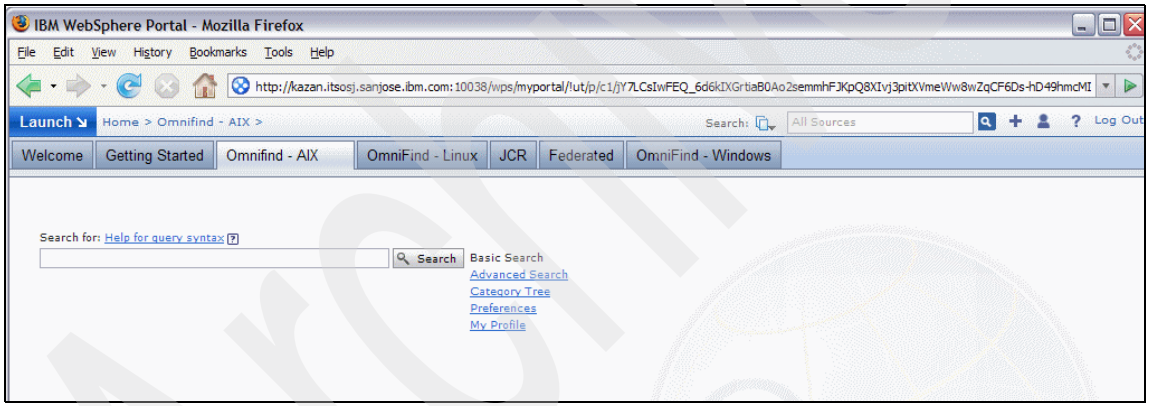

Figure 4-110 Search box

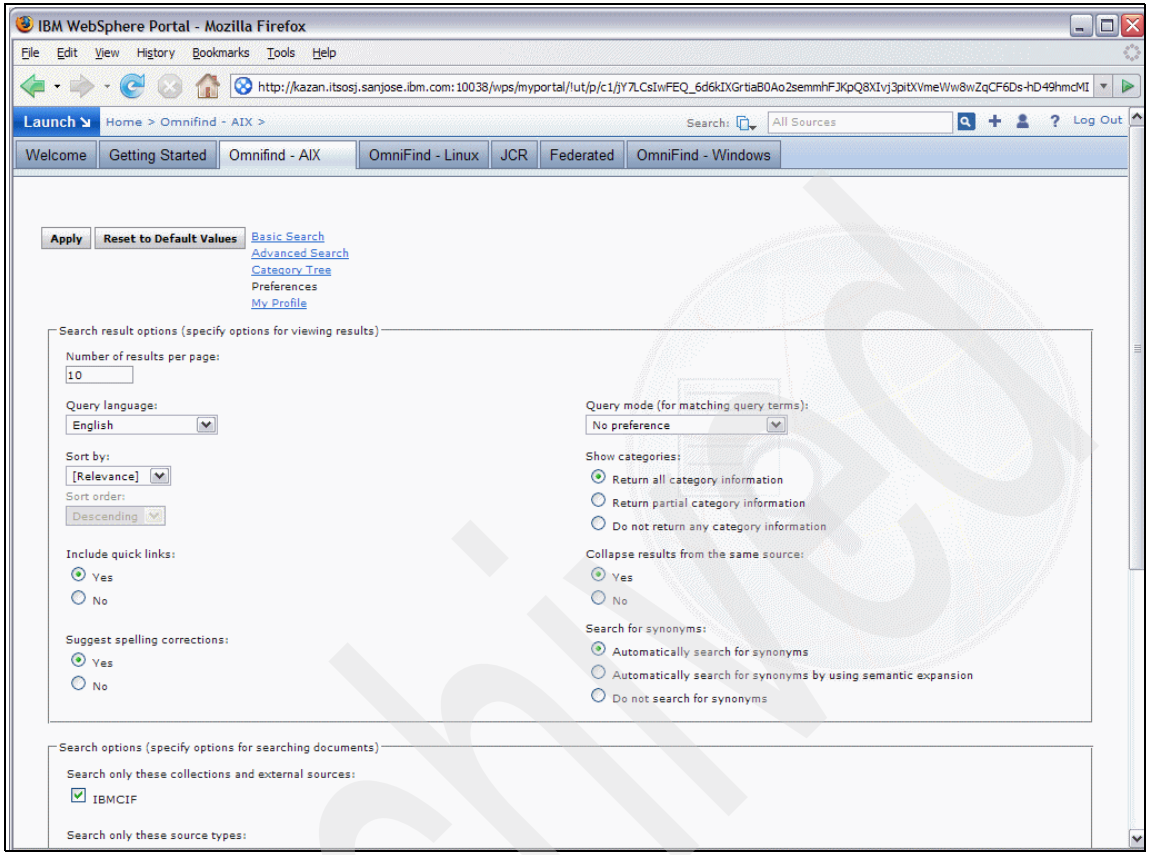

Figure 4-111 Preferences options

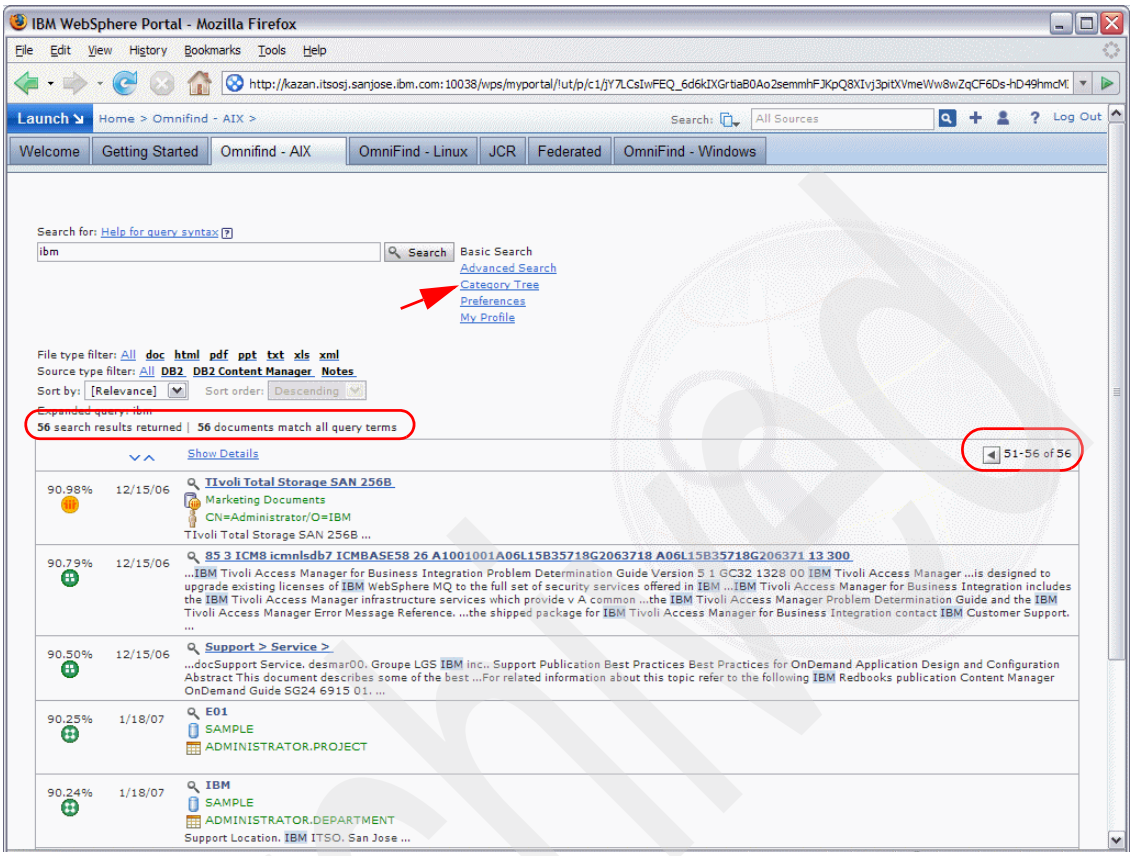

Figure 4-112 Search results for “ibm”

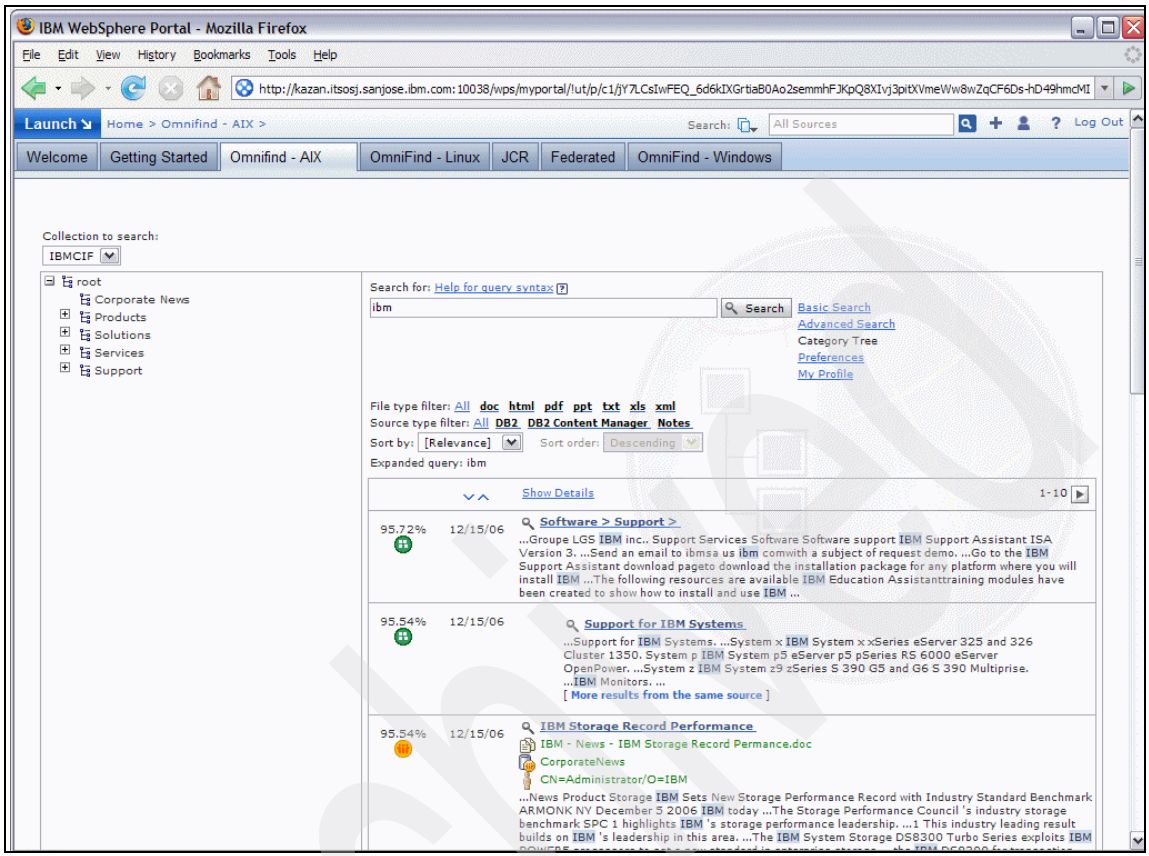

Figure 4-113 Category tree

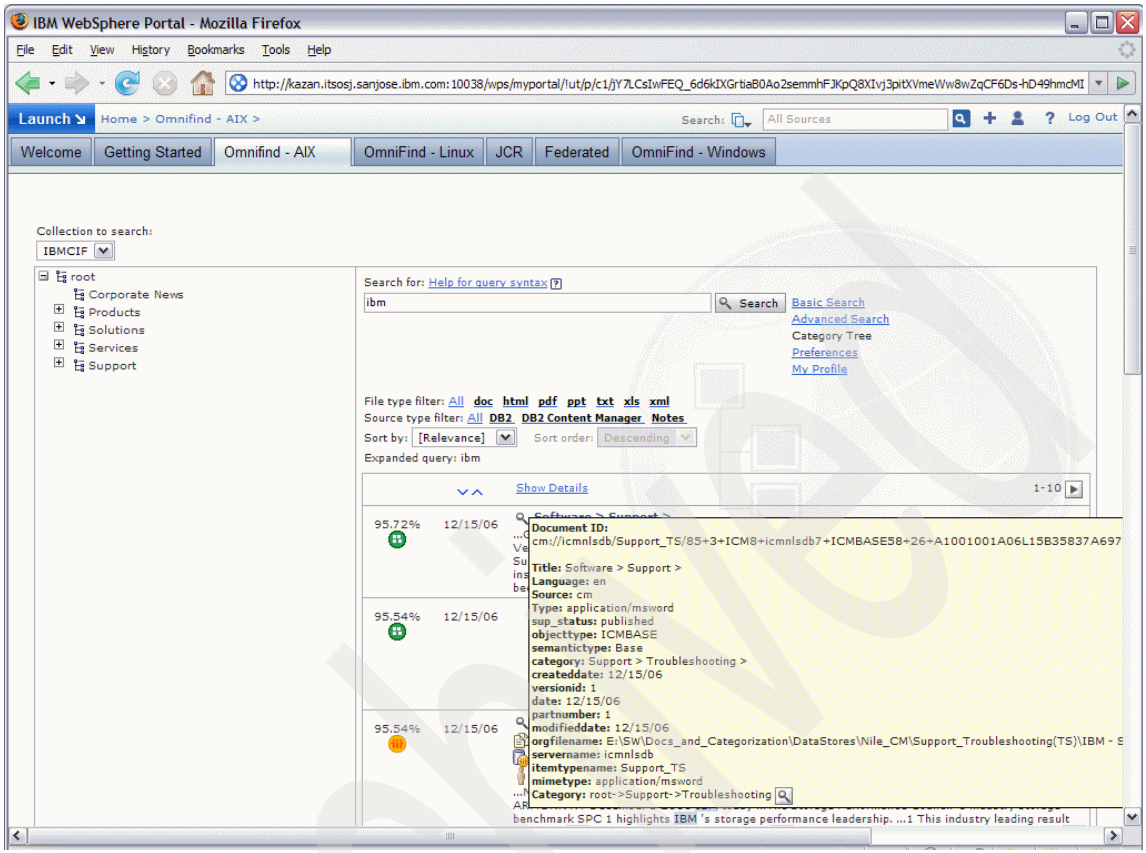

Figure 4-114 Document detail

5

Merger of SMB and medium-size organizations

In this chapter, we describe the development of a custom federation portlet that searches the NWINSURANCE, GENINSINFO, and CUSTINFO collections spanning the Windows and Linux platforms as a consequence of the merger of the Northwest Insurance and SEQUOIA General insurance companies.

The topics are:

- Remote and local federators
- Custom federation portlet
- Search queries using the federatedsearch portlet

5.1 Introduction

As the first steps towards its plans to expand through acquisitions of other insurance companies in different geographic markets and domains,. such as auto insurance and home insurance, Northwest Insurance acquired Sequoia General. This would promote opportunities for cross selling products in each others customer base, and acquiring new customers with a broader set of product offerings.

An important requirement to achieve this objective was to integrate the enterprise search solutions implemented in each environment, such that a single suitably authorized user in the merged organization could seamlessly access relevant documents regardless of whether the information resided in the Sequoia General's enterprise search system or Northwest Insurance's enterprise search system. Fortunately, both organizations had chosen to implement the IBM OmniFind Enterprise Edition V8.4 enterprise search solution, albeit one on a Windows 2003 platform and the other on a Linux Red Hat platform.

IBM OmniFind Enterprise Edition's local and remote federator functionality in its Search and Index API (SIAPI) makes this integration quite simple.

IBM OmniFind Enterprise Edition provides several sets of Java application programming interfaces (APIs) for enterprise search so that you can customize search or administration applications, modify crawled documents, or set up an identity management component. The enterprise search implementation of SIAPI allows the search server to be accessed remotely. The search server stores the collection data for the enterprise search system. With these APIs, you can create applications that submit search requests across multiple collections stored in different servers, process search results, or browse taxonomy trees. You can also use the SIAPI to create administration applications to administer collections and enable indexes to be searched. For detailed information about SIAPIs, refer to *IBM OmniFind Enterprise Edition Version 8.4 Programming Guide and API Reference for Enterprise Search*, SC18-9284.

In the following sections, we provide a brief overview of IBM OmniFind Enterprise Edition's local and remote federator functionality, the implementation of a custom federation portlet that federates over the NWINSURANCE, GENINSINFO, and CUSTINFO collections spanning the Windows and Linux platforms, and search queries using the federatedsearch portlet.

5.2 Remote and local federators

The SIAPIs support the following types of search tasks:

- Searching indexes
- Customizing the information that is returned in search results sets
- Searching and browsing taxonomies
- Searching over several collections as though they were one collection (search federation)
- Viewing results with URIs that you can click and view scoring information (ranking)
- Searching and retrieving documents from a broad range of enterprise data sources, such as IBM Content Edition repositories and Lotus Notes databases

The general steps in creating a search application with the SIAPI are as follows:

1. Instantiate an implementation of a SearchFactory object.

   The SearchFactory can then be used to obtain a SearchService object.

2. Use the SearchFactory object to obtain a SearchService object.

   The SearchService object is configured with the connection information that is necessary to communicate with the search engine.

3. Obtain a Searchable object.

   After you obtain a SearchService object, you can use it to obtain one or more Searchable objects. Each SIAPI searchable object is associated with one enterprise search collection. You can also use the SearchService object to obtain a federator object. A federator object is a special kind of Searchable object that enables you to submit a single query across multiple Searchable objects (collections) at the same time. You need to identify your application by using an application ID.

4. Issue queries.

   The search application passes search queries to the search runtime on the search server.

Attention: The focus of the custom search portlet is search federation.

Our custom federation portlet needs to federate over collections stored in two different servers. We therefore need to use a federator to issue a federatedsearch request across a set of heterogeneous searchable collections and get a unified document result set. Search federators are intermediary components that exist between the requestors of service and the agents that perform that service. They are coordinate resources to manage the multitude of searches that are generated from a single request.

IBM OmniFind Enterprise Edition provides two types of SIAPI federators as follows:

- Local federator

  A local federator runs in the client, such as a search portlet, and federates from the *client* over a set of searchable objects. A local federator is created by using the createLocalFederator method from the SIAPI SearchFactory class. The set of searchable collections on which the query is run is specified when the federator is created. A subset of searchable objects can also be specified during search calls. Before you can create a local federator, you must create or retrieve searchable objects by using a SIAPI SearchFactory. The searchable object that is passed to the local federator must be ready for search without any additional information. The local federator uses the searchable object to issue a federatedsearch request.

- Remote federator

  A remote federator federates from a server over a set of searchable objects. A remote federator is run on the server and consumes server resources. It can only federate results from the server it is running on (there is no proxy-like functionality). A remote federator is created by using the getFederator method from the SearchService class. By default, a remote federator will search and federate across all collections accessible by the application name specified. It is also possible to restrict a particular search request to a subset of accessible collections by passing a string array of collection IDs as a second parameter to the search method.

Search federators are SIAPI searchable objects. Multiple-level[1] federation is allowed. Two levels are shown in Figure 5-1.

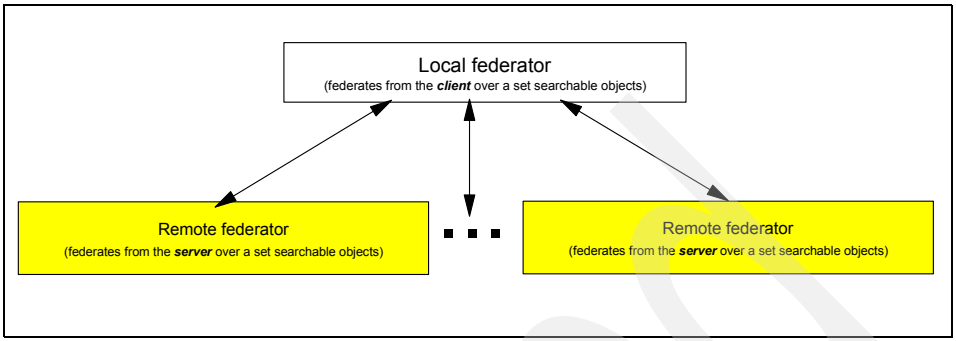

Figure 5-1 Multiple federator levels

5.3 Custom federation portlet

Since we need to develop a federation portlet (FederatedSearch) that federates over collections in two different servers, we need to define a remote federator for the Linux platform that federates over the GENINSINFO and CUSTINFO collections in falcon.itsosj.sanjose.ibm.com, and a local federator that federates over the remote federator and the NWINSURANCE collection in nile.itsosj.sanjose.ibm.com on the Windows platform, as shown in Figure 5-2.

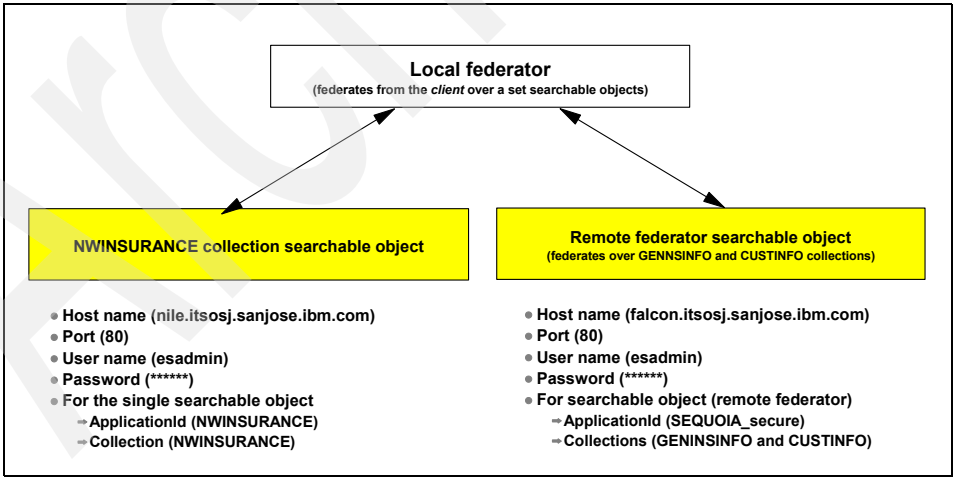

Figure 5-2 Federator design for the custom portlet

[1] Stacking of local federators is possible, but not recommended.

Example 5-1 on page 415 (Portlet descriptor), Example 5-2 on page 415 (Portlet.java), Example 5-3 on page 419 (view.jsp), and Example 5-4 on page 420 (sessionbean.java) show the code developed for the custom federation portlet. The portlet is configured through portlet properties, which can be set using Portal Administration windows.

Note: This portlet uses the recommended JSR168 APIs; the IBM Portlet APIs are deprecated in WebSphere Portal Server V6.

The processing flow is as follows:

1. The portlet descriptor in Example 5-1 on page 415 identifies the name of the portlet as FederatedSearch, and the portlet-class as com.ibm.federatedsearch.Portlet. This portlet only handles View mode; the Help and Edit modes are not configured.

   Based on the contents of the portlet descriptor, the doView method (see Example 5-2 on page 415) of the federatedsearch class is invoked.

2. The doView method uses the SessionBean class for all its search processing, and the view.jsp to render the results.

   a. The getSessionBean method in the doView method creates the local and remote federators (using information obtained from the portlet configuration properties file, as shown in Figure 5-3 on page 418), and sets the user security context (USC) string[2] using the LTPAToken (for QuickPlace and WebSphere Portal), IMC for Windows file system and Portal Document Manager, and the lookup of the Tivoli Directory Server LDAP repository for any DB2 groups[3] associated with the logged in user to populate security tokens in the USC string.

      Note: The federatedsearch portlet assumes that the IMC credentials (Cloudscape) database has been previously primed with the Windows file system and Portal Document Manager credentials, and therefore no IMC credential prompts occur.

   b. The doView method then invokes the view.jsp to create the HTML fragment to be displayed by the portlet container for this portlet.

[2] Uses IMC API for the Windows file system, QuickPlace, DB2 Content Manager, Portal Document Manager and WebSphere Portal. The SSO authentication method is used for QuickPlace and WebSphere Portal, while IMC prompts are used for authentication information for the Windows file system, DB2 Content Manager and Portal Document Manager. The DB2 user groups for the logged in user is obtained by looking up the LDAP repository (Tivoli Directory Server).

[3] This is not really required for this example, but is provided as sample code as to how a lookup could be coded.

3. The view.jsp generates the HTML form, as shown in Figure 5-6 on page 424, for the user to enter a search request. When the query is entered by a user, view.jsp eventually gets control and invokes the createQuery method of the SessionBean class with the search query, USC string, and connection information. This method then processes the request and returns the search results to the view.jsp for rendering.

Note: The FederatedSearchPortlet.war file can be downloaded from the IBM Redbooks Web site http://www.redbooks.ibm.com/abstracts/sg247394.html?Open by clicking **Additional Material**.

Example 5-1 Portlet descriptor WEB_INF/portlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<portlet-app xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_1_0.xsd" version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_1_0.xsd
http://java.sun.com/xml/ns/portlet/portlet-app_1_0.xsd" id="com.ibm.federatedsearch.FederatedSearchPortlet.245812e7f0">
    <portlet>
        <portlet-name>FederatedSearch</portlet-name>
        <display-name>Federated Search</display-name>
        <portlet-class>com.ibm.federatedsearch.Portlet</portlet-class>
        <expiration-cache>0</expiration-cache>
        <supports>
            <mime-type>text/html</mime-type>
            <portlet-mode>view</portlet-mode>
        </supports>
        <portlet-info>
            <title>Federated Search</title>
            <short-title>FederatedSearch</short-title>
        </portlet-info>
    </portlet>
</portlet-app>
```

Example 5-2 Portlet java

```
package com.ibm.federatedsearch;

import java.io.IOException;
import java.util.Iterator;
import java.util.List;
import java.util.Vector;
import java.util.logging.Level;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import javax.portlet.ActionRequest;
import javax.portlet.ActionResponse;
import javax.portlet.GenericPortlet;
import javax.portlet.PortletException;
import javax.portlet.PortletPreferences;
import javax.portlet.PortletRequest;
import javax.portlet.PortletRequestDispatcher;
import javax.portlet.PortletSession;
import javax.portlet.RenderRequest;
import javax.portlet.RenderResponse;
```

```
import com.ibm.portal.portlet.service.PortletServiceHome;
import com.ibm.portal.portlet.service.PortletServiceUnavailableException;
import com.ibm.portal.um.Group;
import com.ibm.portal.um.PumaLocator;
import com.ibm.portal.um.PumaProfile;
import com.ibm.portal.um.User;
import com.ibm.portal.um.exceptions.PumaException;
import com.ibm.portal.um.portletservice.PumaHome;

public class Portlet extends GenericPortlet {
    public static final String SESSION_BEAN  = "SessionBean";

    public void init() throws PortletException{
        super.init();
    }

    public void doView(RenderRequest request, RenderResponse response) throws PortletException, IOException {
        // Set the MIME type for the render response
        response.setContentType(request.getResponseContentType());

        // Check if portlet session bean exists. If not it means problems occured during initialization.
        SessionBean sessionBean = getSessionBean(request);
        if( sessionBean==null ) {
            response.getWriter().println("<b>Federated Search is not configured</b><p>");
            response.getWriter().println("Use administration interface to set parameters.");
            return;
        }

        // Invoke the JSP to render. It will use session bean for all processing.
        PortletRequestDispatcher rd = getPortletContext().getRequestDispatcher("/view.jsp");
        rd.include(request,response);
    }

    /*
     * Action phase in portlet handling. We just pass query parameter to render phase.
     */
    public void processAction(ActionRequest request, ActionResponse response) throws PortletException, java.io.IOException
{
        String query = request.getParameter("query");
        response.setRenderParameter("query", query);
    }

    /*
     * Extract content of LtpaToken cookie from request.
     */
    private static String getLTPAToken(PortletRequest request) {
        String cookies = request.getProperty("cookie");
        String cookieName = "LtpaToken";

        if (cookies.indexOf(cookieName) >= 0) {
            int startValue = cookieName.length() + 1 + cookies.indexOf(cookieName);
            int endValue = cookies.length();
            if (cookies.indexOf(";", startValue) > 0) {
                endValue = cookies.indexOf(";", startValue);
            }
            return cookies.substring(startValue, endValue);
        }
        return "";
    }

    /*
     * This retrieves PUMA (Portal User Manager) service from JNDI.
     * It will be used for getting groups membership for current user.
     */
    private static PumaHome getPumaService(PortletRequest request) {
        PortletServiceHome psh;
        PumaHome service = null;
```

```
        try{
           javax.naming.Context ctx = new javax.naming.InitialContext();
           psh = (PortletServiceHome)
                 ctx.lookup("portletservice/com.ibm.portal.um.portletservice.PumaHome");
           if (psh != null){
              service = (PumaHome) psh.getPortletService(PumaHome.class);
           }
        }
        catch (Exception e){
            System.err.println("Could not initialize PUMA service");
        }

        return service;
    }

    private static SessionBean getSessionBean(PortletRequest request) {
        PortletSession session = request.getPortletSession();

        if( session == null )
            return null;
        SessionBean sessionBean = (SessionBean)session.getAttribute(SESSION_BEAN);
        if( sessionBean == null ) {
            sessionBean = new SessionBean();
            // All configuration parameters are stored in portlet preferences.
            // They are editable in Portal administration screens.
            PortletPreferences prefs = request.getPreferences();
            try {
                // Get PUMA services
                PumaHome service = getPumaService(request);
                PumaProfile pp = service.getProfile(request);
                PumaLocator pl = service.getLocator(request);

                // Translate current user id into LDAP DN
                User user = pp.getCurrentUser();
                String userdn = pp.getIdentifier(user);

                // Get all groups IDs for current user
                List groups = pl.findGroupsByPrincipal(user, true);

                // Pattern probably should be configurable in portlet properties.
                Pattern p = Pattern.compile("cn=token-([^,]+),.*");
                Vector tokens = new Vector();

                // Now check which groups define tokens
                for(Iterator i = groups.iterator();i.hasNext();) {
                Group group = (Group) i.next();

                // Convert group id into group DN
                String groupdn = pp.getIdentifier(group);

                // Check for match with token pattern
                    Matcher m = p.matcher(groupdn);
                    if(m.matches()) {
                        // Just in case convert extracted token into upper case
                        String token = m.group(1).toUpperCase();

                        // And add new token to vector
                        tokens.add(token);
                        System.out.println("ADDING TOKEN: "+token);
                    }
                }

                // Initialize all SIAPI objects from configuration parameters
                sessionBean.initialize(prefs);
                // Create security context based on LTPA key and tokens
                sessionBean.initializeSecurityContext(userdn,(String[])tokens.toArray(new
String[tokens.size()]),getLTPAToken(request));
```

```
                session.setAttribute(SESSION_BEAN,sessionBean);
            } catch(Exception e) {
                System.err.println("Error while initializing seassion bean");
                e.printStackTrace();
                return null;
            }
        }
        return sessionBean;
    }

}
```

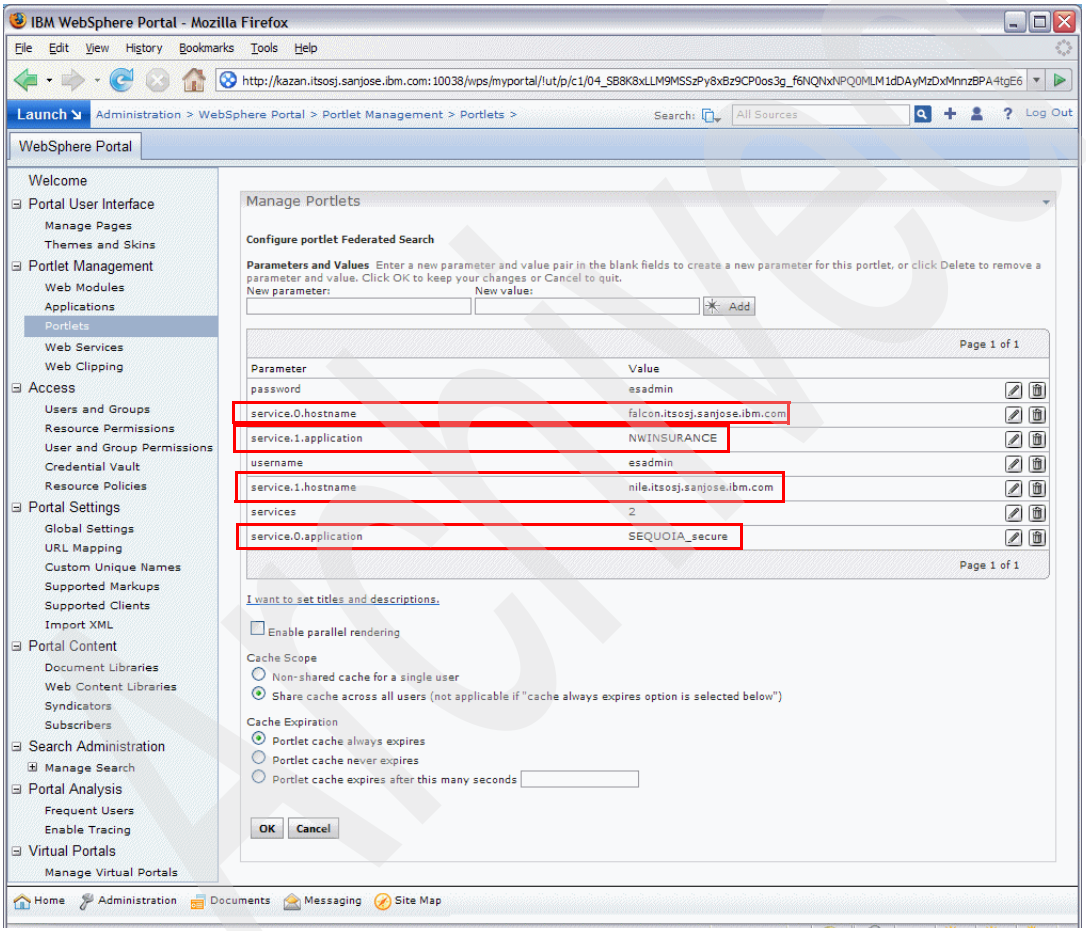

Figure 5-3 Portlet configuration properties

Example 5-3 view.jsp code

```
<%@page session="false" contentType="text/html" pageEncoding="ISO-8859-1"
import="java.util.*,javax.portlet.*,com.ibm.federatedsearch.*,com.ibm.siapi.common.*,com.ibm.siapi.search.*" %>
<%@taglib uri="http://java.sun.com/portlet" prefix="portlet" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<portlet:defineObjects/>

<%
SessionBean sessionBean = (com.ibm.federatedsearch.SessionBean)
renderRequest.getPortletSession().getAttribute(com.ibm.federatedsearch.Portlet.SESSION_BEAN);
String queryString = request.getParameter("query");
if(queryString==null)
  queryString="";
%>
Federated search over following collections:
<%
for(Iterator i = sessionBean.getCollectionIDs().iterator();i.hasNext();) {
  String collid = (String) i.next();
%>
<%= collid %>
<%
}
 %><p>
<form method="POST" action="<portlet:actionURL/>">
<input size="100" name="query" type="text" value="<%= queryString %>"/>
<input name="search" type="submit" value="Search"/>
</form>

<%
if(queryString.length()==0)
    return;

Query query = sessionBean.createQuery(queryString);
ResultSet rset = sessionBean.search(query);
%>
Evaluation time: <%= rset.getQueryEvaluationTime() %>ms<br>
Found <%= rset.getEstimatedNumberOfResults() %> results. Out of those you can see <%= rset.getAvailableNumberOfResults()
%><p>
<%
Result[] results = rset.getResults();
for(int i=0;i<results.length;i++) {
Result result = results[i];
%>
<hr>
<b><%= result.getTitle()%></b> crawled by <b><%= result.getDocumentSource() %></b><br>
<%= result.getDescription()%><br>
<%
NameValuePair[] fields = result.getFields();
for(int j=0;j<fields.length;j++) {
NameValuePair field = fields[j];
%>
<%= field.getName() %> = <%= field.getValue() %><br>
<%
}
%>
<p>
<%
}
%>
```

Example 5-4 SessionBean java

```
package com.ibm.federatedsearch;

import java.util.Iterator;
import java.util.Properties;
import java.util.Vector;

import javax.portlet.PortletPreferences;

import com.ibm.es.api.imc.Identity;
import com.ibm.es.api.imc.IdentityManagementFactory;
import com.ibm.es.api.imc.IdentityManagementService;
import com.ibm.es.api.imc.SecurityContext;
import com.ibm.siapi.SiapiException;
import com.ibm.siapi.common.ApplicationInfo;
import com.ibm.siapi.common.CollectionInfo;
import com.ibm.siapi.search.LocalFederator;
import com.ibm.siapi.search.Query;
import com.ibm.siapi.search.RemoteFederator;
import com.ibm.siapi.search.ResultSet;
import com.ibm.siapi.search.SearchFactory;
import com.ibm.siapi.search.SearchService;
import com.ibm.siapi.search.Searchable;

public class SessionBean {
    private SearchFactory searchFactory;
    private IdentityManagementFactory imFactory;
    private Vector services = new Vector();
    private LocalFederator federator;
    private SecurityContext securityContext;
    private Vector collections = new Vector();

    private class MySearchService {
        private Properties config;
        private RemoteFederator federator;
        private ApplicationInfo application;
        private SearchService searchService;
        private IdentityManagementService imService;

        public MySearchService(Properties c) throws SiapiException {
            config = (Properties) c.clone();
            System.out.println("Initializing search service: "+config);
            searchService = searchFactory.getSearchService(config);
            application = searchFactory.createApplicationInfo(config.getProperty("application","Default"));
            federator = searchService.getFederator(application, application.getId());
            imService = imFactory.getIdentityManagementService(config);

            CollectionInfo[] ci = federator.getCollectionInfos();
            for(int i=0;i<ci.length;i++) {
                collections.add(ci[i].getID());
            }
        }

        public Searchable getSearchable() {
            return federator;
        }

        public SecurityContext getSecurityContext(String userdn) throws SiapiException {
            SecurityContext ctx = imService.getSecurityContext(userdn);
            System.out.println("Retrieved security context: "+ctx.serialize(false));
            return ctx;
        }
    }

    public void initialize(PortletPreferences prefs) throws SiapiException {
        // Create SIAPI factories
```

```
    String searchFactoryClass = prefs.getValue("searchFactory", "com.ibm.es.api.search.RemoteSearchFactory");
    String imFactoryClass = prefs.getValue("imfactory", "com.ibm.es.api.imc.IdentityManagementFactory");

    try {
        Class cls = Class.forName(searchFactoryClass);
        searchFactory = (SearchFactory) cls.newInstance();
    } catch(Exception e) {
        System.err.println("SAPI search factory initialization failed");
    }

    try {
        Class cls = Class.forName(imFactoryClass);
        imFactory = (IdentityManagementFactory) cls.newInstance();
    } catch(Exception e) {
        System.err.println("SAPI identity management factory initialization failed");
    }

    // Get default configuration values
    String username = prefs.getValue("username","");
    String password = prefs.getValue("password","");
    String appname = prefs.getValue("application","Default");

    // Overwrite defaults with settings for each remote service
    int scount = Integer.parseInt(prefs.getValue("services","0"));

    // Options for services will start with "service.X." prefix
    // Some options like username, password and application name can be
    // "inherited" from global defaults
    System.out.println("Number of services: "+scount);
    for(int i=0;i<scount;i++) {
        String prefix = "service." + String.valueOf(i) + ".";
        Properties config = new Properties();
        config.setProperty("hostname", prefs.getValue(prefix+"hostname", ""));
        config.setProperty("username", prefs.getValue(prefix+"username", username));
        config.setProperty("password", prefs.getValue(prefix+"password", password));
        config.setProperty("application", prefs.getValue(prefix+"application", appname));
        config.setProperty("port", prefs.getValue(prefix+"port", "80"));
        System.out.println("SERVICE CONFIG: "+config);

        // Now create new search service
        try {
            services.add(new MySearchService(config));
        } catch (Exception e) {
            System.err.println("Failed to initialize remote search: " + config);
            e.printStackTrace();
        }
    }

    // For debugging purposes fetch all collection IDs from each federator
    // They will be displayed in view.jsp code.
    Searchable[] searchables = new Searchable[services.size()];
    for(int i=0;i<services.size();i++) {
        searchables[i] = ((MySearchService)services.get(i)).getSearchable();
    }

    // Create local federator spanning all configured remote federators
    try {
        federator = searchFactory.createLocalFederator(searchables);
    } catch (SiapiException e) {
        System.err.println("Failed to create local federator");
        throw e;
    }
}

public Query createQuery(String q) throws SiapiException {
    Query query = searchFactory.createQuery(q);
```

```
    // Set options enabling returning of attributes in result lists
    query.setReturnedAttribute(Query.RETURN_RESULT_DESCRIPTION,true);
    query.setReturnedAttribute(Query.RETURN_RESULT_FIELDS,true);
    query.setReturnedAttribute(Query.RETURN_RESULT_SOURCE,true);
    query.setReturnedAttribute(Query.RETURN_RESULT_TYPE,true);
    query.setReturnedAttribute(Query.RETURN_RESULT_URI,true);
    query.setReturnedAttribute(Query.RETURN_RESULT_TITLE,true);

    // Set security context on query
    if(securityContext!=null)
        query.setACLConstraints("@SecurityContext::'" + securityContext.serialize(true) + "'");

    return query;
}

public Searchable getSearchable() {
    return federator;
}

public ResultSet search(Query query) throws SiapiException {
    ResultSet rset = federator.search(query);
    System.out.println("SEARCHING: "+query.getText());
    System.out.println("Results: "+rset.getAvailableNumberOfResults());
    System.out.println("Time: "+rset.getQueryEvaluationTime());
    return rset;
}

public void initializeSecurityContext(String userdn, String[] tokens, String ltpa) {
    if(userdn==null)
        return;

    // Create empty security context for username
    SecurityContext basectx = new SecurityContext();
    basectx.setUserID(userdn);

    // Set LTPA token for single sign on if available
    if(ltpa!=null)
        basectx.setSSOToken(ltpa);

    // Set tokens if provided
    if(tokens!=null)
        basectx.setNativeTokens(tokens);

    // Now merge contexts from each service. For initialization of IMC context user have to
    // use standard search application.
    for(Iterator i = services.iterator(); i.hasNext();) {
        MySearchService s = (MySearchService) i.next();
        SecurityContext newctx;
        try {
            newctx = s.getSecurityContext(userdn);
            mergeSecurityContext(basectx,newctx);
        } catch (SiapiException e) {
            System.err.println("Cannot retrieve security context.");
            e.printStackTrace();
        }
    }
    System.out.println("Initialized security context: " + basectx.serialize(false));
    securityContext = basectx;
}

/*
 * Merge two security contexts into first one.
 */
private void mergeSecurityContext(SecurityContext base, SecurityContext add) {
    Identity[] baseids = base.getIdentities();
    Identity[] newids = add.getIdentities();
    Identity[] ids = new Identity[baseids.length + newids.length];
```

```
        for(int i=0;i<baseids.length;i++) {
            ids[i] = baseids[i];
        }

        for(int i=0;i<newids.length;i++) {
            ids[i + baseids.length] = newids[i];
        }

        base.setIdentities(ids);
    }

    public Vector getCollectionIDs() {
        return collections;
    }
}
```

5.4 Search queries using federatedsearch portlet

In this step, we describe a few search query interactions using the federatedsearch portlet.

Figure 5-4 on page 424 through Figure 5-9 on page 427 describe the login to WebSphere Portal as user wpsadmin (Figure 5-4 on page 424), and the invocation of the federatedsearch portlet by selecting the **Federated** tab (Figure 5-5 on page 424). This displays the search box (Figure 5-6 on page 424), which also shows the three collection IDs GENINSINFO, CUSTINFO, and col_34035 (NWINSURANCE collection). Figure 5-7 on page 425 and Figure 5-8 on page 426 show the seven documents in the search results for the string “smith”, while Figure 5-9 on page 427 shows one document in the search results for the string “sarah”.

Figure 5-10 on page 427 through Figure 5-12 on page 429 describe the login to WebSphere Portal as a different user esadmin (Figure 5-10 on page 427) with different access privileges, and therefore a different number of documents in the search results for the same search strings. Figure 5-11 on page 428 now shows six documents (as opposed to seven with user ID wpsadmin) in the search results for the string “smith”, while Figure 5-12 on page 429 now shows two documents (as opposed to one with user ID wpsadmin) in the search results for the string “sarah”.

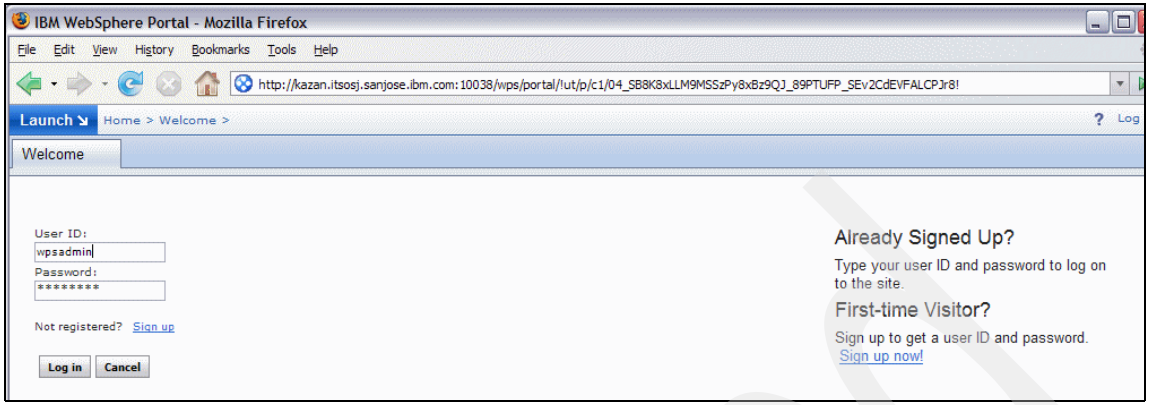

Figure 5-4 Log in to WebSphere Portal as wpsadmin

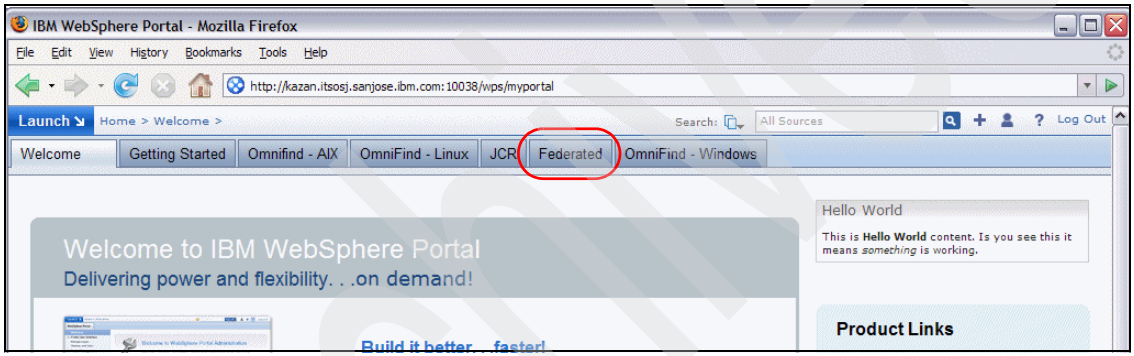

Figure 5-5 Click the Federated tab to invoke the federatedsearch portlet

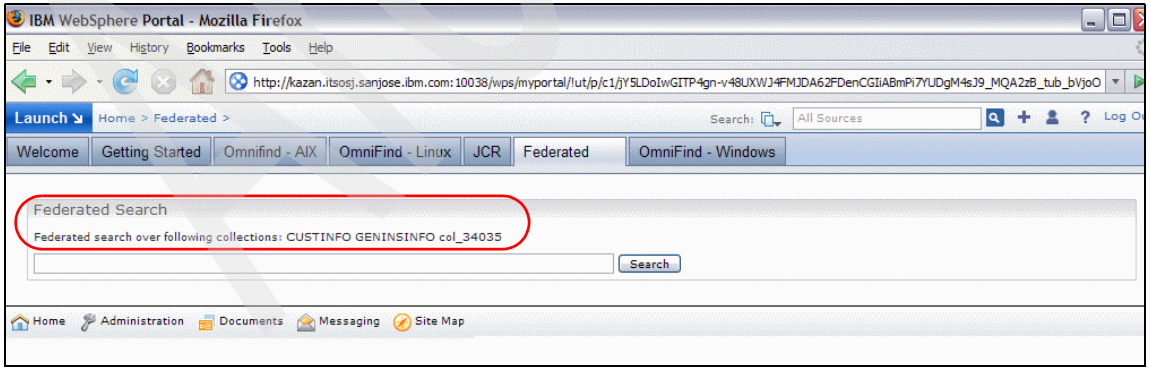

Figure 5-6 Search box

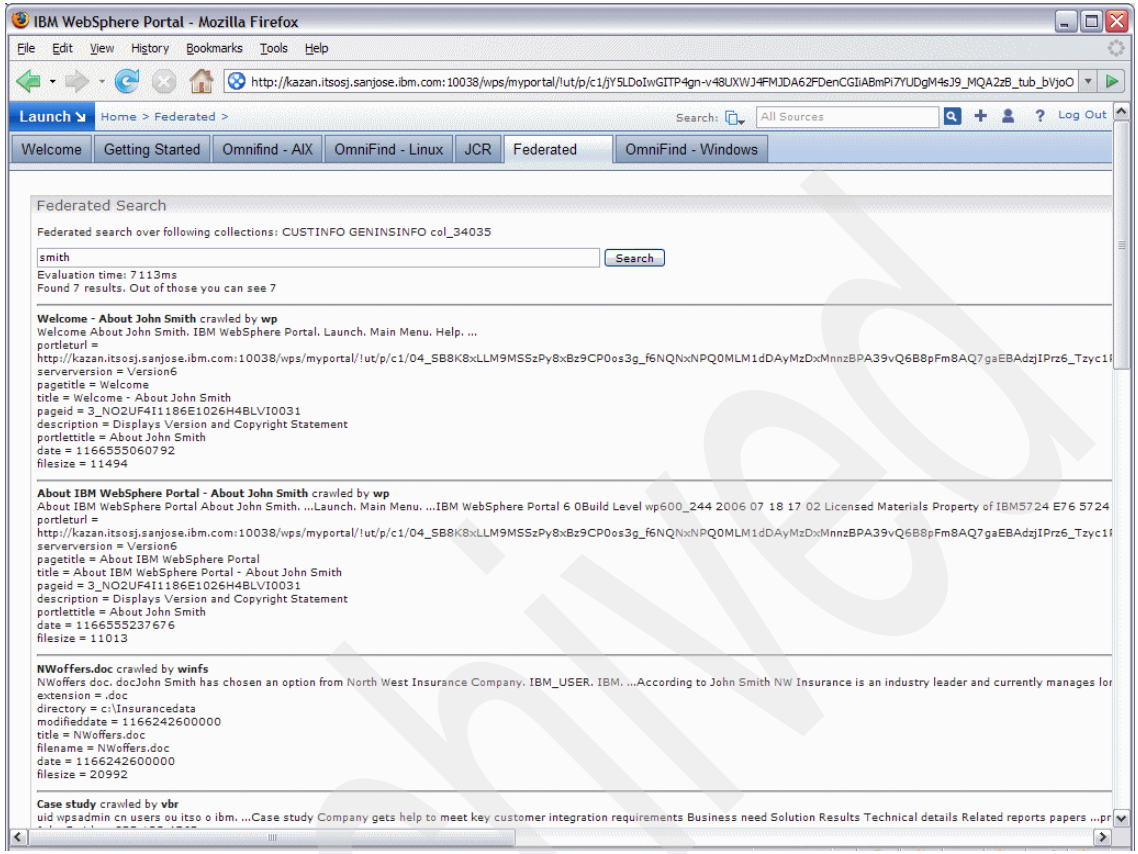

Figure 5-7 Search results for “smith” 1/2

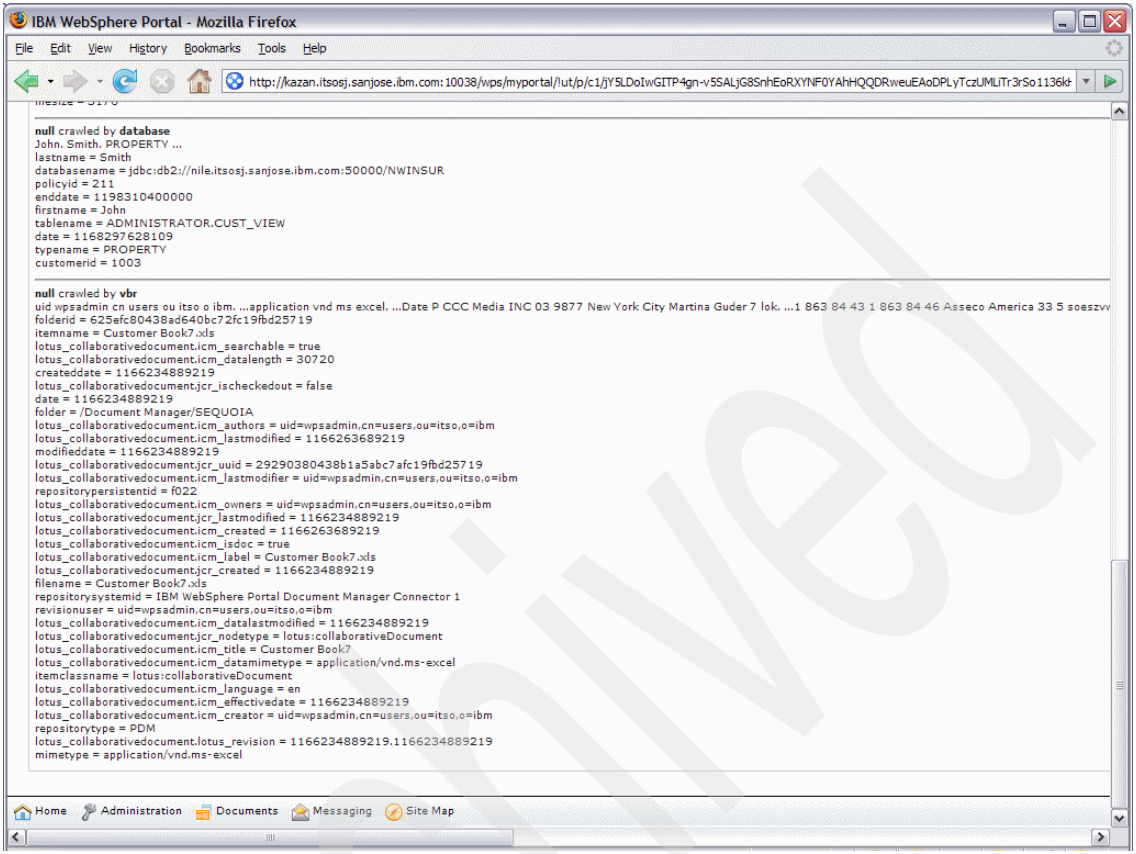

Figure 5-8 Search results for “smith” 2/2

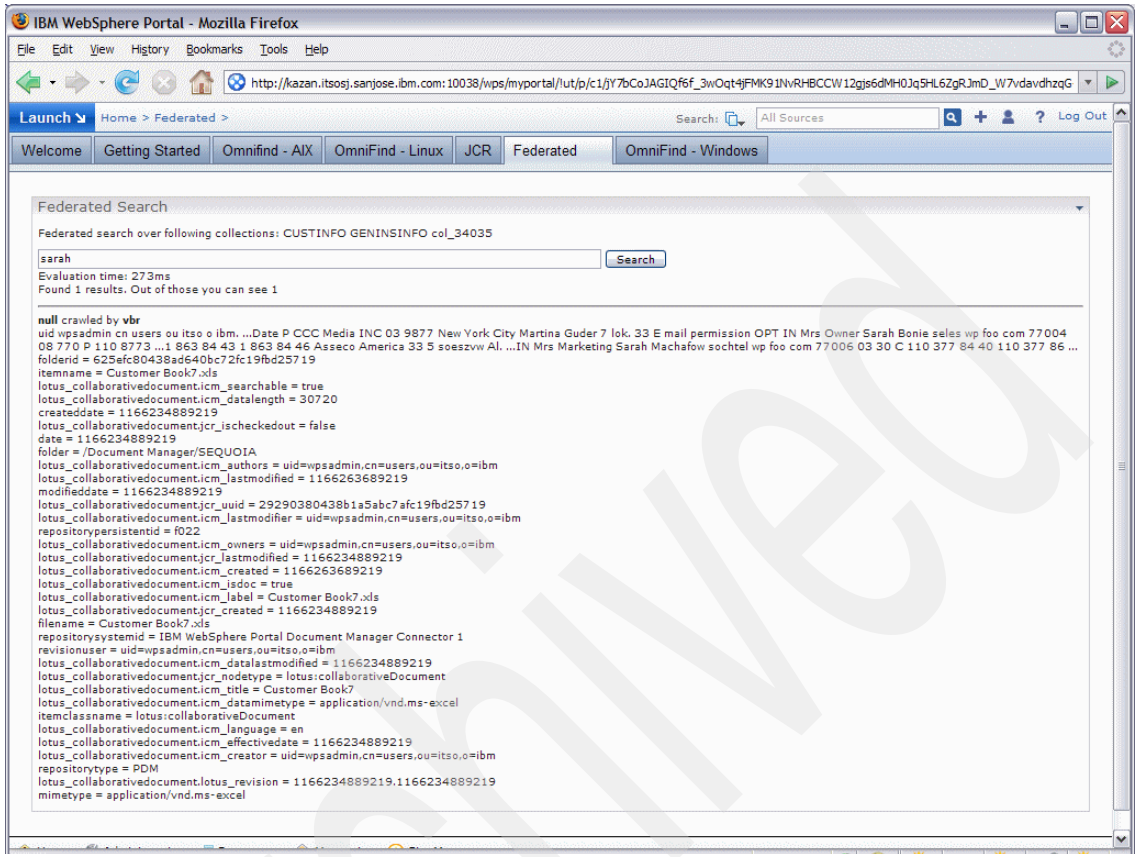

Figure 5-9 Search results for "sarah"

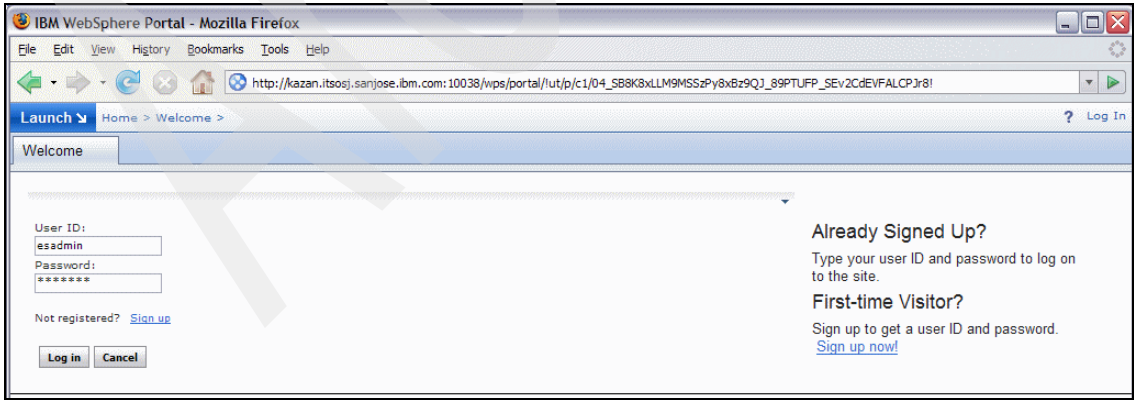

Figure 5-10 Log in as esadmin

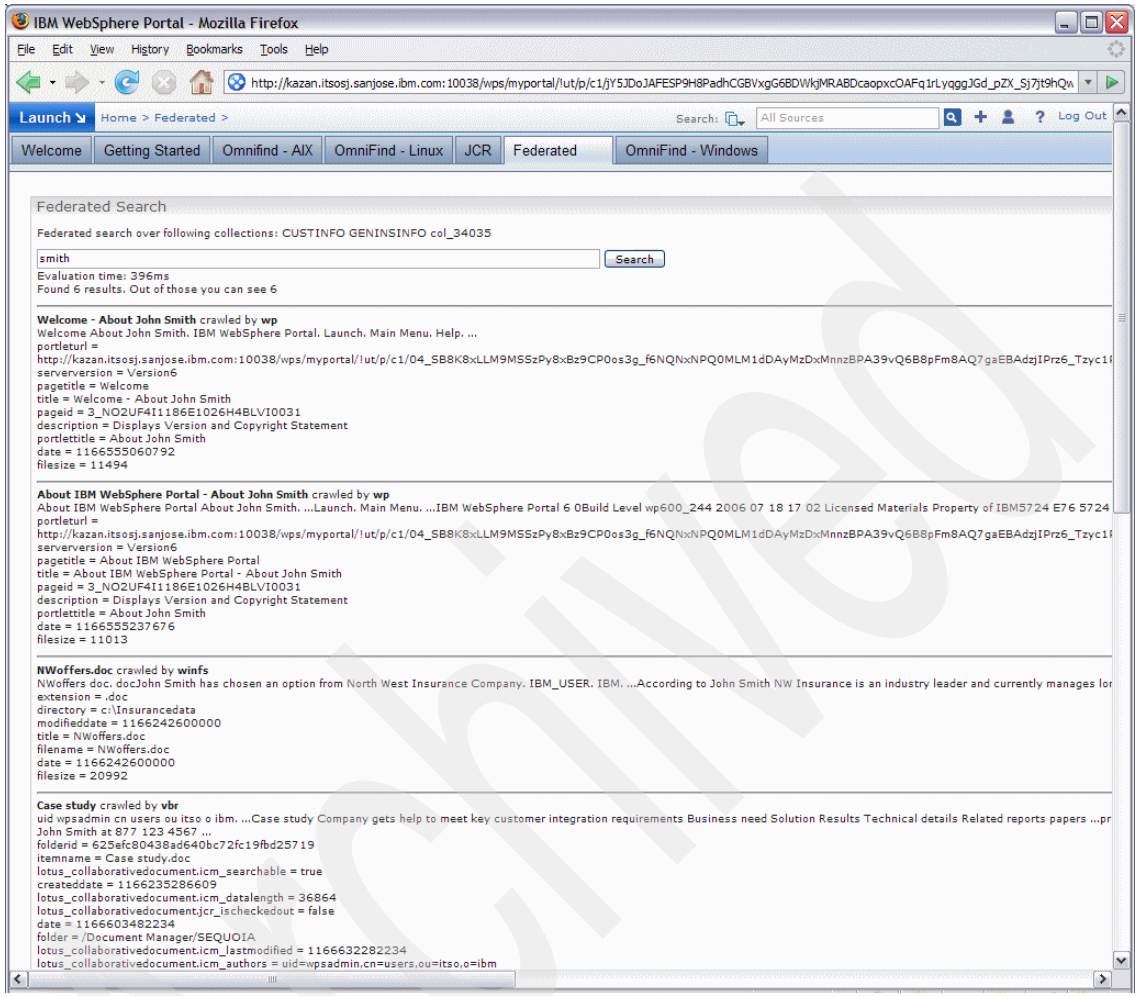

Figure 5-11 Search results for “smith”

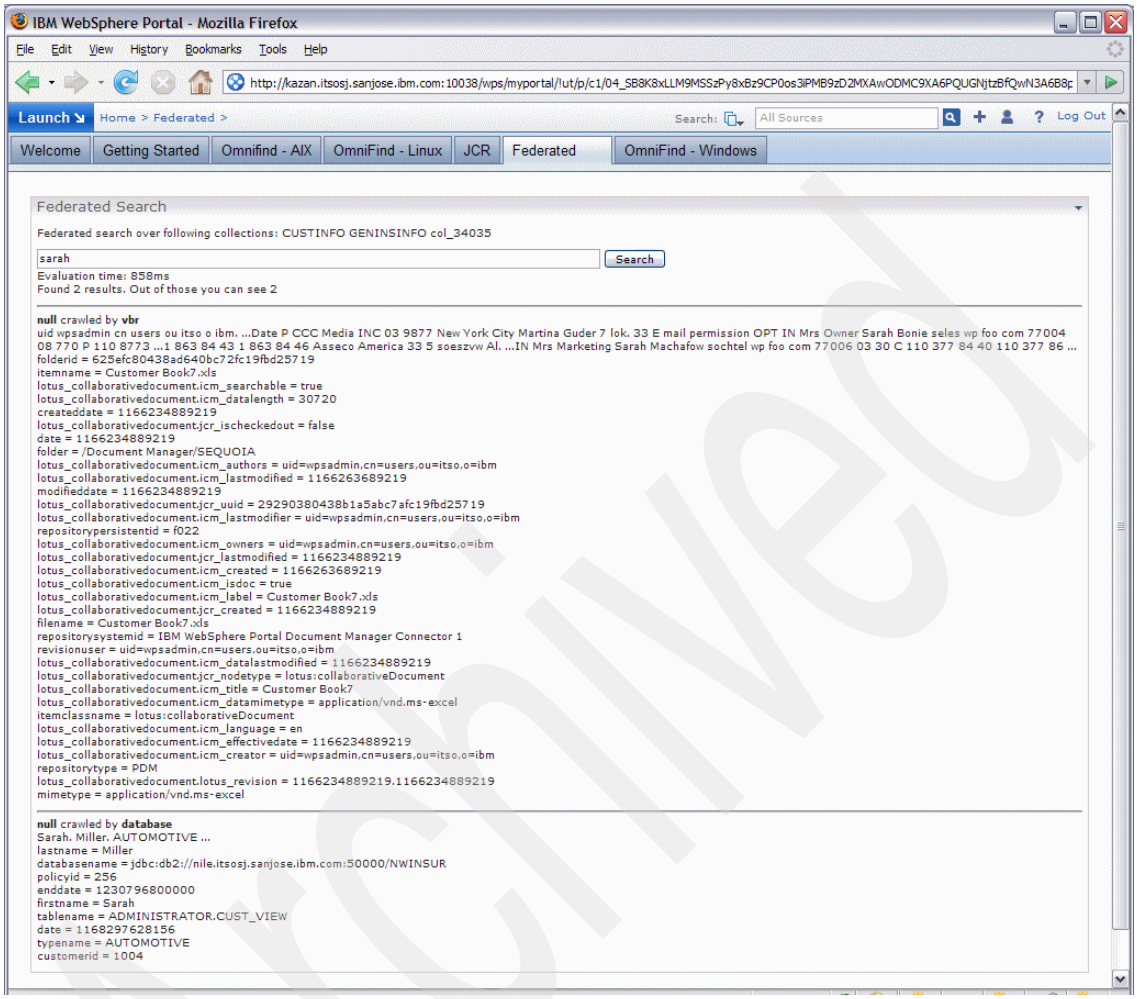

Figure 5-12 Search results for “sarah”

A

Install Sample Search application portlet

In this appendix, we describe the steps in installing the Sample Search application portlet in WebSphere Portal Server.

Introduction

To integrate enterprise search with IBM WebSphere Portal, you can:

1. Run the setup scripts that are provided with the IBM OmniFind Edition installation program.
2. Use the WebSphere Portal administration interface to update search portlet properties and specify information about the search server for enterprise search.

These two steps are described in the following sections.

Run the setup scripts

Before you can run the setup script (wps6_install in our case) provided, you must copy the JAR file (es.wp6.install.jar in our case, located in $ES_INSTALL_ROOT/bin directory) that contains the setup scripts for your version of WebSphere Portal (Version 6 in our case) from the enterprise search server (nile.itsosj.sanjose.ibm.com) to the server where WebSphere Portal (kazan.itsosj/sanjose.ibm.com, which is a Windows 2003 platform) is installed. You must then unpack the file using the Java JAR command (or the TAR command), which extracts the following files:

- ESSearchPortlet.war
- ESPACServer.ear
- esapi.jar
- es.search.provider.jar
- es.security.jar
- wp6_install.bat or wp6_install.sh
- wp6_uninstall.bat or wp6_uninstall.sh
- Search application source type icons that are used in the search provider results page
- XML and JACL files that are needed by the installation

Note: The copying and unpacking of the es.wp6.install.jar file is not shown here.

The setup script wps6_install (with all the options[1] specified) performs the following tasks, as shown in Figure A-1 on page 434 through Figure A-9 on page 438:

1. Deploys EAR files that enable you to use enterprise search within WebSphere Portal and create crawlers for adding WebSphere Portal and IBM Workplace Web Content Management content to enterprise search collections.
2. Deploys WAR files that are required by the enterprise search portlet.
3. Creates pages in WebSphere Portal and assign the enterprise search portlet files to those pages.
4. Copies all required JAR files to the WebSphere Portal installation directories (JAR files already in the installation directories are backed up before the JAR files used for enterprise search are copied).
5. Provides an integration point for WebSphere Information Integrator Content Edition to search Portal Document Manager documents.

Guidelines for using the setup script are as follows:

- The scripts set up all integration points between enterprise search and WebSphere Portal. For example, you cannot selectively install the portlet and not install EAR files that support the WebSphere Portal and Web Content Management crawlers.
- If you do not set up WebSphere Information Integrator Content Edition, and later decide that you want to use a portlet to search Portal Document Manager documents, you can run the setup script again and specify the WebSphere Information Integrator Content Edition installation path.
- The scripts stop and restart WebSphere Portal. You might want to run the scripts after normal working hours to ensure that your user community is not affected by unavailability of portal services.

Attention: If an error occur while the setup scripts are running, run the setup script again. Tasks that completed successfully during the first attempt might report errors, but the setup process continues and completes the remaining tasks.

Note: The first time that you access the Enterprise Search portlet page after you run the setup script, the page might be slow to appear because the system must compile Java Server Pages (JSP™ files) for the portlet.

[1] For a detailed description of the options, refer to *IBM OmniFind Enterprise Edition V8.4 Administering Enterprise Search*, SC18-9283.

```
C:\IBM\es.wp6.install>wp6_install.bat -WPSProfileDir "C:\\IBM\\WebSphere\\profil
es\\wp_profile" -WASUser wasadmin -WASPassword wasadmin -WPSDir "C:\\IBM\\wps" -
WPSUser wpsadmin -WPSPassword wpsadmin -WPSHost "kazan.itsosj.sanjose.ibm.com:10
038/wps/portal/" -IICEDir "C:\\IBM\\IICE"

C:\IBM\es.wp6.install>call "C:\IBM\WebSphere\profiles\wp_profile\bin\ws_ant.bat"
 -Duser=wasadmin -Dpassword=wasadmin -buildfile wp6_install.xml stopServer
Buildfile: wp6_install.xml

stopServer:
     [echo] WSMSG12I: ** Stopping the WebSphere_Portal application server **
```

Figure A-1 Execute wp6_install.bat command 1/9

```
C:\WINDOWS\system32\cmd.exe - wp6_install.bat -WPSProfileDir "C:\\IBM\\WebSphere\\profiles\\w...
[stopServer] ADMU0116I: Tool information is being logged in file
[stopServer]            C:\ibm\WebSphere\profiles\wp_profile\logs\WebSphere_Port
al\stopServer.log
[stopServer] ADMU0128I: Starting tool with the wp_profile profile
[stopServer] ADMU3100I: Reading configuration for server: WebSphere_Portal
[stopServer] ADMU3201I: Server stop request issued. Waiting for stop status.
[stopServer] ADMU4000I: Server WebSphere_Portal stop completed.


BUILD SUCCESSFUL
Total time: 50 seconds
C:ESSearchPortlet.war
1 File(s) copied
C:es.search.provider.jar
C:es.security.jar
C:esapi.jar
3 File(s) copied
Buildfile: wp6_install.xml

copyImages:
     [echo] WSMSG12I: ** Copying icons to Portal Server icons directory **

BUILD SUCCESSFUL
Total time: 2 seconds
```

Figure A-2 Execute wp6_install.bat command 2/9

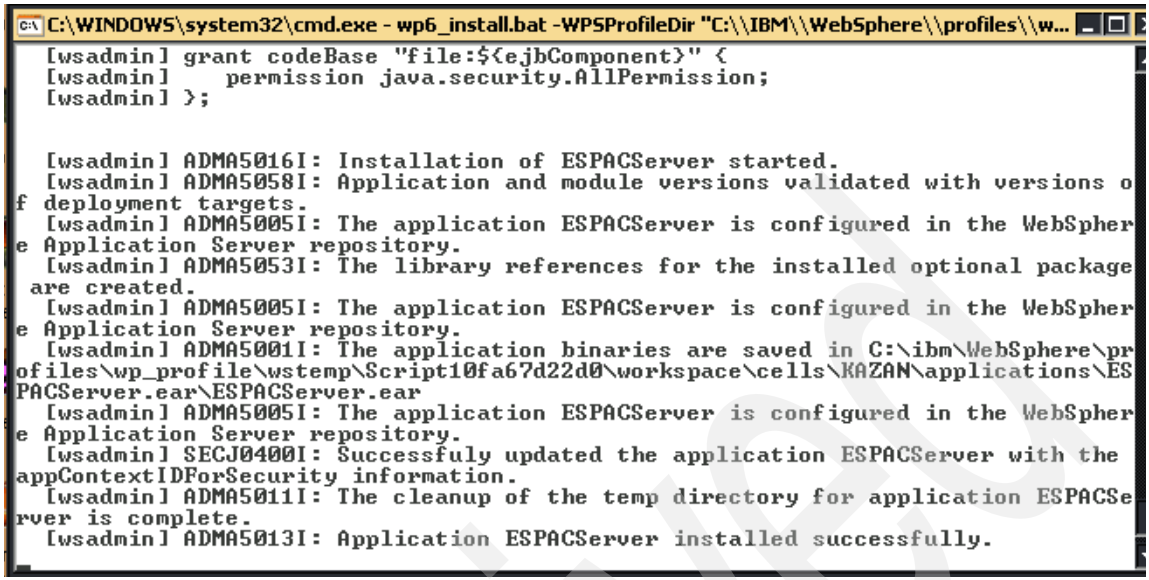

Figure A-3 Execute wp6_install.bat command 3/9

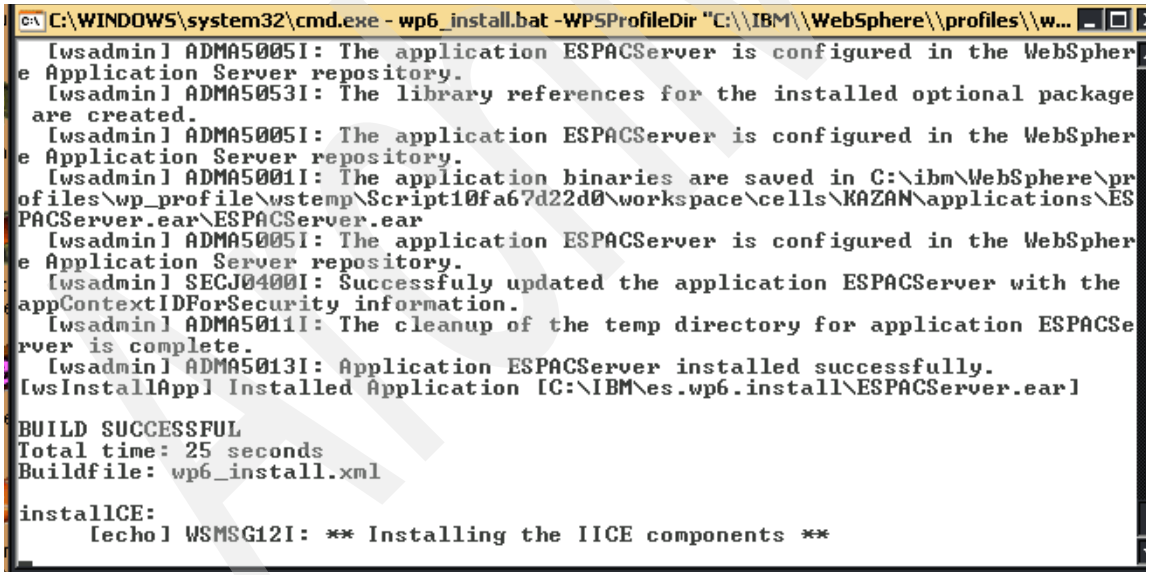

Figure A-4 Execute wp6_install.bat command 4/9

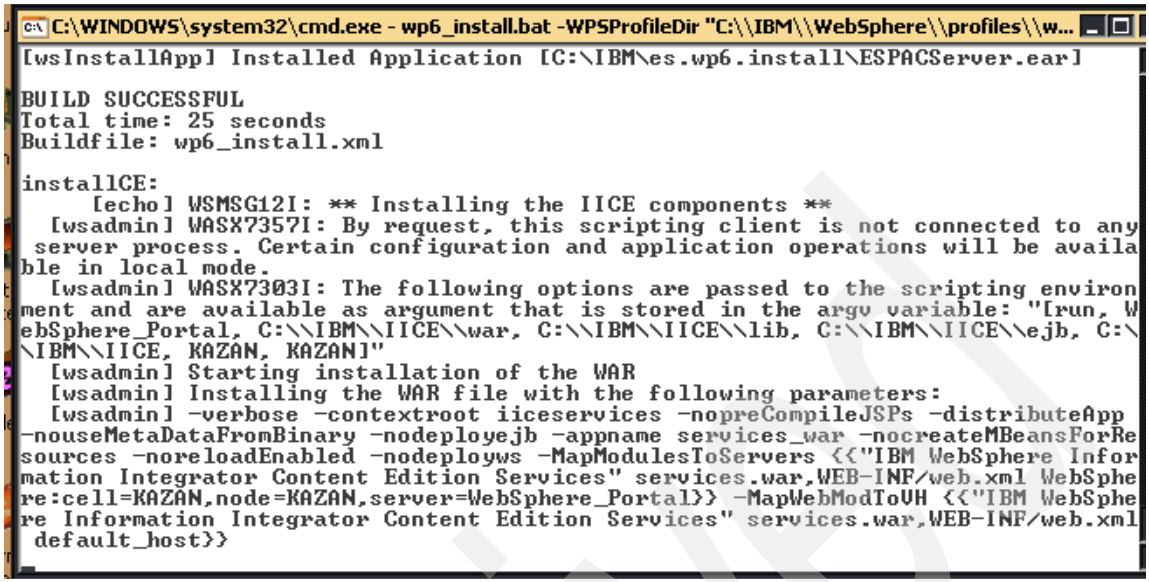

Figure A-5 Execute wp6_install.bat command 5/9

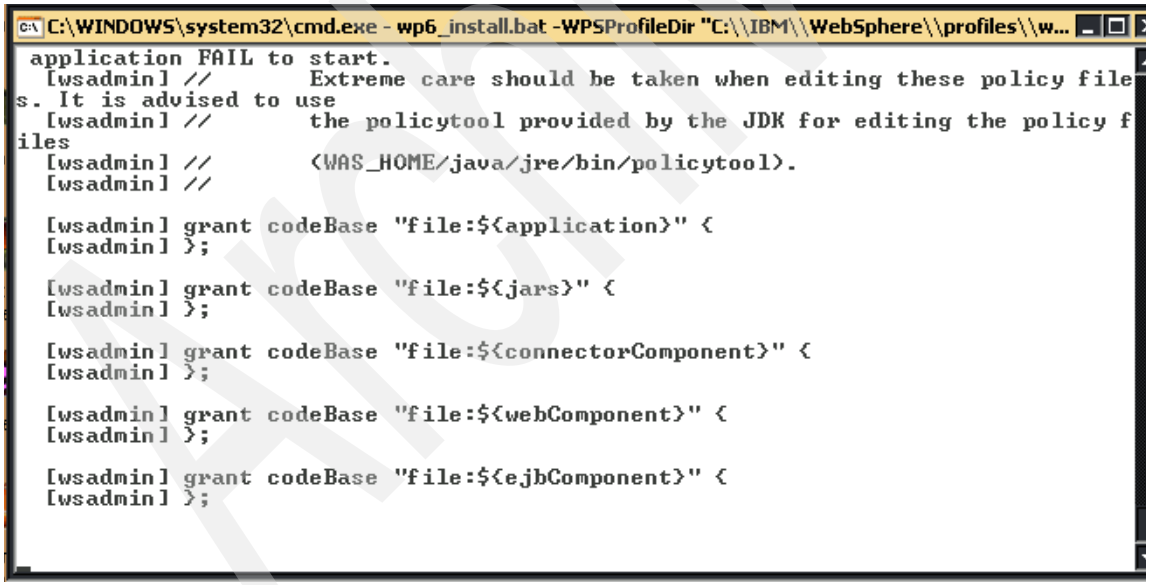

Figure A-6 Execute wp6_install.bat command 6/9

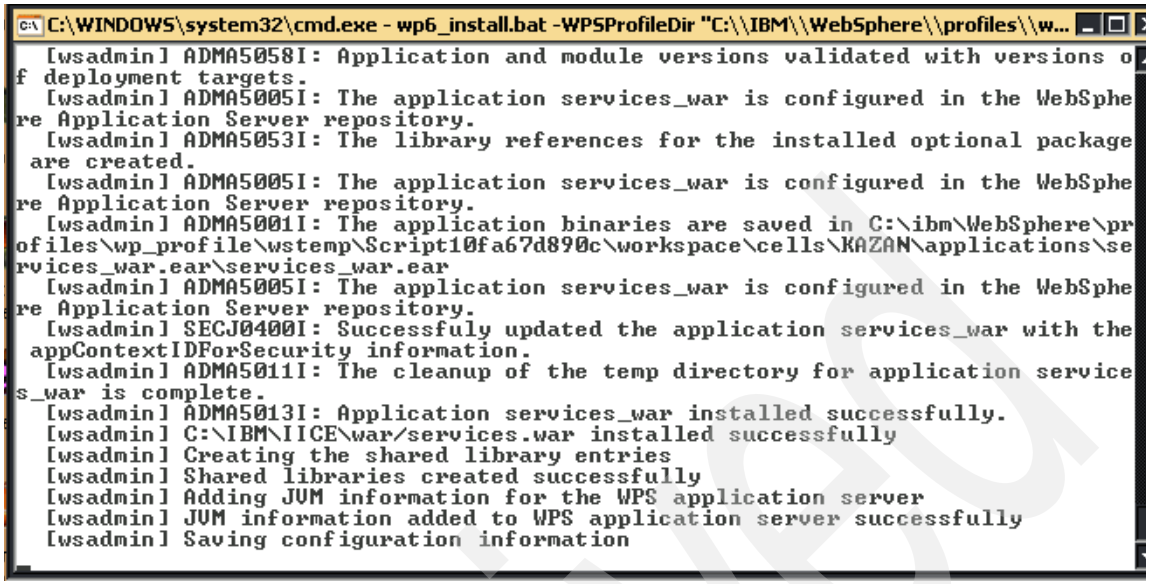

Figure A-7 Execute wp6_install.bat command 7/9

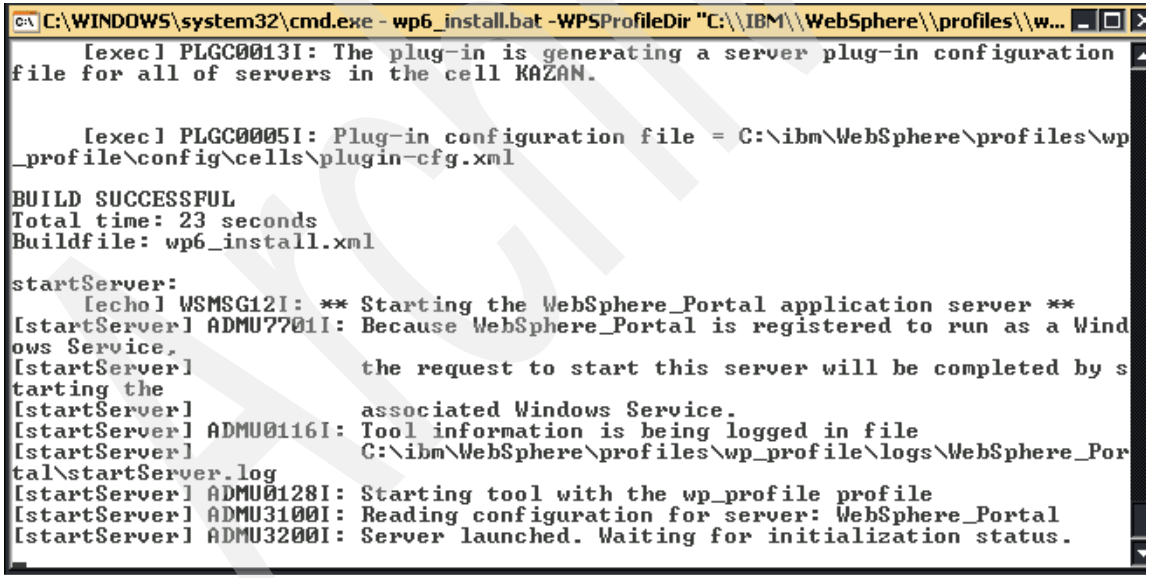

Figure A-8 Execute wp6_install.bat command 8/9

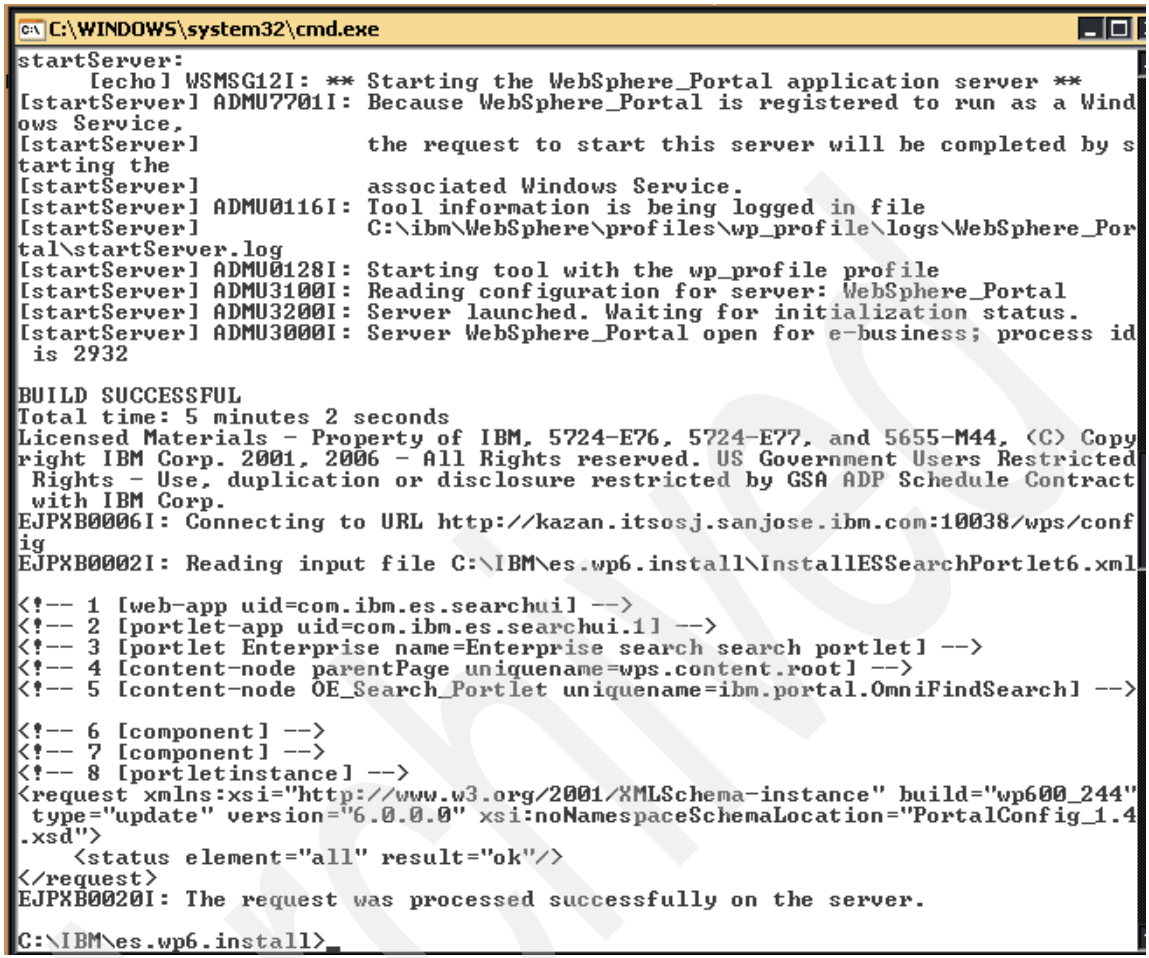

Figure A-9 Execute wp6_install.bat command 9/9

Configure WebSphere Portal

After you run the script (WebSphere Portal is stopped and restarted), we need to update the search portlet properties and specify information about the search server for enterprise search.

Update search portlet properties

Figure A-10 on page 440 through Figure A-21 on page 451 describe the steps to update the enterprise search portlet properties.

After logging in to the WebSphere Portal with the Portal administrator ID (wpsadmin) and password in Figure A-10 on page 440, click **Administration** in the lower left corner, click **Portlet Management** in the navigation area to the left, and then click **Portlets**, as shown in Figure A-11 on page 441. Change the **Search by option to Title contains**, and in the Search field, type enterprise search and then click the **Search** button.

When the search results appeared, we chose to create a copy of the enterprise search portlet as Nile Portlet, as shown in Figure A-12 on page 442 and Figure A-13 on page 443. This step is optional.

Click the wrench icon for the Nile Portlet in Figure A-14 on page 444 to configure the Nile Portlet. In the list of portlet parameters, modify the following parameters:

- Host name (nile.itsosj.sanjose.ibm.com), as shown in Figure A-15 on page 445 and Figure A-16 on page 446.
- Since global security is enabled in WebSphere Application Server on the search server, specify a user name (esadmin) that is valid in the WebSphere Application Server user registry (Tivoli Directory Server), as shown in Figure A-17 on page 447 and Figure A-19 on page 449.
- Password (esadmin) for the user name, as shown in Figure A-19 on page 449 and Figure A-20 on page 450.

The defaults for the other parameters being acceptable, click **OK** to save all the changes made, as shown in Figure A-21 on page 451.

Figure A-21 on page 451 through Figure A-28 on page 458 describe the configuration of the Nile Portlet's page properties and layout.

This completes the installation and configuration of the Sample Search application portlet in WebSphere Portal.

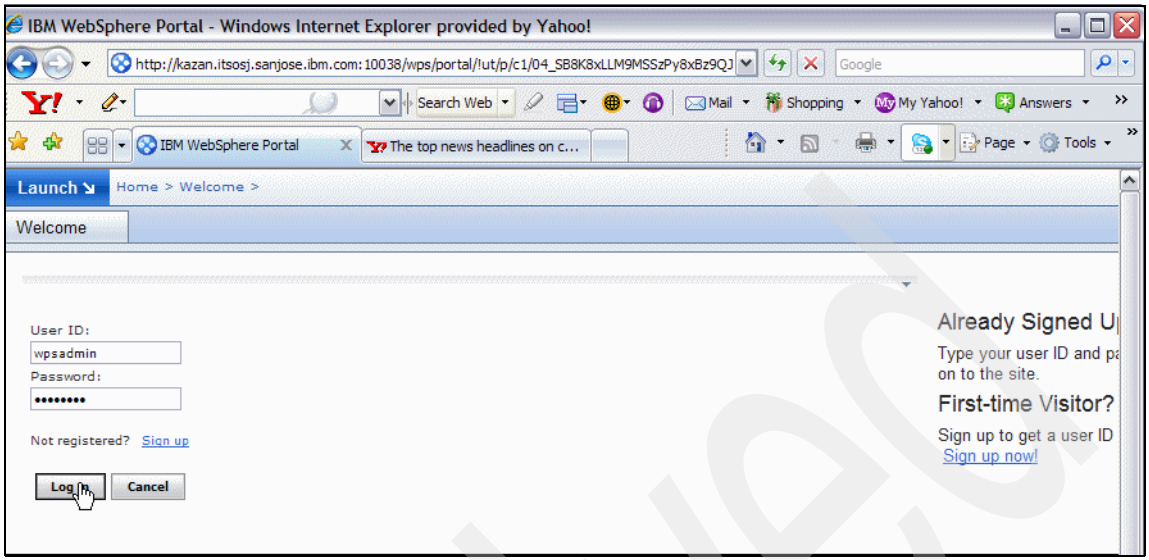

Figure A-10 Log in to WebSphere Portal Server

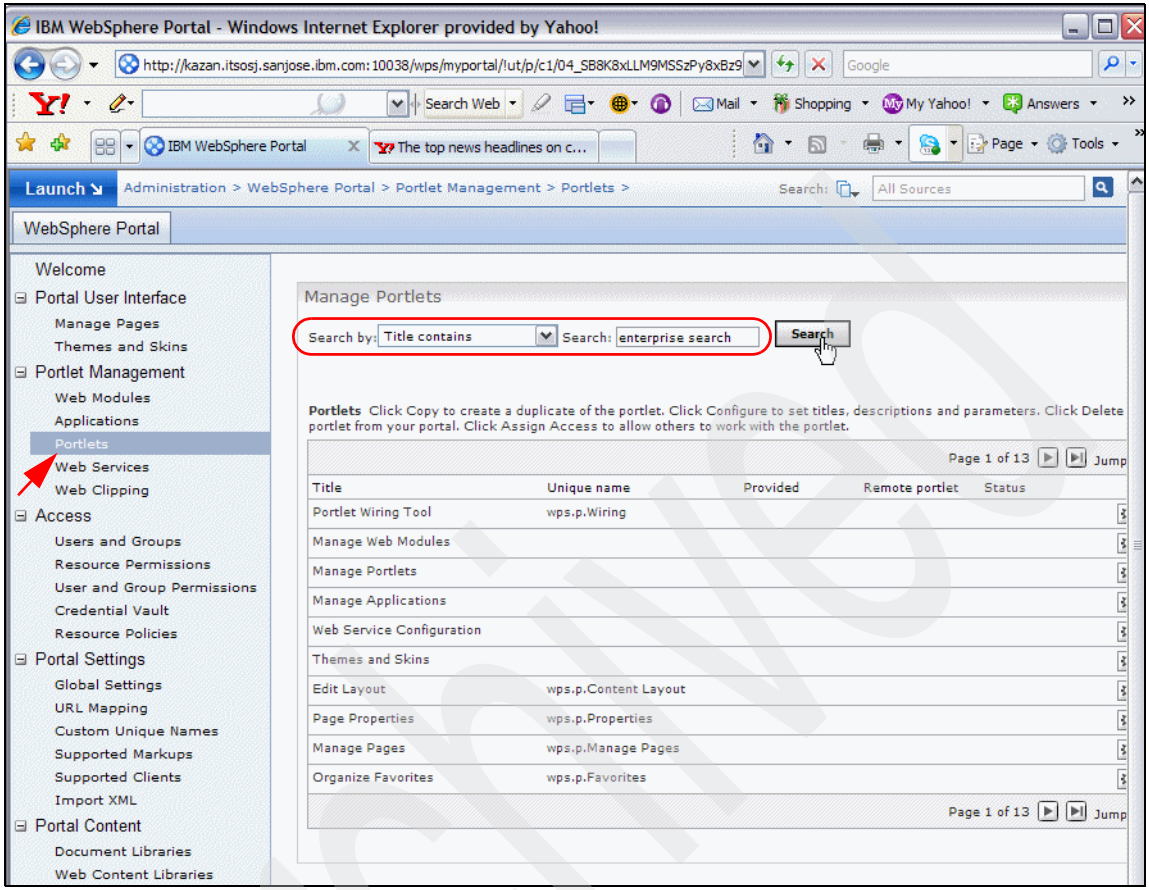

Figure A-11 Search for "enterprise search" in the portlet title

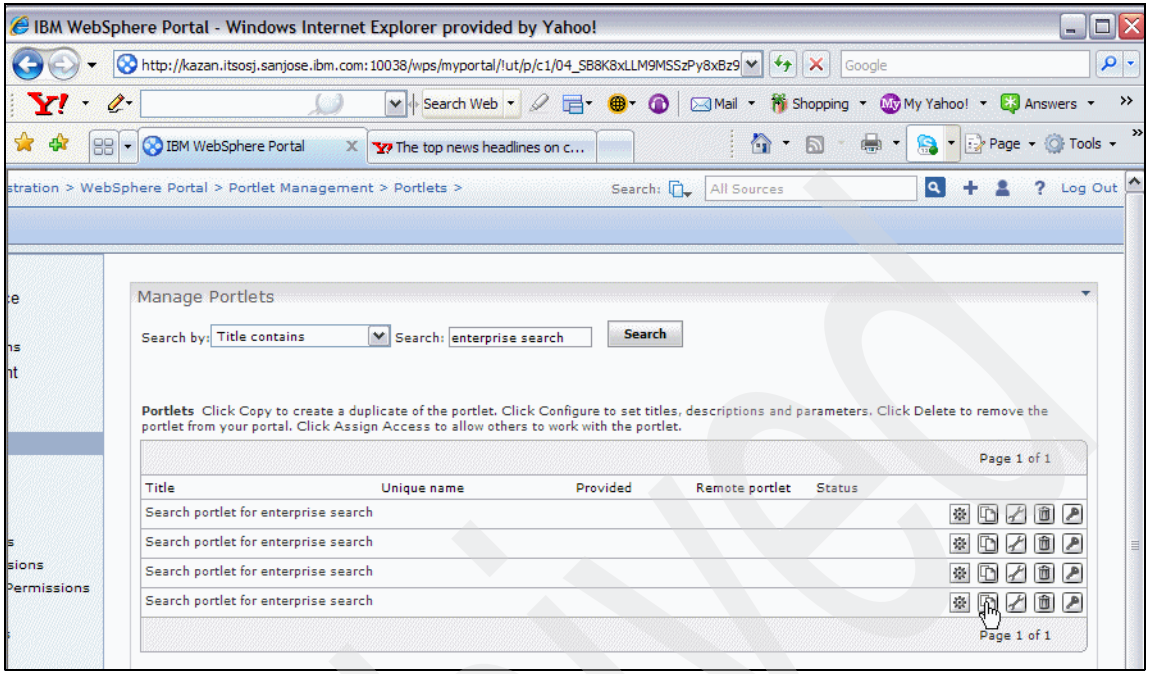

Figure A-12 Make a copy of the enterprise search portlet as Nile Portlet 1/2

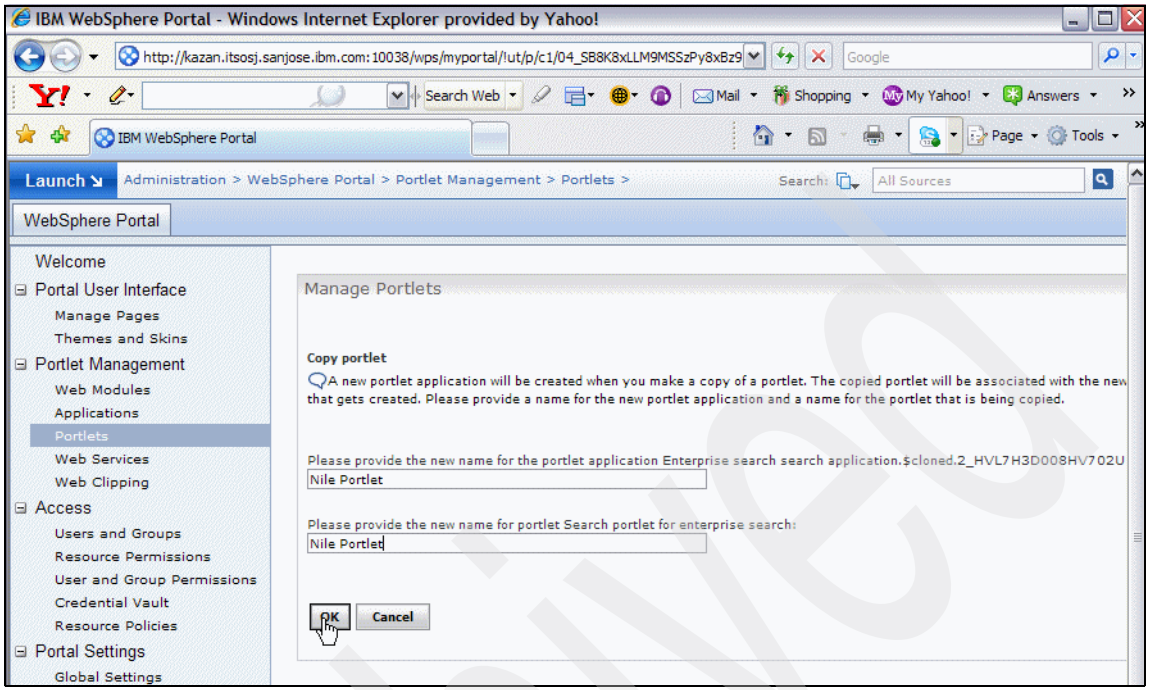

Figure A-13 Make a copy of the enterprise search portlet as Nile Portlet 2/2

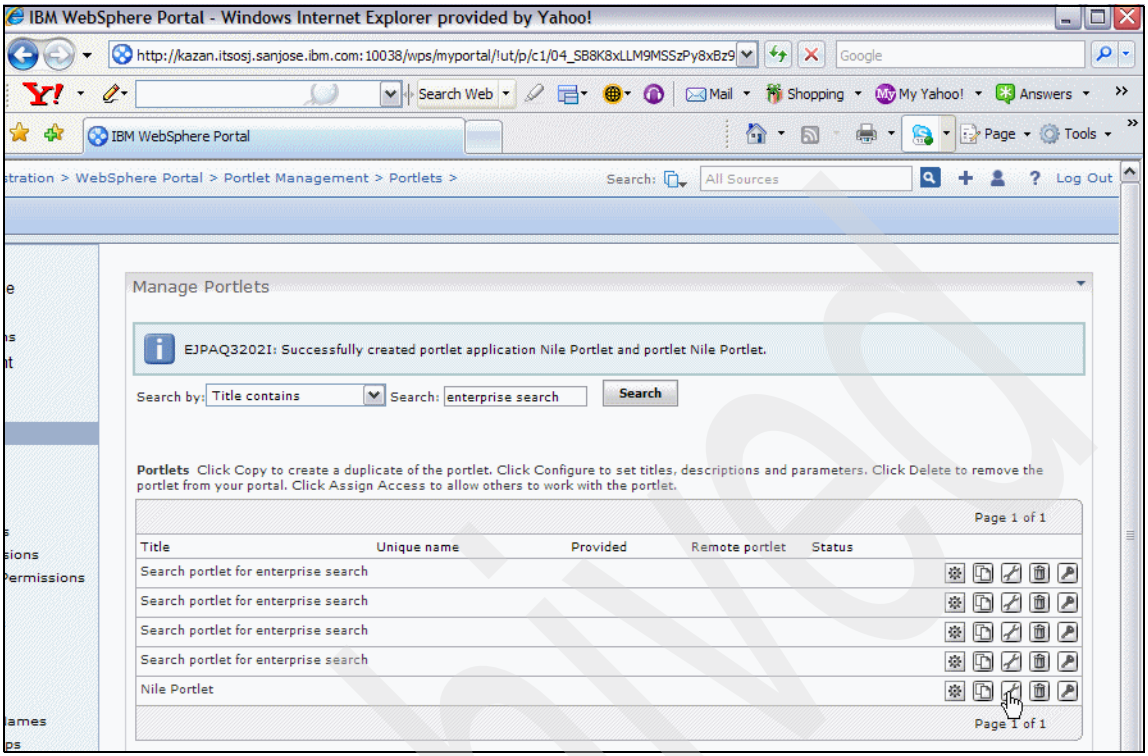

Figure A-14 Configure Nile Portlet 1/15

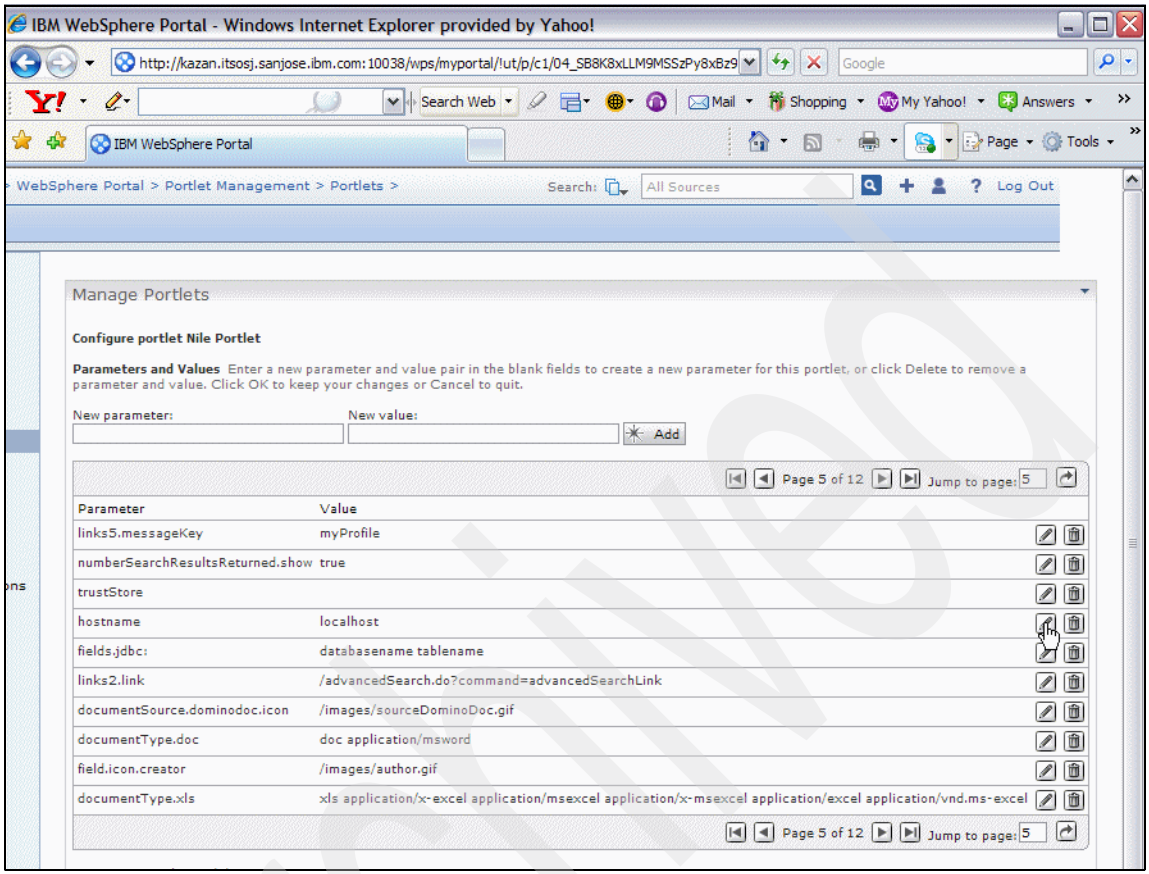

Figure A-15 Configure Nile Portlet 2/15

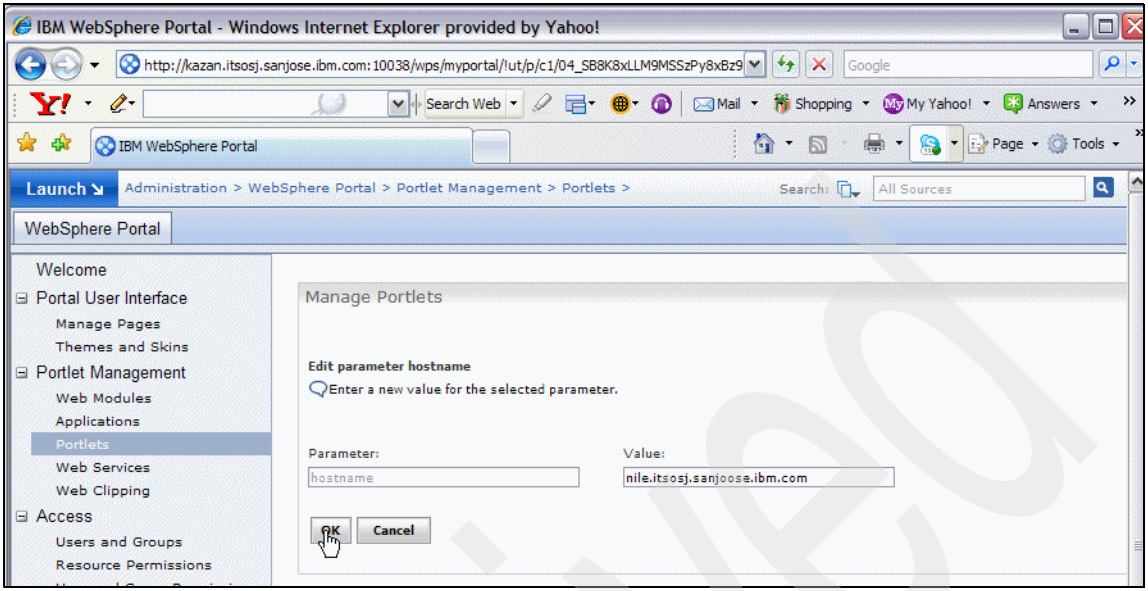

Figure A-16 Configure Nile Portlet 3/15

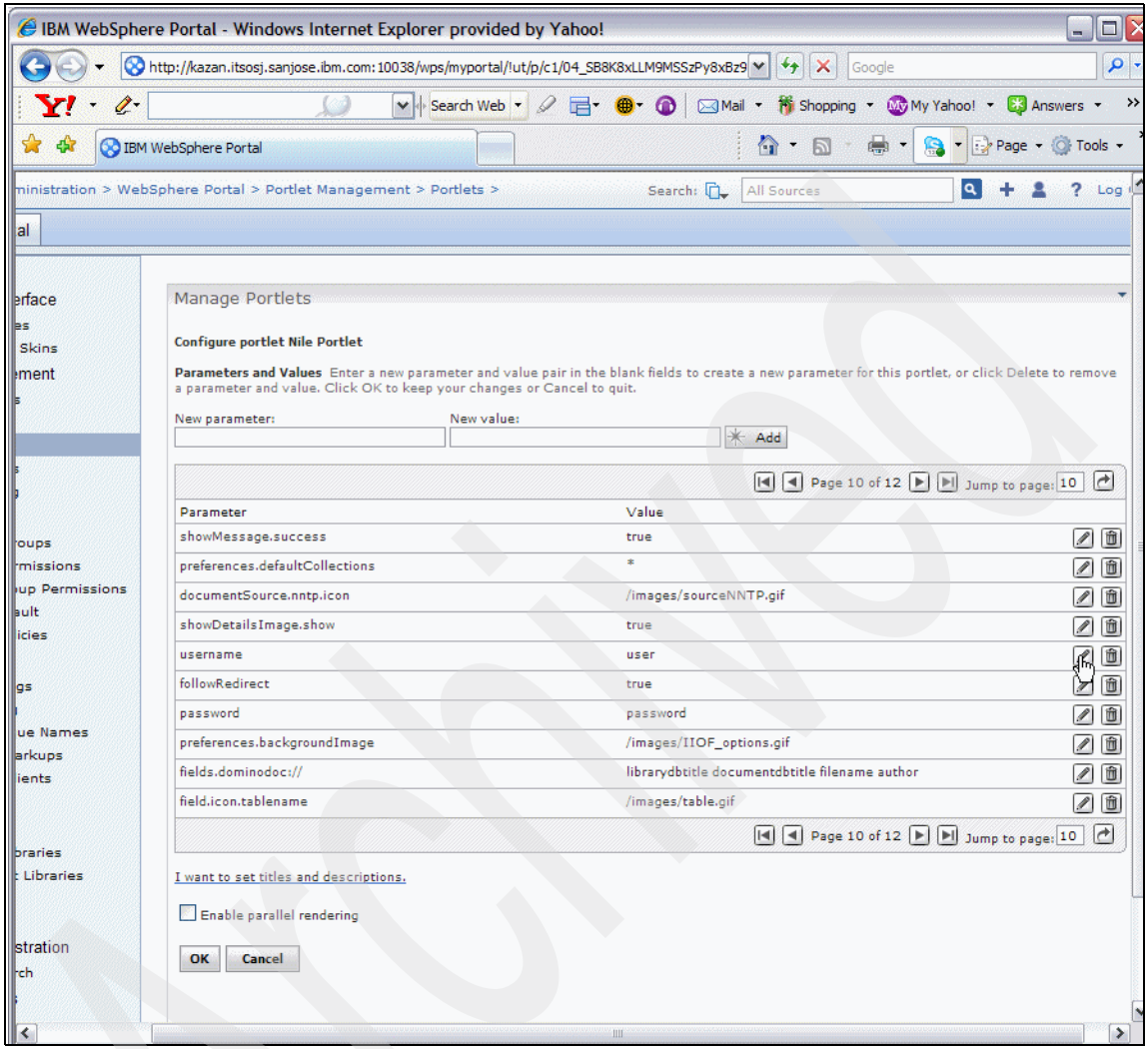

Figure A-17 Configure Nile Portlet 4/15

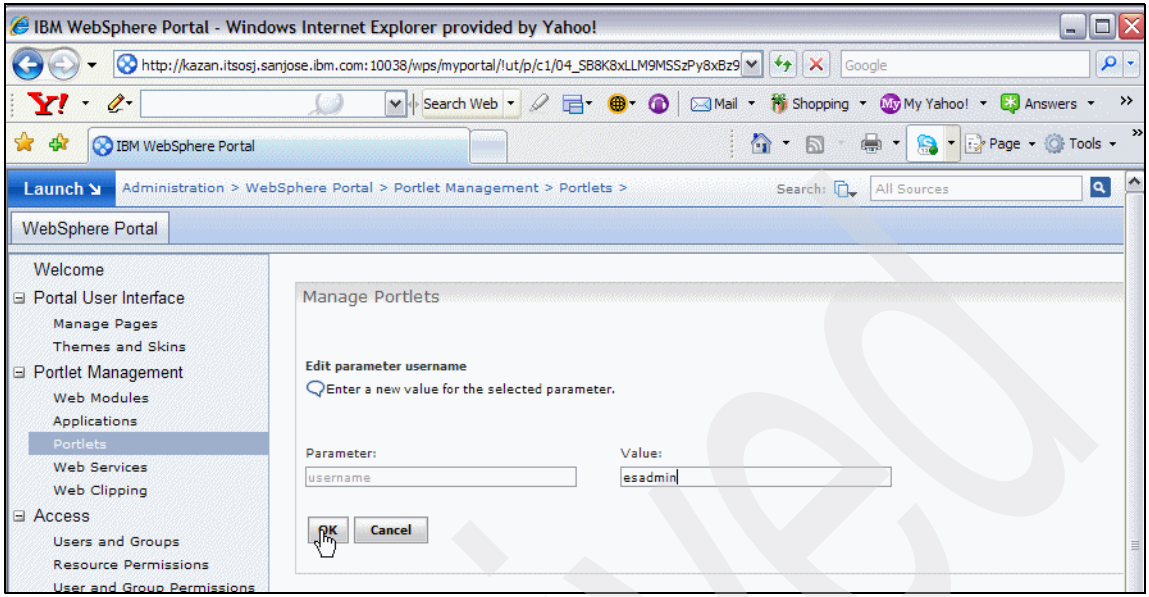

Figure A-18 Configure Nile Portlet 5/15

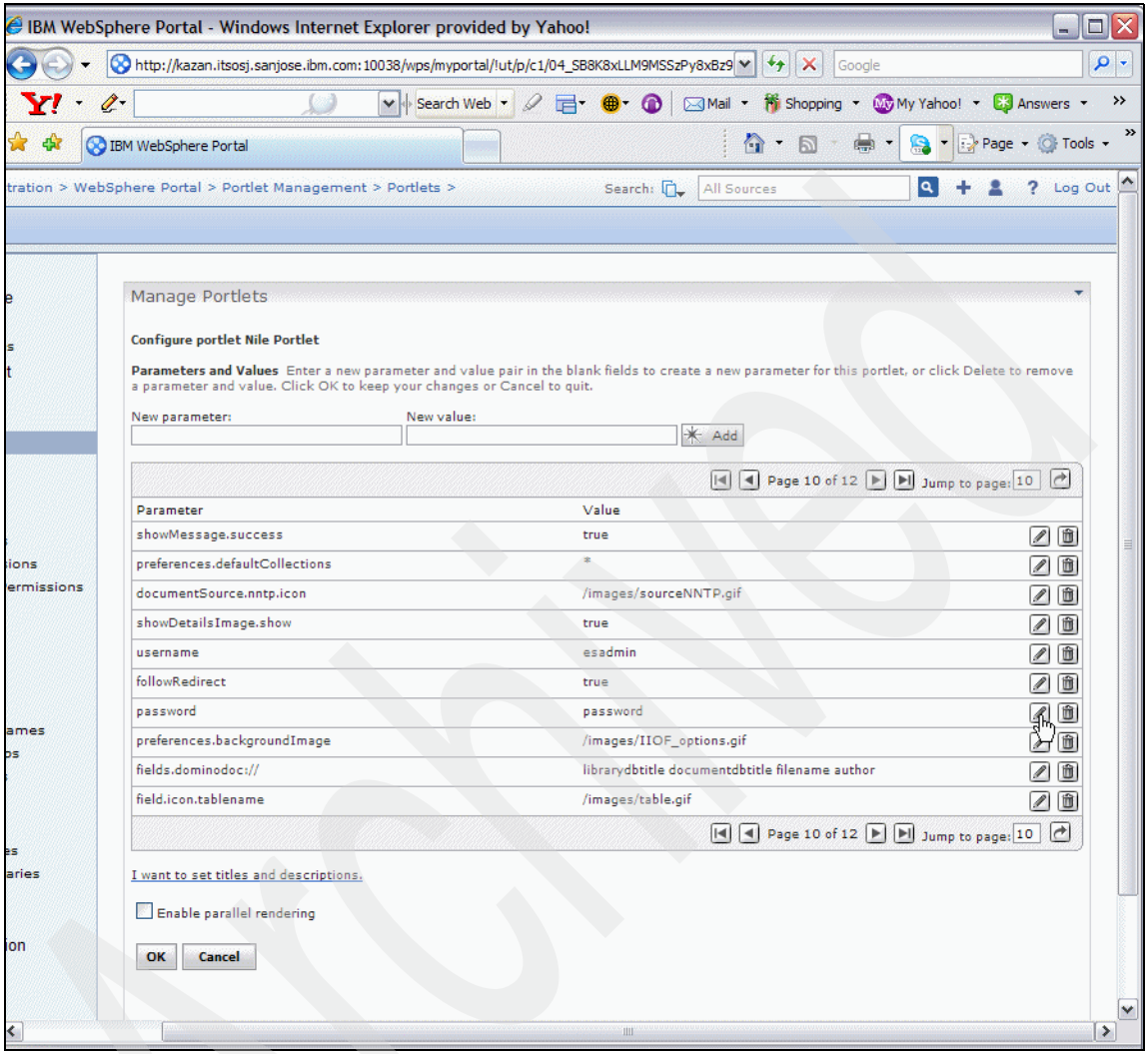

Figure A-19 Configure Nile Portlet 6/15

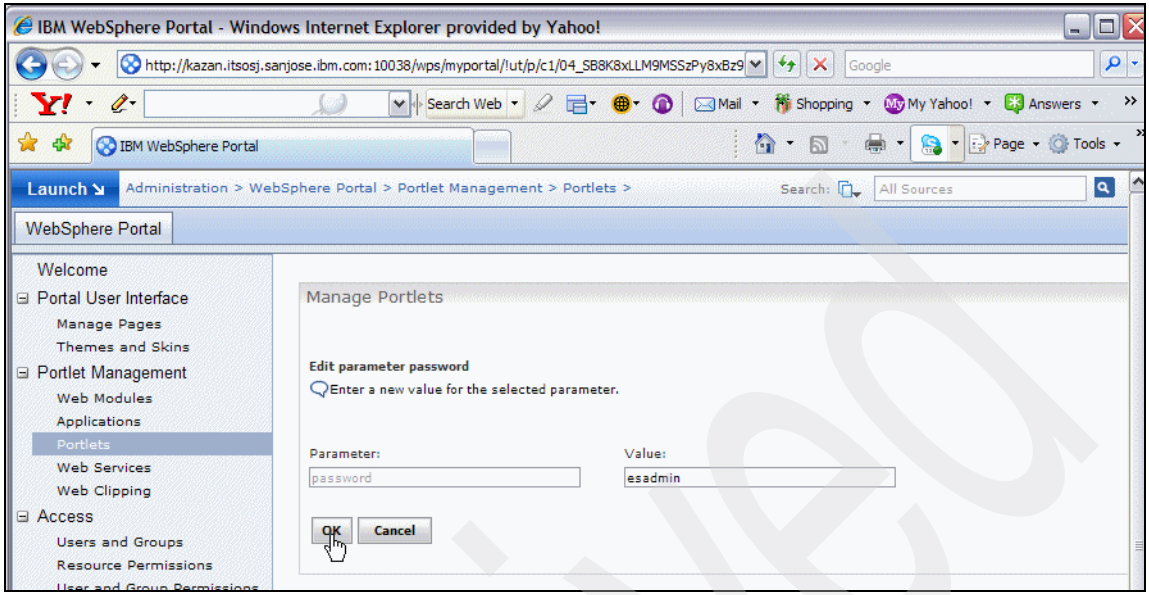

Figure A-20 Configure Nile Portlet 7/15

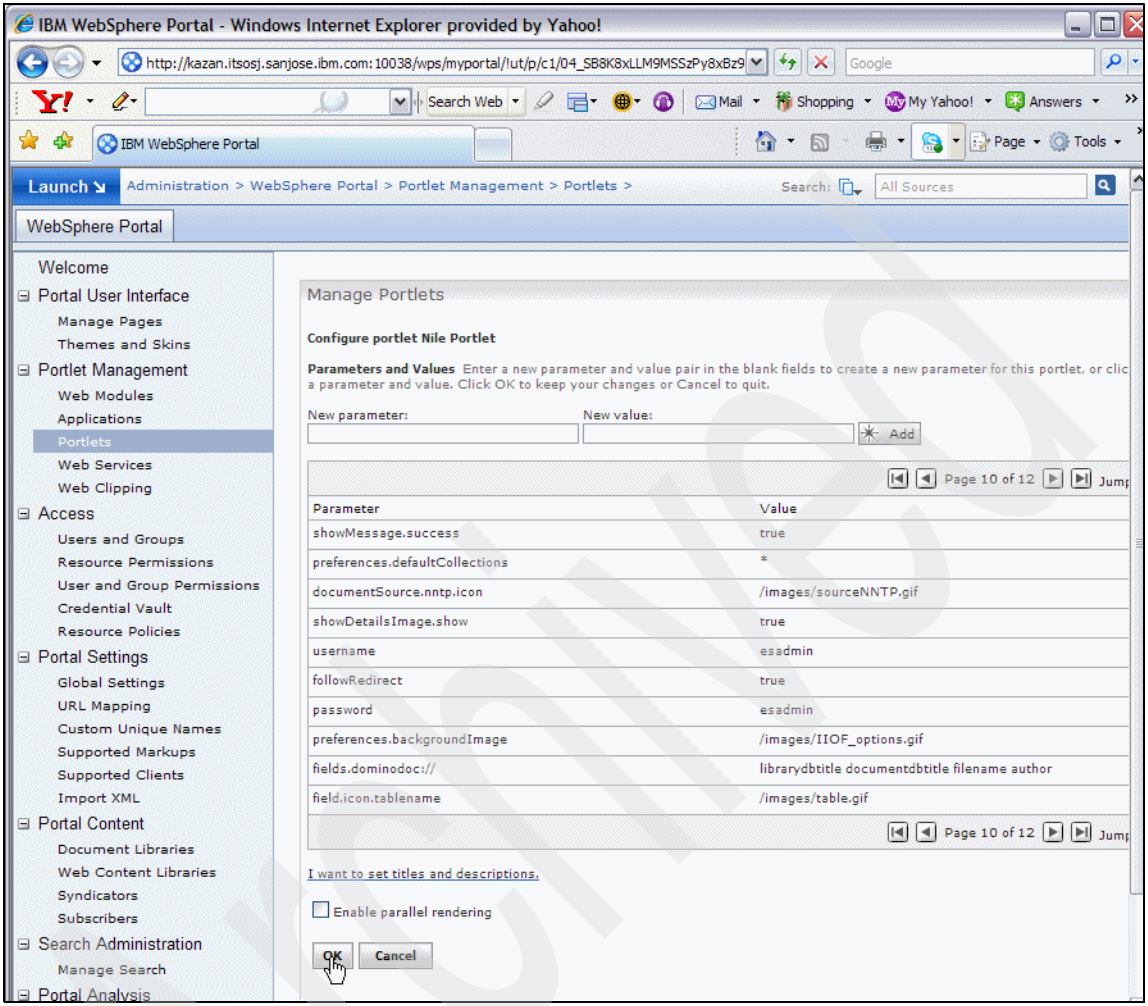

Figure A-21 Configure Nile Portlet 8/15

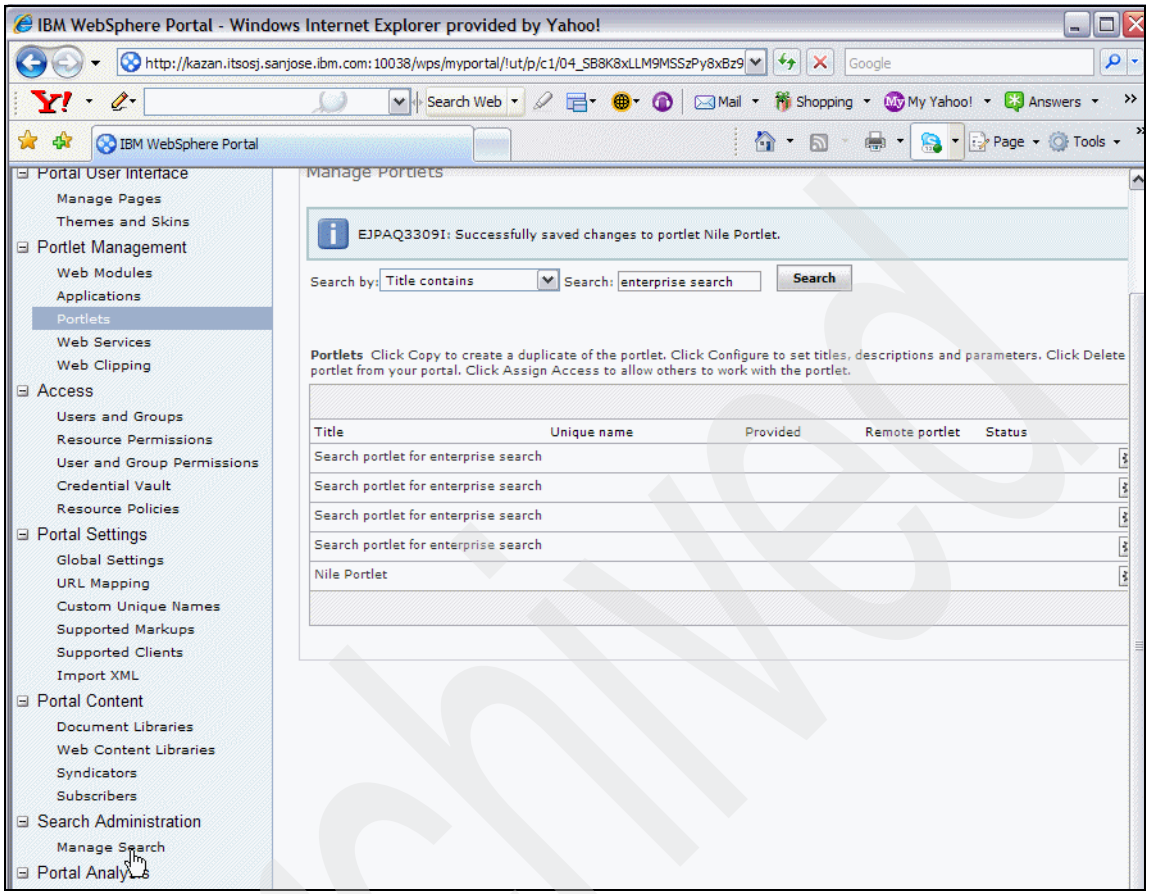

Figure A-22 Configure Nile Portlet 9/15

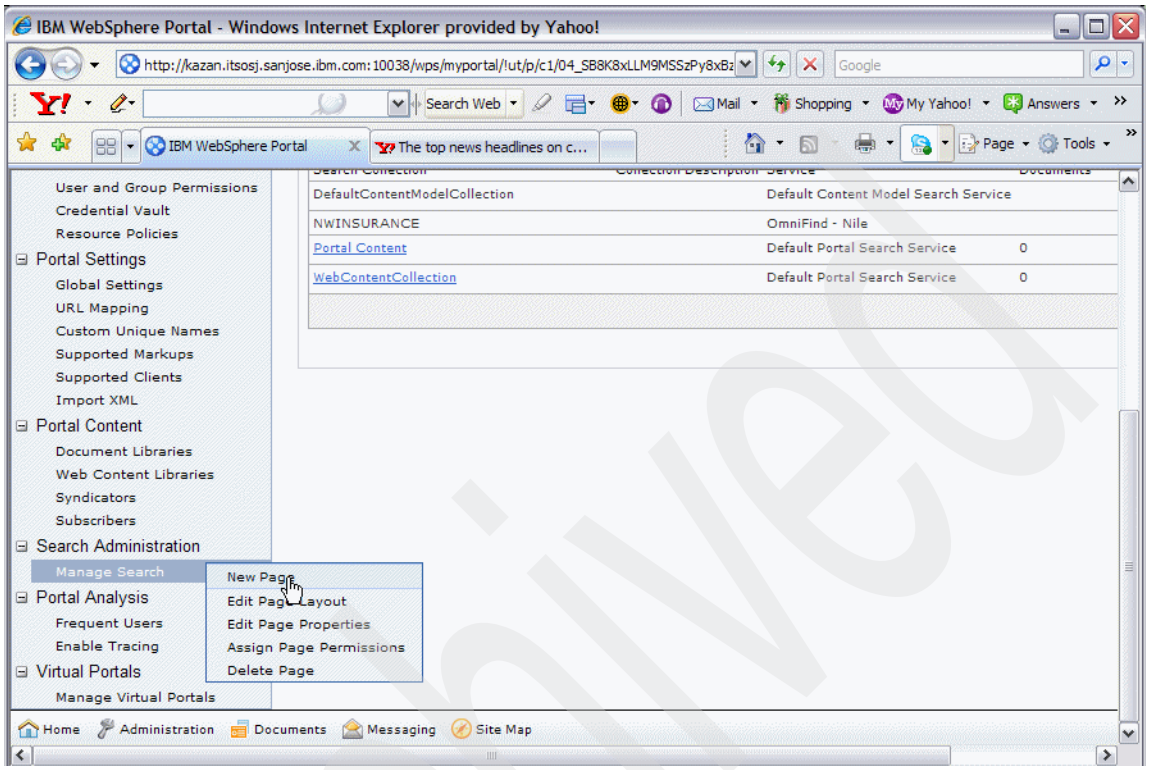

Figure A-23 Configure Nile Portlet 10/15

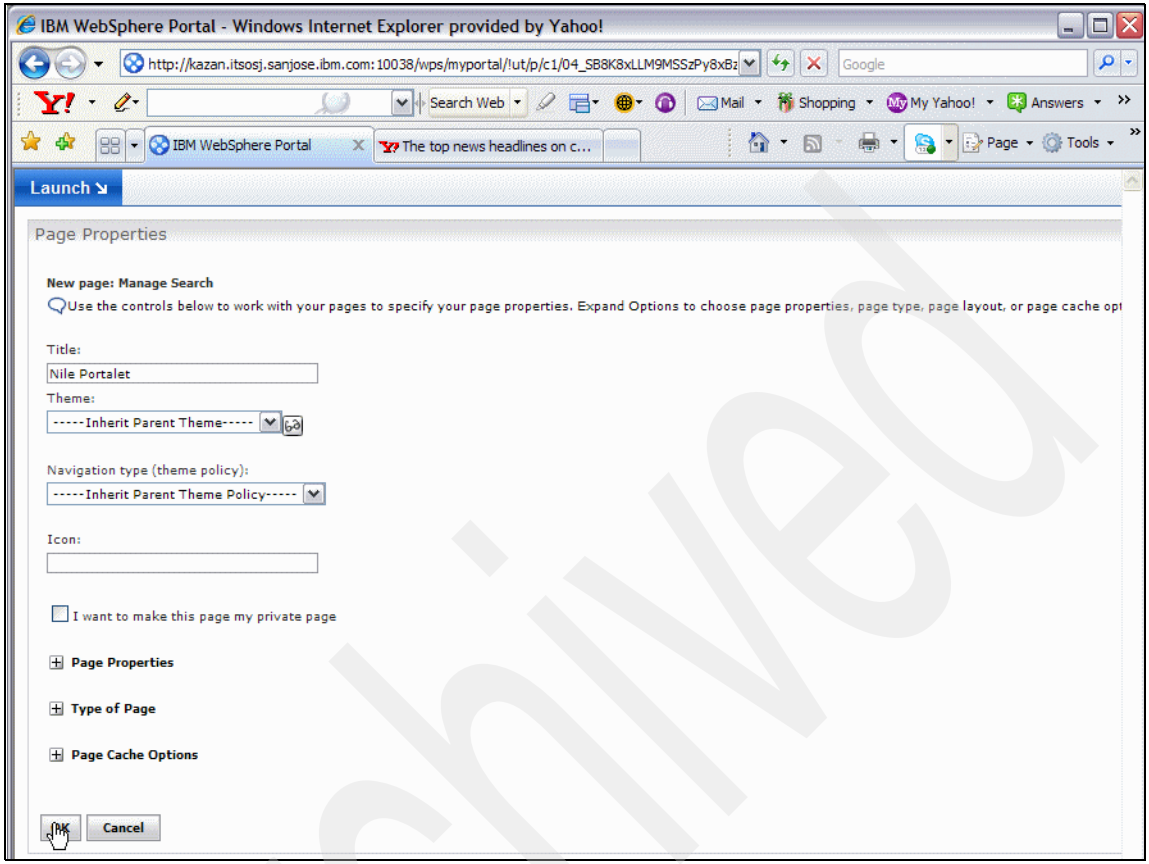

Figure A-24 Configure Nile Portlet 11/15

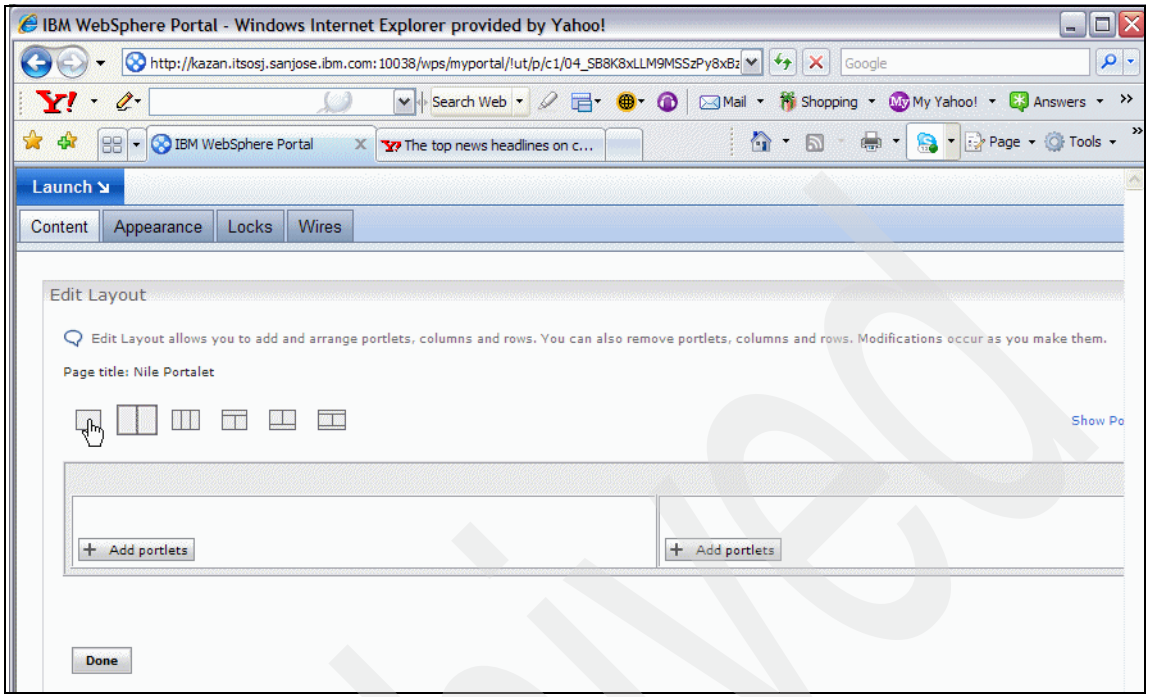

Figure A-25 Configure Nile Portlet 12/15

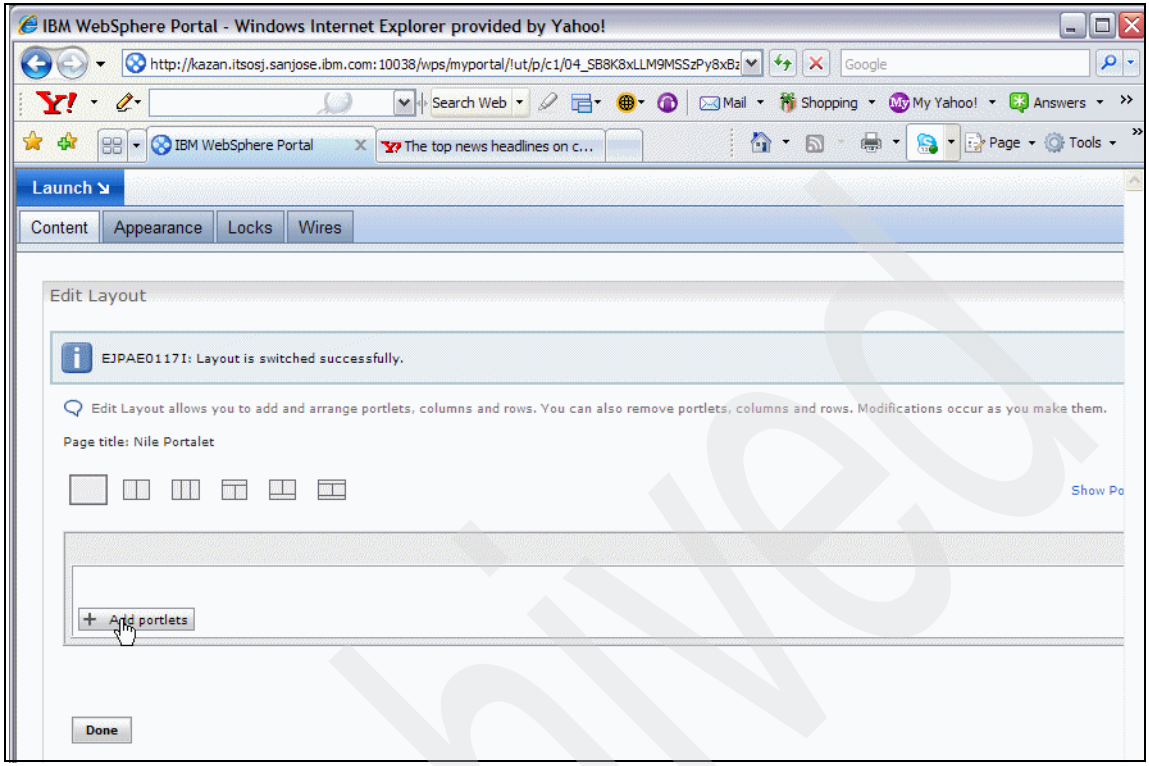

Figure A-26 Configure Nile Portlet 13/15

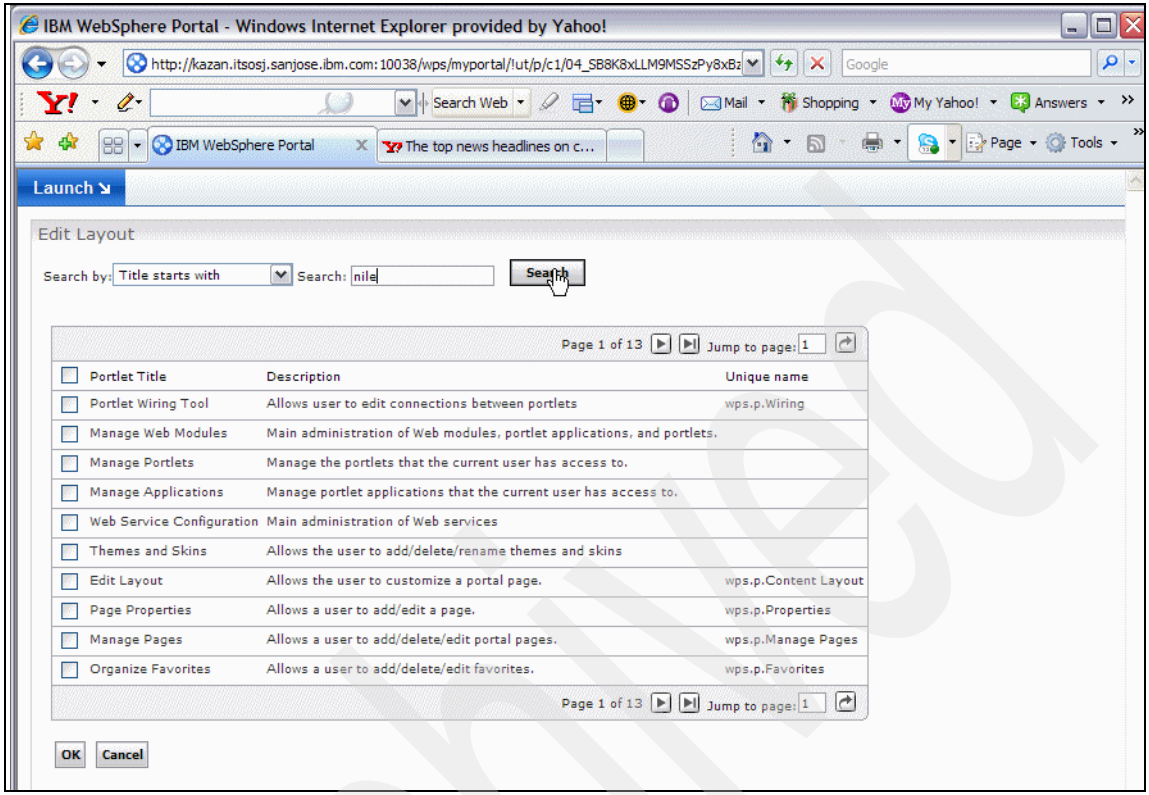

Figure A-27 Configure Nile Portlet 14/15

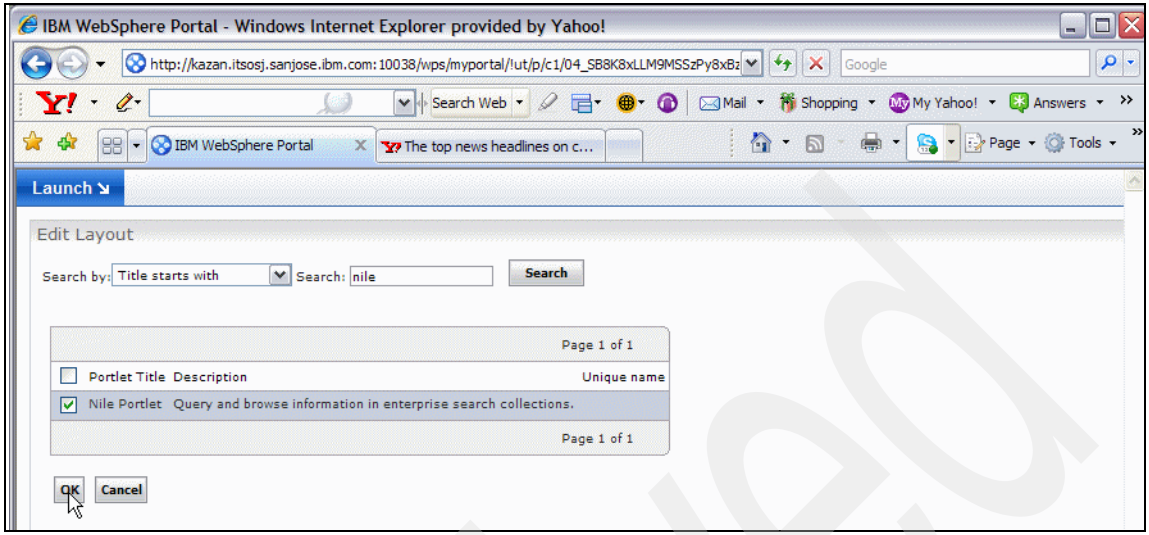

Figure A-28 Configure Nile Portlet 15/15

B

Search Application Customizer

In this appendix, we show an example of using the Search Application Customizer.

Introduction

Prior to OmniFind V8.4, you could edit the configuration file (config.properties in the ES_INSTALL_ROOT/installedApps/ESSearchApplication.ear/ESSearchApplication.war/WEB-INF/ directory) for a search application to specify options for your environment, change the appearance of the application, and control the options that are available to users. The types of properties that could be modified included environment parameters, data source icons, document titles, default values for user preferences, default collections and external sources, extra information for the search results, custom banner and logo, and the custom background image. The ESSearchApplication caches the config.properties file contents when started, and therefore in order for the manual changes to take effect, the contents of the cache must be refreshed with the changes.

For the changes to take effect, you could either:

- Stop and start the ESSearchApplication through the WebSphere Application Server admin console (if WebSphere Application Server Network Deployment (ND) is installed), as shown in Figure B-1, or using WebSphere Application Server commands.

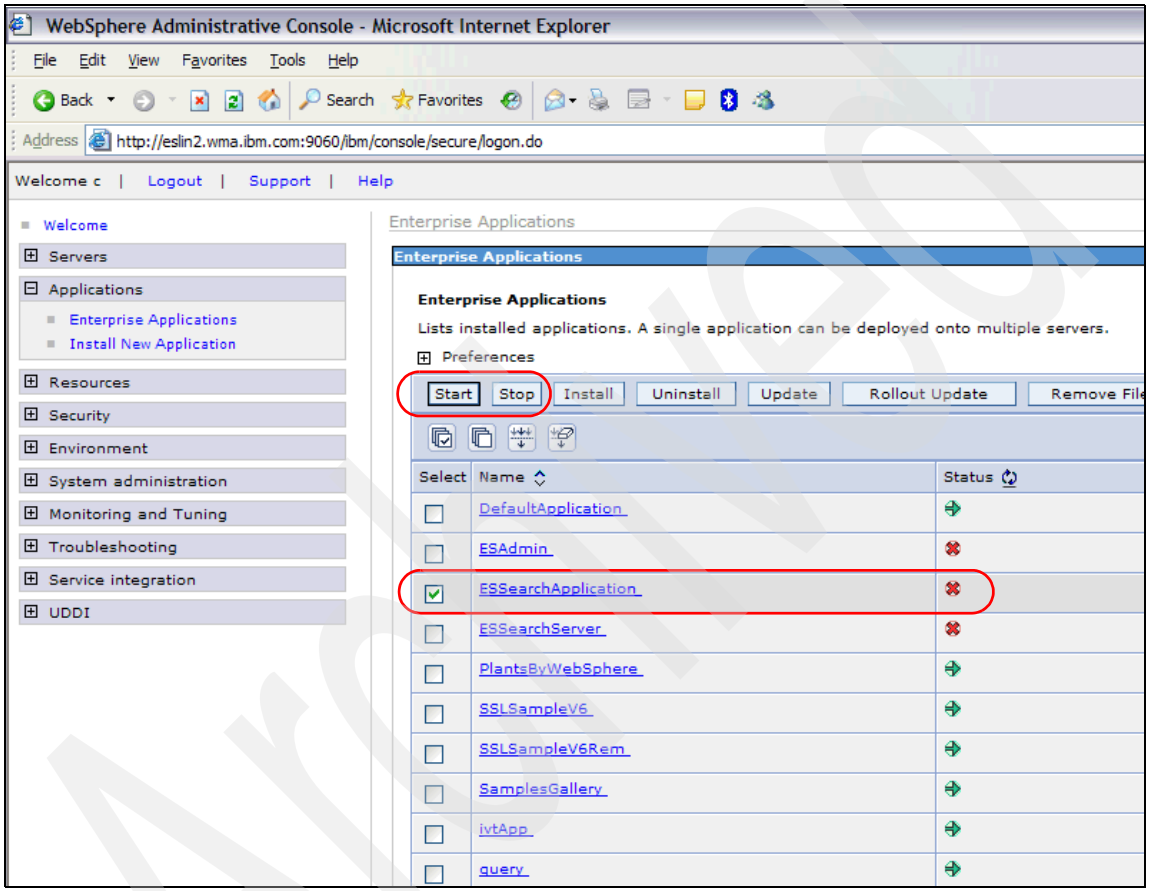

Figure B-1 WebSphere Administrative Console - Enterprise Applications

- Hit the **Refresh** button in the Search Application user interface, as shown in Figure B-25 on page 482.

In OmniFind Enterprise Edition V8.4, you can also edit properties by using the Search Application Customizer, which is a graphical user interface that enables you to see the effects of your changes as you make them. When you are satisfied with the options that you specify for searching collections and viewing search results, you can save the options to update the configuration file for the search application. However, unlike OmniFind V8.3, you do not need to stop and start the ESSearchApplication for the changes to take effect.

In OmniFind Enterprise Edition V8.4, if you choose to edit the config.properties manually instead of using the Search Application Customizer, you need to refresh the contents of the cache, just as you did in OmniFind V8.3 by recycling the ESSearchApplication or hitting the **Refresh** button in the Search Application user interface.

Attention: In all the following sections, for the purposes of avoiding screen capture overload, we have *not* included all the windows that you would typically navigate through in order to perform the desired function. Instead, we have focused on including select screen captures (and in some cases portions of selected screen captures) that highlight the key items of interest, thereby skipping both initial as well as intervening screen captures in the process.

Search Application Customizer

As mentioned earlier, the Search Application Customizer enables you to visualize changes that you want to make and to modify a search application without editing the configuration file.

Restriction: The Search Application Customizer is available as a stand-alone application. You cannot launch the Search Application Customizer as a portlet within WebSphere Portal.

Figure B-2 on page 463 shows the main steps to customize a search application using Search Application Customizer. In the following sections, we describe a very simple customization of the sample search application following these steps.

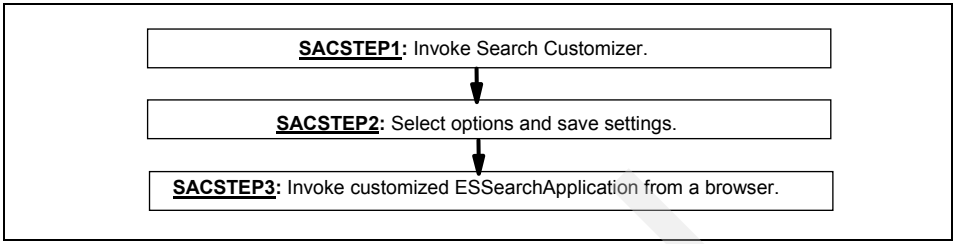

Figure B-2 Steps to customize a search application using Search Application Customizer

SACSTEP1: Invoke Search Customizer

The Search Customizer may be invoked from a Web browser or from the GUI admin console.

- Invoke from a Web browser (not shown here)

  Type the URL for the Search Application Customizer in a Web browser. For example, `http://SearchServer.com/ESSearchApplication/palette.do`, where SearchServer.com is the host name of the search server. If your Web server is not configured to use port 80, you also need to specify the correct port number. For example, `http://SearchServer.com:9080/ESSearchApplication/palette.do`.

  To customize a custom search application, type the URL for the Search Application Customizer, and append the name of the configuration file for your search application. For example, `http://SearchServer.com/ESSearchApplication/palette.do?configFile=/WEB-INF/myConfig.properties`.

  If the file that you specify (myConfig.properties file, in this case) does not exist, values in the config.properties file for the sample search application are displayed.

  Tip: You can also specify the configuration file that you want to use with a search application by clicking **Load** after you start the Search Application Customizer and specifying the name of the file.

  If global security is enabled in WebSphere Application Server, you need to log in with a valid user ID and password.

- Invoke from GUI admin console

  This is the option shown here.

Figure B-3 through Figure B-12 on page 470 show the invocation of the Search Customizer and the various configuration properties available for modification.

After logging in to the admin console, click the **Search Customizer** link to invoke it, as shown in Figure B-3.

Figure B-4 on page 465 shows the various options available. It shows the configuration properties file and the various categories of properties that can be modified, such as Server settings, Screen navigation, Messages, Query options, Results, Images, and Theme. Each of these categories are expanded to show the default parameter values (Figure B-5 on page 465 through Figure B-12 on page 470) and the search application display interface in the right hand pane that reflect these default settings.

Tip: Use the **Help for the customizer** link as needed for an explanation of the options available.

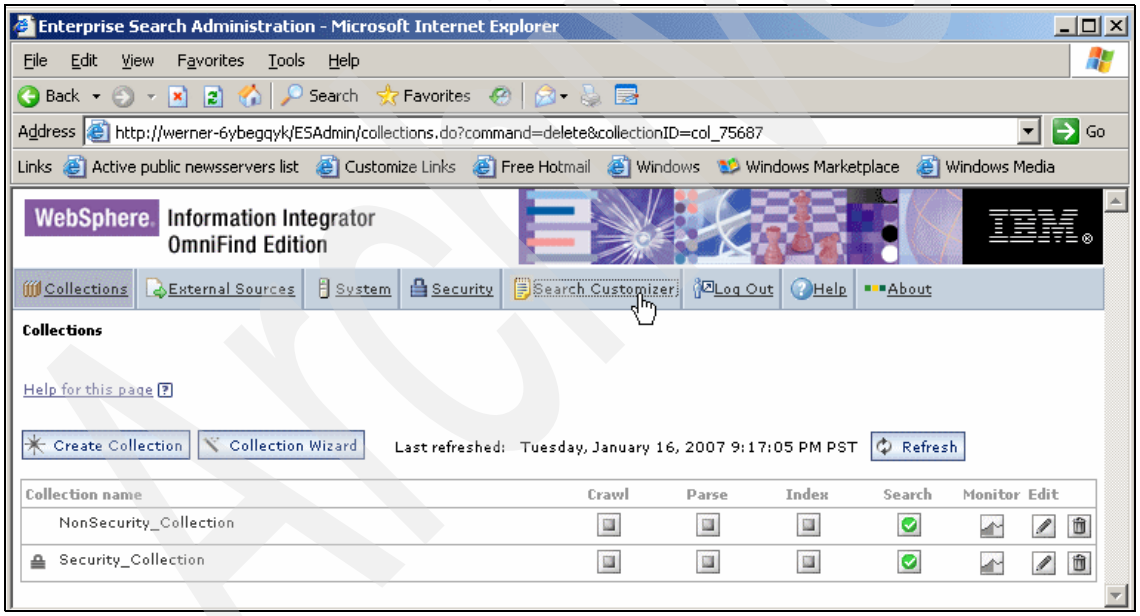

Figure B-3 Invoke Search Customizer

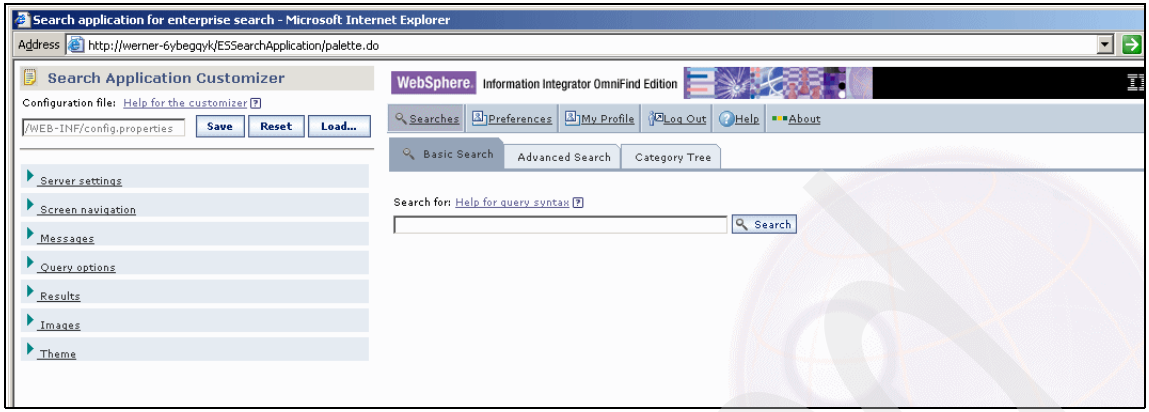

Figure B-4 Customizable properties in the config.properties file

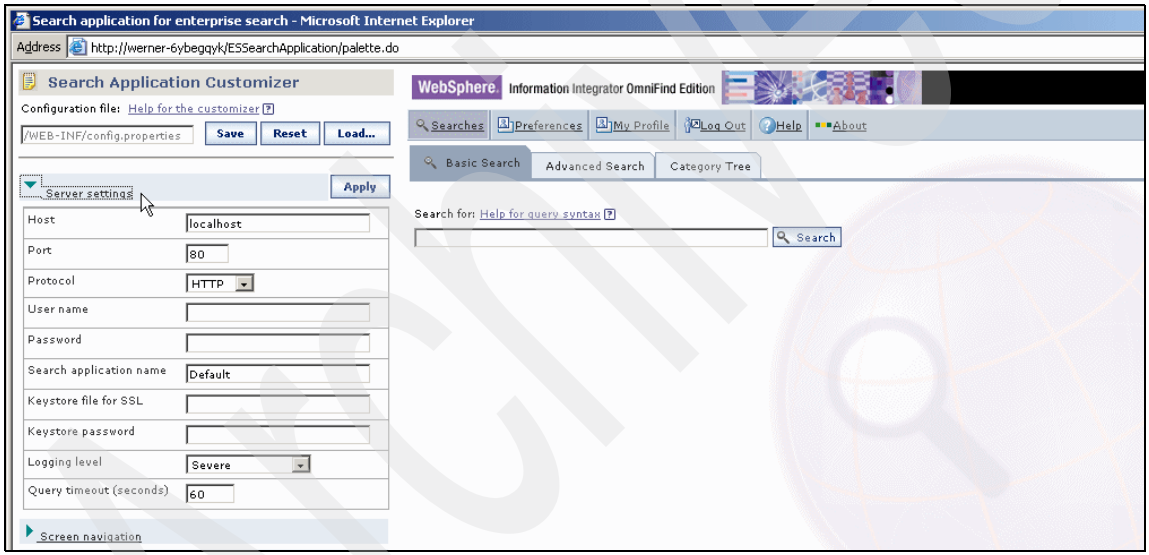

Figure B-5 Server settings parameters

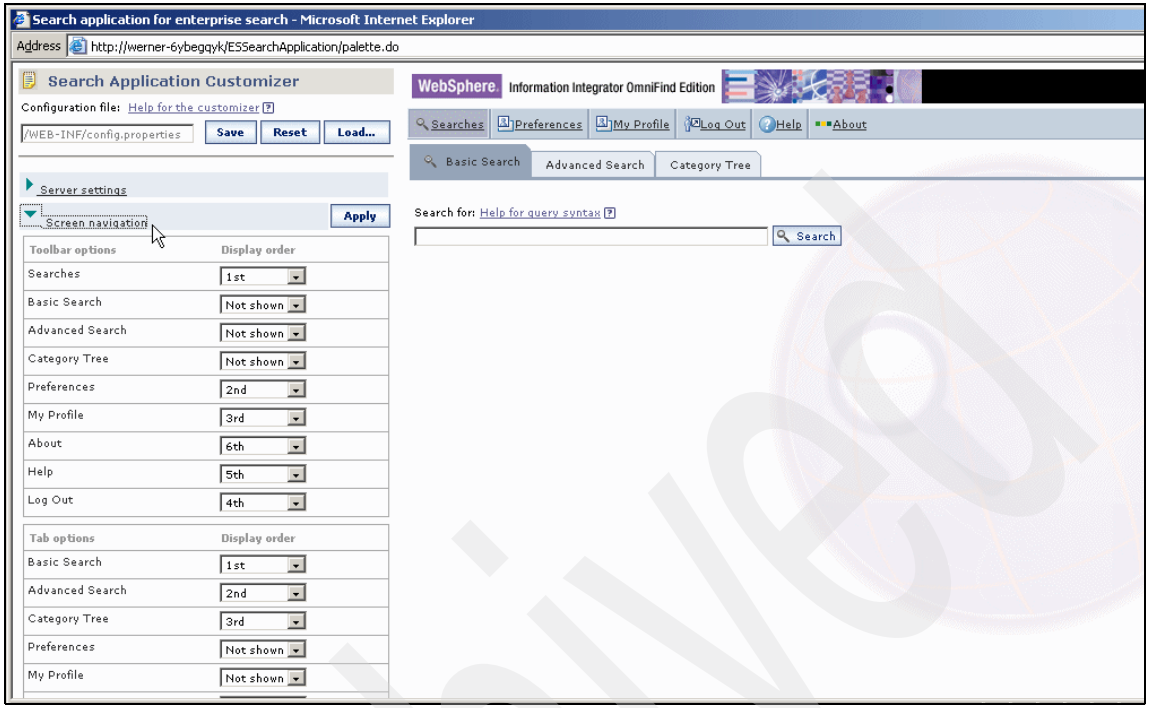

Figure B-6 Screen navigation parameters

Figure B-7 Messages parameters

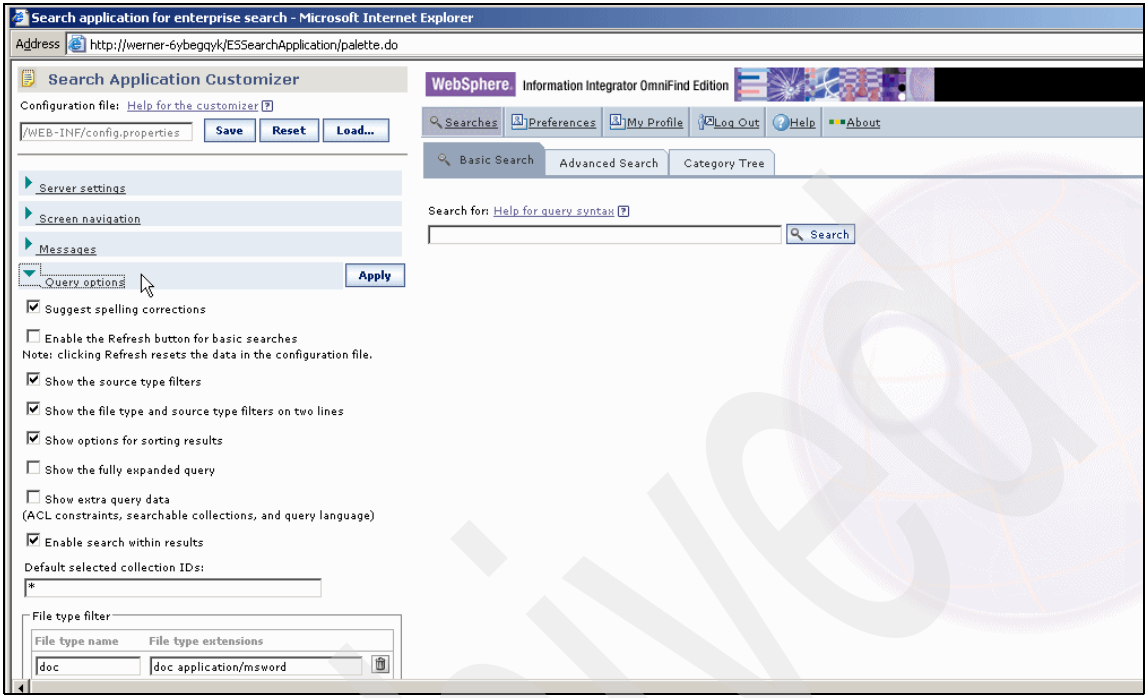

Figure B-8 Query options parameters

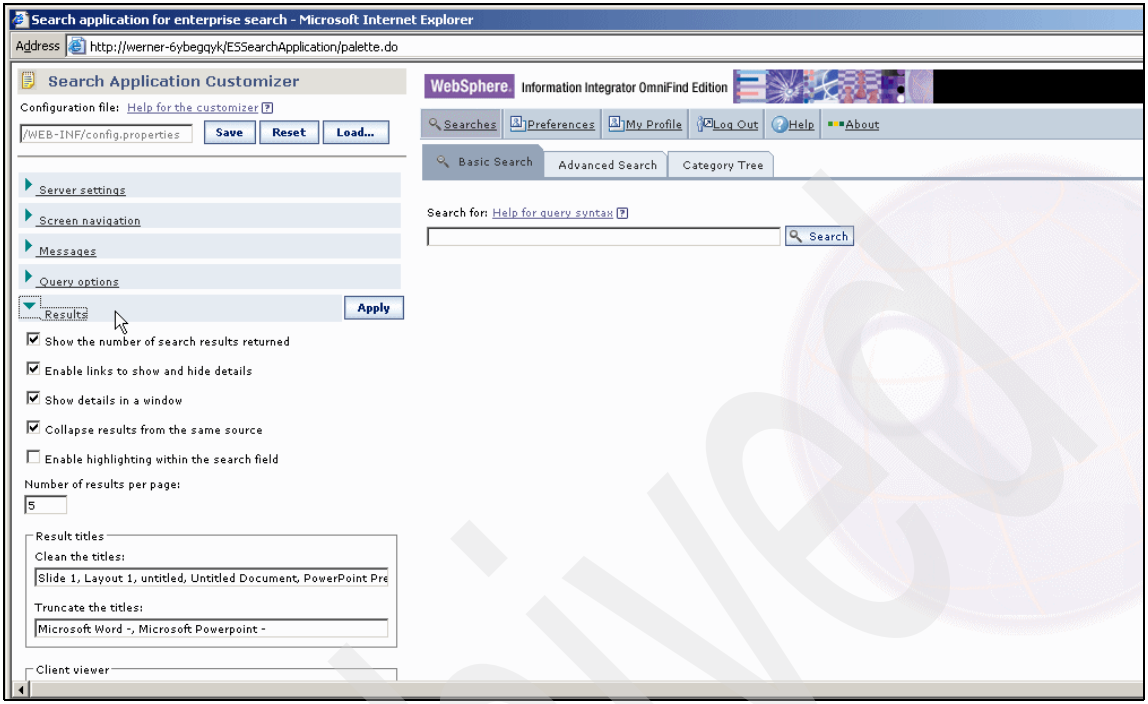

Figure B-9 Results parameters

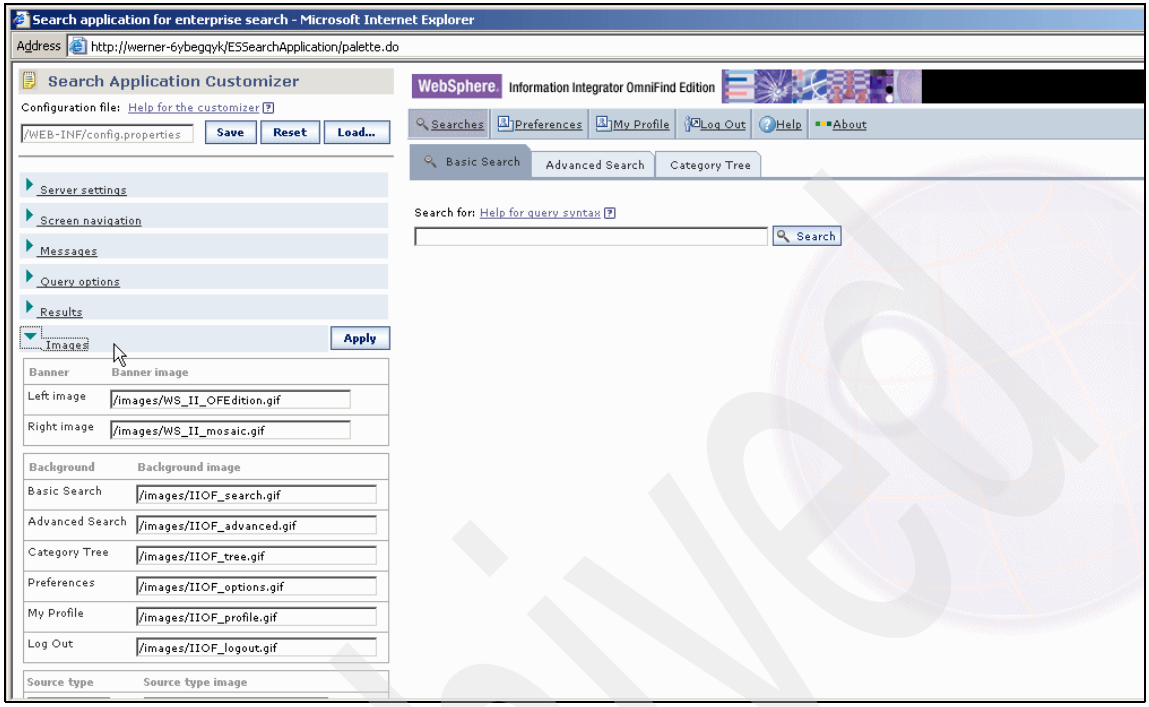

Figure B-10 Images parameters

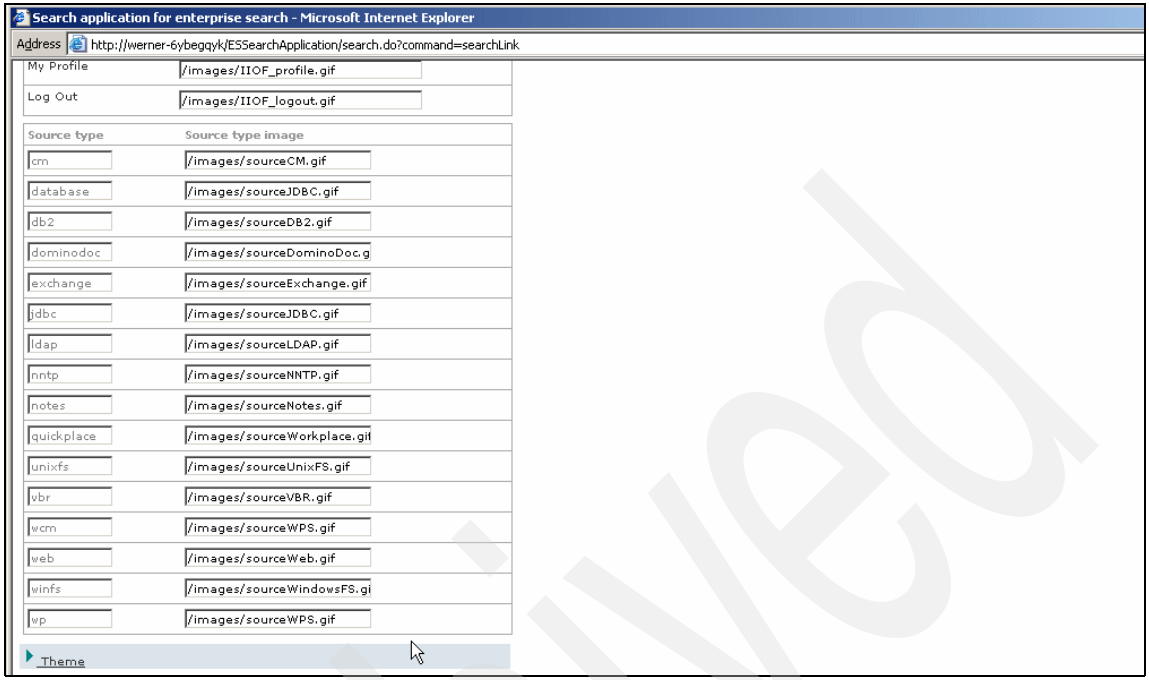

Figure B-11 Other images that can be changed in Images

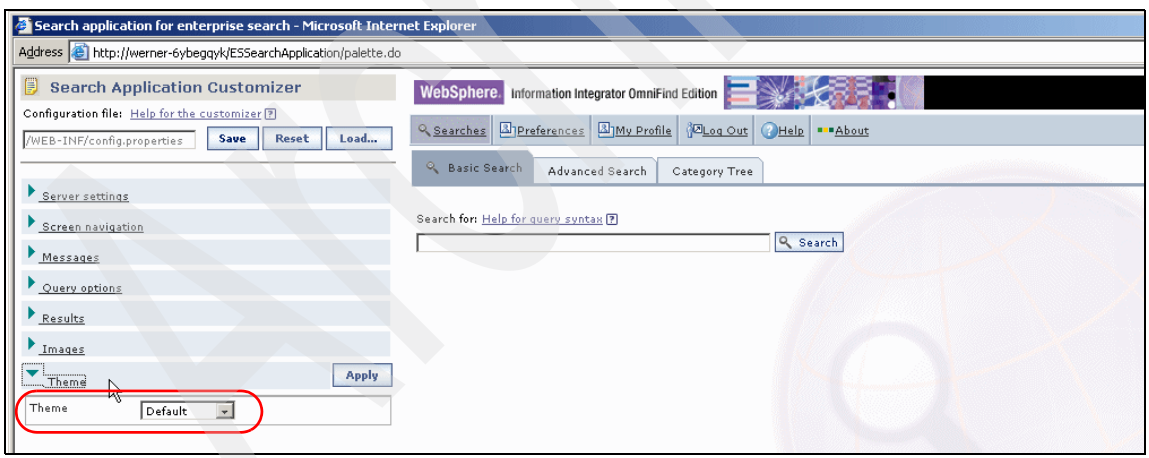

Figure B-12 Default theme

SACSTEP2: Select options and save settings

In this step, we display and modify some of the parameters for each of the categories, and show the corresponding effect in the right hand pane.

Note: To see the effect of some changes, such as how search results are presented, type a query and click **Search**. We do not show this here.

After we were satisfied with our selections, we click **Save** to update the configuration file (not shown here).

If you click **Reset**, the options displayed in the Search Application Customizer are restored to values in the last saved version of the configuration file.

Figure B-13 on page 472 shows the selection of **Science** from the Theme drop-down list of the Theme category to view the pale blue theme in the window.

Figure B-14 on page 473 shows the selection of an appropriate image for the Right image field in the Images category to view the new image in the right pane as highlighted. Figure B-15 on page 474 shows the selection of an appropriate image for the Left image field as well in the Images category to view the new images in the right pane as highlighted.

Figure B-16 on page 475 and Figure B-17 on page 476 show the parameters that can be modified in the Results category, including such items as the Number of results per page, Enable highlighting within the search field, and Show the number of search results returned. You will need to type in a search query and click **Search** to view the effect of these changes (not shown here).

Figure B-18 on page 477 and Figure B-19 on page 478 show the parameters that can be modified in the Query options category, including such items as Suggest spelling corrections, Enable search within results, and Default selected collection IDs. Here again, you will need to type in a search query and click **Search** to view the effect of these changes (not shown here).

Figure B-20 on page 479 through Figure B-22 on page 480 show the Screen navigation options, such as Toolbar options, Tab options, and Link options. The changes made to the default window navigation (Figure B-12 on page 470) is as follows:

- For Toolbar options, remove Preferences and About by selecting **Not shown** from the drop-down list, and move **Help** to third place in the drop-down list. The new navigation window for Toolbars is shown in Figure B-20 on page 479.

- For Tab options, remove Advanced Search by selecting **Not shown** from the drop-down list, and move Preferences to second place in the drop-down list (Figure B-21 on page 479). The new navigation window for Tabs is shown shown in Figure B-20 on page 479.
- For Link options, add Advanced Search by selecting first from the drop-down list in Figure B-21 on page 479. The new navigation window for Links is shown shown in Figure B-22 on page 480.

Figure B-23 on page 480 shows the parameters in the Server settings category, including such items as Search application name, and Host and Logging level. The effect of this is not seen until you issue a query (not shown here). Click **Save** to write the changes to the configuration file, as shown in Figure B-24 on page 481.

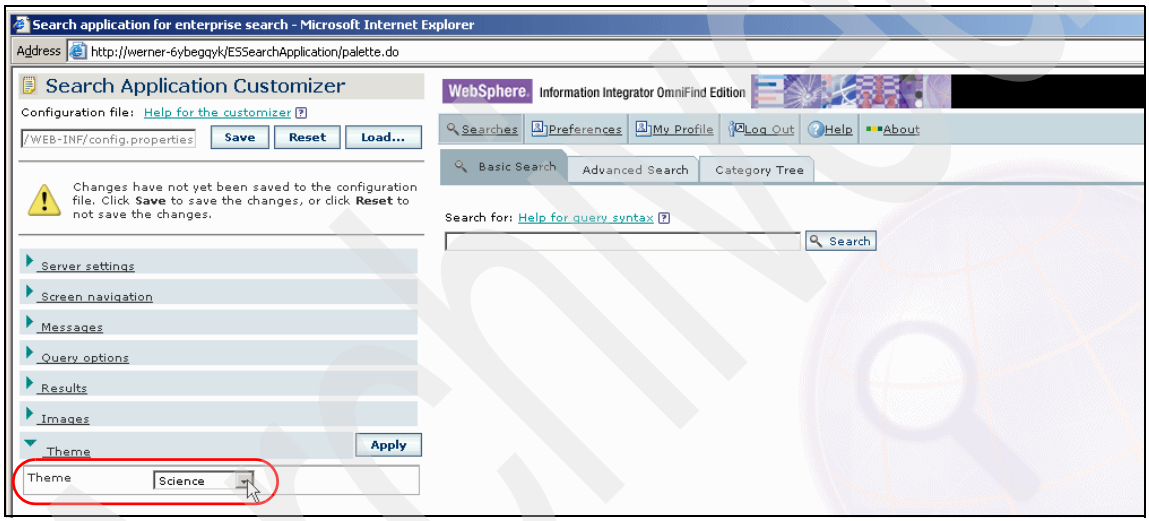

Figure B-13 Science Theme in Theme

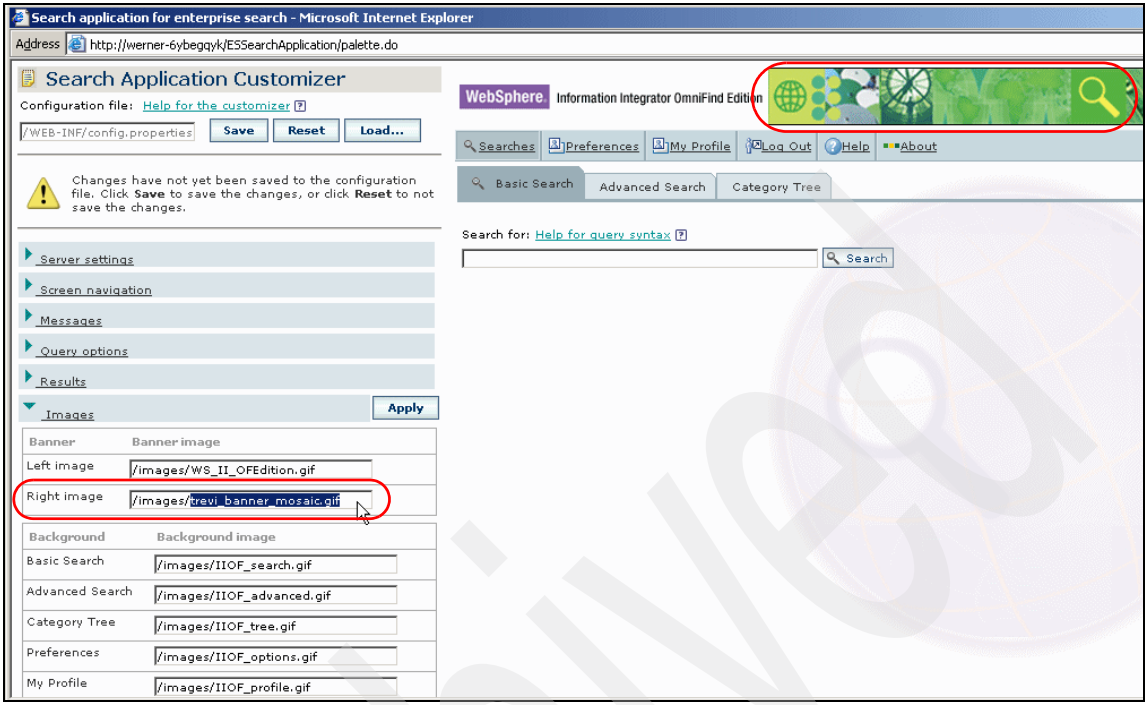

Figure B-14 Right image in Images

Figure B-15 Left image and Right image in Images

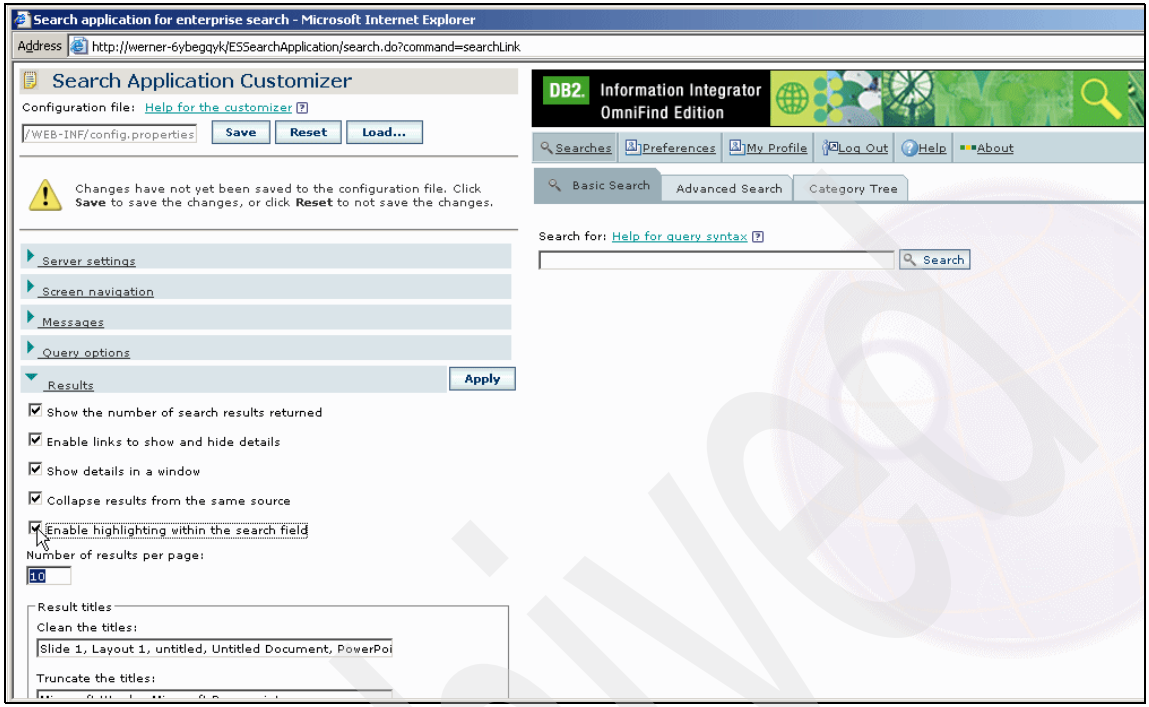

Figure B-16 Results parameters 1/2

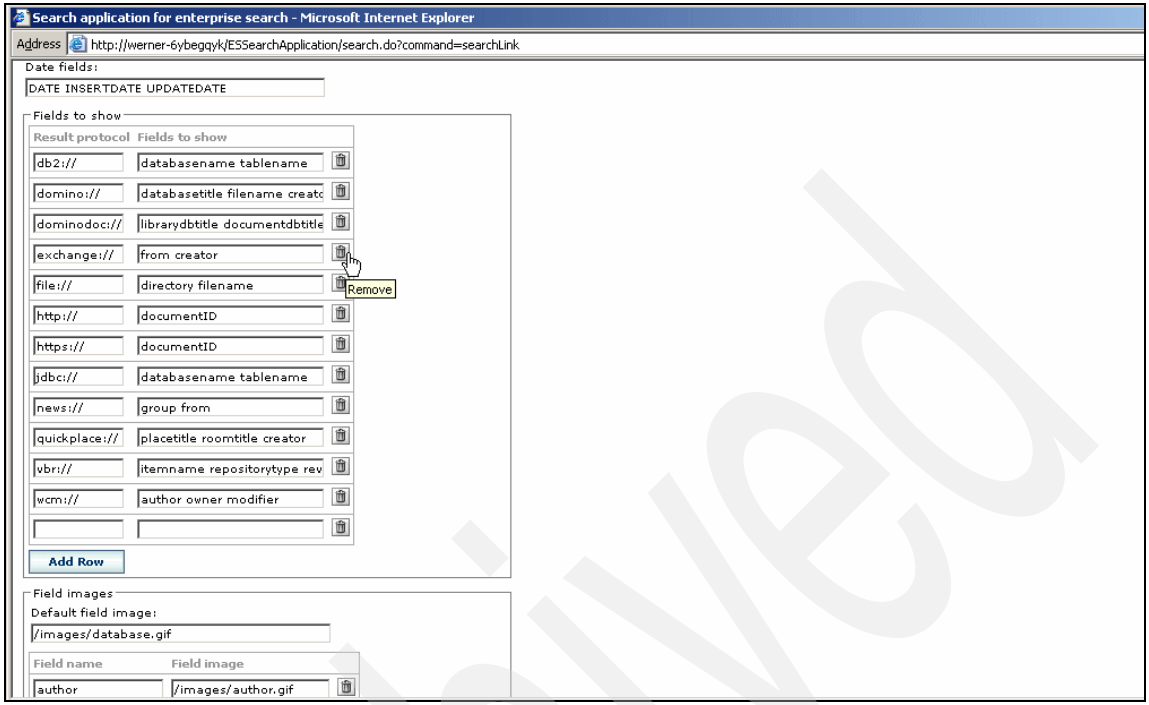

Figure B-17 Results parameters 2/2

Figure B-18 Query options 1/2

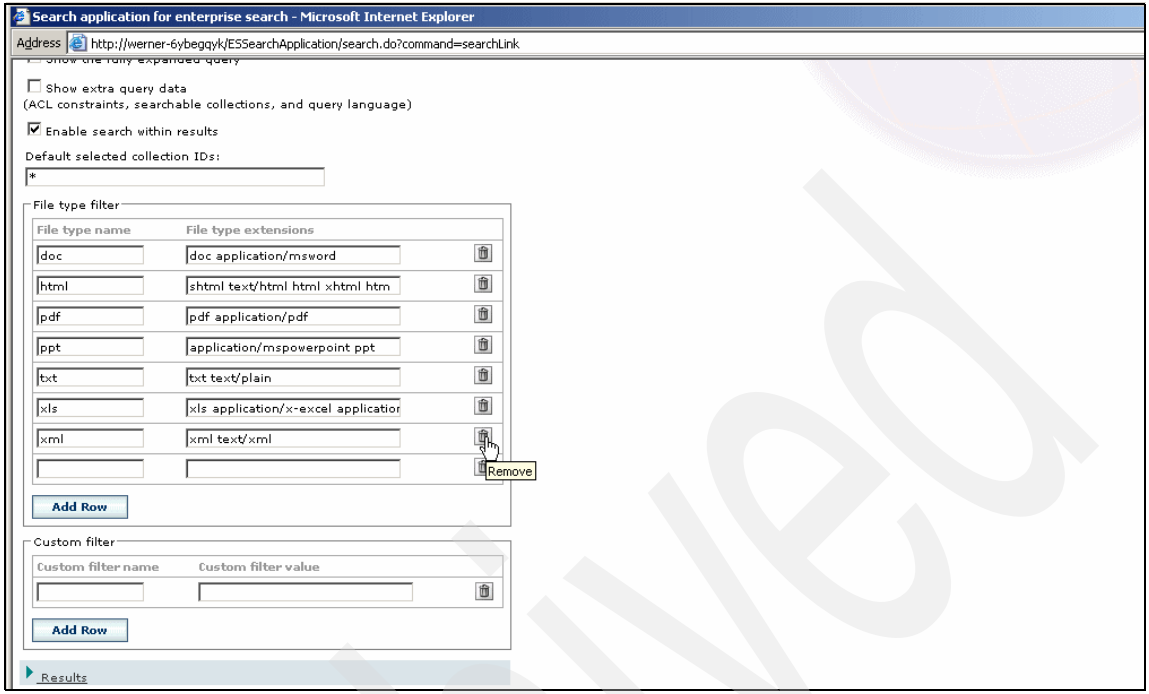

Figure B-19 Query options 2/2

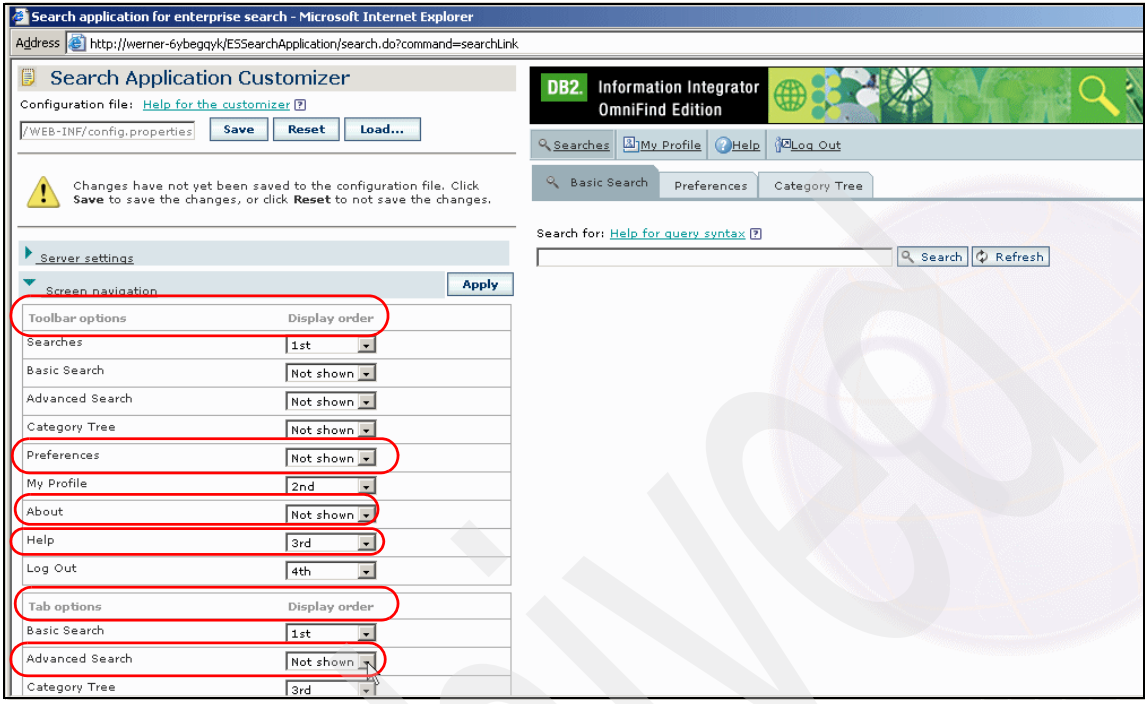

Figure B-20 Screen navigation Toolbar and Tab options

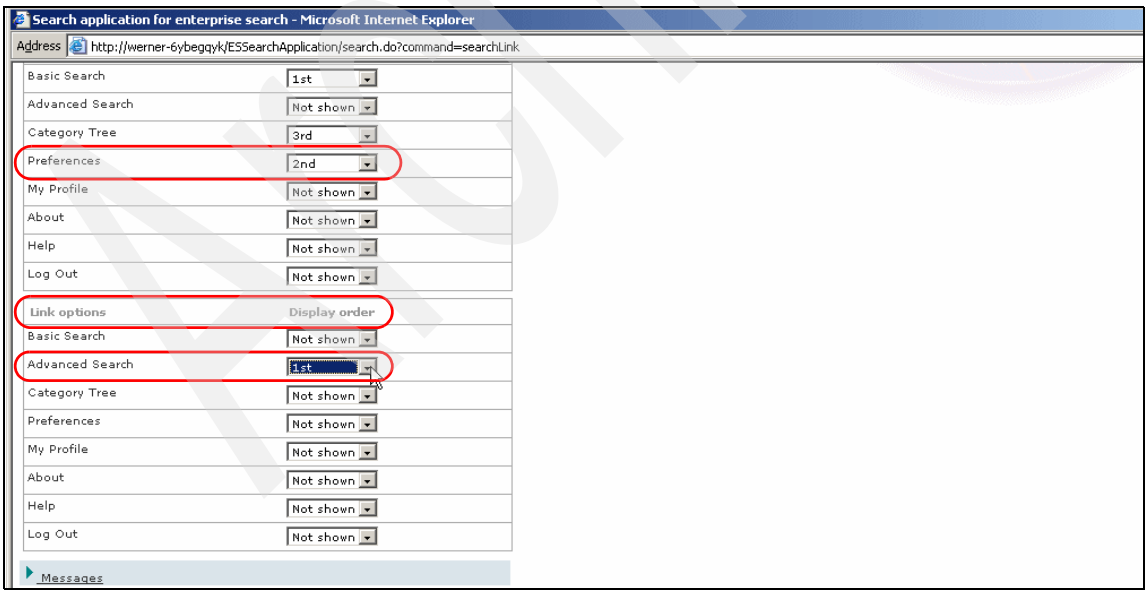

Figure B-21 Screen navigation Link options

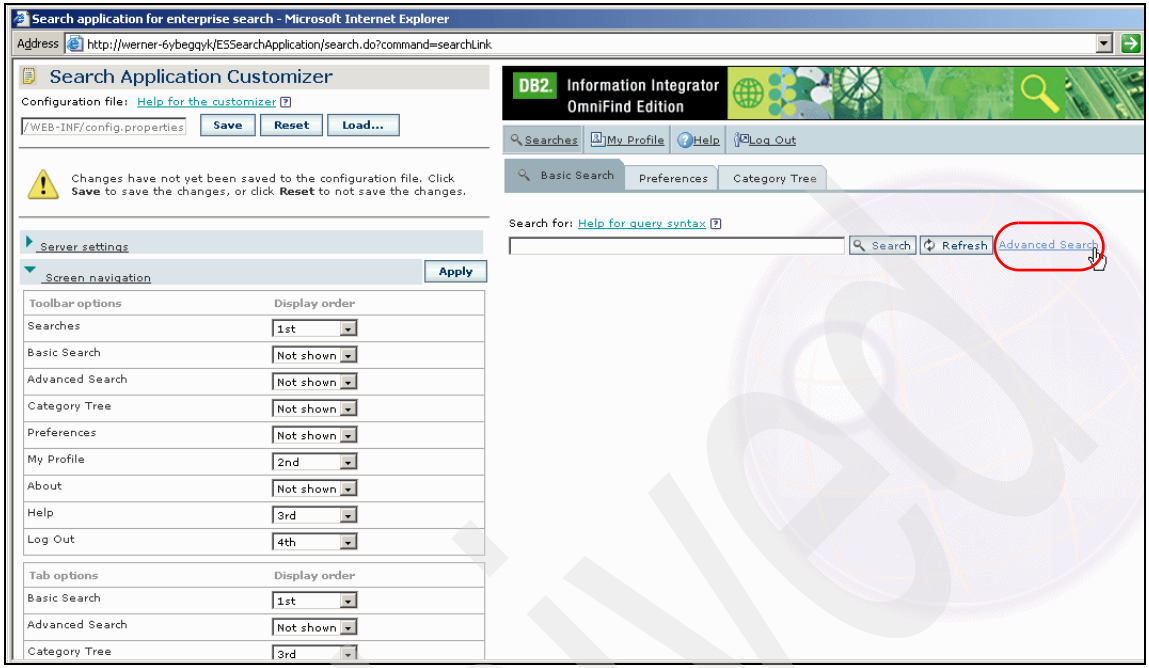

Figure B-22 Screen navigation settings

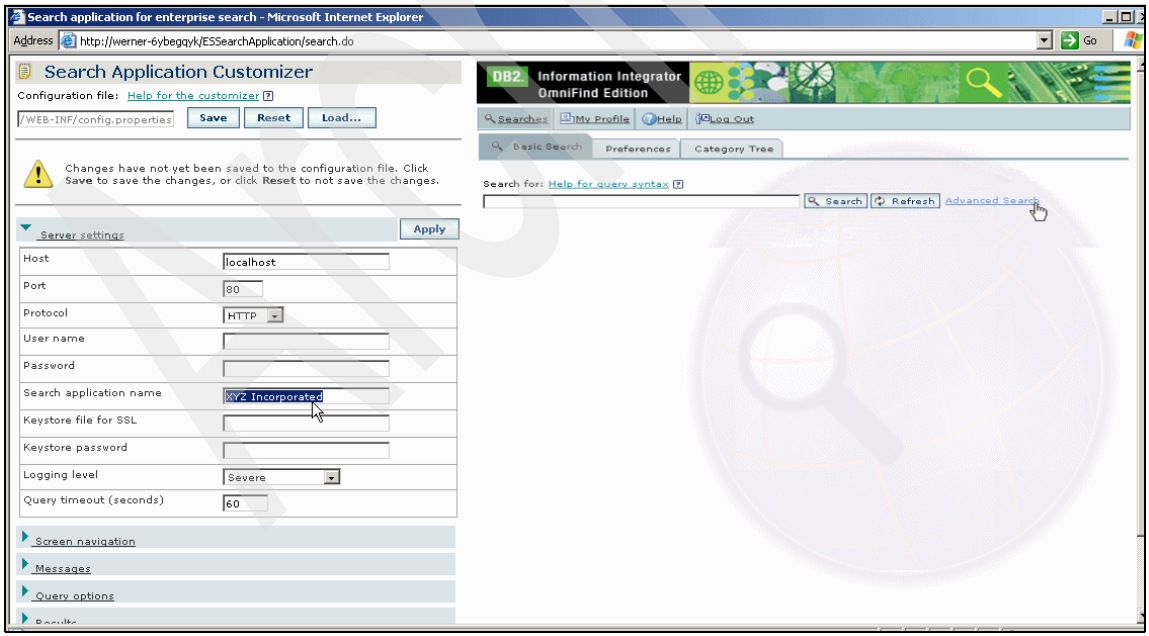

Figure B-23 Server settings

Figure B-24 Save all the settings

SACSTEP3: Invoke customized ESSearchApplication from a browser

Invoke the customized ESSearchApplication from a browser, as shown in Figure B-25.

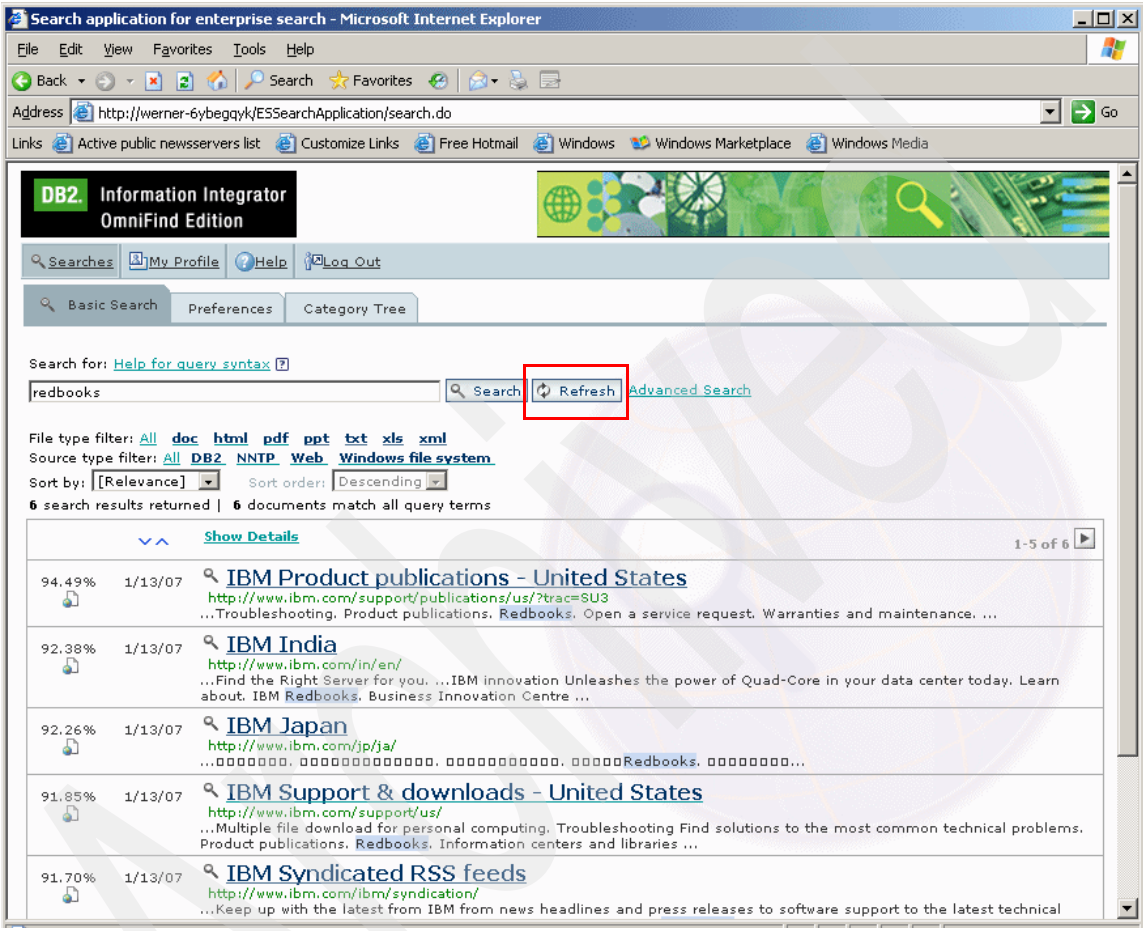

Figure B-25 Customized ESSearchApplication

C

IBM OmniFind Enterprise Edition V8.4 control tables

In this appendix, we provide an overview of the Cloudscape tables used by OmniFind Enterprise Edition V8.4.

Introduction

Prior to OmniFind Edition V8.4, the FOUNTAIN DB2 database contained tables that stored metadata information for each crawler, a raw data store for each crawler, and the IMC credentials. This database was installed on the crawler server.

In OmniFind V8.4, the FOUNTAIN DB2 database has been replaced by multiple Cloudscape databases that are also stored on the crawler server. Moreover, raw data store information is now stored in a file system on the indexer server, as described in 1.4.4, “Usability and performance enhancements” on page 34.

In this appendix, we provide:

- A brief overview of the Derby database engine and the ij script that we will be using to identify and query the structure and content of these Cloudscape control tables.
- Examples of accessing the Cloudscape database for the Windows file system and Web crawler.

Important: The information presented here is aimed at the sophisticated enterprise search administrator interested in obtaining a deeper understanding of the inner workings of the OmniFind product. While you may choose to browse the schema[a] and content of Cloudscape tables, you are strongly advised not to modify the contents of these tables, as this might produce unpredictable results.

a. The schema of these tables is likely to change without notice.

Derby database and ij script overview

IBM Cloudscape is a relational database management system that is based on Java and SQL. Cloudscape is a commercial release of the Apache Software Foundation's (ASF) open source relational database project called Derby.

The Cloudscape product includes Derby without any modification whatsoever to the underlying source code. Cloudscape includes the same core Derby engine, but provides a few features you will not find in the Derby software, such as installers (with a JRE™) and translated manuals and error messages. In addition, technical support is available for purchase for the Cloudscape product through IBM.

Note: Because Cloudscape and Derby have the same functionality, the Cloudscape documentation refers to the core functionality as Derby. All references to "Derby" in this documentation refer to the Derby core engine included with the Cloudscape product.

You can deploy Derby in a number of different ways:

- Embedded in a single-user Java application. Derby can be almost invisible to the user because it requires no administration and runs in the same Java virtual machine (JVM™) as the application.

  Note: This is how it is used in IBM OmniFind Enterprise Edition by the crawlers.

- Embedded in a multiuser application, such as a Web server, an application server, or a shared development environment.
- Embedded in a server framework. You can use the Network Server with the Network Client driver or a server of your own choice.

  Note: This is how the IMC credentials database is used in IBM OmniFind Enterprise Edition.

ij script overview

ij is Derby's interactive JDBC scripting tool. It is a simple utility for running scripts against a Derby database. You can also use it interactively to run *ad hoc* queries. ij provides several commands for ease in accessing a variety of JDBC features. ij can be used in an embedded or a client/server environment.

ij is a Java application, which you start from a command window, such as an MS-DOS® command window or the UNIX shell. ij provides several commands to easily access a variety of JDBC features through scripts.

In the following sections, we provide a brief overview of starting ij, connecting to a Derby database, and running the ij script.

Starting ij

Derby provides batch and shell scripts for users in Windows and UNIX environments. If you put the appropriate script in your path, you will be able to start ij with a simple command. These scripts use the ij.protocol property, which automatically loads a driver and simplifies the process of connecting to a

database. The scripts are found in the %DERBY_INSTALL%/bin/ directory. You can also customize the ij scripts to suit your environment.

If you are starting ij from a command line, be sure that the derbytools.jar file is in your classpath.

You can start ij by running the ij scripts in the /frameworks/embedded/bin/ directory or in the /frameworks/NetworkServer/bin/ directory.

To start ij, run the script provided or use this command:

```
java [<options>] org.apache.derby.tools.ij [-p <propertyFile>]
[<inputFile>]
```

Where:

- java is the name of the JVM program.
- options is what the JVM uses.
- propertyFile sets the ij properties (instead of the -D command).
- inputFile sets the file from which commands are read.

Note: The ij tool exits at the end of the file or an exit command. Using an input file causes ij to print out the commands as it runs them. If you reroute standard input, ij does not print out the commands. If you do not supply an input file, ij reads from the standard input.

Connecting to a Derby database

To connect to a Derby database, you need to perform the following steps:

1. Load the appropriate driver.
2. Provide a database connection URL for the database.

In ij, there are three ways of accomplishing these steps:

- Full database connection URL

  ij can work with any JDBC driver. For drivers supplied by other vendors, you need to load the driver separately. For drivers supplied by Derby, you can load the driver by specifying the full database connection URL in the connection. You do not need to load the driver explicitly in a second step.

```
D:>java org.apache.derby.tools.ij
      ij version 10.1
      ij> connect 'jdbc:derby:sample';
      ij>
```

- Protocol and short database connection URL

  For drivers supplied by Derby, specifying a protocol automatically loads the appropriate driver. You do not need to load the driver explicitly in a separate step. You specify a protocol with a property (ij.protocol or ij.protocol.protocolName) or command (Protocol).

  To connect, specify the "short form" of the database connection URL in a Connect command, ij.connection.connectionName property, or ij.database property. A short form of the database connection URL eliminates the protocol.

```
D:>java org.apache.derby.tools.ij
   ij version 10.1
   ij> protocol 'jdbc:derby:';
   ij> connect 'sample';
   ij>
D:>java -Dij.protocol.myprotocolName=jdbc:derby:
       org.apache.derby.tools.ij
   ij version 10.1
   ij> connect 'sample' protocol myprotocolName;
   ij>
```

- Driver and full database connection URL

  If you are using the drivers supplied by Derby, use the driver names listed in the JDBC drivers overview. The Derby drivers are implicitly loaded when a supported protocol is used. Any other driver has to be explicitly loaded. You can load a driver explicitly with an ij property (ij.Driver), a system property (jdbc.drivers), or a command (Driver).

  To connect, specify the full database connection URL in a Connect command, ij.connection.connectionName property, or ij.database property.

```
D:>java org.apache.derby.tools.ij
   ij version 10.1
   ij> driver 'sun.jdbc.odbc.JdbcOdbcDriver';
   ij> connect 'jdbc:odbc:myOdbcDataSource';
```

Run the ij scripts

You can run scripts in ij in any of the following ways:

- Name an input file as a command-line argument. For example:

```
java -Djdbc.drivers=org.apache.derby.jdbc.EmbeddedDriver
org.apache.derby.tools.ij <myscript.sql>
```

- Redirect standard input to come from a file. For example:

```
java -Djdbc.drivers=org.apache.derby.jdbc.EmbeddedDriver
org.apache.derby.tools.ij <myscript.sql>
```

- Use the Run command from the ij command line. For example:

```
ij> run 'myscript.sql';
```

Note: If you name an input file as a command-line argument or if you use the Run command, ij echoes input from a file. If you redirect standard input to come from a file, ij does not echo commands.

You can save output in any of the following ways:

- By redirecting output to a file:

```
java -Djdbc.drivers=org.apache.derby.jdbc.EmbeddedDriver
org.apache.derby.tools.ij <myscript.sql> <myoutput.txt>
```

- By setting the ij.outfile property:

```
java -Dij.outfile=<myoutput.txt> org.apache.derby.tools.ij
<myscript.sql>
```

ij exits when you enter the **`Exit`** command or if you enter a command file on the Java invocation line when the end of the command file is reached. When you use the **`Exit`** command, ij automatically shuts down an embedded Derby system by issuing a **`connect jdbc:derby:;shutdown=true`** request. It does not shut down Derby if it is running in a server framework.

Note: For full details on using the ij script, refer to *Derby Tools and Utilities Guide*, found at:

`http://db.apache.org/derby/docs/dev/tools/derbytools.pdf`

Also refer to `http://db.apache.org/derby/papers/DerbyTut/ns_intro.html` for information about accessing Cloudscape databases.

Cloudscape control tables

The Cloudscape database for crawlers, data listener, and IMC user credentials are stored on the Crawler server in the $ES_NODE_ROOT/cloudscape directory and sub-directories as follows:

- Crawled document metadata is stored in $ES_NODE_ROOT/data/cloudscape/OmniFind_crawlers/<collection id>.<crawler id>.
- Add/remove URIs collection data is stored in $ES_NODE_ROOT/data/cloudscape/OmniFind_datalistener/.

- IMC credentials user profile data is stored in $ES_NODE_ROOT/data/cloudscape/OmniFind_imc/.
- $ES_NODE_ROOT/data/cloudscape/OmniFind_fs/ is the name of a temporary database used by the FirstSteps application and is required to validate the Cloudscape installation. After FirstSteps application is run to completion, this directory should automatically get deleted. If FirstSteps failed while validating the Cloudscape installation, then this directory might linger around, but a rerun of the FirstSteps application should clean it up.

Figure C-1 on page 490 shows the directory structure on a Windows platform for a collection with QuickPlace, JDBC, and two Windows file system crawlers. Each of the folders shown corresponds to an embedded Cloudscape database associated with that crawler. The crawlers access the Cloudscape database in embedded mode. Accessing the crawler Cloudscape databases is described in “Accessing crawler Cloudscape databases” on page 490.

The Cloudscape IMC database (OmniFind-imc), on the other hand, is a network database, and accessing it is described in “Accessing IMC Cloudscape database” on page 496.

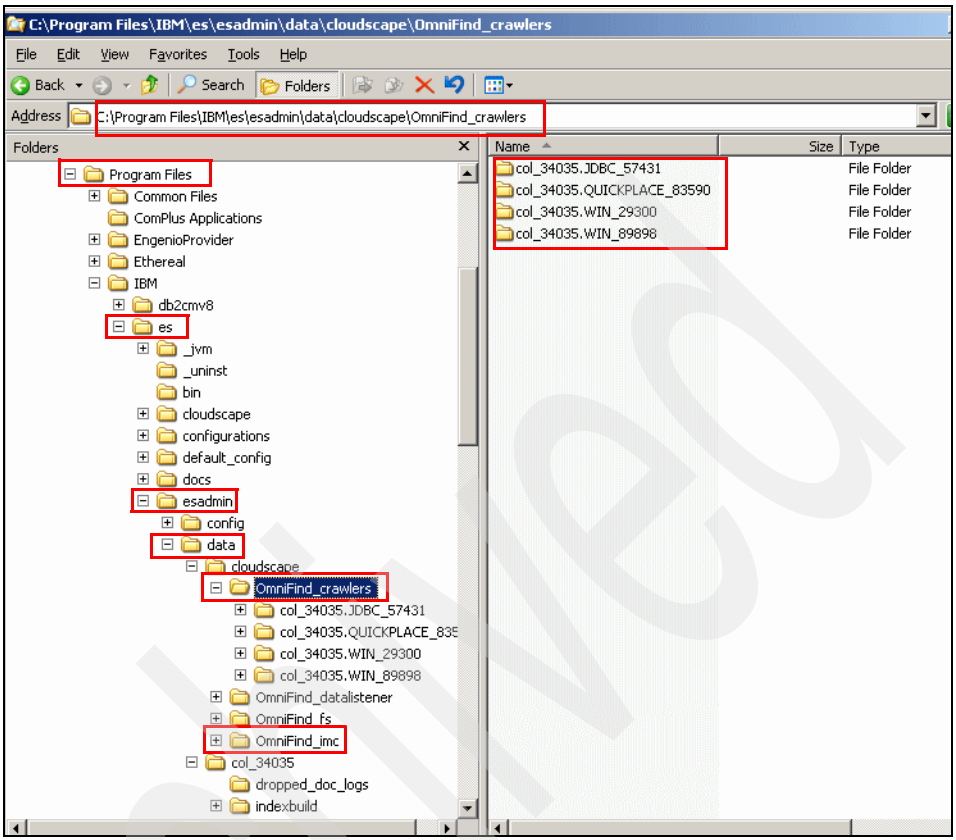

Figure C-1 Cloudscape database directory structure

Accessing crawler Cloudscape databases

The crawlers use the Cloudscape in embedded mode, and therefore only a single process can access it at a time. Therefore, you need to stop the crawlers to access a given database. These databases also do not have any user ID / password associated with them.

A number of approaches are available to connect and navigate a Cloudscape database, including the Cloudview GUI interface, the command-line interface using the **`ij`** command, as described in “ij script overview” on page 485, and a user-written program.

We use the command line interface to connect and query the crawler Cloudscape database. In the following sections, we first describe the preparatory work that is required, followed by access to the Cloudscape database and tables.

Prepare the Cloudscape environment

As mentioned earlier, OmniFind accesses all the Cloudscape databases (crawler or IMC) with no user ID / password, and the supported Cloudscape version ID 10.1. Therefore, the preparatory work required includes stopping all other access to the Cloudscape database, modifying the **ij.bat** command, and modifying the **dblookup.bat** command, as shown below

Stop access to all the Cloudscape databases

Stop the OmniFind crawler when you want to access the Cloudscape database you want to access. This is not shown here.

Modify the ij.bat file

The ij.bat file is located in the c:\ibm\es\cloudscape\frameworks\embedded\bin\ directory. The commands to set the user ID / password should be commented out, as shown in Example C-1. Also, modify the environment variables as required.

Example: C-1 Modify the ij.bat file

```
@echo off
@if "%1" == "shortcut" set CLASSPATH=
@if "%1" == "shortcut" shift

set DERBY_INSTALL=C:\Program Files\IBM\es\cloudscape
set JAVA_HOME=C:\Program Files\IBM\es\_jvm\jre

@REM  if "%DERBY_USER%" != "" set IJ_USER="-Dij.user=%DERBY_USER%"
@REM  if "%DERBY_PASSWORD%" != "" set IJ_PASSWORD="-Dij.password=%DERBY_PASSWORD%"

@REM set PARAM="%IJ_USER% %IJ_PASSWORD%"
set OF_CLASSPATH=%DERBY_INSTALL%\lib\derby.jar;%DERBY_INSTALL%\lib\derbytools.jar;%CLASSPATH%

@REM ------------------------------------------------------------
@REM -- start ij
@REM ------------------------------------------------------------
"%JAVA_HOME%\bin\java.exe" -cp "%OF_CLASSPATH%" %PARAM% -Dij.protocol=jdbc:derby: org.apache.derby.tools.ij


@REM ------------------------------------------------------------
@REM -- To use a different JVM with a different syntax, simply edit
@REM -- this file
@REM ------------------------------------------------------------
```

Modify the dblook.bat file

The dblook.bat[1] file is located in the c:\ibm\es\cloudscape\frameworks\embedded\bin\ directory. The commands to set the user ID / password should be commented out, as shown in Example C-2. Also, modify the environment variables as required. This command is used to list the CREATE SQL statement for a given table in the Cloudscape database.

Example: C-2 Modify the dblook.bat file

```
.............................
set DERBY_INSTALL=C:\Program Files\IBM\es\cloudscape
set JAVA_HOME=C:\Program Files\IBM\es\_jvm\jre

@REM if "%DERBY_USER%" != "" set IJ_USER="-Dij.user=%DERBY_USER%"
@REM if "%DERBY_PASSWORD%" != "" set IJ_PASSWORD="-Dij.password=%DERBY_PASSWORD%"

@REM set PARAM="-DES_CFG=%ES_NODE_ROOT%/nodeinfo/es.cfg %IJ_USER% %IJ_PASSWORD%"
@REM set PARAM="%IJ_USER% %IJ_PASSWORD%"

set
OF_CLASSPATH=%DERBY_INSTALL%\lib\derby.jar;%DERBY_INSTALL%\lib\derbytools.jar;%ES_INSTALL_ROOT%\lib\esctrl.jar;%CLASSPATH%

@REM ------------------------------------------------------------
@REM -- start dblook
@REM ------------------------------------------------------------
"%JAVA_HOME%\bin\java" -cp "%OF_CLASSPATH%" %PARAM% org.apache.derby.tools.dblook %*
```

Accessing the Cloudscape databases and tables

With the preparatory work done, we can proceed to identify the OmniFind tables in a crawler database, determine the CREATE statement for a table, and list its contents as follows.

The various crawler Cloudscape databases for our environment are shown in Figure C-1 on page 490.

Invoke the ij environment and connect to a database

Example C-3 shows the establishment of the ij environment (by executing the **ij.bat** command in the c:\ibm\es\cloudscape\frameworks\embedded\bin\ directory) and connection to a the crawler database col_34035.QUICKPLACE_83590 (using the connect SQL statement).

Example: C-3 Invoke the ij environment and connect to a crawler database

```
C:\Program Files\IBM\es\cloudscape\frameworks\embedded\bin>ij.bat
ij version 10.1
ij> connect 'jdbc:derby:c:/progra~1/ibm/es/esadmin/data/cloudscape/omnifind_crawlers/col_34035.QUICKPLACE_83590';
ij>
```

[1] dblook is used because the SQL DESCRIBE statement is not supported in Cloudscape Version 10.1.

Identify OmniFind tables in the crawler database

After successfully connecting to the crawler database, as shown in Example C-3 on page 492, we can identify the OmniFind tables by executing the query shown in Example C-4. The schema name of the OmniFind tables is ESADMIN.

Example C-4 shows four OmniFind tables in the QUICKPLACE crawler database: CDSR, TDATAREC, TOPTDATA, and TSERVERREC.

Example: C-4 Identify OmniFind tables in this crawler database

```
ij> select schemaname, tablename from sys.systables a, sys.sysschemas b where a.schemaid=b.schemaid and
schemaname='ESADMIN';

SCHEMANAME                                                      TABLENAME
------------------------------------------------------------------------
ESADMIN                                    CDSR

ESADMIN TDATAREC

ESADMIN TOPTDATA

ESADMIN TSERVERREC

4 rows selected
```

Determine the CREATE SQL statement for a table

As mentioned earlier, Cloudscape Database Version 10.1 does not support the DESCRIBE SQL statement to view the columns and associated data types of a table. We therefore used the **dblook** command to determine the CREATE SQL statement, as shown in Example C-5.

Example: C-5 Determine the CREATE SQL statement for the CDSR table

```
C:\Program Files\IBM\es\cloudscape\frameworks\embedded\bin>dblook.bat -z ESADMIN
  -t cdsr -d "jdbc:derby:c:/progra~1/ibm/es/esadmin/data/cloudscape/omnifind_cra
wlers/col_34035.QUICKPLACE_83590";

C:\Program Files\IBM\es\cloudscape\frameworks\embedded\bin>set DERBY_INSTALL=C:\
Program Files\IBM\es\cloudscape

C:\Program Files\IBM\es\cloudscape\frameworks\embedded\bin>set JAVA_HOME=C:\Prog
ram Files\IBM\es\_jvm\jre

C:\Program Files\IBM\es\cloudscape\frameworks\embedded\bin>set OF_CLASSPATH=C:\P
rogram Files\IBM\es\cloudscape\lib\derby.jar;C:\Program Files\IBM\es\cloudscape\
lib\derbytools.jar;C:\PROGRA~1\IBM\es\CLOUDS~1\lib\derby.jar;C:\PROGRA~1\IBM\es\
CLOUDS~1\lib\derbytools.jar;C:\PROGRA~1\IBM\es\lib\esctrl.jar;C:\Program Files\I
BM\SQLLIB\java\db2jcc.jar;C:\Program Files\IBM\SQLLIB\java\db2jcc_license_cu.jar
;C:\PROGRA~1\IBM\es\lib;.;C:\PROGRA~1\IBM\SQLLIB\java\db2java.zip;C:\PROGRA~1\IB
M\SQLLIB\java\db2jcc.jar;C:\PROGRA~1\IBM\SQLLIB\java\sqlj.zip;C:\PROGRA~1\IBM\SQ
LLIB\java\db2jcc_license_cisuz.jar;C:\PROGRA~1\IBM\SQLLIB\java\db2jcc_license_cu
.jar;C:\PROGRA~1\IBM\SQLLIB\bin;C:\PROGRA~1\IBM\SQLLIB\java\common.jar;c:\Progra
m Files\IBM\es\cloudscape\lib\derbytools.jar;

C:\Program Files\IBM\es\cloudscape\frameworks\embedded\bin>"C:\Program Files\IBM
\es\_jvm\jre\bin\java" -cp "C:\Program Files\IBM\es\cloudscape\lib\derby.jar;C:\
Program Files\IBM\es\cloudscape\lib\derbytools.jar;C:\PROGRA~1\IBM\es\CLOUDS~1\l
ib\derby.jar;C:\PROGRA~1\IBM\es\CLOUDS~1\lib\derbytools.jar;C:\PROGRA~1\IBM\es\l
ib\esctrl.jar;C:\Program Files\IBM\SQLLIB\java\db2jcc.jar;C:\Program Files\IBM\S
QLLIB\java\db2jcc_license_cu.jar;C:\PROGRA~1\IBM\es\lib;.;C:\PROGRA~1\IBM\SQLLIB
\java\db2java.zip;C:\PROGRA~1\IBM\SQLLIB\java\db2jcc.jar;C:\PROGRA~1\IBM\SQLLIB\
java\sqlj.zip;C:\PROGRA~1\IBM\SQLLIB\java\db2jcc_license_cisuz.jar;C:\PROGRA~1\I
BM\SQLLIB\java\db2jcc_license_cu.jar;C:\PROGRA~1\IBM\SQLLIB\bin;C:\PROGRA~1\IBM\
SQLLIB\java\common.jar;c:\Program Files\IBM\es\cloudscape\lib\derbytools.jar;"
org.apache.derby.tools.dblook -z ESADMIN  -t cdsr -d "jdbc:derby:c:/progra~1/ibm
/es/esadmin/data/cloudscape/omnifind_crawlers/col_34035.QUICKPLACE_83590";
-- Time Stamp: 2007-01-22 17:17:21.984
-- Source database is: c:/progra~1/ibm/es/esadmin/data/cloudscape/omnifind_crawl
ers/col_34035.QUICKPLACE_83590
-- Connection URL is: jdbc:derby:c:/progra~1/ibm/es/esadmin/data/cloudscape/omni
find_crawlers/col_34035.QUICKPLACE_83590;
-- The dblook utility will consider only specified tables.
-- Specified schema is: ESADMIN
-- appendLogs: false

-- ----------------------------------------------
-- DDL Statements for schemas
-- ----------------------------------------------

CREATE SCHEMA "ESADMIN";

-- ----------------------------------------------
-- DDL Statements for tables
-- ----------------------------------------------

CREATE TABLE "ESADMIN"."CDSR" ("HASH" BIGINT NOT NULL, "TARGETID" BIGINT NOT NULL, "URI" VARCHAR(2048) NOT NULL,
"CHECKSUM" VARCHAR (16) FOR BIT DATA, "LASTMODIFIED" BIGINT, "LASTACCESSED" BIGINT, "PRHASH" BIGINT, "EXCLUDED" SMALLINT,
"ACLHASH" BIGINT);
```

```
-- ------------------------------------------------
-- DDL Statements for indexes
-- ------------------------------------------------

CREATE INDEX "ESADMIN"."CDSRX" ON "ESADMIN"."CDSR" ("PRHASH");

-- ------------------------------------------------
-- DDL Statements for keys
-- ------------------------------------------------

-- primary/unique
ALTER TABLE "ESADMIN"."CDSR" ADD CONSTRAINT "SQL061214031044150" PRIMARY KEY ("HASH");
```

List the contents of a table

Example C-6 shows the SQL statement to list the contents of the CDSR table.

Example: C-6 List the contents of the CDSR table

```
ij> select * from ESADMIN.CDSR;
HASH                |TARGETID            |URI

          |CHECKSUM                        |LASTMODIFIED        |LASTACCESSED
     |PRHASH               |EXCLU&|ACLHASH
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
------------------------------------------------------
-7747088274088376446|2                   |quickplace://kazan.itsosj.sanjose.ibm.
com/8825723E007E49B9/QuickPlace%5Cw3knowyourindustry%5CMain.nsf/A7986FD2A9CD4709
052567080&|NULL                           |1165948503000       |1166205919500
     |NULL                 |0     |NULL
3873056970272087946 |2                   |quickplace://kazan.itsosj.sanjose.ibm.
com/8825723E007E49B9/QuickPlace%5Cw3knowyourindustry%5CMain.nsf/A7986FD2A9CD4709
052567080&|NULL                           |1165948756000       |1166205919625
     |NULL                 |0     |NULL
-9144532275002361060|2                   |quickplace://kazan.itsosj.sanjose.ibm.
com/8825723E007E49B9/QuickPlace%5Cw3knowyourindustry%5CMain.nsf/A7986FD2A9CD4709
052567080&|NULL                           |1165948902000       |1166205919703
     |NULL                 |0     |NULL
6392548167513277556 |2                   |quickplace://kazan.itsosj.sanjose.ibm.
com/8825723E007E49B9/QuickPlace%5Cw3knowyourindustry%5CMain.nsf/A7986FD2A9CD4709
052567080&|NULL                           |1165948975000       |1166205919797
     |NULL                 |0     |NULL
-6426157463296059021|2                   |quickplace://kazan.itsosj.sanjose.ibm.
com/8825723E007E49B9/QuickPlace%5Cw3knowyourindustry%5CMain.nsf/A7986FD2A9CD4709
052567080&|NULL                           |1165949035000       |1166205919875
     |NULL                 |0     |NULL
-5246679202682452664|2                   |quickplace://kazan.itsosj.sanjose.ibm.
com/8825723E007E49B9/QuickPlace%5Cw3knowyourindustry%5CMain.nsf/A7986FD2A9CD4709
052567080&|NULL                           |1165949094000       |1166205919969
     |NULL                 |0     |NULL
-181521754300285356 |2                   |quickplace://kazan.itsosj.sanjose.ibm.
com/8825723E007E49B9/QuickPlace%5Cw3knowyourindustry%5CMain.nsf/A7986FD2A9CD4709
052567080&|NULL                           |1165949114000       |1166205920359
     |NULL                 |0     |NULL
2889260838872131738 |2                   |quickplace://kazan.itsosj.sanjose.ibm.
com/8825723E007E49B9/QuickPlace%5Cw3knowyourindustry%5CMain.nsf/A7986FD2A9CD4709
052567080&|NULL                           |1165949430000       |1166205920656
     |NULL                 |0     |NULL

8 rows selected
ij>
```

Note: This process can be repeated for all the OmniFind tables for all the crawlers.

Accessing IMC Cloudscape database

The IMC Cloudscape database is accessed in network mode, and therefore multiple processes can access it at one time. Like the embedded crawlers databases, this database does not have any user ID / password associated with it.

A number of approaches are available to connect and navigate a Cloudscape database, including the Cloudview GUI interface, the command-line interface using the **ij** command, as described in “ij script overview” on page 485, and a user-written program.

We use the command-line interface to connect and query the crawler Cloudscape database. In the following sections, we first describe the preparatory work that is required, followed by access to the Cloudscape database and tables.

Prepare the Cloudscape environment

As mentioned earlier, OmniFind accesses the IMC Cloudscape database with no user ID / password in network server mode. Also, the supported Cloudscape version ID is 10.1. Therefore, the preparatory work required includes ensuring that the network server is running, modifying the **ij.bat** command, and modifying the **dblookup.bat** command as follows.

Ensure that the network server is running

The Cloudscape network server must be running to access this database. The network server is automatically started when the common communications layer (CCL[2]) is started. Execute the **sysinfo.bat** command in the c:\Program Files\IBM\es\cloudscape\frameworks\NetworkServer\bin directory. If you obtain the information shown in Example C-7 on page 497, it means that the network server is up and running. If you do not receive any output, then you may have to stop and start the CCL again to start up the network server.

[2] IBM WebSphere Information Integrator OmniFind Edition service

Example: C-7 sysinfo.bat output

```
C:\PROGRA~1\IBM\es\cloudscape\frameworks\NetworkServer\bin>sysinfo.bat
--------- Derby Network Server Information --------
Version: CSS10011/10.1.2.4  Build: 396056  DRDA Product Id: CSS10011
-- listing properties --
derby.drda.maxThreads=0
derby.drda.keepAlive=true
derby.drda.minThreads=0
derby.drda.portNumber=1527
derby.drda.logConnections=false
derby.drda.timeSlice=0
derby.drda.startNetworkServer=false
derby.drda.host=0.0.0.0
derby.drda.traceAll=false
------------------ Java Information ------------------
Java Version:    1.4.2
Java Vendor:     IBM Corporation
Java home:       C:\Program Files\IBM\es\_jvm\jre
Java classpath:  C:\Program Files\IBM\es\lib\ccl.jar;C:\Program Files\IBM\es\lib
\es.oss.jar;C:\Program Files\IBM\es\lib\esctrl.jar;C:\Program Files\IBM\es\lib\L
UMClient.jar;C:\Program Files\IBM\es\lib\mail.jar;C:\Program Files\IBM\es\lib\ac
tivation.jar;C:\Program Files\IBM\es\lib\ccl.jar;C:\Program Files\IBM\es\lib\es.
oss.jar;C:\Program Files\IBM\es\lib\esctrl.jar;C:\Program Files\IBM\es\lib\mail.
jar;C:\Program Files\IBM\es\lib\activation.jar;C:\Program Files\IBM\es\lib\LUMCl
ient.jar;C:\Program Files\IBM\es\cloudscape\lib\derby.jar;C:\Program Files\IBM\e
s\cloudscape\lib\derbynet.jar;C:\Program Files\IBM\es\lib\siapi.jar;C:\Program F
iles\IBM\es\lib\esapi.jar;C:\Program Files\IBM\SQLLIB\java\db2jcc.jar;C:\Program
 Files\IBM\SQLLIB\java\db2jcc_license_cu.jar;C:\PROGRA~1\IBM\es\lib;.;C:\PROGRA~
1\IBM\SQLLIB\java\db2java.zip;C:\PROGRA~1\IBM\SQLLIB\java\db2jcc.jar;C:\PROGRA~1
\IBM\SQLLIB\java\sqlj.zip;C:\PROGRA~1\IBM\SQLLIB\java\db2jcc_license_cisuz.jar;C
:\PROGRA~1\IBM\SQLLIB\java\db2jcc_license_cu.jar;C:\PROGRA~1\IBM\SQLLIB\bin;C:\P
ROGRA~1\IBM\SQLLIB\java\common.jar
OS name:         Windows 2003
OS architecture: x86
OS version:      5.2
Java user name:  essearch
Java user home:  C:\Documents and Settings\essearch
Java user dir:   C:\Program Files\IBM\es\esadmin\logs
java.specification.name: Java Platform API Specification
java.specification.version: 1.4
--------- Derby Information --------
JRE - JDBC: J2SE 1.4.2 - JDBC 3.0
[C:\Program Files\IBM\es\cloudscape\lib\derby.jar] 10.1.2.4 - (396056)
[C:\Program Files\IBM\es\cloudscape\lib\derbynet.jar] 10.1.2.4 - (396056)
[C:\Program Files\IBM\SQLLIB\java\db2jcc.jar] 2.5 - (33)
[C:\Program Files\IBM\SQLLIB\java\db2jcc_license_cu.jar] 2.5 - (33)
[C:\Program Files\IBM\SQLLIB\java\db2jcc.jar] 2.5 - (33)
[C:\Program Files\IBM\SQLLIB\java\db2jcc_license_cisuz.jar] 2.5 - (33)
[C:\Program Files\IBM\SQLLIB\java\db2jcc_license_cu.jar] 2.5 - (33)
------------------------------------------------------
----------------- Locale Information -----------------
------------------------------------------------------

C:\PROGRA~1\IBM\es\cloudscape\frameworks\NetworkServer\bin>
```

Modify the ij.bat file

The ij.bat file is located in the c:\ibm\es\cloudscape\frameworks\NetworkServer\bin\ directory. The commands to set the user ID / password should be commented out, as shown in Example C-8. Also, modify the environment variables as required.

Example: C-8 Modify the ij.bat file

```
.........................
@echo off

@if "%1" == "shortcut" set CLASSPATH=
@if "%1" == "shortcut" shift
@REM ------------------------------------------------------
@REM -- This batch file is an example of how to start ij in
@REM -- an NetworkServer environment.
@REM --
@REM -- REQUIREMENTS:
@REM -- You must have the Derby and DB2 JCC libraries in your classpath
@REM --
@REM -- See the setNetworkClientCP.bat for an example of
@REM -- how to do this.
@REM --
@REM -- You may need to modify the values below for a different
@REM -- host, port, user, or password
@REM --
@REM -- This file for use on Windows systems
@REM ------------------------------------------------------------

set DERBY_INSTALL=C:\Program Files\IBM\es\cloudscape
set JAVA_HOME=C:\Program Files\IBM\es\_jvm\jre

set IJ_HOST=NILE.itsosj.sanjose.ibm.com
@if "%DERBY_SERVER_HOST%" NEQ "" set IJ_HOST=%DERBY_SERVER_HOST%

set IJ_PORT=1527
@if "%DERBY_SERVER_PORT%" NEQ "" set IJ_PORT=%DERBY_SERVER_PORT%

@REM if "%DERBY_USER%" NEQ "" set IJ_USER="-Dij.user=%DERBY_USER%"
@REM if "%DERBY_PASSWORD%" NEQ "" set IJ_PASSWORD="-Dij.password=%DERBY_PASSWORD%"

@REM set PARAM="-DES_CFG=%ES_NODE_ROOT%\nodeinfo\es.cfg %IJ_USER% %IJ_PASSWORD%"

@setlocal

set
OF_CLASSPATH=%DERBY_INSTALL%\lib\derbyclient.jar;%DERBY_INSTALL%\lib\derbytools.jar;%ES_INSTALL_ROOT%\lib\esctrl.jar;%ES_I
NSTALL_ROOT%\lib\es.oss.jar;%CLASSPATH%

@REM ------------------------------------------------------------
@REM -- start ij
@REM -- host, port, user and password may need to be changed
@REM ------------------------------------------------------------
"%JAVA_HOME%\bin\java.exe" -cp "%OF_CLASSPATH%" %PARAM% -Dij.driver=org.apache.derby.jdbc.ClientDriver
-Dij.protocol=jdbc:derby://%IJ_HOST%:%IJ_PORT%/ org.apache.derby.tools.ij

@REM ------------------------------------------------------------
@REM -- To use a different JVM with a different syntax, simply edit
@REM -- this file
@REM ------------------------------------------------------------

@endlocal
```

Modify the dblook.bat file

The **dblook.bat** file is located in the c:\ibm\es\cloudscape\frameworks\NetworkServer\bin\ directory, as shown in Example C-9. Modify the environment variables as required. This command is used to list the CREATE SQL statement for ESUSER table in the IMC Cloudscape database.

Example: C-9 Modify the dblook.bat file

```
...................
..............
set ES_INSTALL_ROOT=C:\Program Files\IBM\es
set DERBY_INSTALL=C:\Program Files\IBM\es\cloudscape
set JAVA_HOME=C:\Program Files\IBM\es\_jvm\jre

@setlocal

set
OF_CLASSPATH=%DERBY_INSTALL%\lib\derbyclient.jar;%DERBY_INSTALL%\lib\derbytools.jar;%ES_INSTALL_ROOT%\lib\esctrl.jar;%ES_I
NSTALL_ROOT%\lib\es.oss.jar;%CLASSPATH%
@REM ------------------------------------------------------
@REM -- start dblook
@REM ------------------------------------------------------
"%JAVA_HOME%\bin\java" -cp "%OF_CLASSPATH%" org.apache.derby.tools.dblook %*

@endlocal
```

Accessing the Cloudscape databases and tables

With the preparatory done, we can proceed to identify the OmniFind tables in a IMC database, determine the CREATE statement for the ESUSER table, and list its contents as follows:

The only IMC Cloudscape database is OmniFind_imc, as shown in Figure C-1 on page 490.

Invoke ij environment and connect to a database

Example C-10 shows the establishment of the ij environment (by executing the **ij.bat** command in the c:\ibm\es\cloudscape\frameworks\NetworkServer\bin\ directory) and connection to an IMC database OmniFind_imc (using the connect SQL statement).

Example: C-10 Invoke the ij environment and connect to a crawler database

```
C:\PROGRA~1\IBM\es\cloudscape\frameworks\NetworkServer\bin>ij.bat
ij version 10.1
ij> connect 'jdbc:derby://localhost:1527/c:/progra~1/ibm/es/esadmin/data/cloudscape/omnifind_imc';
ij>
```

Identify OmniFind tables in the IMC database

After successfully connecting to the IMC database, as shown in Example C-10 on page 499, we can identify the OmniFind tables by executing the query shown in Example C-11. The schema name of the OmniFind tables is APP.

Example C-11 shows a single OmniFind table ESUSER in the IMC database.

Example: C-11 Identify OmniFind tables in this crawler database

```
ij> select schemaname, tablename from sys.systables a, sys.sysschemas b where a.
schemaid=b.schemaid and schemaname='APP';
SCHEMANAME TABLENAME
--------------------------------------------------------------------------------
APP ESUSER

1 row selected
ij>
```

Determine the CREATE SQL statement for a table

As mentioned earlier, the Cloudscape Database Version 10.1 does not support the DESCRIBE SQL statement to view the columns and associated data types of a table. We therefore used the **dblook** command to determine the CREATE SQL statement, as shown in Example C-12.

Example: C-12 Determine the CREATE SQL statement for the ESUSER table

```
C:\Program Files\IBM\es\cloudscape\frameworks\NetworkServer\bin>dblook.bat  -z APP -t ESUSER -d
"jdbc:derby://localhost:1527/c:/progra~1/ibm/es/esadmin/data/cloudscape/omnifind_imc";
-- Time Stamp: 2007-01-23 20:53:02.797
-- Source database is: c:/progra~1/ibm/es/esadmin/data/cloudscape/omnifind_imc
-- Connection URL is: jdbc:derby://localhost:1527/c:/progra~1/ibm/es/esadmin/data/cloudscape/omnifind_imc;
-- The dblook utility will consider only specified tables.
-- Specified schema is: APP
-- appendLogs: false

-- ----------------------------------------------
-- DDL Statements for tables
-- ----------------------------------------------

CREATE TABLE "APP"."ESUSER" ("USERID" VARCHAR(256) NOT NULL, "CREDENTIALS" CLOB(
65536) NOT NULL);

-- ----------------------------------------------
-- DDL Statements for keys
-- ----------------------------------------------

-- primary/unique
ALTER TABLE "APP"."ESUSER" ADD CONSTRAINT "SQL061121015328180" PRIMARY KEY ("USE
RID");


C:\Program Files\IBM\es\cloudscape\frameworks\NetworkServer\bin>
```

List the contents of a table

Example C-13 shows the SQL statement to list the contents of the ESUSER table, which shows the credentials stored for nine different user IDs, as highlighted.

Example: C-13 List the contents of the ESUSER table

```
ij> select * from  APP.ESUSER;
USERID |CREDENTIALS
--------------------------------------------------------------------------------
esadmin |<identities id="ZXNhZG1pbg==">
identity id="TklMRQ=="><username>d2luZnNfdXNy</username><type>winfs</type><passw
ord encrypt="yes&
icmadmin |<identities id="aWNtYWRtaW4=">
identity id="TklMRQ=="><username>d2luZnNfdXNy</username><type>winfs</type><passw
ord encrypt="yes&
wpsadmin |<identities id="d3BzYWRtaW4=">
identity id="aWNtbmxzZGI="><username>aWNtYWRtaW4=</username><type>cm</type><pass
word encrypt="ye&
winfs_usr |<identities id="d2luZnNfdXNy">
identity id="TklMRQ=="><username>d2luZnNfdXNy</username><type>winfs</type><passw
ord encrypt="yes&
qpadmin |<identities id="cXBhZG1pbg==">
identity id="TklMRQ=="><username>null</username><type>winfs</type><password encr
ypt="yes"></pass&
wasadmin |<identities id="d2FzYWRtaW4=">
identity id="TklMRQ=="><username>null</username><type>winfs</type><password encr
ypt="yes"></pass&
testusr |<identities id="dGVzdHVzcg==">
identity id="TklMRQ=="><username>dGVzdHVzcg==</username><type>winfs</type><passw
ord encrypt="yes&
uid=wpsadmin,cn=users,ou=itso,o=ibm |<identities id="dWlkPXdwc2FkbWl
uLGNuPXVzZXJzLG91PWl0c28sbz1pYmO="><identity id="TklMRQ=="><username>ZXNhZG1pbg=
=</username><typ&
uid=esadmin,cn=users,ou=itso,o=ibm |<identities id="dWlkPWVzYWRtaW4
sY249dXNlcnMsb3U9aXRzbyxvPWlibQ=="><identity id="TklMRQ=="><username>QWRtaW5pc3R
yYXRvcg==</usern&

9 rows selected
```

D

Configuring typical data sources

In this appendix, we describe the configuration of certain data sources not included in the scenarios.

Introduction

OmniFind Enterprise Edition V8.4 added support for a number of data sources and corresponding crawlers to access these data sources. In this appendix, we include the configuration of the Web Content Management (WCM) and Portal Document Manager (PDM) with SSO enabled.

While the scenarios described earlier included WCM and PDM data sources, the WCM was crawled using the Web crawler in Sequoia General's GENINSINFO collection, while PDM was configured without SSO in Sequoia General's CUSTINFO collection.

Attention: In all the following sections, for the purposes of avoiding screen capture overload, we have *not* included all the windows that you would typically navigate through in order to perform the desired function. Instead, we have focused on including select screen captures (and in some cases, portions of selected screen captures) that highlight the key items of interest, thereby skipping both initial as well as intervening screen captures in the process.

Configure Web Content Management (WCM) crawler

The Web Content Management crawler can crawl any number of WCM sites. When you configure the crawler, you specify the URLs for the sites to be crawled. The crawler then downloads the pages that are linked from the specified sites. The sites to be crawled must be accessible by the same WebSphere Portal administrator ID and password. To crawl sites that use different credentials, you must configure a separate WCM crawler.

Creating the crawler involves the following tasks:

- Before you create a WCM crawler, you must run a script to set up the enterprise search environment on WebSphere Portal. This script (wp6_install.sh[1] on AIX, Linux, or Solaris, or wp6_install.bat on Windows) is installed on the search servers when IBM OmniFind Enterprise Edition is installed.

  This step is not shown here.

  When you specify the URLs to crawl, you must use the following format:

  ```
  http_protocol://portal_hostname:port_number/portal_prefix
  /WCM_search_seed_servlet_path/searchseed?site=WCM_site_name&lib=WCM_
  library_name
  ```

[1] For WebSphere Portal Version 6

The following example shows the URL for a site at the default installation path of Workplace WCM on WebSphere Portal:

```
http://portal.server.ibm.com:80/wps/wcmsearchseed/searchseed?site=SiteTest01&lib=Web+Content
```

Note: If the site name or library name contains spaces, you must replace the space with a plus sign (+) character. For example, replace Web Content with Web+Content.

- Specify properties that control how the crawler operates and uses system resources. The crawler properties control how the crawler crawls all documents in the crawl space.
- Specify the URLs for the sites to be crawled and information that enables the crawler to connect to the sites. When you create or edit the crawler, you can test the crawler's ability to connect to the URLs to be crawled. Messages tell you whether the crawler can access the documents to be crawled before you start the crawler.
- Specify document-level security options. If security was enabled when the collection was created, the crawler can associate security data with documents in the index. This data enables access controls to be enforced based on the stored access control lists or security tokens. You can also select an option to validate user credentials at the time a user submits a query. In this case, instead of comparing the user's credentials to indexed security data, the system compares the credentials to current access control lists that are maintained by the original data source.

Important: To search secure WCM pages, you must submit searches by using the Search portlet for enterprise search from within WebSphere Portal. Searches submitted from the sample search Web application, ESSearchApplication, will not have the proper credentials and cannot verify the user's authority to access documents.

- Specify information that enables the crawler to communicate with a proxy server, if the WCM sites use a proxy server to serve documents.
- Specify authentication data that enables the crawler to access documents that are protected by single sign-on (SSO) security.
- Specify information about a keystore file so that the crawler can use the Secure Sockets Layer (SSL) protocol to connect to the WCM sites.
- Specify the language and code page of the documents to be crawled.
- Specify options for crawling and searching metadata in Web Content Management documents.

- Specify schedules for crawling the Web Content Management sites.

Figure D-1 on page 507 through Figure D-15 on page 521 describe the creation and configuration of the WCM crawler.

After logging in to the administration console, from the **Collections** view in Edit mode, under the **Crawl** tab for the NWINSURANCE collection, click **Create Crawler**, as shown in Figure D-1 on page 507. Select **Web Content Management** from the drop-down list for Crawler type and click **Next** in Figure D-2 on page 508.

Provide details of the WCM crawler in Figure D-3 on page 509, such as the Crawler name (NW_INSU_WCM) and Maximum number of documents to crawl (2000). Click **Next** to provide further details in Figure D-4 on page 510, such as the WCM site URLs (nile.itsosj.sanjose.ibm.com:10038/wps/wcmsearchseed/searchseed?site=insurance&lib=web+content), and the credentials to access it, such as the User DN (uid=wpsadmin,cn=users,ou=itso,o=ibm) and Password. Click **Next** to proceed to the WCM Crawl Space page.

Click **Edit document-level security** (in Figure D-5 on page 511) to view and optionally modify the Document-level security options, such as the Validate current credentials during query processing and the Options for indexing access control, as shown in Figure D-6 on page 512. Click **OK** to save any changes made.

Since we want to leverage WCM support for SSO, click **Specify the SSO authentication type** in Figure D-7 on page 513 to configure the required information to access WCM documents that are protected by single sign-on (SSO) security, as described in Figure D-8 on page 514 through Figure D-14 on page 520. Select **Form-based authentication** from the drop-down list for SSO authentication type and the WebSphere Portal Server Login form URL (http://kazan.itsosj.sanjose.ibm.com/wps/portal) and Form name (login.jsp) in Figure D-8 on page 514. Click **Add Field** to identify each field in the form that the crawler must provide to be authenticated; in this case, it is the user ID (wpsadmin) and password fields with the corresponding values, as shown in Figure D-9 on page 515 through Figure D-12 on page 518. Click **OK** in Figure D-12 on page 518 to test the success of the SSO authentication configuration.

Click **Test the configuration** in Figure D-13 on page 519 to test the crawler's ability to connect to the WCM URL specified. A successful connection results in the "FFQM0350I Success..." message shown in Figure D-14 on page 520. Click **Next** to specify the crawl schedule.

Since we are going to manually schedule this crawler, we chose the defaults and clicked **Finish** in Figure D-15 on page 521 to complete the configuration of the WCM crawler.

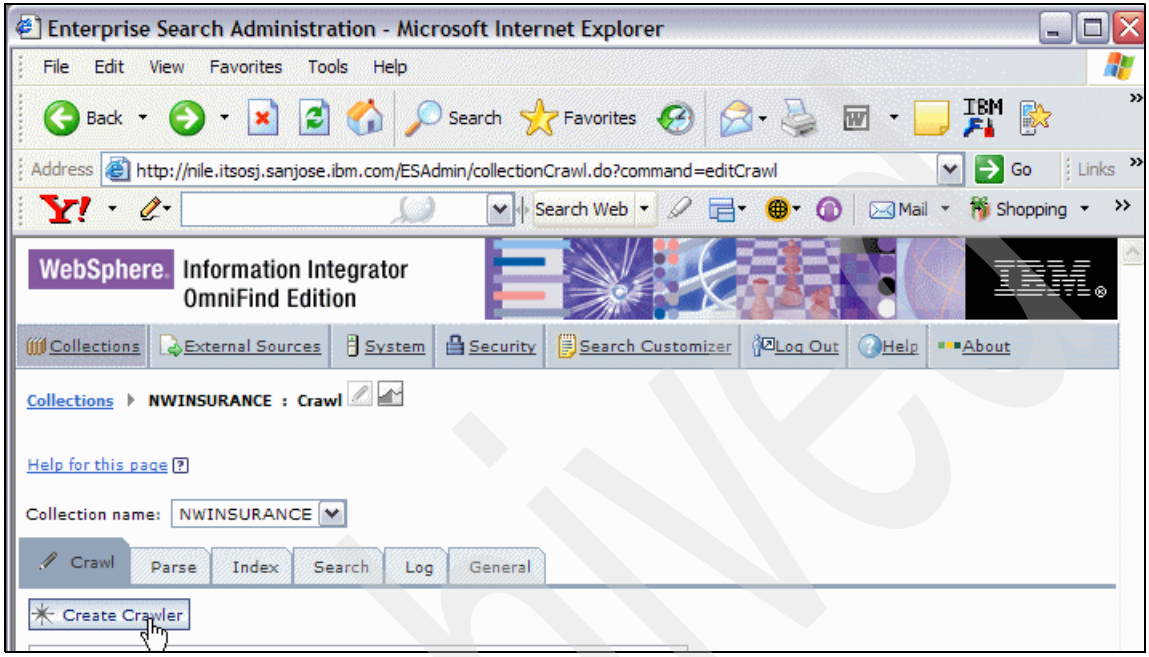

Figure D-1 Create Crawler

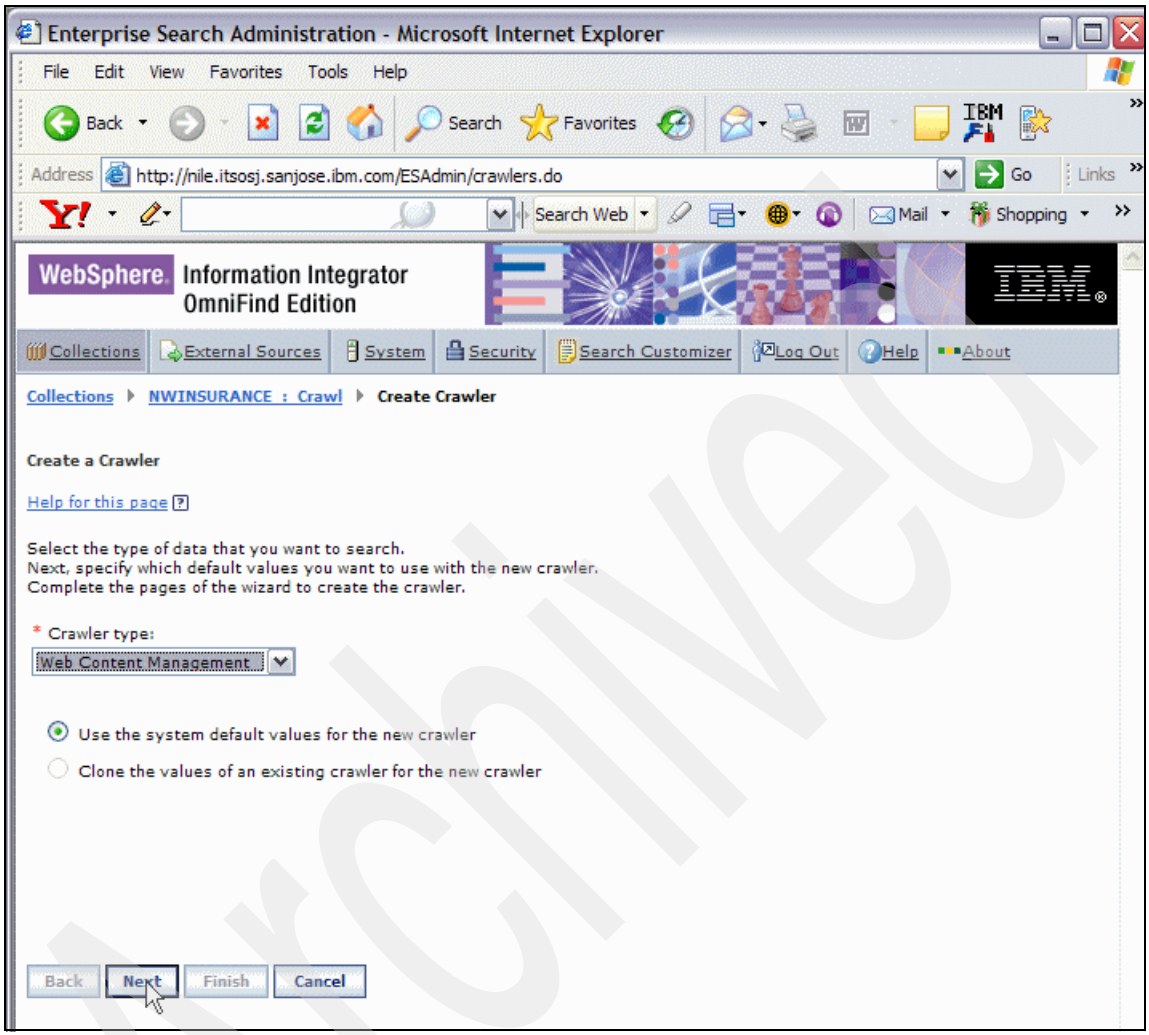

Figure D-2 Web Content Management crawler type

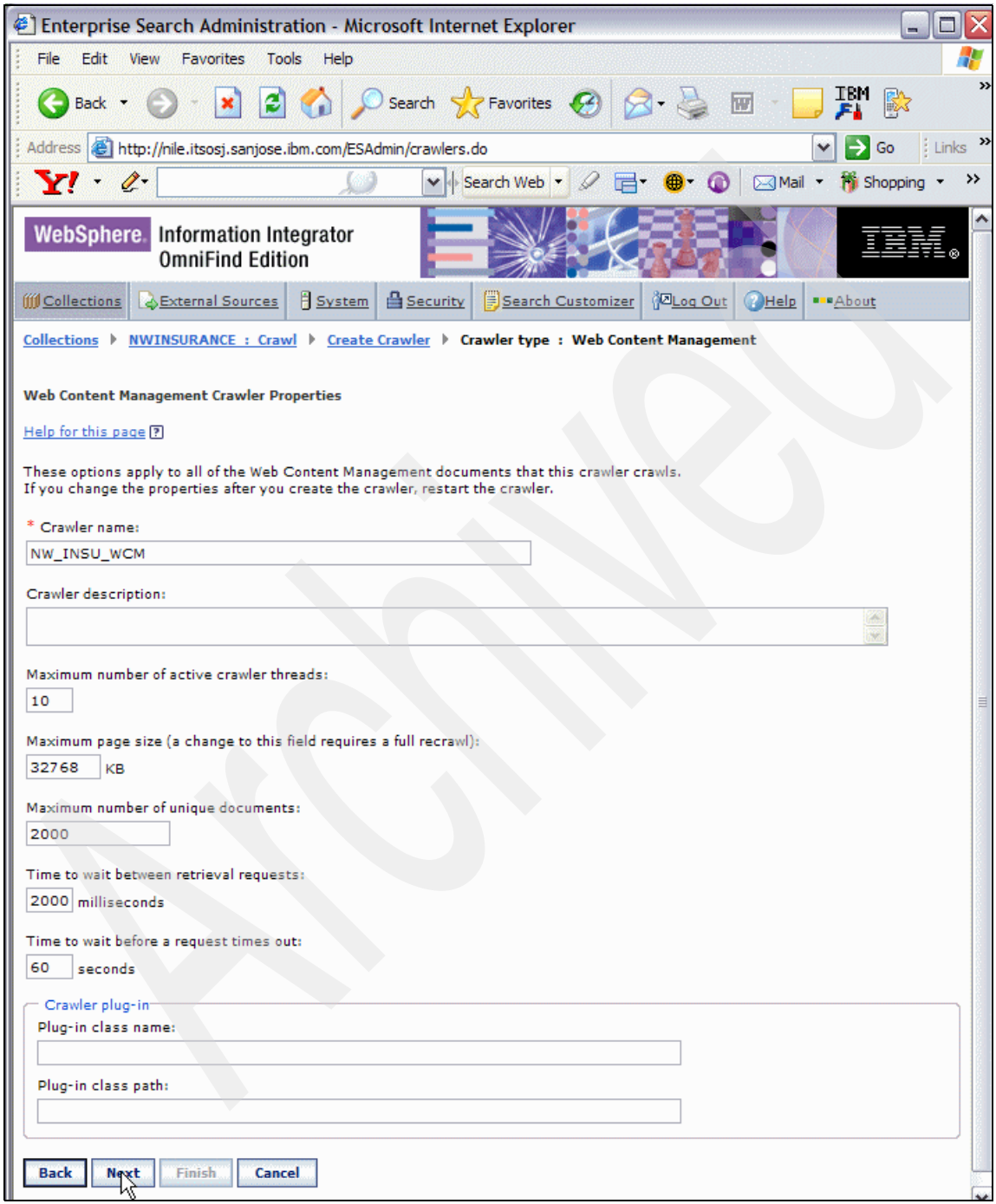

Figure D-3 Crawler properties

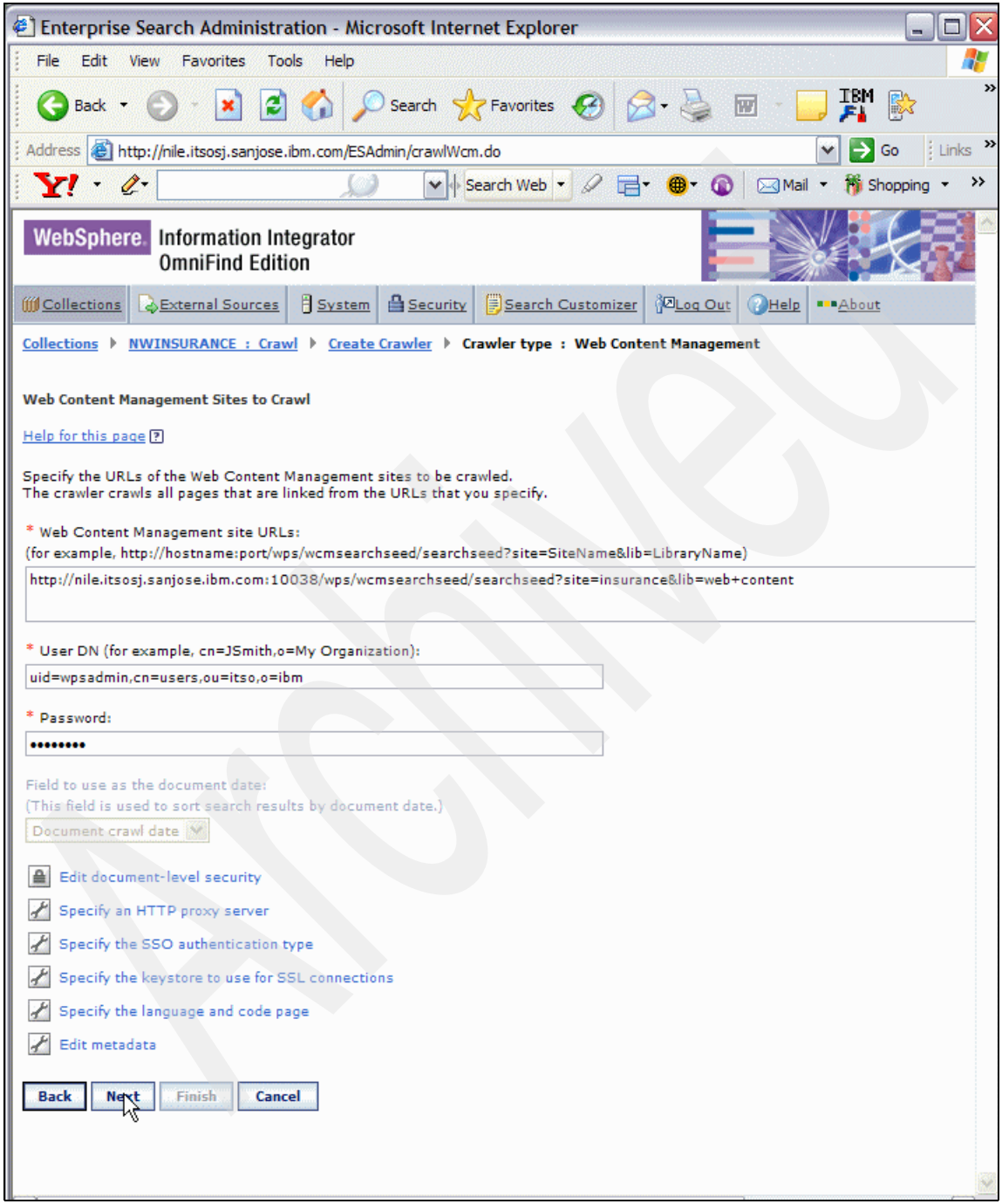

Figure D-4 WCM Sites to Crawl

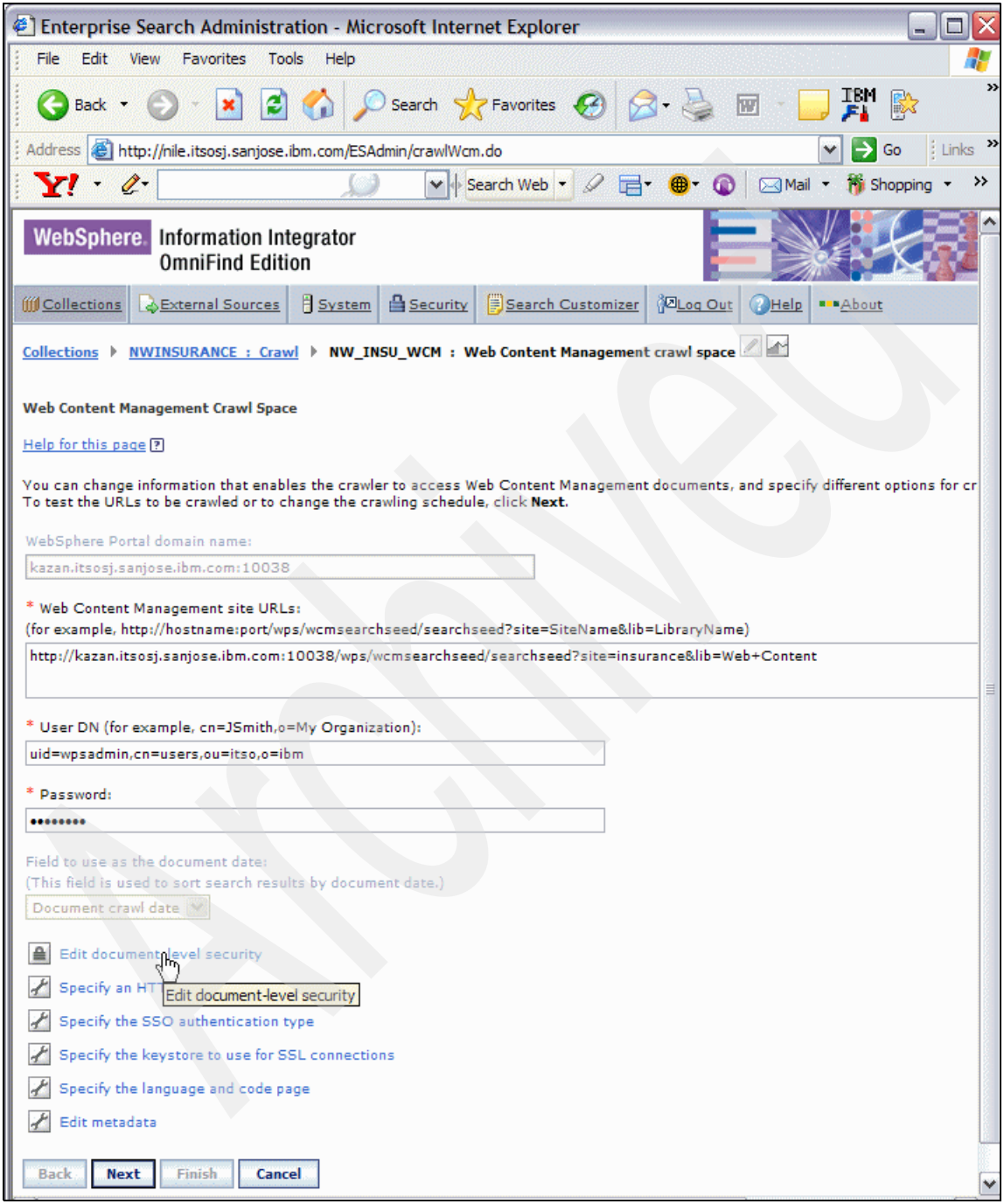

Figure D-5 WCM Crawl Space

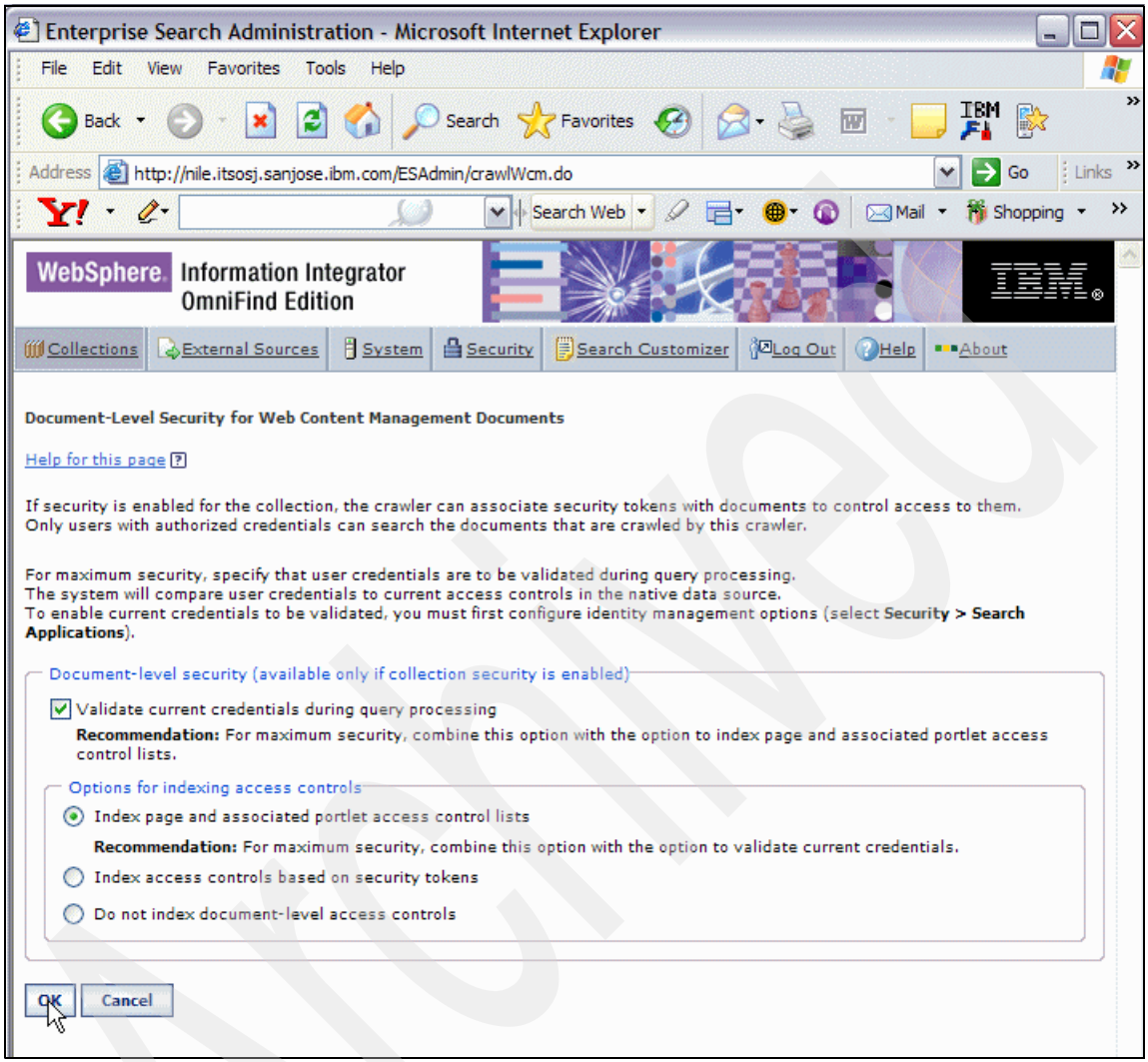

Figure D-6 Document-Level Security for WCM Documents

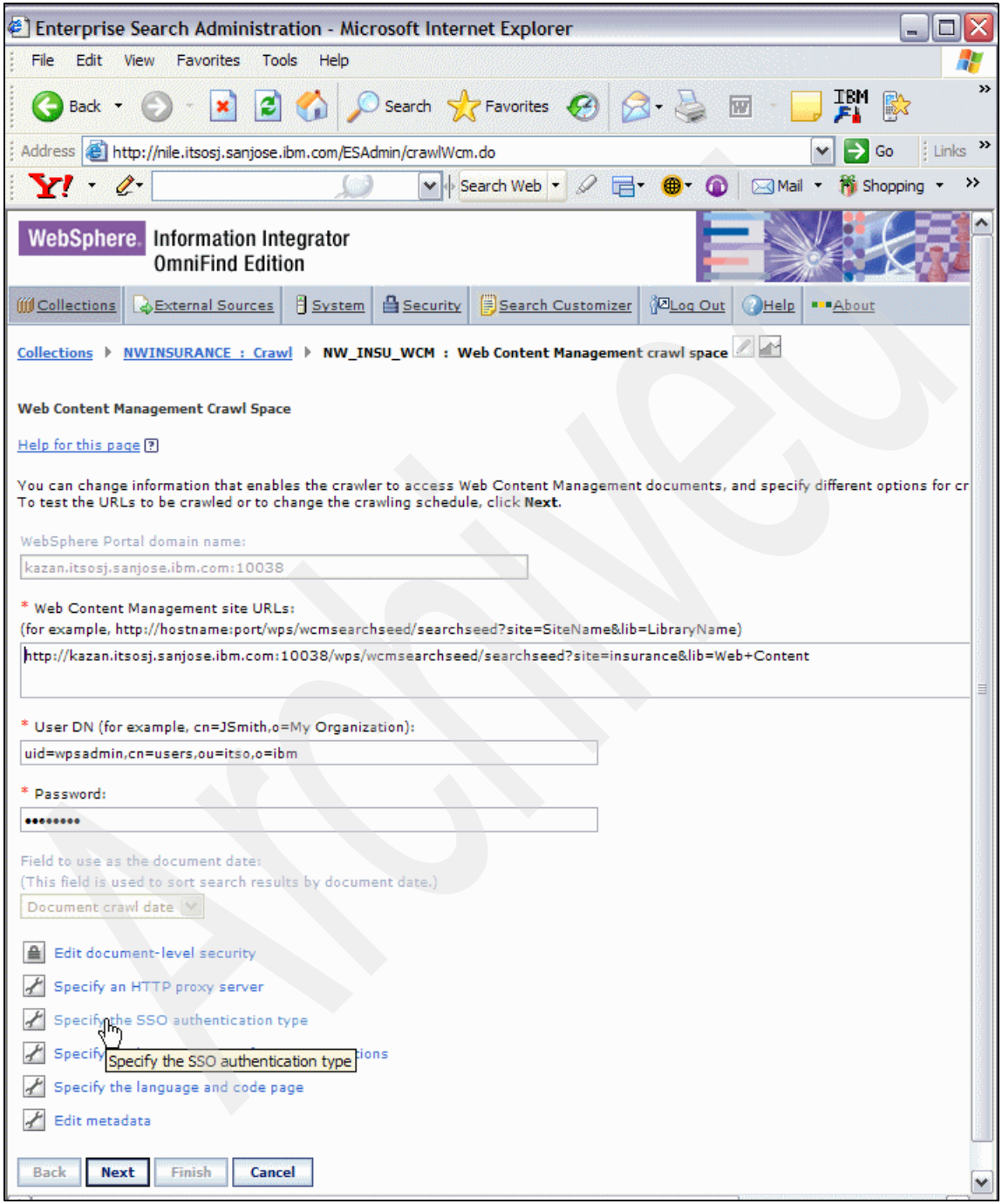

Figure D-7 Specify SSO authentication type 1/6

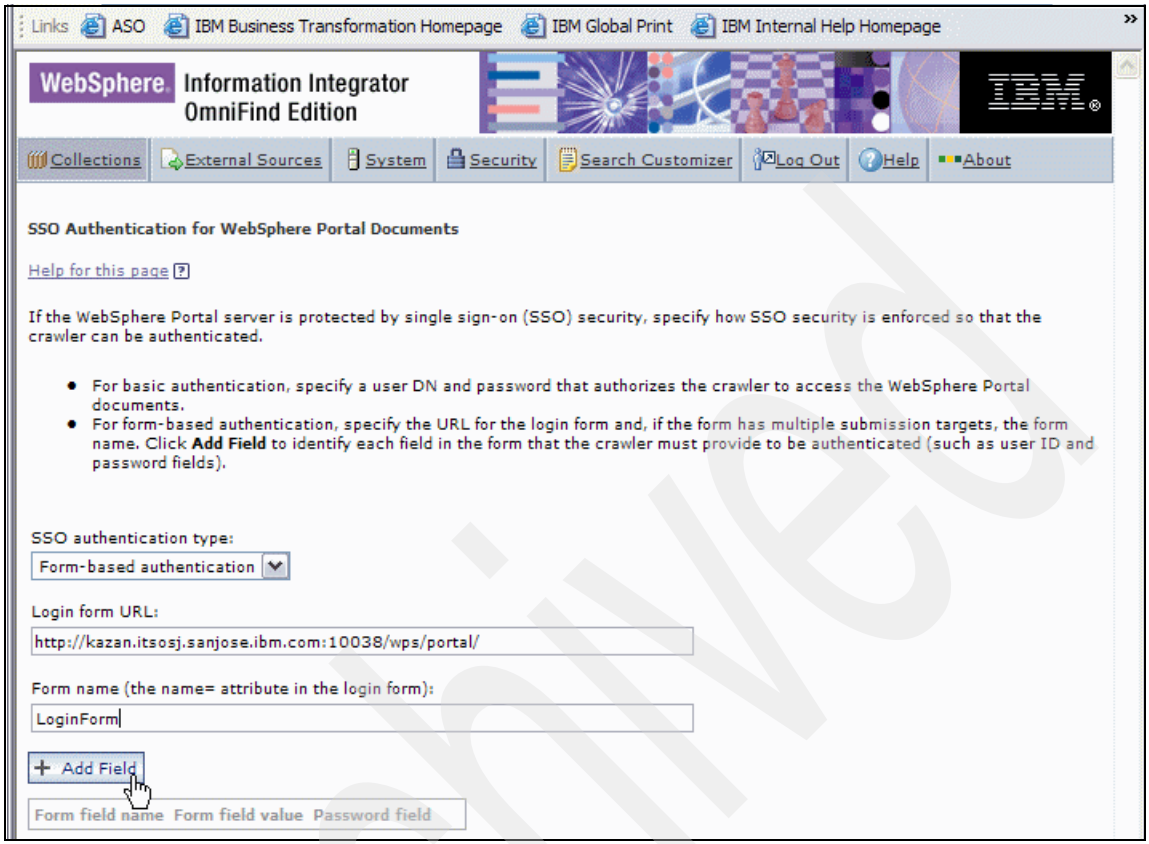

Figure D-8 Specify SSO authentication type 2/6

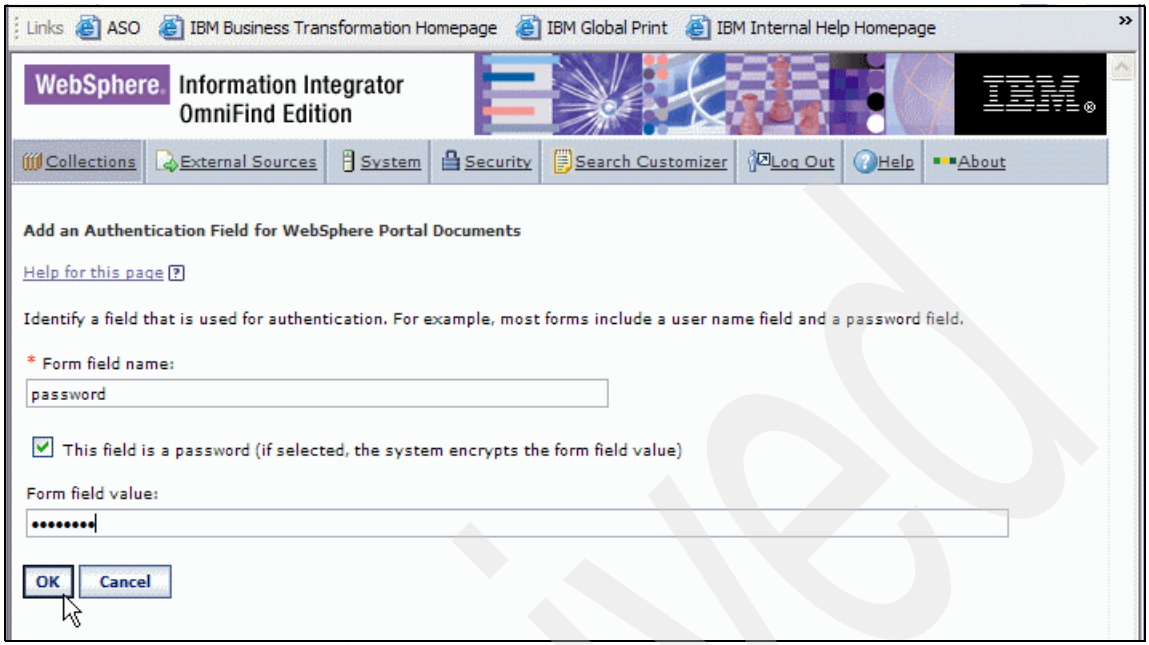

Figure D-9 Specify SSO authentication type 3/6

Figure D-10 Specify SSO authentication type 4/6

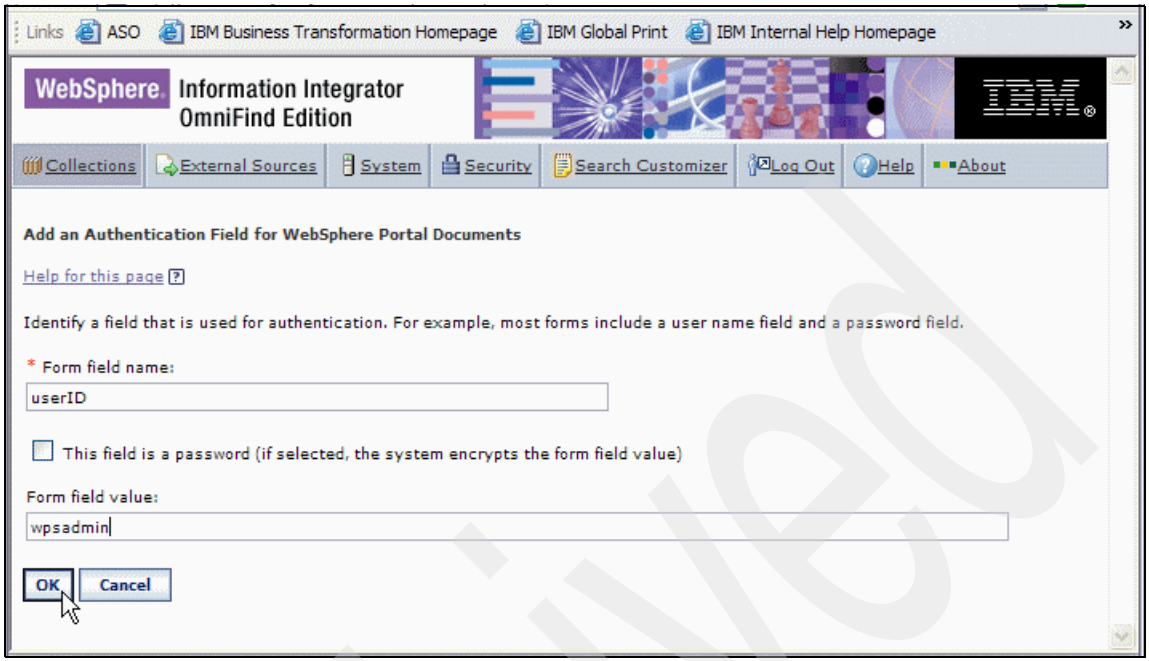

Figure D-11 Specify SSO authentication type 5/6

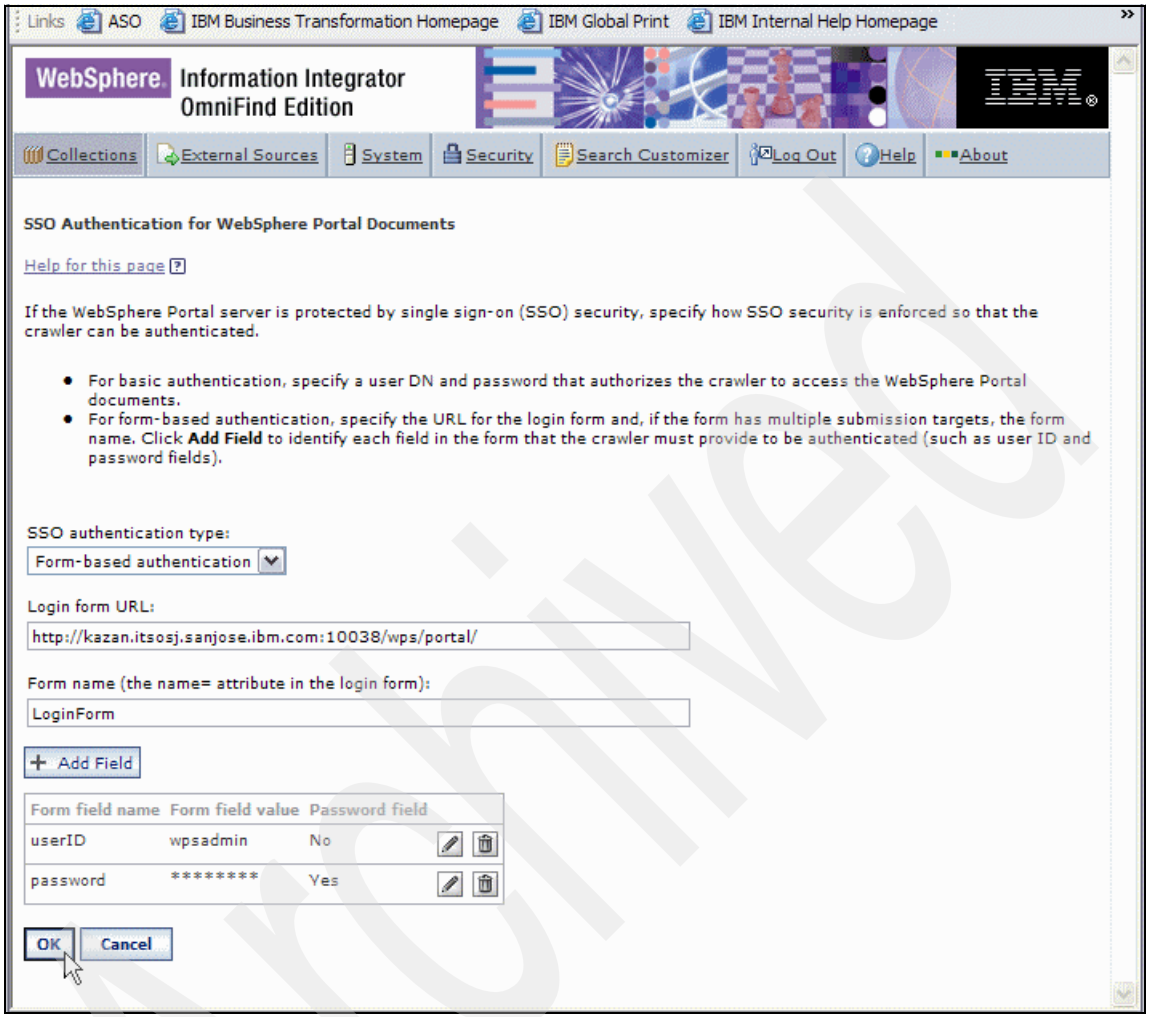

Figure D-12 Specify SSO authentication type 6/6

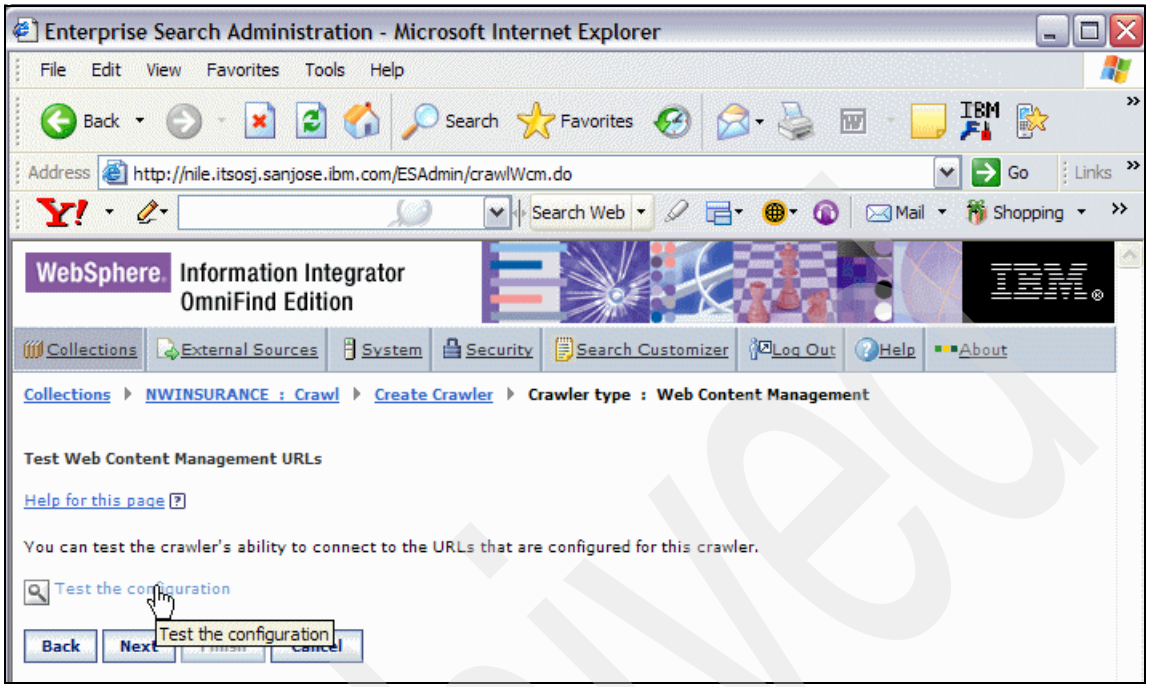

Figure D-13 Test the configuration 1/2

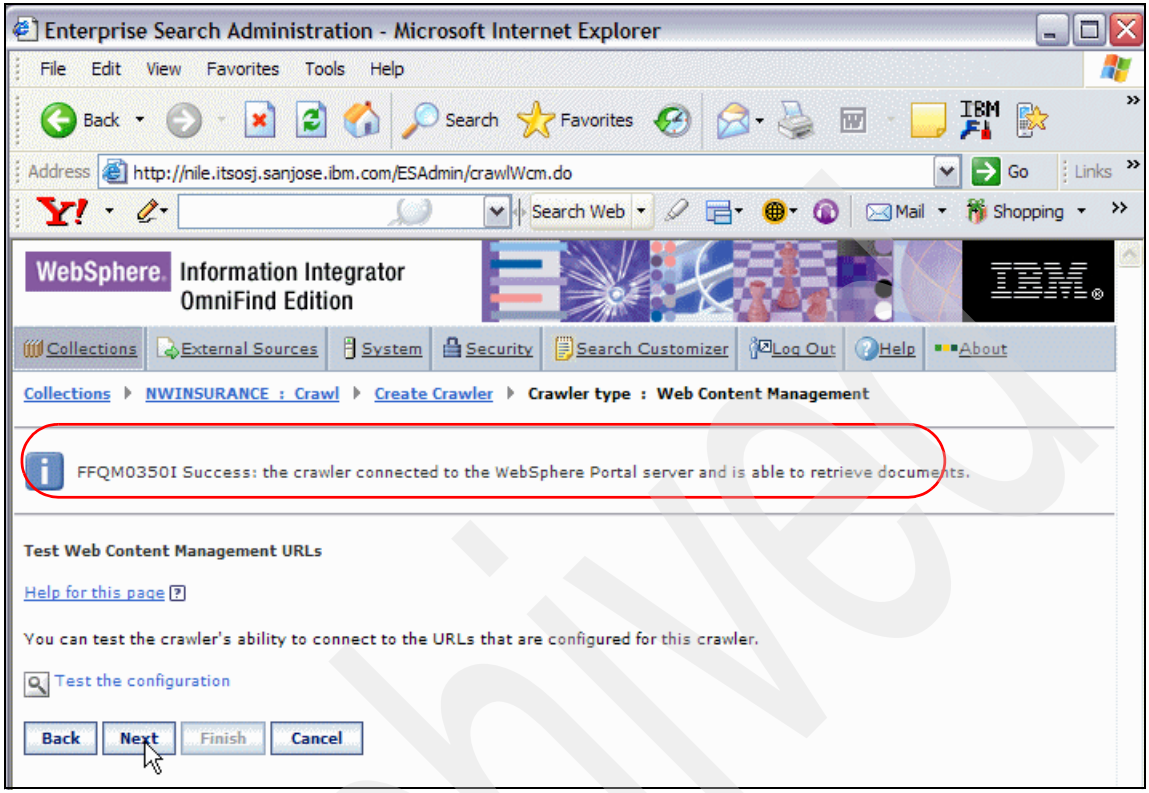

Figure D-14 Test the configuration 2/2

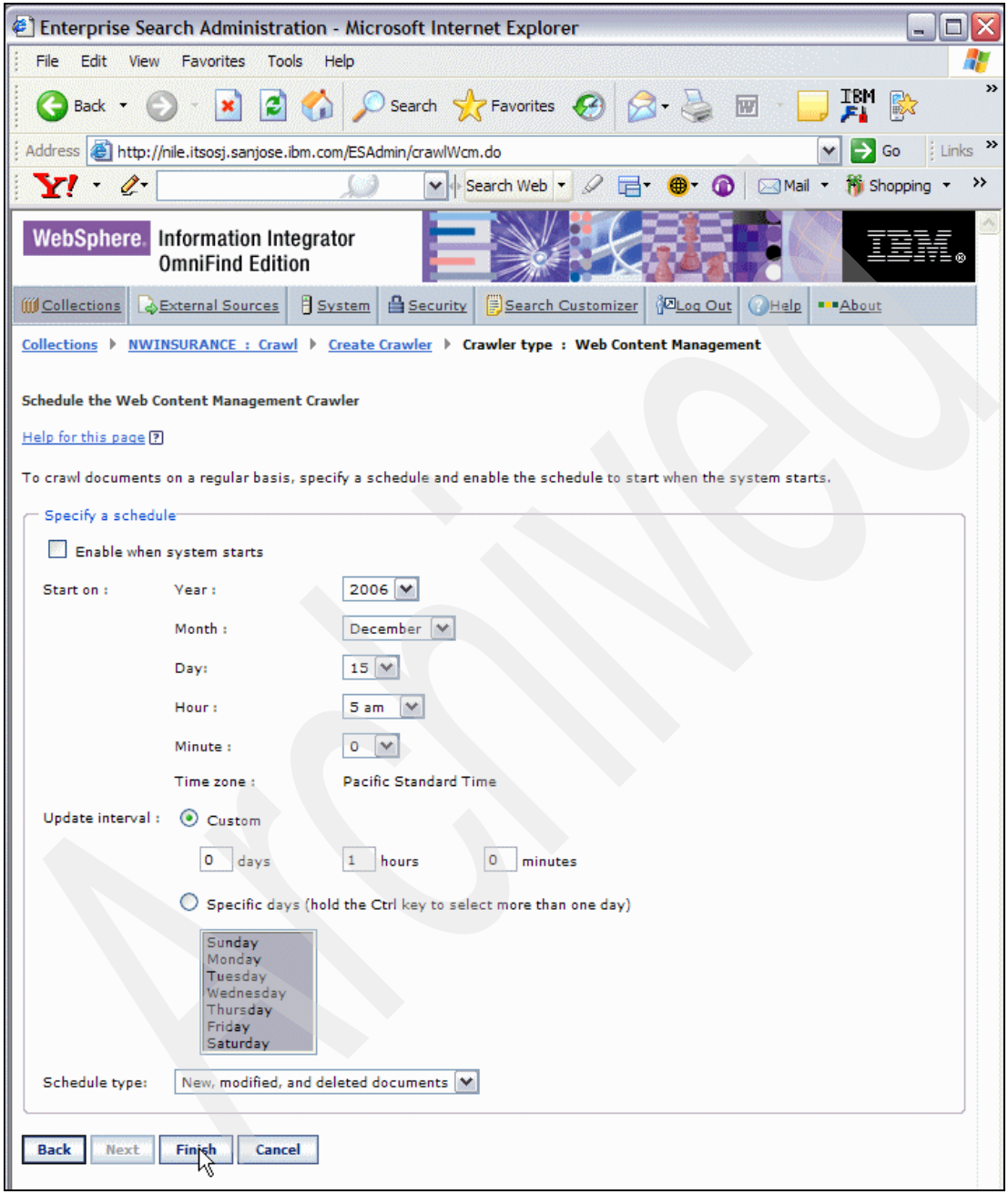

Figure D-15 Crawl schedule

Configure Portal Document Manager (with SSO) crawler

You can use the Content Edition crawler to crawl a number of repository types, such as Documentum, FileNet Panagon Content Services, FileNet P8 Content Manager, Hummingbird Document Management (DM), Microsoft SharePoint, OpenText Livelink, and Portal Document Manager (PDM).

When you configure the crawler, you specify options for how the crawler is to crawl all repositories in the crawl space. You also select the item classes that you want to crawl in each repository. How you prepare for crawling repositories depends on whether you plan to use direct mode or server mode to connect to the data to be crawled.

- If you use direct mode, you must configure a connector in WebSphere Information Integrator Content Edition.

  In direct mode, the crawler uses a WebSphere Information Integrator Content Edition connector that is installed on the crawler server when IBM OmniFind Enterprise Edition is installed. The crawler uses content integration APIs to connect directly to the repositories to be crawled. Not all content integration server functionality is available when the content integration server runs in direct mode.

  To configure the system so that the crawler can access repositories in direct mode:

  a. Confirm that the VBR_HOME and JAVA_HOME environment variables in the iice_install_root/bin/config.sh file (on UNIX) or iice_install_root\bin\config.bat file (on Microsoft Windows) specify the correct directory.

  b. To configure the WebSphere Information Integrator Content Edition administration console to run in direct mode, add the vbr.as.operationMode=direct Java system property to the iice_install_root/bin/Admin.bat file (on UNIX) or iice_install_root\bin\Admin.bat file (on Windows).

  c. Start the WebSphere Information Integrator Content Edition administration console in direct mode and configure the connector for the IBM OmniFind Enterprise Edition crawler server.

  d. Select the direct mode option when you use the enterprise search administration console to configure the Content Edition crawler.

  See the WebSphere Information Integrator Content Edition documentation for information about running the content integration server in direct mode and how the functionality differs from a content integration server that runs in server mode.

- If you use server mode, you must run a script on the crawler server. This script, which is provided with WebSphere Information Integrator OmniFind Edition, enables the Content Edition crawler to communicate with WebSphere Information Integrator Content Edition servers.

  In server mode, the WebSphere Information Integrator Content Edition connector that the crawler uses to access data is installed as an enterprise application on WebSphere Application Server, and the crawler accesses repositories through the server. This approach enables you to take advantage of J2EE™ application server environments.

  The Content Edition crawler uses Java libraries of WebSphere Information Integrator Content Edition as a Java client. In server mode, these Java libraries require EJB™-related Java libraries of WebSphere Application Server. To ensure that the Content Edition crawler can work with the Java libraries, you must run a setup script that WebSphere II OmniFind Edition provides on the crawler server after you install WebSphere Application Server. WebSphere Information Integrator Content Edition is installed on the crawler server when IBM OmniFind Enterprise Edition is installed. To be able to use the Content Edition crawler in server mode, you must copy the vbr_access_services.jar file from the WebSphere Information Integrator Content Edition server to the crawler server.

  To configure the crawler server so that it can crawl WebSphere Information Integrator Content Edition repositories:

  a. If WebSphere II OmniFind Edition is installed in a multiple server configuration, install and bind the WebSphere Application Server Java libraries.

  b. On the crawler server, run the setup script for the Content Edition crawler:

     i. Log in as the enterprise search administrator.

     ii. Start the script escrvbr.sh, which is installed in the $ES_INSTALL_ROOT/bin directory), and answer the prompts.

  c. On the crawler server, stop and restart the enterprise search system:

     ```
     esadmin system stopall esadmin system startall
     ```

  d. Copy the vbr_access_services.jar file from the WebSphere Information Integrator Content Edition server to the crawler server.

Creating the crawler involves the following tasks:

- Specify properties that control how the crawler operates and uses system resources. The crawler properties control how the crawler crawls all of the repositories in the crawl space.

- Specify whether the crawler uses direct mode or server mode to access repositories. For server mode, you must also specify information that enables the crawler to access the Web application server.
- Select the repositories that you want to crawl.
- Specify user IDs and passwords that enable the crawler to access content in the selected repositories.
- Set up a schedule for crawling the repositories. Select the item classes that you want to crawl in each repository.
- Specify options for making the properties of item classes searchable. For example, you can exclude certain types of documents from the crawl space or specify that you want to crawl a particular version of a repository.
- Configure document-level security options. If security was enabled when the collection was created, the crawler can associate security data with documents in the index. This data enables search applications to enforce access controls based on the access control lists or security tokens.

  For Documentum, FileNet Panagon Content Services, Hummingbird DM, Portal Document Manager, and SharePoint item classes, you can also select an option to validate user credentials when a user submits a query. In this case, instead of comparing user credentials to indexed security data, the system compares the credentials to current access control lists that are maintained by the original data source. This type of current credential validation is not available for the other repository types.

Figure D-16 on page 525 through Figure D-24 on page 533 describe the creation and configuration of the Content Edition crawler.

After logging in to the administration console, navigate to the Collections view in Edit mode, and from the Crawl tab in Figure D-16 on page 525, click **Create Crawler**. Select **Content Edition Crawler** type and click **Next** in Figure D-17 on page 526.

Provide details of the Content Edition crawler in Figure D-18 on page 527, such as the Crawler name (pdm_sso) and Maximum number of documents to crawl (2000). Click **Next** to specify the Content Edition access mode. Select **Direct mode (access repositories through a connector on the crawler server)** in Figure D-19 on page 528 and click **Next** to select the Content Edition repositories to crawl.

Figure D-20 on page 529 shows the selected Repositories to crawl (IBM WebSphere Portal Document Manager Connector Portal Document Manager) obtained by first discovering available repositories ('*' in the Repository name or pattern followed by a click of **Search for repositories**, which lists all those found with the matching criteria in the Available repositories box and then copying

those of interest to the Repositories to crawl box). Click **Next** in Figure D-20 on page 529 to specify Content Edition Repository User IDs (and password) to access the selected repository, and select **Enabled for SSO** from the Single sign-on (SSO) field and click **Apply** in the Repository User ID box, as shown in Figure D-21 on page 530. Click **Next** to specify the crawl schedule (Figure D-22 on page 531). Since we chose to schedule the crawls manually, click **Next** in Figure D-22 on page 531. The next step is to identify all the item classes to be crawled. Figure D-23 on page 532 shows the selected Item Classes to crawl [lotus:collaborativeDocument(Content) and icm:documentLibrary(Folder)] obtained by first discovering available item classes ('*' in the Item class name or pattern followed by clicking **Search for item classes**, which lists all those found with the matching criteria in the Available item classes box and then copying those of interest to the Item classes to crawl box). Click **Next** in Figure D-23 on page 532 to view and optionally modify the Edit options and Edit security parameters for the selected item classes to crawl, as shown in Figure D-24 on page 533. Click **Finish** to complete the creation and configuration of this crawler.

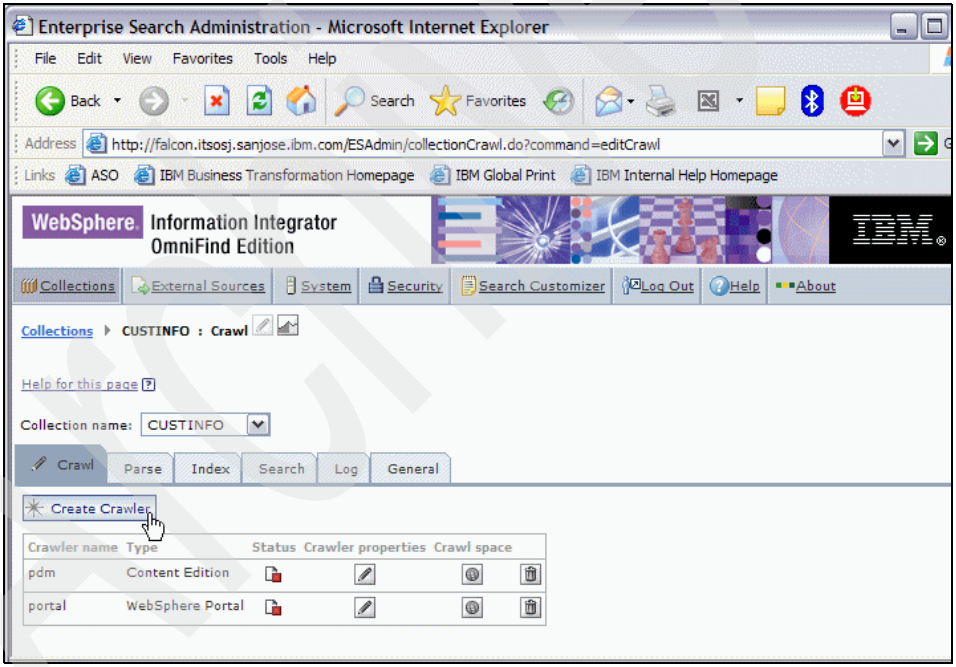

Figure D-16 Create Crawler

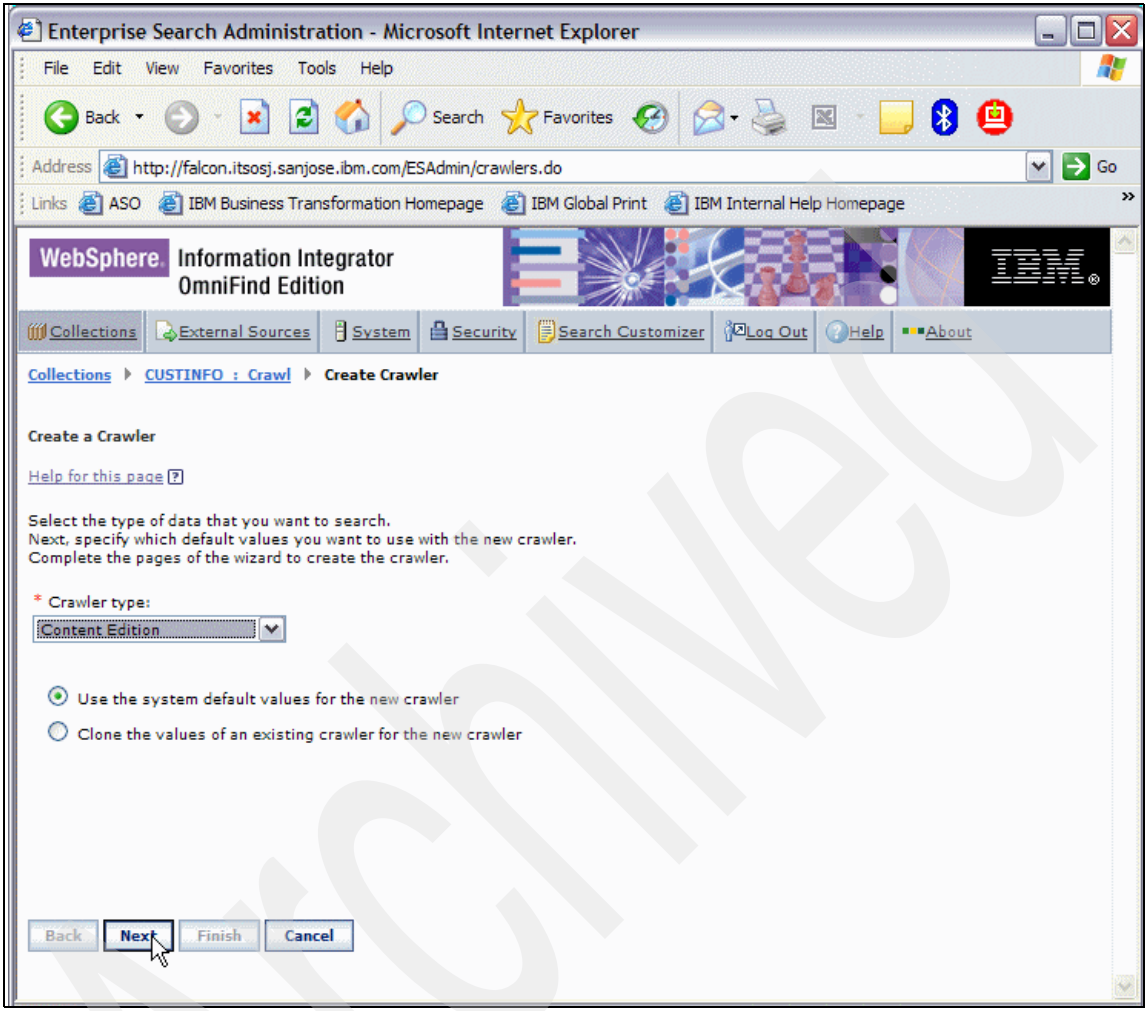

Figure D-17 Content Edition crawler type

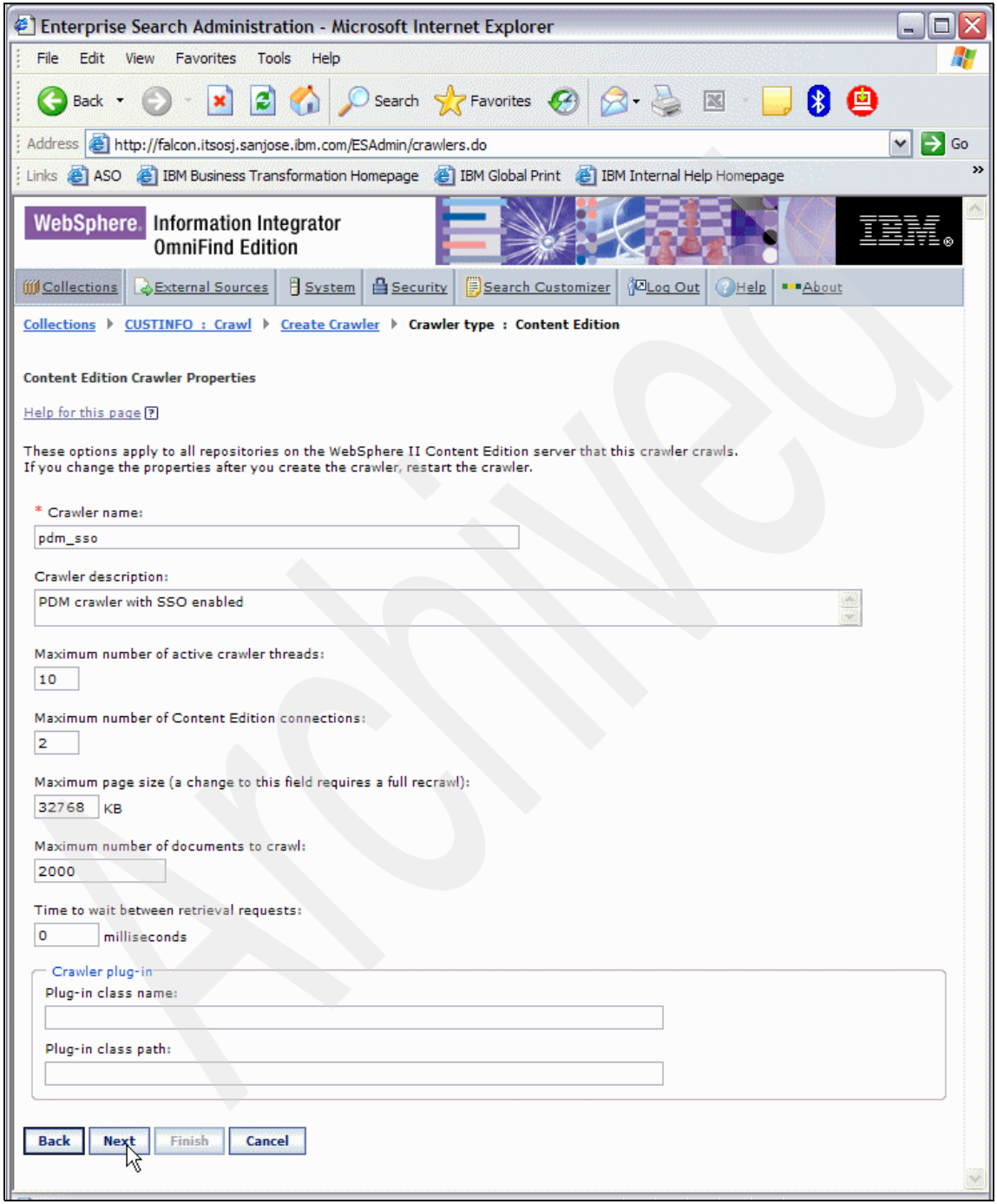

Figure D-18 Crawler properties

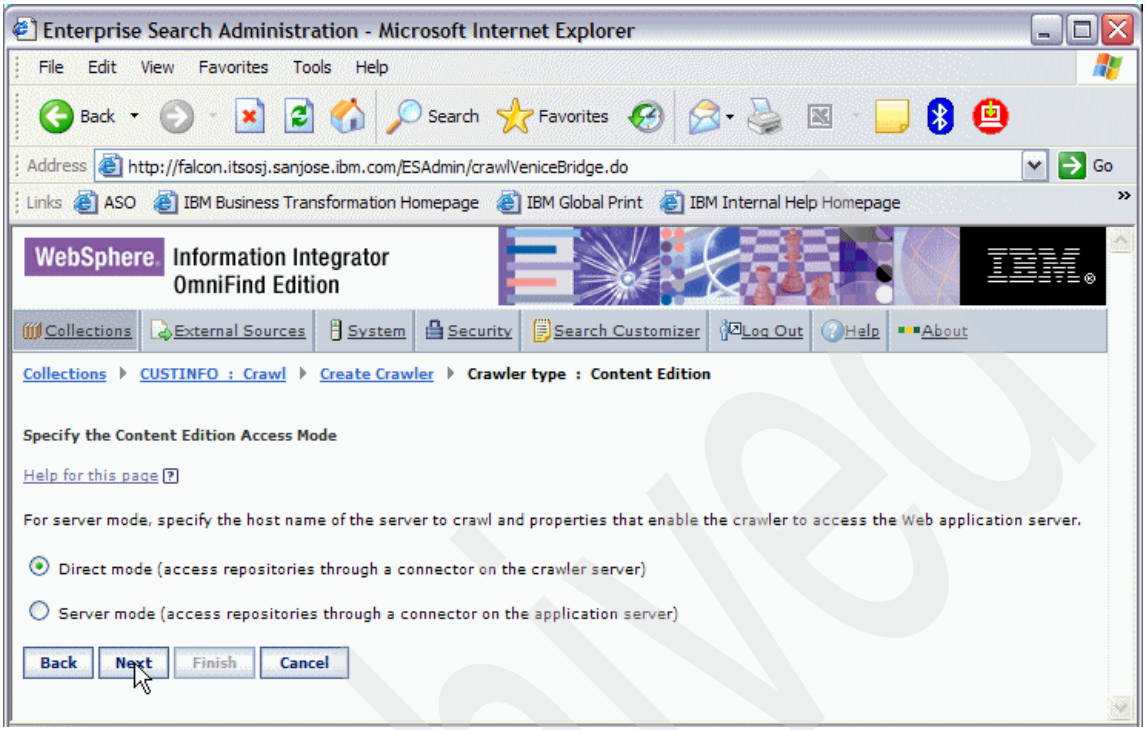

Figure D-19 Specify the Content Edition Access Mode

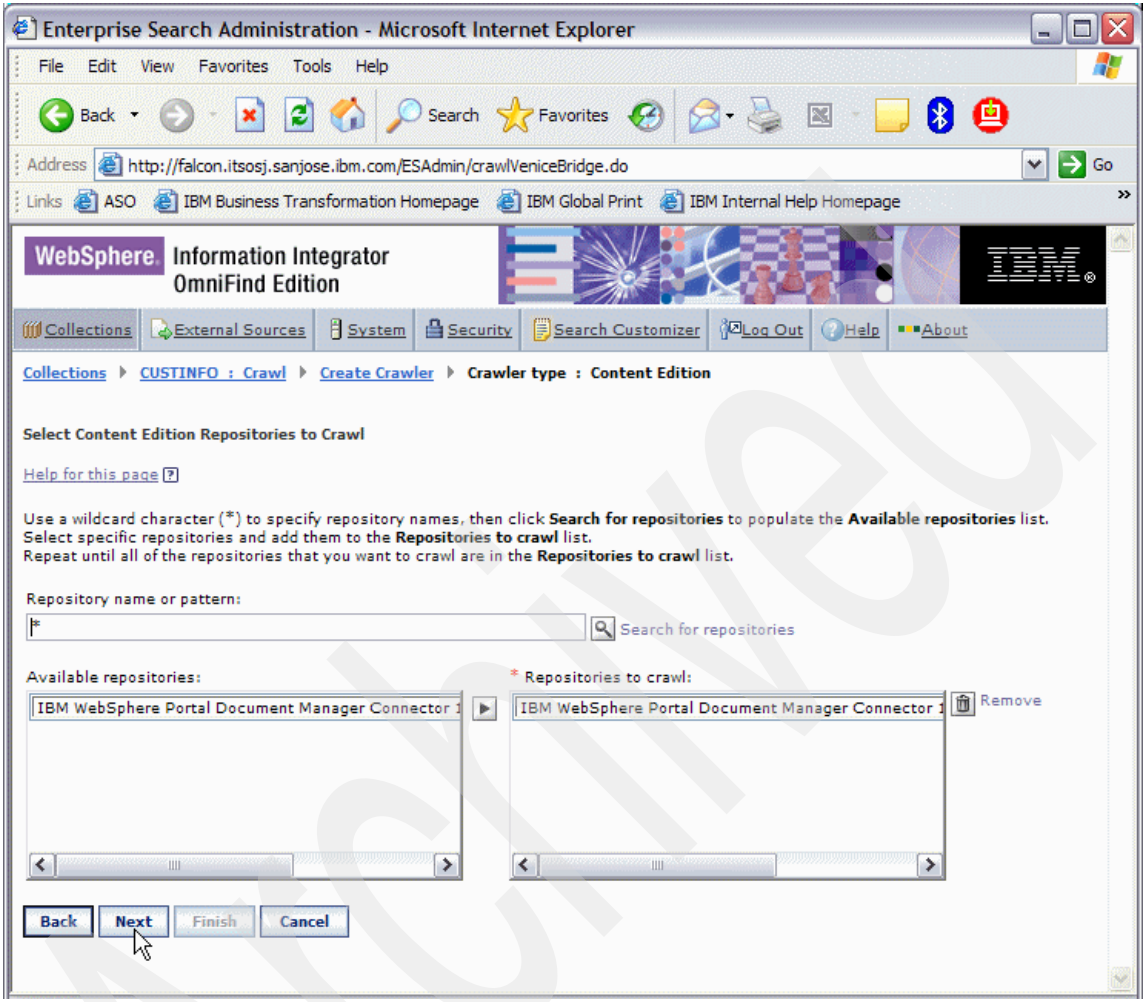

Figure D-20 Select Content Edition Repositories to Crawl

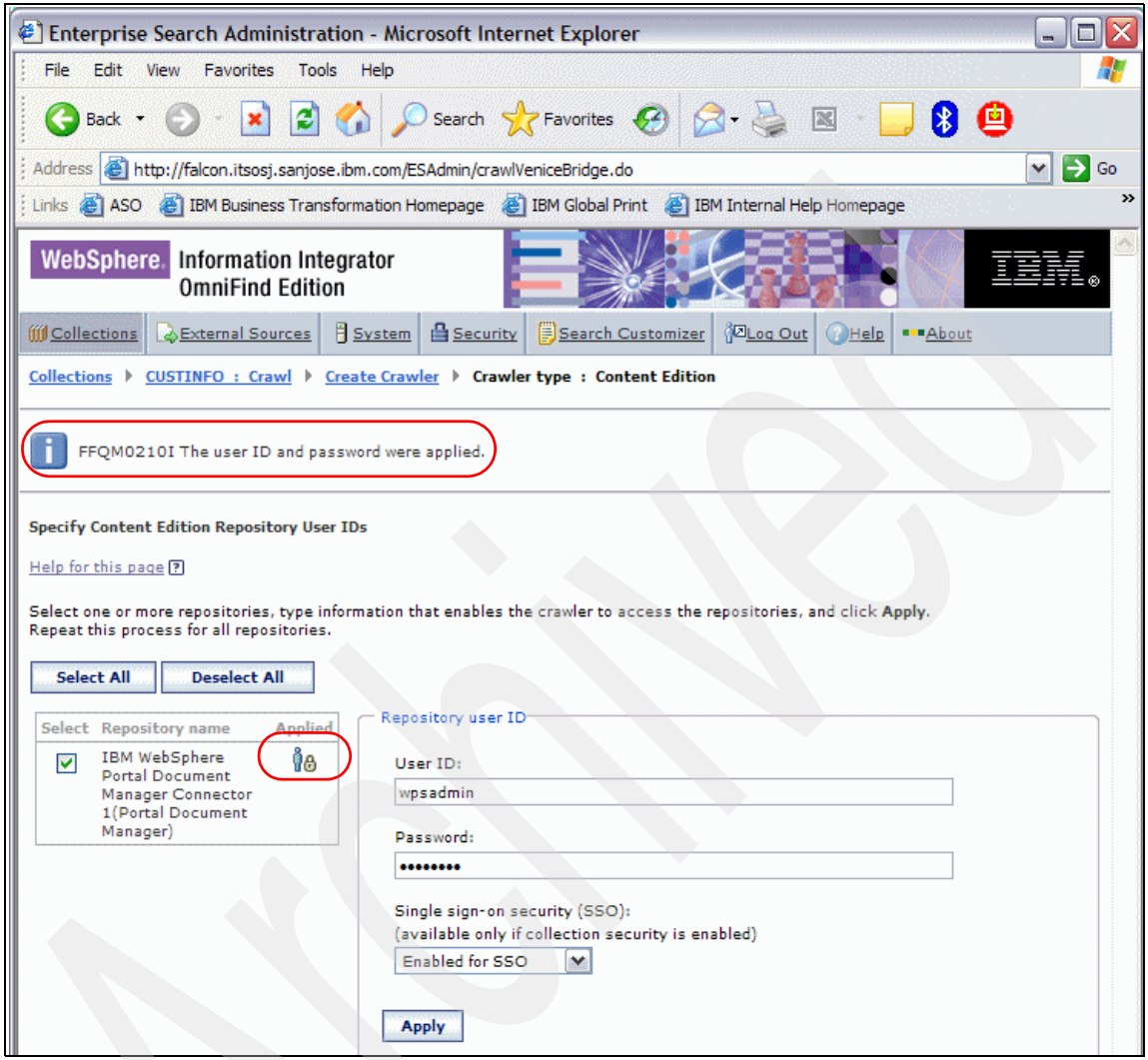

Figure D-21 Specify Content Edition Repository User IDs

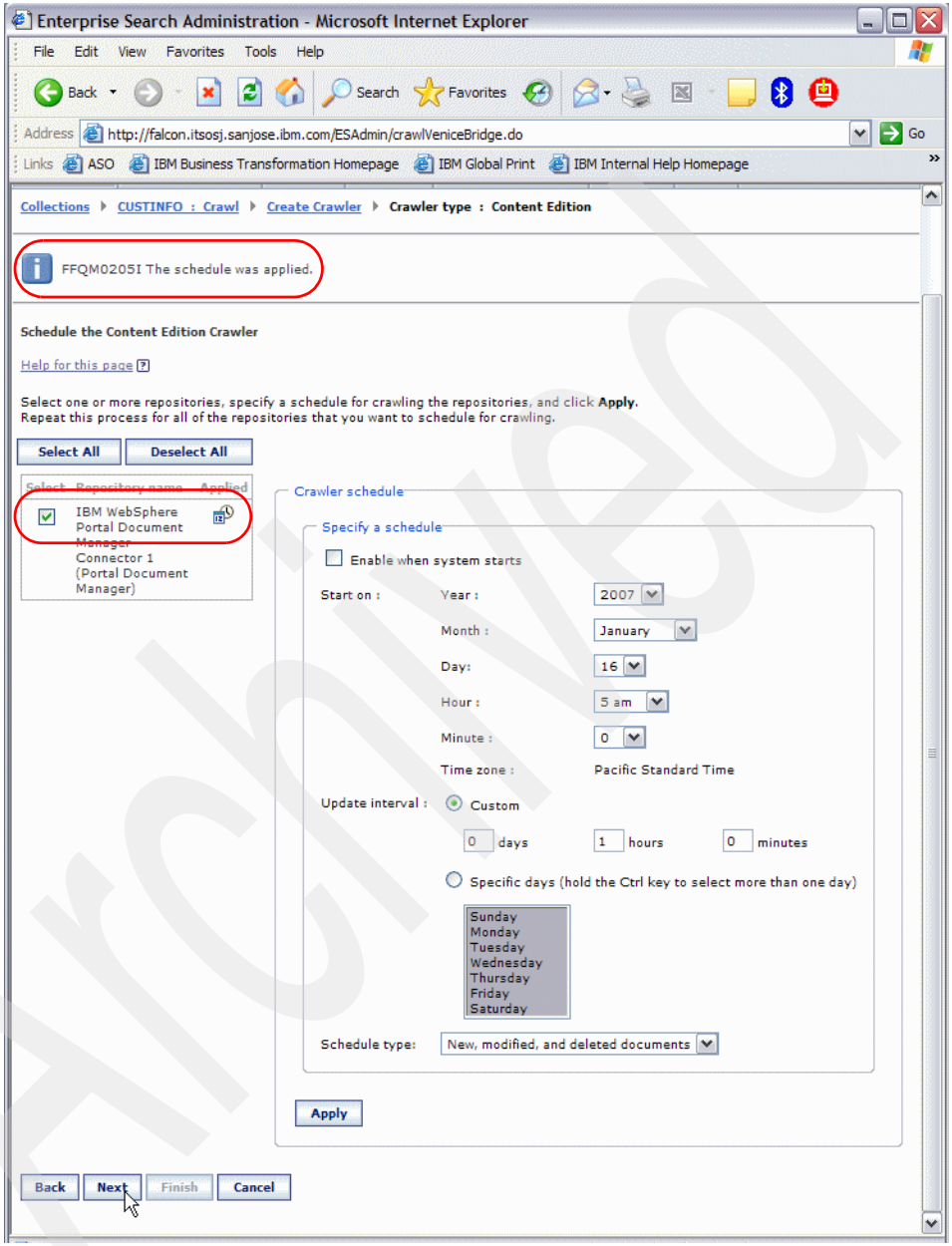

Figure D-22 Crawl schedule

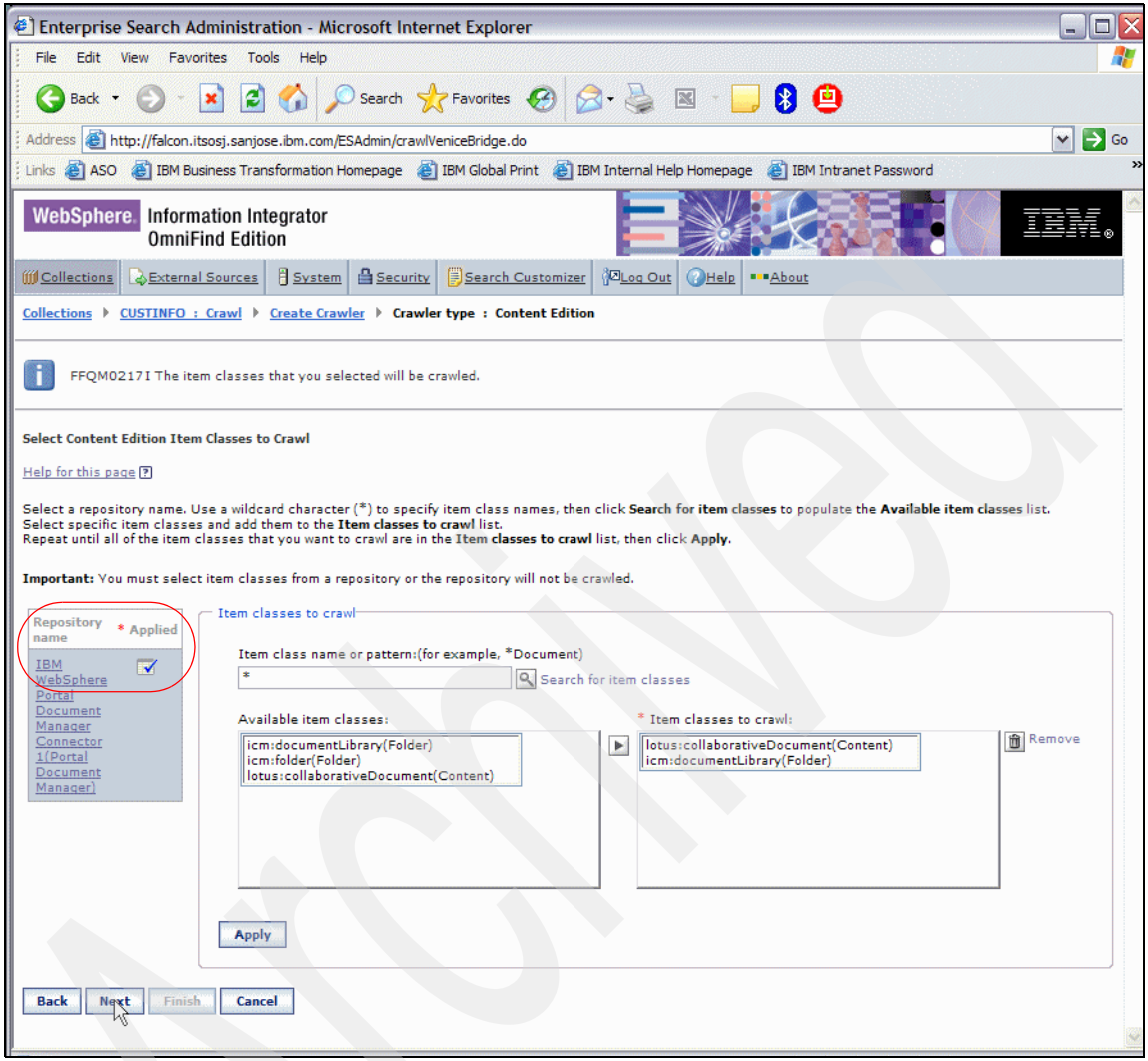

Figure D-23 Select Content Edition Item Classes to Crawl

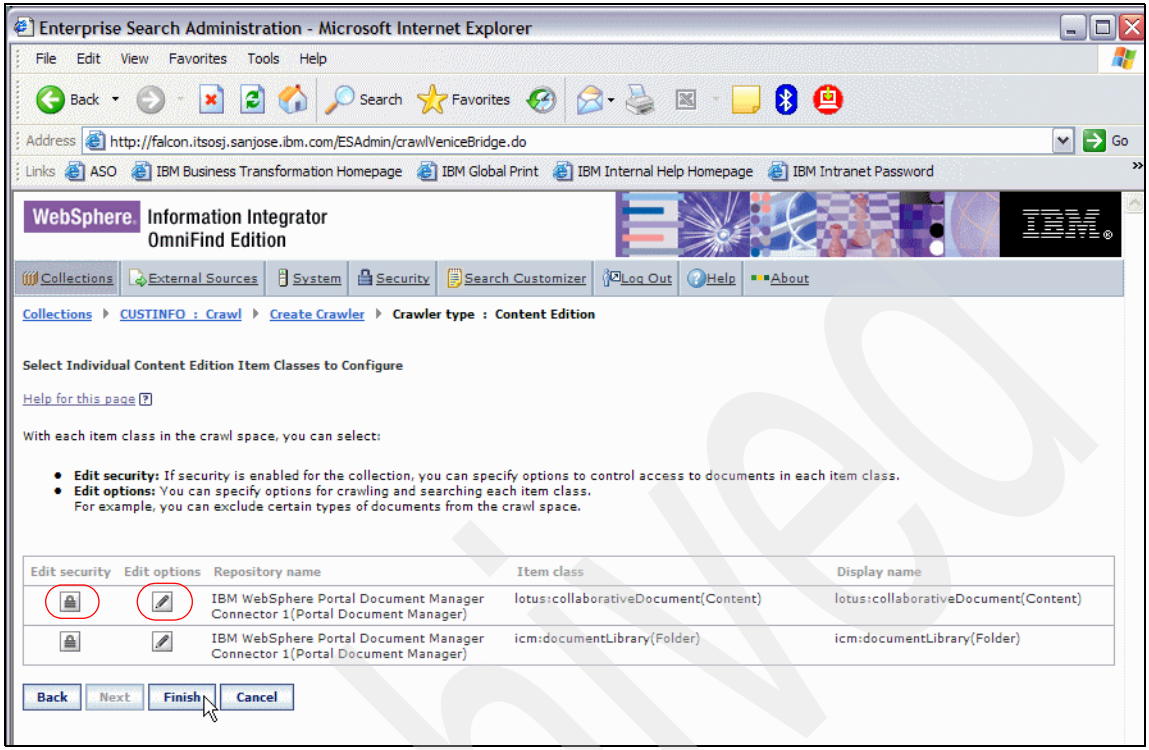

Figure D-24 Select Individual Content Edition Item Classes to Configure

E

Migration considerations

In this appendix, we describe the steps in migrating a single server OmniFind V8.3 Windows system to OmniFind Enterprise Edition V8.4.

Introduction

IBM OmniFind Enterprise Edition V8.4 has significant new functionality enhancements, as described in 1.4, “What is new in V8.4” on page 31, that provides current OmniFind installations a great incentive to migrate to the new version.

However, there are a few considerations in migrating to IBM OmniFind Enterprise Edition V8.4:

- You can upgrade only from Version 8.3 to Version 8.4.
- You cannot upgrade to a different system configuration:
  - If you currently run enterprise search on a single server, you must install the new software on a single server.
  - If you currently run enterprise search in a four server configuration, you must install the new software on four servers.
  - You cannot upgrade from a single server or multiple server configuration to a two server configuration.
- To install enterprise search in a two server configuration, you must install a new system.
- NNTP crawler is not migrated. You have to delete and recreate this crawler.
- Documents that were crawled but not parsed before migration are not migrated when you install OmniFind Enterprise Edition V8.4. Therefore, you should ensure that all crawled documents have been parsed before you install OmniFind Enterprise Edition V8.4.
- If you want to take advantage of Single Sign-On, or Notes advanced IOR settings, then you must recreate your crawlers and re-parse and re-index the crawled data.
- In OmniFind Enterprise Edition V8.4, the default query behavior has been changed. In OmniFind V8.3, there was a query relaxation phase that was enabled by default; this is now disabled by default when you create a new collection in OmniFind Enterprise Edition V8.4.

  This setting is not automatically updated during migration. To update a V8.3 collection after migrating to V8.4, you can do so manually as follows:

  a. Stop the search process for the collection.
  b. Open the %ES_NODE_ROOT%/master_config/<collection id>.runtime.node3/runtime-generic.properties file.
  c. Search for the "trevi.runtime.forceGreedyEvaluation" property and change the value to "1" and save the file.

d. Repeat for the %ES_NODE_ROOT%/master_config/<collection id>.runtime.node4/runtime-generic.properties file.

e. Finally, repeat all of the above for the other collections.

- After upgrading, you cannot return to Version 8.3.

Note: The OmniFind V8.3 search application and homegrown applications built around the OmniFind V8.3 SIAPI should be completely compatible with an OmniFind Enterprise Edition V8.4 search server in terms of search and the "security" implementation. You can therefore choose to migrate your existing applications over time and not at the same time that you migrate your OmniFind V8.3 system to OmniFind Enterprise Edition V8.4.

There are several upgrade paths that you can take to upgrade to OmniFind Enterprise Edition V8.4. The path you choose depends on the versions of prerequisite software installed on your system and whether you use WebSphere Application Server and DB2 Universal Database (DB2) for purposes other than enterprise search.

One possible option we recommend is to uninstall OmniFind V8.3 and checking the option to *leave* the data in place. Then install OmniFind Enterprise Edition V8.4 with the upgrade path. This will get rid of obsolete information, such as the older samples directories. For more information about this option and several upgrade paths, refer to *IBM OmniFind Enterprise Edition Version 8.4 Installation Guide for Enterprise Search*, GC18-9282.

Note: DB2 is not required to run OmniFind Enterprise Edition Version 8.4. You need DB2 only if you want to crawl DB2 data sources. If you do not need to crawl DB2 data sources, you can remove DB2 from your system after the OmniFind Enterprise Edition upgrade is complete.

In this appendix, we describe the steps in migrating a single server Windows 2000 platform with the following characteristics:

- OmniFind Edition V8.3 installed on:
  - Windows 2000 Advanced Server with Service Pack 4
  - WebSphere Application Server V5.1.1.3 and Network Deployment
- Two collections (NonSecurity_Collection and Security_Collection) defined as shown in Figure E-1 on page 538.
  - The NonSecurity_Collection has two crawlers defined: an NNTP crawler (NNTP_CRAWLER) and a Web crawler (Web_Crawler), as shown in Figure E-2 on page 539.

– The Security_Collection has two crawlers defined: a DB2 crawler (DB_Crawler) and a Windows file system crawler (Windows Crawler), as shown in Figure E-3 on page 539.

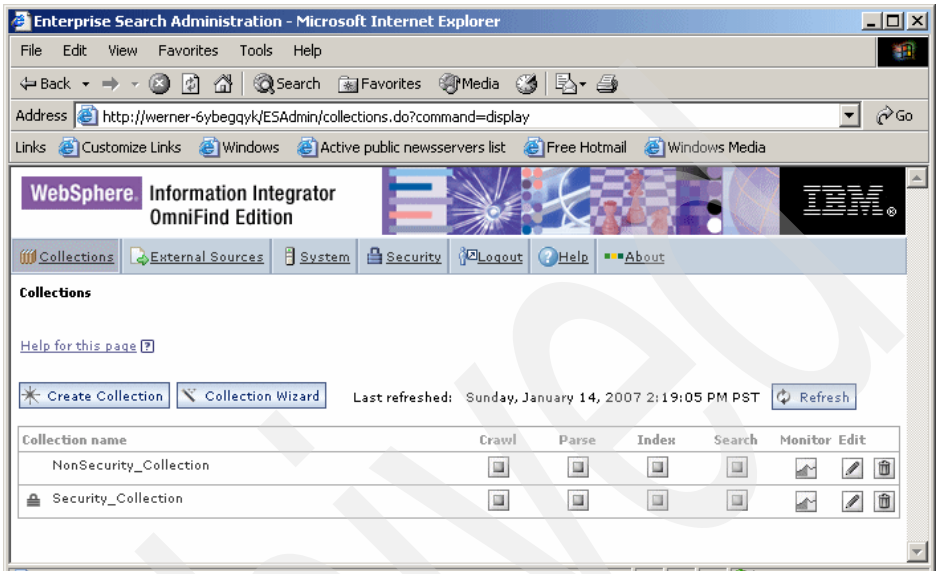

Figure E-1 Collections defined in OmniFind Edition V8.3

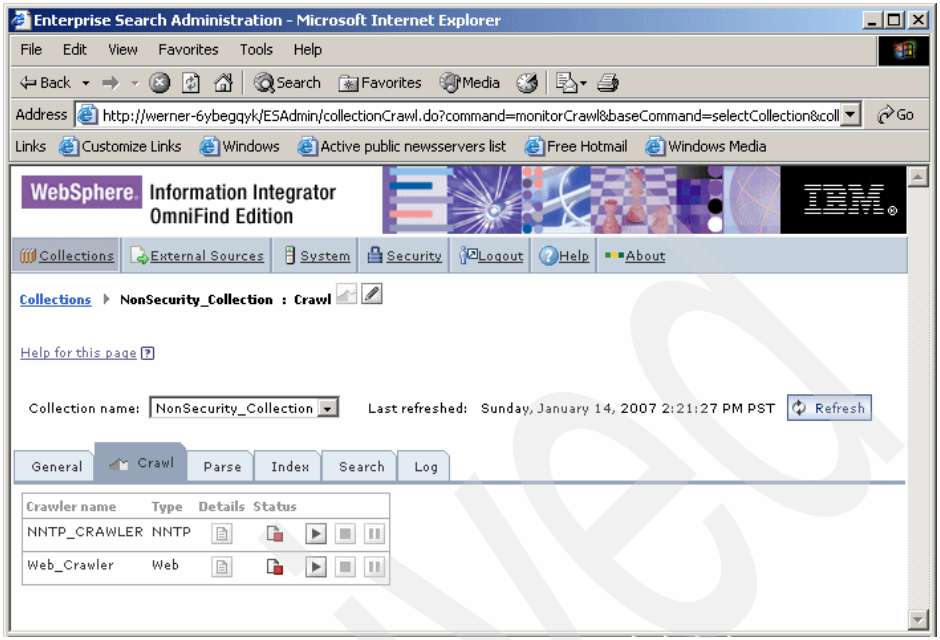

Figure E-2 NonSecurity_Collection crawlers

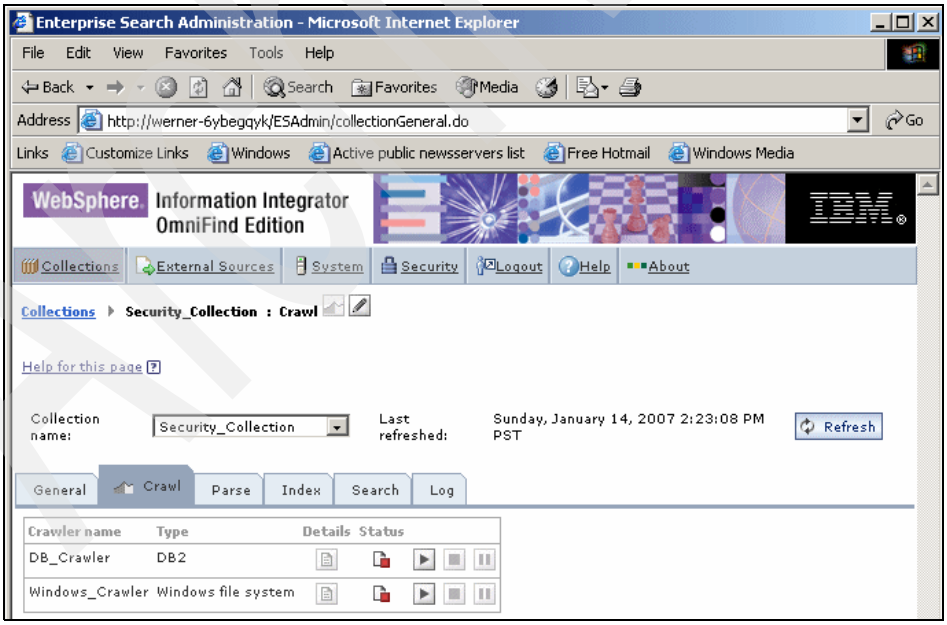

Figure E-3 Security_Collection crawlers

Migrating the single server OmniFind Edition V8.3 system

Figure E-4 shows the main steps involved in migrating an existing single server OmniFind Edition V8.3 system to OmniFind Enterprise Edition V8.4 system. These steps are described briefly in the following sections, as it applies to our Windows 2000 Advanced Server WebSphere Information Integrator OmniFind Edition V8.3 system that has two collections defined, as described in "Introduction" on page 536.

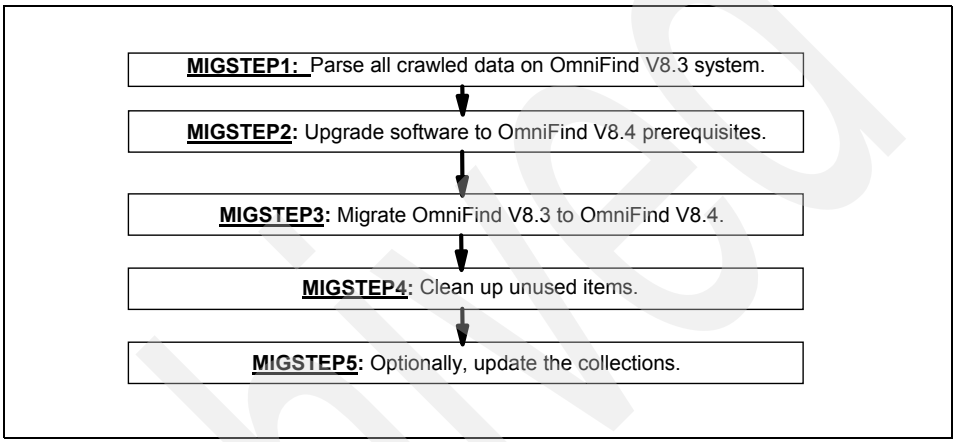

Figure E-4 Steps in migrating an OmniFind V8.3 system to OmniFind V8.4

MIGSTEP1: Parse all crawled data on the OmniFind V8.3 system

As mentioned earlier, documents that were crawled but not parsed are not migrated when OmniFind V8.4 is installed. Therefore, to ensure that all crawled documents are parsed:

- Stop all the crawlers defined in each collection; the red **Status** icon indicates the stopped status of the crawlers, as shown in Figure E-2 on page 539 and Figure E-3 on page 539.
- Start the parser the parser for each collection and ensure that al the documents crawled have been parsed (this is not shown here). We then stopped the parsers for each collection; the red **Status** icon indicates the stopped status of the parser, as shown in Figure E-5 on page 541 and Figure E-7 on page 543.

Note: While it is not required, we chose to chose to refresh the index for the NonSecurity_Collection (Figure E-6 on page 542) and reorganize the index for the Security_Collection (Figure E-8 on page 544).

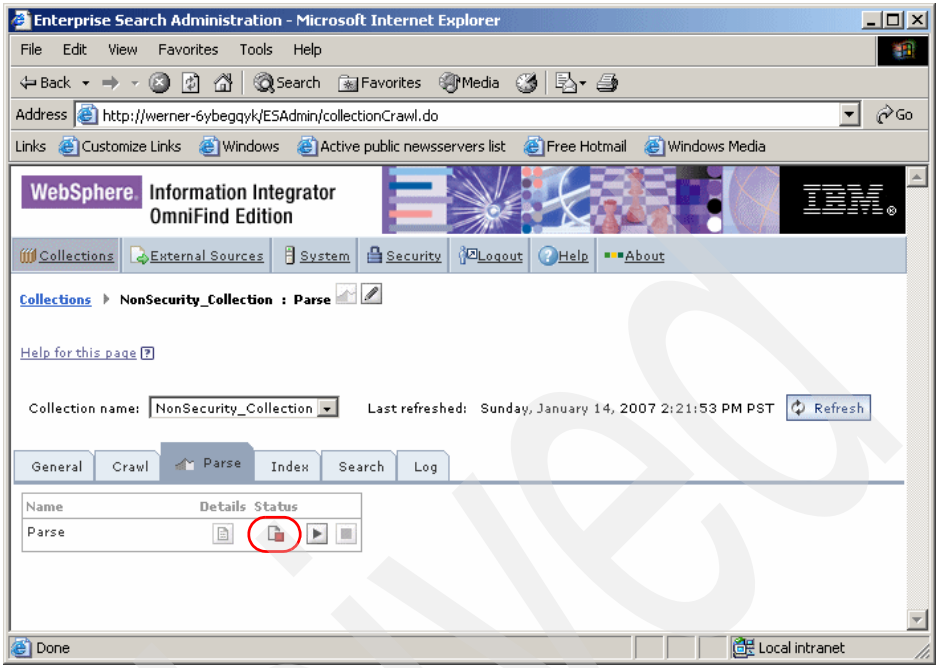

Figure E-5 Stopped status of NonSecurity_Collection parser

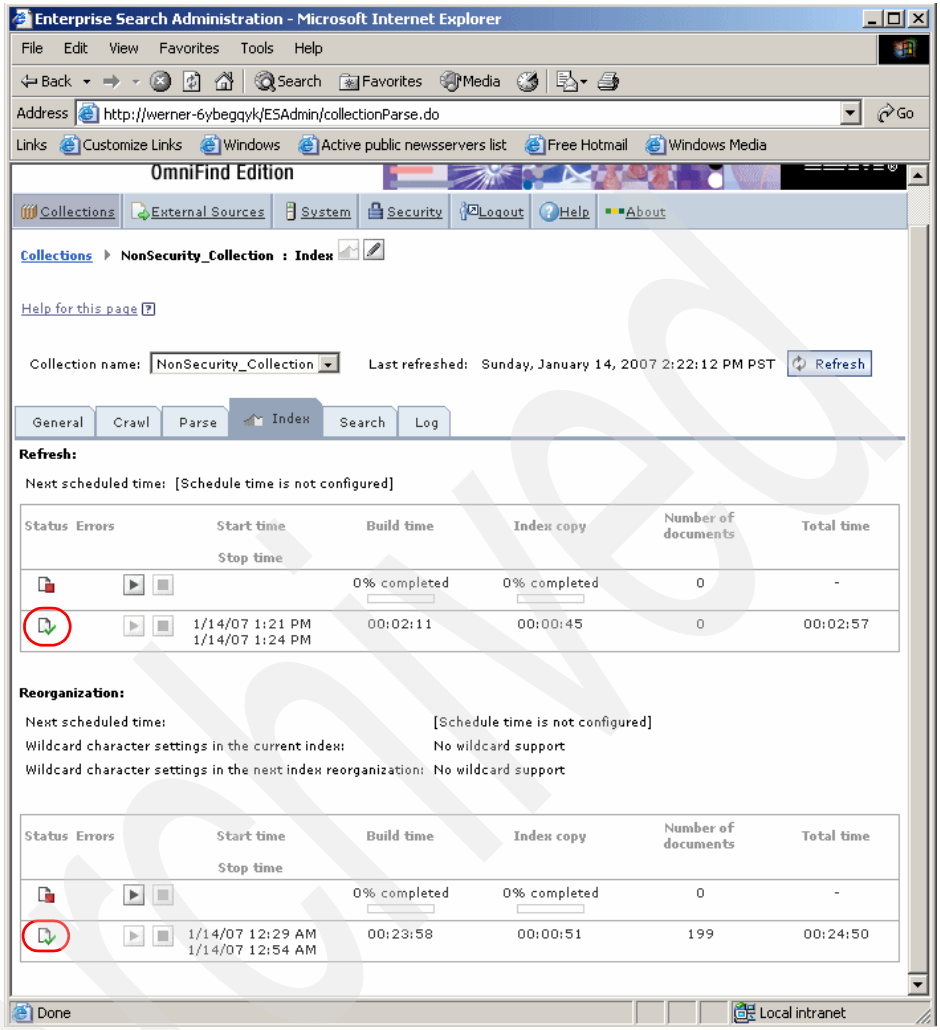

Figure E-6 Refresh Index completion status

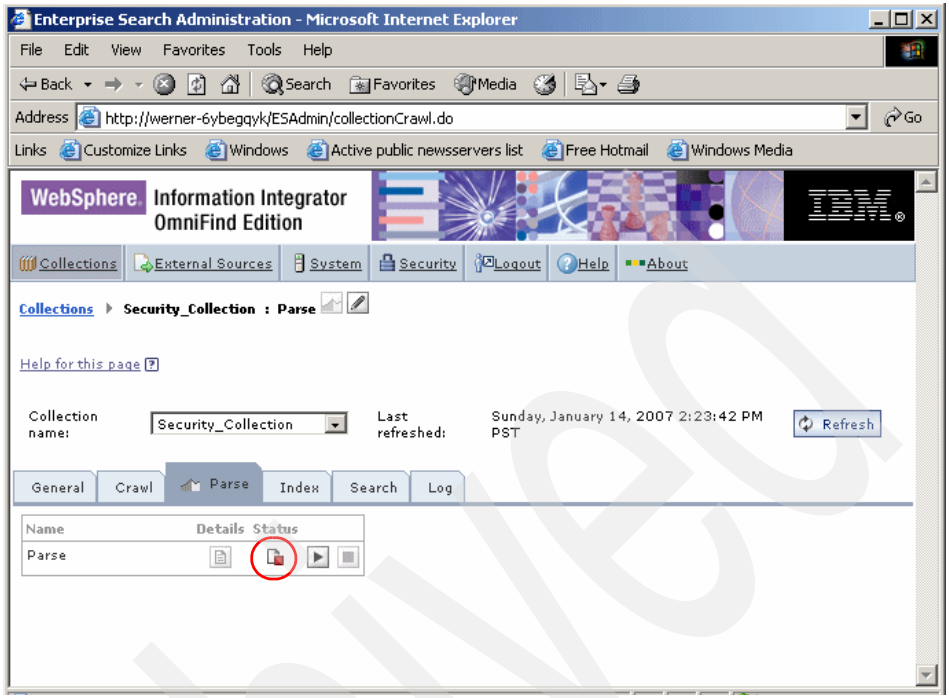

Figure E-7 Stopped status of Security_Collection parser

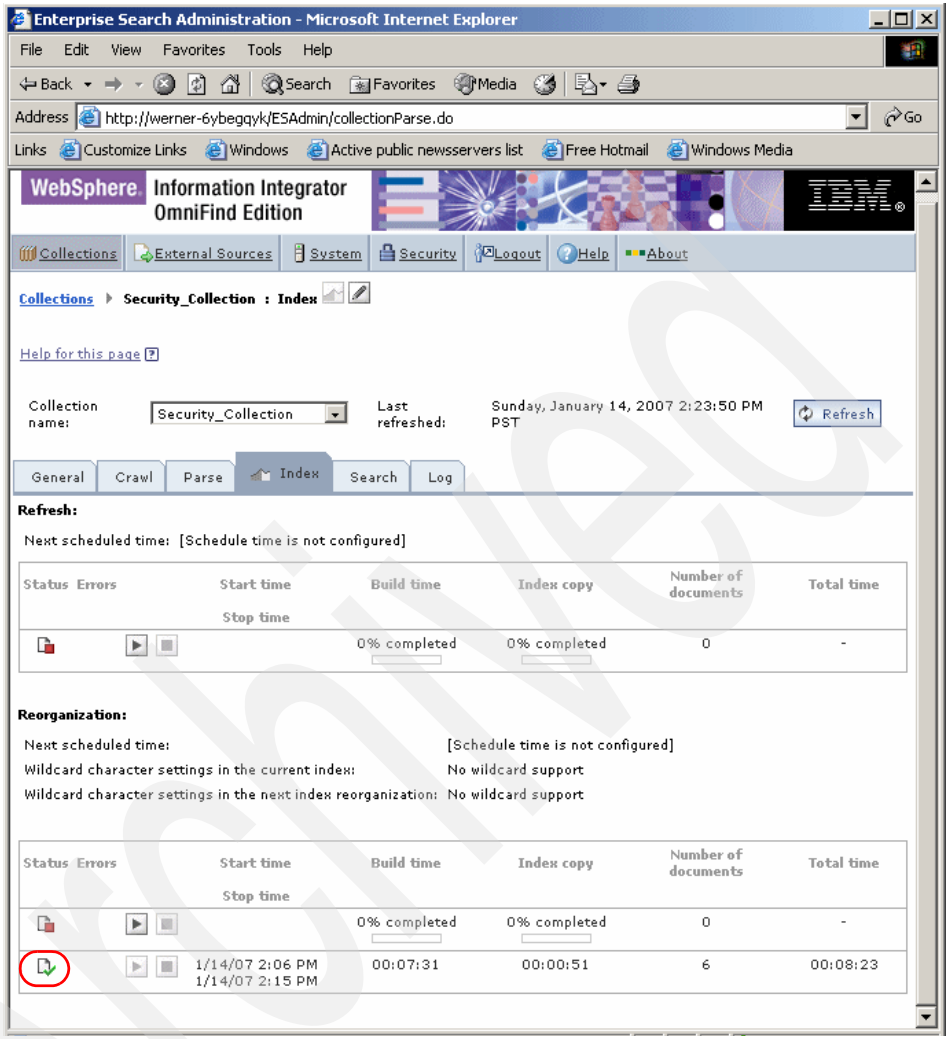

Figure E-8 Reorganize Index completion status

MIGSTEP2: Upgrade software to OmniFind V8.4 prerequisites

Our existing OmniFind V8.3 environment was installed on a single server Windows 2000 Advanced Server with Service Pack 4, with WebSphere Application Server V5.1.1.3 and Network Deployment.

Since OmniFind V8.4 has different prerequisites and no longer supported Windows 2000 Advanced Server with Service Pack 4, we had to upgrade the operating system to Windows Server 2003 Enterprise Edition with Service Pack 1.

Note: Even though OmniFind V8.4 supports WebSphere Application Server V5.1.1.3 and Network Deployment, we chose to upgrade it to WebSphere Application Server V6 Refresh Pack 2, since that is the recommended option for performance reasons. Since our WebSphere Application Server environment was only used by enterprise search applications, we could then remove the WebSphere Application Server V5.1.1.3 and Network Deployment version.

If WebSphere Application Server is used for applications other than enterprise search, you should upgrade to WebSphere Application Server V6 Refresh Pack 2 and migrate your applications to this new version.

MIGSTEP3: Migrate OmniFind V8.3 to OmniFind V8.4

We then installed OmniFind Enterprise Edition V8.4 and migrated the OmniFind V8.3 system.

Restriction: The restriction is that you must use the graphical or silent method to install the enterprise search software so that you can specify the WebSphere Application Server V6.0.2 path. You cannot use the console mode to upgrade your system.

Figure E-9 on page 547 shows the upgrade option being selected during the installation of OmniFind Enterprise Edition V8.4.

During installation, configuration errors are detected, as indicated in Figure E-10 on page 547, directing you to the MigrateConfigurationFiles.txt file for more details. This file is located in the $ES_NODE_ROOT/logs/install directory and its contents is shown in Example E-1 on page 548, which indicates a problem migrating the col_43339.NNTP_94594 crawler metadata; this error is generated because NNTP crawler metadata migration is not supported. Migration messages are written to the migration_20070115.log in the $ES_NODE_ROOT/logs/install directory, as shown in Example E-2 on page 550.

It shows the Java exceptions occurring as a consequence of not finding the NNTP crawler metadata when trying to copy it to the Cloudscape database.

Figure E-11 on page 551 and Figure E-12 on page 552 show the crawlers for the migrated collections.

Attention: Even though the NNTP crawler metadata is not migrated, the crawler still appears in the NonSecurity_Collection, as shown in Figure E-11 on page 551.

It is your responsibility to delete this NNTP crawler, and then execute the Main index process to remove all the NNTP data from the store and index. You can then recreate the NNTP crawler, crawl the NNTP data source, and then parse and execute the Delta index process to re-establish the NNTP data in the store and index.

Even though the NNTP crawler metadata is not migrated, since the Omnifind V8.3 NNTP data is not removed from the OmniFind V8.4, you can continue to access the NNTP data, as shown in Figure E-14 on page 554, unless you delete the NNTP crawler, and then execute the Main Index process.

Figure E-15 on page 555 through Figure E-17 on page 556 show the deletion of the NNTP crawler, followed by a successful execution of the Main Index process in Figure E-18 on page 557 and Figure E-19 on page 558.

Figure E-20 on page 559 shows a search query for the string “ibm”, which shows 170 documents in the results (as compared to 184 in Figure E-13 on page 553), and the NNTP data source no longer present in the index, as highlighted in the Source type filter field.

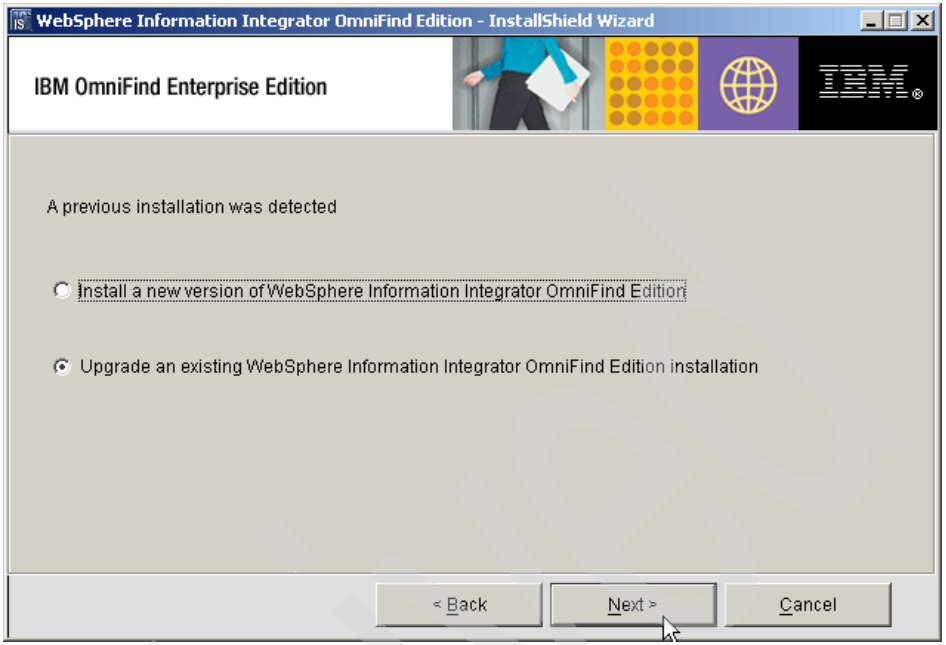

Figure E-9 Select upgrade option during OmniFInd V8.4 installation

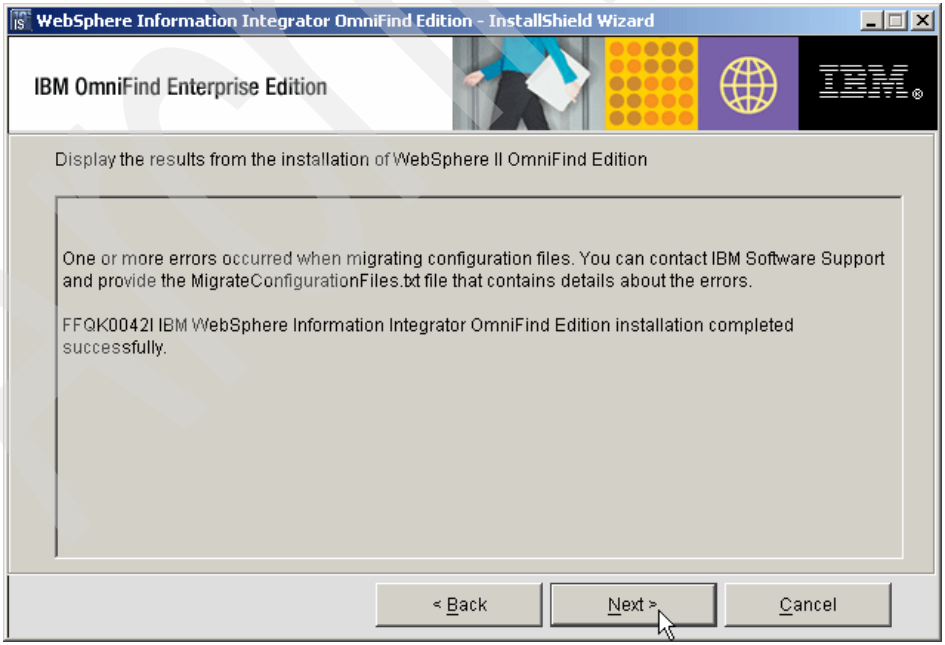

Figure E-10 Errors during migration of configuration files

Example: E-1 MigrateConfigurationFilesTo84.txt file contents

```
Starting migration for version 8.4

Updated file C:\IBM\es\esadmin\master_config\col_20994.parserdriver\specifiers\dlt_extension_typesystem.xml file for
collection col_20994
Updated file C:\IBM\es\esadmin\master_config\col_43339.parserdriver\specifiers\dlt_extension_typesystem.xml file for
collection col_43339
Status = 0
Updated file C:\IBM\es\esadmin\master_config\col_20994.parserdriver\specifiers\tt_extension_typesystem.xml file for
collection col_20994
Updated file C:\IBM\es\esadmin\master_config\col_43339.parserdriver\specifiers\tt_extension_typesystem.xml file for
collection col_43339
Status = 0
Updated file C:\IBM\es\esadmin\master_config\col_20994.parserdriver\specifiers\tt_core_typesystem.xml file for collection
col_20994
Updated file C:\IBM\es\esadmin\master_config\col_43339.parserdriver\specifiers\tt_core_typesystem.xml file for collection
col_43339
Status = 0
Updated file C:\IBM\es\esadmin\master_config\col_20994.parserdriver\specifiers\jtok.xml file for collection col_20994
Updated file C:\IBM\es\esadmin\master_config\col_43339.parserdriver\specifiers\jtok.xml file for collection col_43339
Status = 0
Updated file C:\IBM\es\esadmin\master_config\col_20994.parserdriver\specifiers\jfrost_dict_lookup.xml file for collection
col_20994
Updated file C:\IBM\es\esadmin\master_config\col_43339.parserdriver\specifiers\jfrost_dict_lookup.xml file for collection
col_43339
Status = 0
Updated file C:\IBM\es\esadmin\master_config\col_20994.parserdriver\specifiers\jfrost_ngram.xml file for collection
col_20994
Updated file C:\IBM\es\esadmin\master_config\col_43339.parserdriver\specifiers\jfrost_ngram.xml file for collection
col_43339
Status = 0
Updated file C:\IBM\es\esadmin\master_config\col_20994.parserdriver\specifiers\jfrost.xml file for collection col_20994
Updated file C:\IBM\es\esadmin\master_config\col_43339.parserdriver\specifiers\jfrost.xml file for collection col_43339
Status = 0
Updated file C:\IBM\es\esadmin\master_config\col_20994.parserdriver\specifiers\es_tok_with_stw.xml file for collection
col_20994
Updated file C:\IBM\es\esadmin\master_config\col_43339.parserdriver\specifiers\es_tok_with_stw.xml file for collection
col_43339
Status = 0
Updated file C:\IBM\es\esadmin\master_config\col_20994.parserdriver\specifiers\es_tok_no_stw.xml file for collection
col_20994
Updated file C:\IBM\es\esadmin\master_config\col_43339.parserdriver\specifiers\es_tok_no_stw.xml file for collection
col_43339
Status = 0
Updated file C:\IBM\es\esadmin\master_config\col_20994.parserdriver\specifiers\cas2jdbc.xml file for collection col_20994
Updated file C:\IBM\es\esadmin\master_config\col_43339.parserdriver\specifiers\cas2jdbc.xml file for collection col_43339
Status = 0
Updated file C:\IBM\es\esadmin\master_config\col_20994.parserdriver\specifiers\of_typesystem.xml file for collection
col_20994
Updated file C:\IBM\es\esadmin\master_config\col_43339.parserdriver\specifiers\of_typesystem.xml file for collection
col_43339
Status = 0
Status = 0
Updating C:\IBM\es\esadmin\master_config\col_20994.runtime.node1\runtime.properties
Updated C:\IBM\es\esadmin\master_config\col_20994.runtime.node1\runtime.properties
Updating C:\IBM\es\esadmin\master_config\col_43339.runtime.node1\runtime.properties
Updated C:\IBM\es\esadmin\master_config\col_43339.runtime.node1\runtime.properties
Status = 0
Updated file C:\IBM\es\esadmin\master_config\col_20994.parserdriver\casToIndex\defaultFields.prp file for collection
col_20994
Updated file C:\IBM\es\esadmin\master_config\col_43339.parserdriver\casToIndex\defaultFields.prp file for collection
col_43339
Status = OUpdated property es_special_field.default_metadata_field=16 in file
C:\IBM\es\esadmin\master_config\col_20994.parserdriver\casToIndex\fieldAttr.prp
Updated property es_special_field.default_field=7 in file
C:\IBM\es\esadmin\master_config\col_20994.parserdriver\casToIndex\fieldAttr.prp
```

```
Updated property es_special_field.default_metadata_field=16 in file
C:\IBM\es\esadmin\master_config\col_43339.parserdriver\casToIndex\fieldAttr.prp
Updated property es_special_field.default_field=7 in file
C:\IBM\es\esadmin\master_config\col_43339.parserdriver\casToIndex\fieldAttr.prp
Status = 0
Updated CPM parser configuration files for collection col_20994.
Updated CPM parser configuration files for collection col_43339.
Status = 0
Copied files from directory C:\IBM\es\esadmin\master_config\col_20994.parserdriver\specifiers to
C:\IBM\es\esadmin\master_config\col_20994.runtime.node1\specifiers
Copied files from directory C:\IBM\es\esadmin\master_config\col_43339.parserdriver\specifiers to
C:\IBM\es\esadmin\master_config\col_43339.runtime.node1\specifiers
Status = 0
Updated max_heap=1000 for session col_43339.WEB_46372.
Updated init_heap=32 for session col_43339.WEB_46372.
Status = 0
Relative config dir resourcemanager
Session(s) of type resourcemanager (resourcemanager) has been removed from file
C:\IBM\es\esadmin\master_config\services.ini
Removed directory C:\IBM\es\esadmin\master_config\resourcemanager
Removed directory C:\IBM\es\esadmin\config\master_config\resourcemanager
Status = 0
Calling com.ibm.es.crawler.Migrator84.migrate fountain AC000005  esadmin tRKn/NvOHJzbmdBMr+lPoA== col_20994.DB2_6633 false
Migrated col_20994.DB2_6633 crawler metadata from DB2 to Cloudscape
Calling com.ibm.es.crawler.Migrator84.migrate fountain AG000006  esadmin tRKn/NvOHJzbmdBMr+lPoA== col_20994.WIN_11306
false
Migrated col_20994.WIN_11306 crawler metadata from DB2 to Cloudscape
Calling com.ibm.es.crawler.Migrator84.migrate fountain AB000007  esadmin tRKn/NvOHJzbmdBMr+lPoA== col_43339.NNTP_94594
false
FFQD3403E An error occurred during migrating internal state table.
Calling com.ibm.es.wc.dbt.Migrator84.migrate fountain AA000004  esadmin tRKn/NvOHJzbmdBMr+lPoA== col_43339.WEB_46372 false
Migrated col_43339.WEB_46372 crawler metadata from DB2 to Cloudscape
Status = 1
Removed RDS ini files from directory C:\IBM\es\esadmin\master_config\col_20994.DB2_6633
Removed RDS ini files from directory C:\IBM\es\esadmin\master_config\col_20994.WIN_11306
Removed RDS ini files from directory C:\IBM\es\esadmin\master_config\col_43339.NNTP_94594
Removed RDS ini files from directory C:\IBM\es\esadmin\master_config\col_43339.WEB_46372
Removed log.ini and log.prp files from C:\IBM\es\esadmin\master_config\col_43339.WEB_46372
Status = 1
Removed RDS ini files from directory C:\IBM\es\esadmin\master_config\col_20994.parserdriver
Removed RDS ini files from directory C:\IBM\es\esadmin\master_config\col_43339.parserdriver
Status = 1
Removed rds.ini files from C:\IBM\es\default_config
Removed log.ini and log.prp files from C:\IBM\es\default_config\crawler.WEB
Removed directory C:\IBM\es\default_config\searchmanager
Status = 1
Updated flags=17 for session searchmanager.node1.
Status = 1
Updated node ID for session datalistener from node1 to node1
Status = 1
MigrateConfigurationFiles.MigrateError
```

Example: E-2 Migration_2007015.log file contents

```
<OFMsg>251658517"1"1168916047078"603987427"0"636518954" " " werner-6ybegqyk" BaseException.java"-1"3
com.ibm.db2.jcc.c.SqlException: DB2 SQL error: SQLCODE: -204, SQLSTATE: 42704, SQLERRMC: AB000007.TSERVERREC"3
com.ibm.db2.jcc.c.SqlException: DB2 SQL error: SQLCODE: -204, SQLSTATE: 42704, SQLERRMC: AB000007.TSERVERREC
    at com.ibm.db2.jcc.c.tf.e(tf.java:1680)
    at com.ibm.db2.jcc.c.tf.a(tf.java:1239)
    at com.ibm.db2.jcc.b.jb.h(jb.java:139)
    at com.ibm.db2.jcc.b.jb.a(jb.java:43)
    at com.ibm.db2.jcc.b.w.a(w.java:30)
    at com.ibm.db2.jcc.b.cc.f(cc.java:161)
    at com.ibm.db2.jcc.c.tf.n(tf.java:1219)
    at com.ibm.db2.jcc.c.uf.gb(uf.java:1816)
    at com.ibm.db2.jcc.c.uf.d(uf.java:2298)
    at com.ibm.db2.jcc.c.uf.X(uf.java:508)
    at com.ibm.db2.jcc.c.uf.executeQuery(uf.java:491)
    at com.ibm.es.crawler.migrate.Migrator84.copyServerStatus(Unknown Source)
    at com.ibm.es.crawler.migrate.Migrator84.migrate(Unknown Source)
    at com.ibm.es.crawler.migrate.Migrator84.migrate(Unknown Source)
    at com.ibm.es.migration.config.MigrateConfigFiles84.migrateCrawlerMetadata(Unknown Source)
    at com.ibm.es.migration.config.MigrateConfigFiles84.migrate(Unknown Source)
    at com.ibm.es.migration.config.MigrateConfigFiles.migrate(Unknown Source)
    at com.ibm.es.migration.config.MigrateConfigFiles.migrate(Unknown Source)
    at com.ibm.es.install.v84.MigrateConfigurationFiles.main(Unknown Source)
""</OFMsg>
<OFMsg>67112267"1"1168916047078"603987427"0"636518954" " " werner-6ybegqyk" BaseException.java"-1""</OFMsg>
<OFMsg>251658517"1"1168916047172"603987427"0"636518954" " " werner-6ybegqyk" BaseException.java"-1"3
com.ibm.db2.jcc.c.SqlException: DB2 SQL error: SQLCODE: -204, SQLSTATE: 42704, SQLERRMC: AB000007.TSERVERREC"3
com.ibm.db2.jcc.c.SqlException: DB2 SQL error: SQLCODE: -204, SQLSTATE: 42704, SQLERRMC: AB000007.TSERVERREC
    at com.ibm.db2.jcc.c.tf.e(tf.java:1680)
    at com.ibm.db2.jcc.c.tf.a(tf.java:1239)
    at com.ibm.db2.jcc.b.jb.h(jb.java:139)
    at com.ibm.db2.jcc.b.jb.a(jb.java:43)
    at com.ibm.db2.jcc.b.w.a(w.java:30)
    at com.ibm.db2.jcc.b.cc.f(cc.java:161)
    at com.ibm.db2.jcc.c.tf.n(tf.java:1219)
    at com.ibm.db2.jcc.c.uf.gb(uf.java:1816)
    at com.ibm.db2.jcc.c.uf.d(uf.java:2298)
    at com.ibm.db2.jcc.c.uf.X(uf.java:508)
    at com.ibm.db2.jcc.c.uf.executeQuery(uf.java:491)
    at com.ibm.es.crawler.migrate.Migrator84.copyServerStatus(Unknown Source)
    at com.ibm.es.crawler.migrate.Migrator84.migrate(Unknown Source)
    at com.ibm.es.crawler.migrate.Migrator84.migrate(Unknown Source)
    at com.ibm.es.migration.config.MigrateConfigFiles84.migrateCrawlerMetadata(Unknown Source)
    at com.ibm.es.migration.config.MigrateConfigFiles84.migrate(Unknown Source)
    at com.ibm.es.migration.config.MigrateConfigFiles.migrate(Unknown Source)
    at com.ibm.es.migration.config.MigrateConfigFiles.migrate(Unknown Source)
    at com.ibm.es.install.v84.MigrateConfigurationFiles.main(Unknown Source)
""</OFMsg>
<OFMsg>251658517"1"1168916047188"603987427"0"636518954" " " werner-6ybegqyk" BaseException.java"-1"3
com.ibm.db2.jcc.c.SqlException: DB2 SQL error: SQLCODE: -204, SQLSTATE: 42704, SQLERRMC: AB000007.TSERVERREC"3
com.ibm.db2.jcc.c.SqlException: DB2 SQL error: SQLCODE: -204, SQLSTATE: 42704, SQLERRMC: AB000007.TSERVERREC
    at com.ibm.db2.jcc.c.tf.e(tf.java:1680)
    at com.ibm.db2.jcc.c.tf.a(tf.java:1239)
    at com.ibm.db2.jcc.b.jb.h(jb.java:139)
    at com.ibm.db2.jcc.b.jb.a(jb.java:43)
    at com.ibm.db2.jcc.b.w.a(w.java:30)
    at com.ibm.db2.jcc.b.cc.f(cc.java:161)
    at com.ibm.db2.jcc.c.tf.n(tf.java:1219)
    at com.ibm.db2.jcc.c.uf.gb(uf.java:1816)
    at com.ibm.db2.jcc.c.uf.d(uf.java:2298)
    at com.ibm.db2.jcc.c.uf.X(uf.java:508)
    at com.ibm.db2.jcc.c.uf.executeQuery(uf.java:491)
    at com.ibm.es.crawler.migrate.Migrator84.copyServerStatus(Unknown Source)
    at com.ibm.es.crawler.migrate.Migrator84.migrate(Unknown Source)
    at com.ibm.es.crawler.migrate.Migrator84.migrate(Unknown Source)
    at com.ibm.es.migration.config.MigrateConfigFiles84.migrateCrawlerMetadata(Unknown Source)
```

```
    at com.ibm.es.migration.config.MigrateConfigFiles84.migrate(Unknown Source)
    at com.ibm.es.migration.config.MigrateConfigFiles.migrate(Unknown Source)
    at com.ibm.es.migration.config.MigrateConfigFiles.migrate(Unknown Source)
    at com.ibm.es.install.v84.MigrateConfigurationFiles.main(Unknown Source)
""</OFMsg>
<OFMsg>67112267"1"1168916047188"603987427"0"636518954" " " werner-6ybegqyk" BaseException.java"-1""</OFMsg>
<OFMsg>67112267"1"1168916047172"603987427"0"636518954" " " werner-6ybegqyk" BaseException.java"-1""</OFMsg>
<OFMsg>67113758"4"1168916048156"1090521529"0"636518954" " " werner-6ybegqyk" WCException.java"127"3 Migrator84 connect to
DB2 database jdbc:db2:fountain""</OFMsg>
<OFMsg>67113758"4"1168916052234"1090521529"0"636518954" " " werner-6ybegqyk" WCException.java"127"3 Migrator84 create
database jdbc:derby:C:\IBM\es\esadmin\data\cloudscape\OmniFind_crawlers\col_43339.WEB_46372""</OFMsg>
<OFMsg>67113758"4"1168916053922"1090521529"0"636518954" " " werner-6ybegqyk" WCException.java"127"3 Migrator84 create
tables in database jdbc:derby:C:\IBM\es\esadmin\data\cloudscape\OmniFind_crawlers\col_43339.WEB_46372""</OFMsg>
<OFMsg>67113758"4"1168916054000"1090521529"0"636518954" " " werner-6ybegqyk" WCException.java"127"3 Migrator84 copy BUCKET
data to database jdbc:derby:C:\IBM\es\esadmin\data\cloudscape\OmniFind_crawlers\col_43339.WEB_46372""</OFMsg>
<OFMsg>67113758"4"1168916054016"1090521529"0"636518954" " " werner-6ybegqyk" WCException.java"127"3 Migrator84 copy IPV4
data to database jdbc:derby:C:\IBM\es\esadmin\data\cloudscape\OmniFind_crawlers\col_43339.WEB_46372""</OFMsg>
<OFMsg>67113758"4"1168916054047"1090521529"0"636518954" " " werner-6ybegqyk" WCException.java"127"3 Migrator84 copy IPV6
data to database jdbc:derby:C:\IBM\es\esadmin\data\cloudscape\OmniFind_crawlers\col_43339.WEB_46372""</OFMsg>
<OFMsg>67113758"4"1168916054094"1090521529"0"636518954" " " werner-6ybegqyk" WCException.java"127"3 Migrator84 copy ROBOTS
data to database jdbc:derby:C:\IBM\es\esadmin\data\cloudscape\OmniFind_crawlers\col_43339.WEB_46372""</OFMsg>
<OFMsg>67113758"4"1168916057047"1090521529"0"636518954" " " werner-6ybegqyk" WCException.java"127"3 Migrator84 copy URL
data to database jdbc:derby:C:\IBM\es\esadmin\data\cloudscape\OmniFind_crawlers\col_43339.WEB_46372""</OFMsg>
<OFMsg>67113758"4"1168916057625"1090521529"0"636518954" " " werner-6ybegqyk" WCException.java"127"3 Migrator84 create
tables in database jdbc:derby:C:\IBM\es\esadmin\data\cloudscape\OmniFind_crawlers\col_43339.WEB_46372""</OFMsg>
<OFMsg>67113758"4"1168916057625"1090521529"0"636518954" " " werner-6ybegqyk" WCException.java"127"3 Migrator84 all
operations in database jdbc:derby:C:\IBM\es\esadmin\data\cloudscape\OmniFind_crawlers\col_43339.WEB_46372""</OFMsg>
```

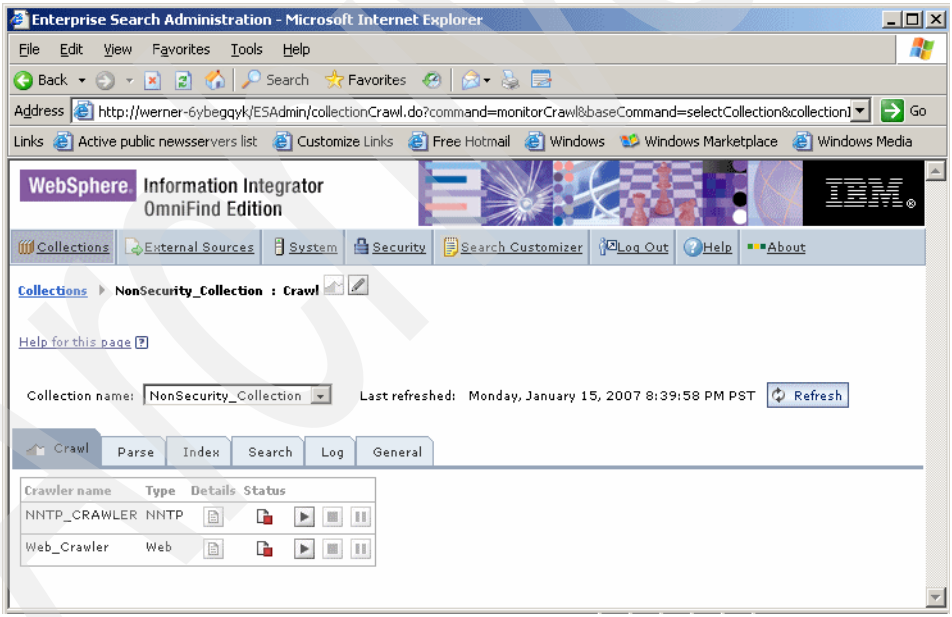

Figure E-11 NonSecurity_Collection crawlers NNTP_CRAWLER and Web_Crawler

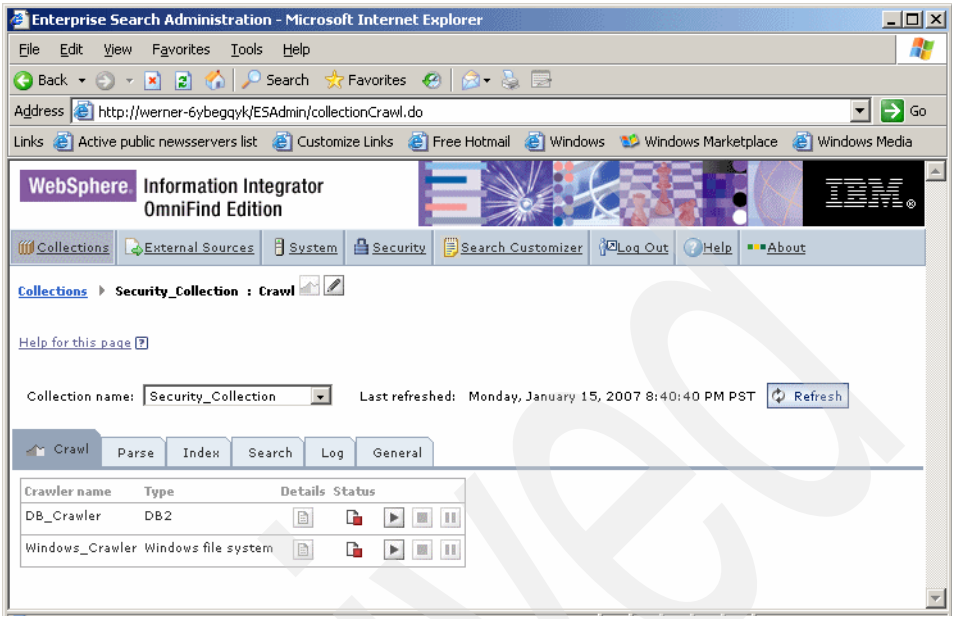

Figure E-12 Security_Collection crawlers DB_CRAWLER and Windows_Crawler

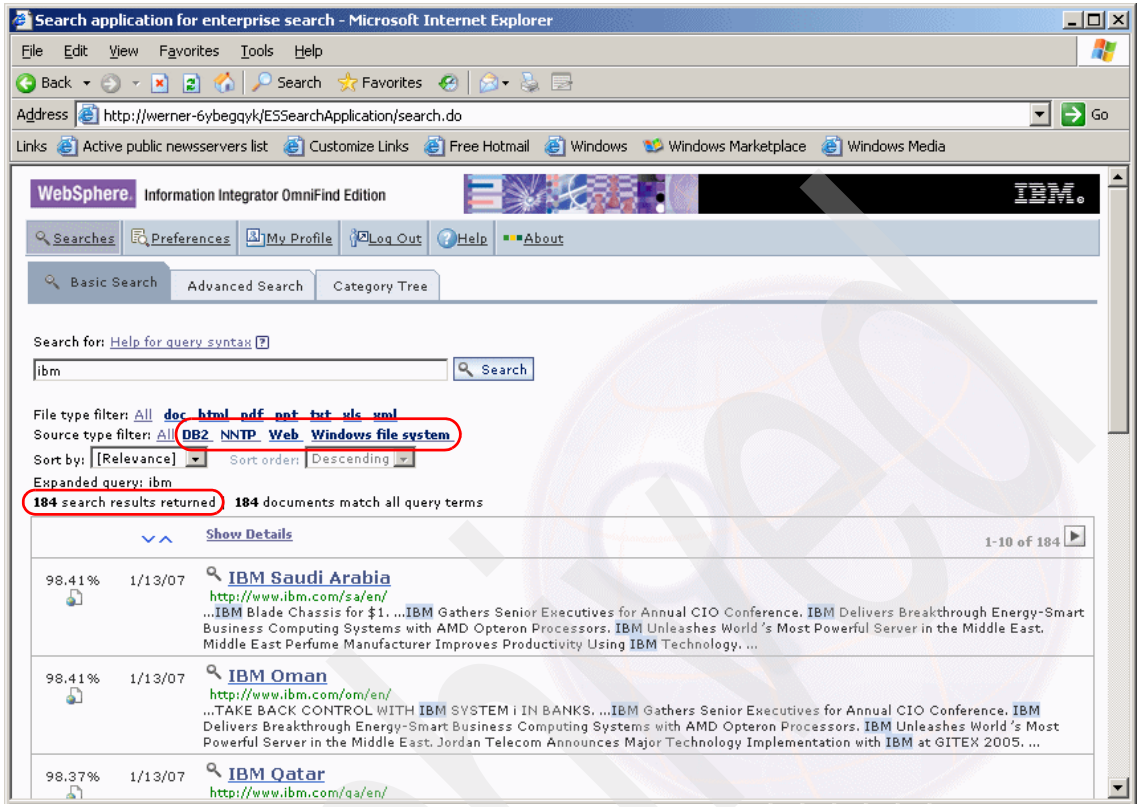

Figure E-13 Search results for "ibm" accessing all sources

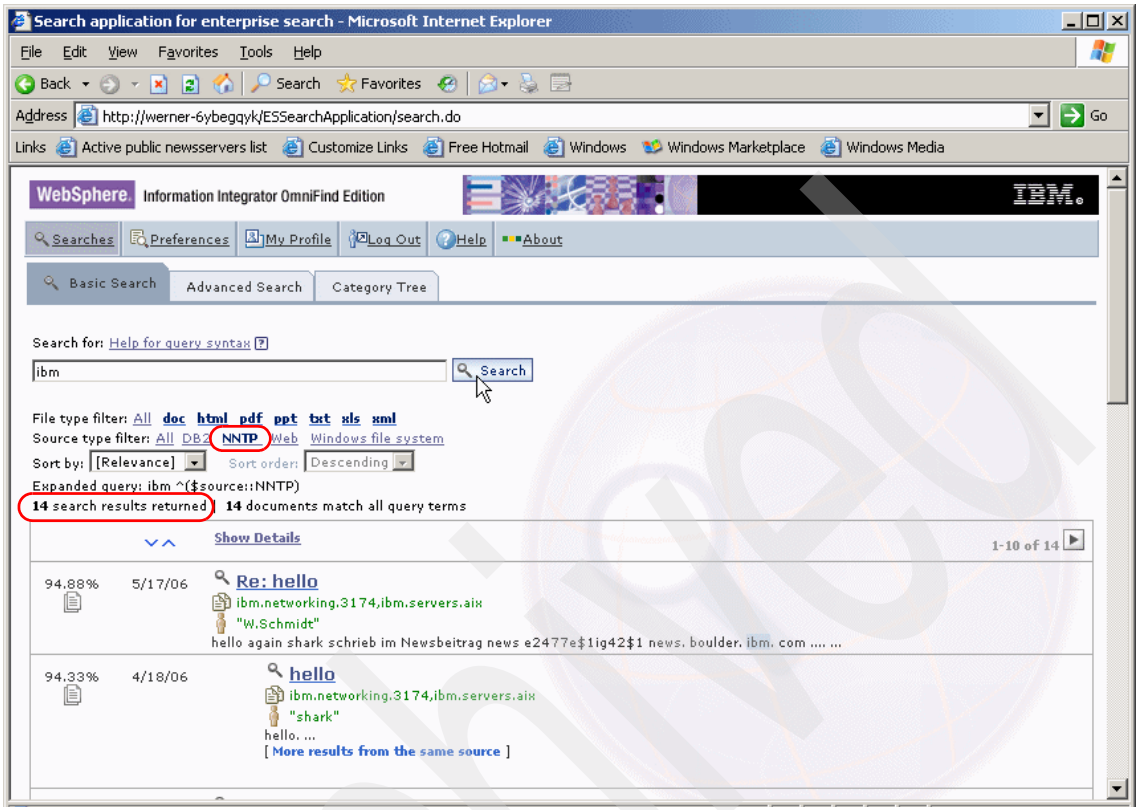

Figure E-14 Search results for “ibm” accessing all NNTP source only

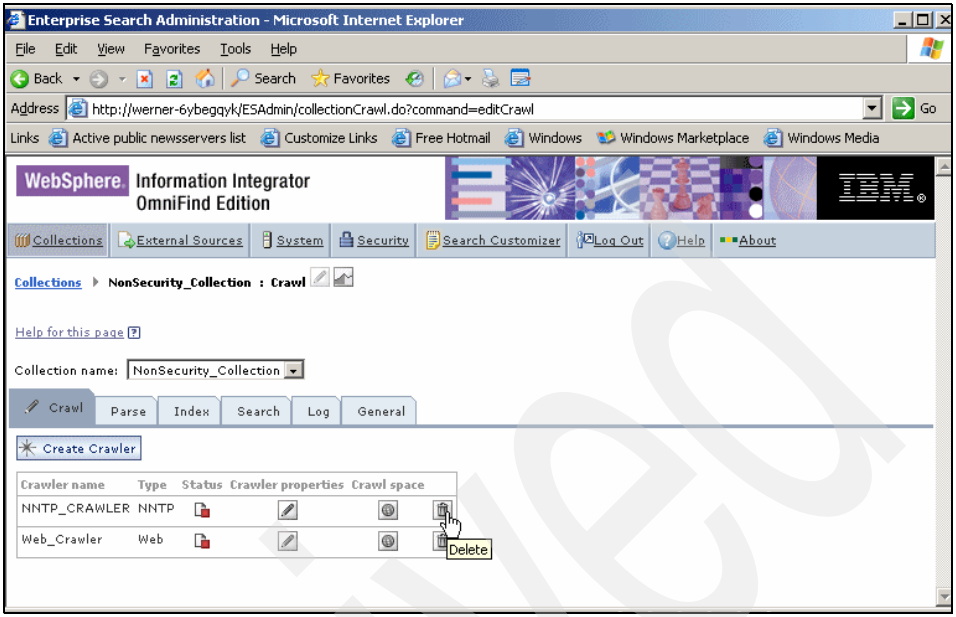

Figure E-15 Delete NNTP crawler 1/3

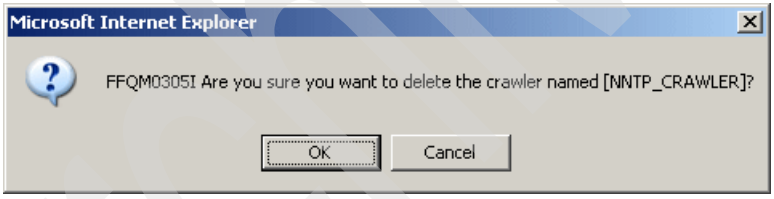

Figure E-16 Delete NNTP crawler 2/3

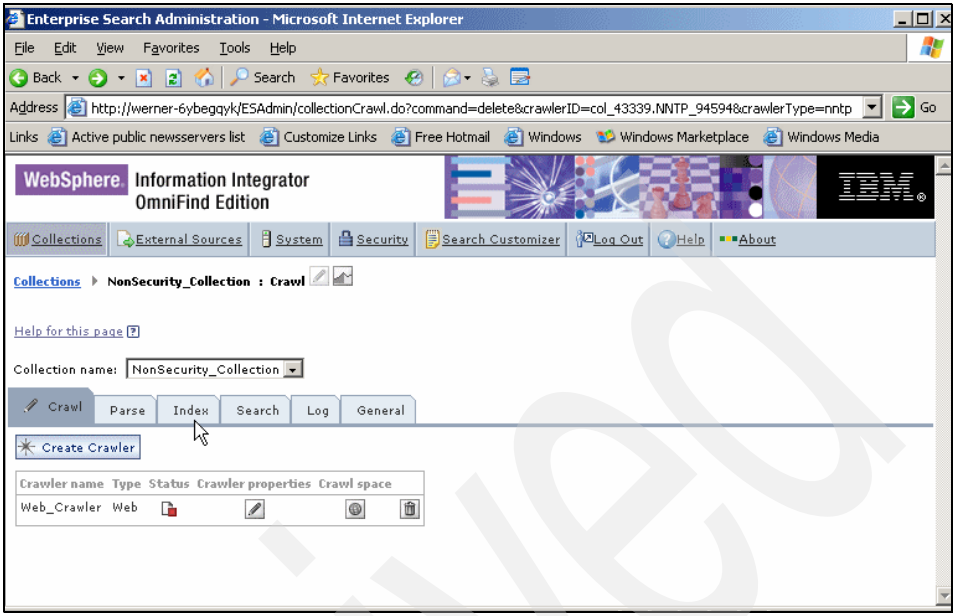

Figure E-17 Delete NNTP crawler 3/3

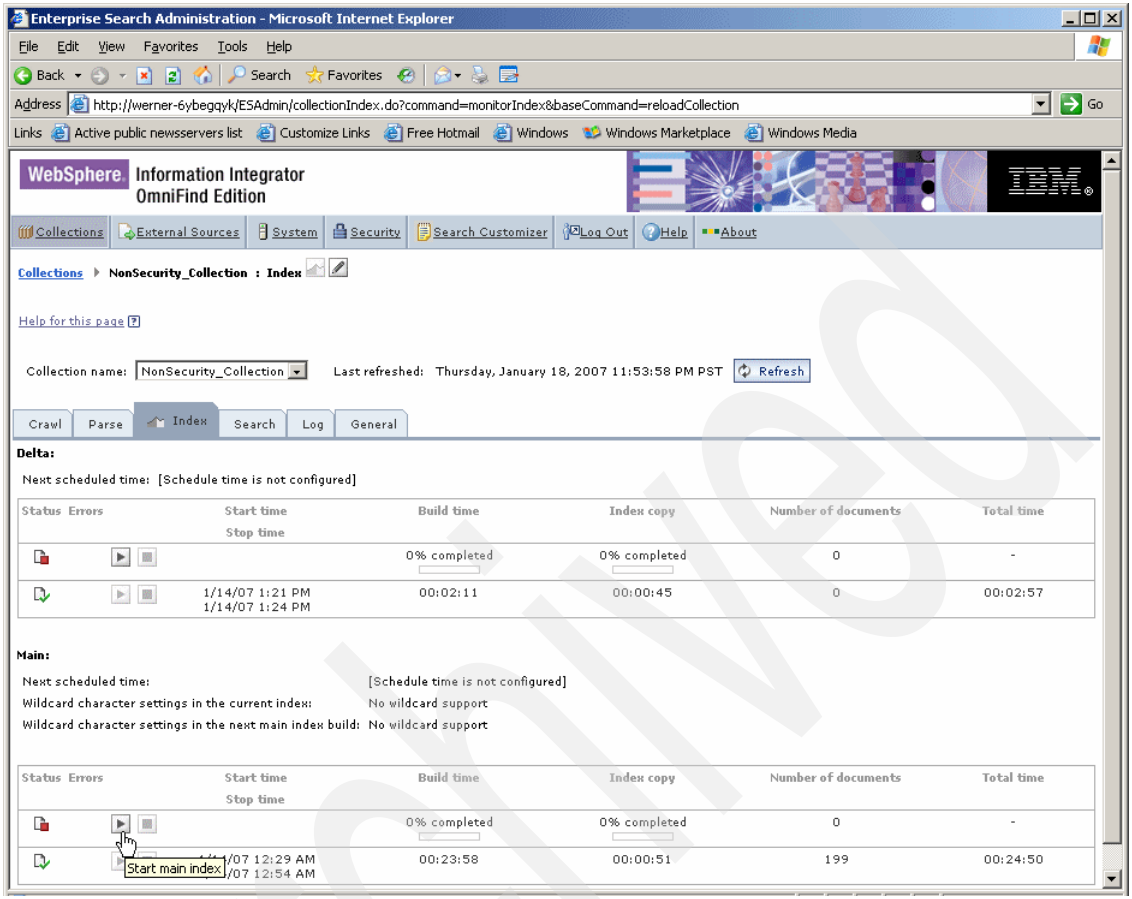

Figure E-18 Start main index 1/2

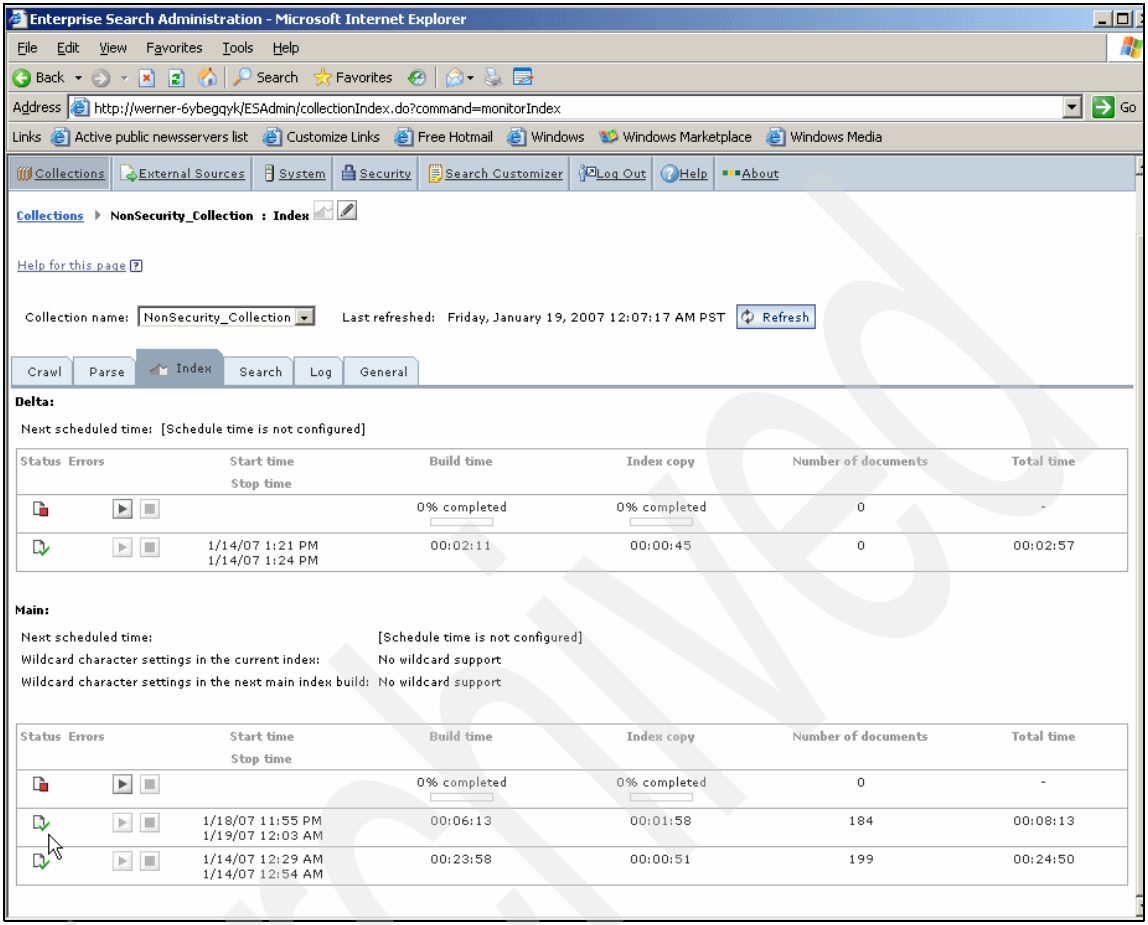

Figure E-19 Start main index 2/2

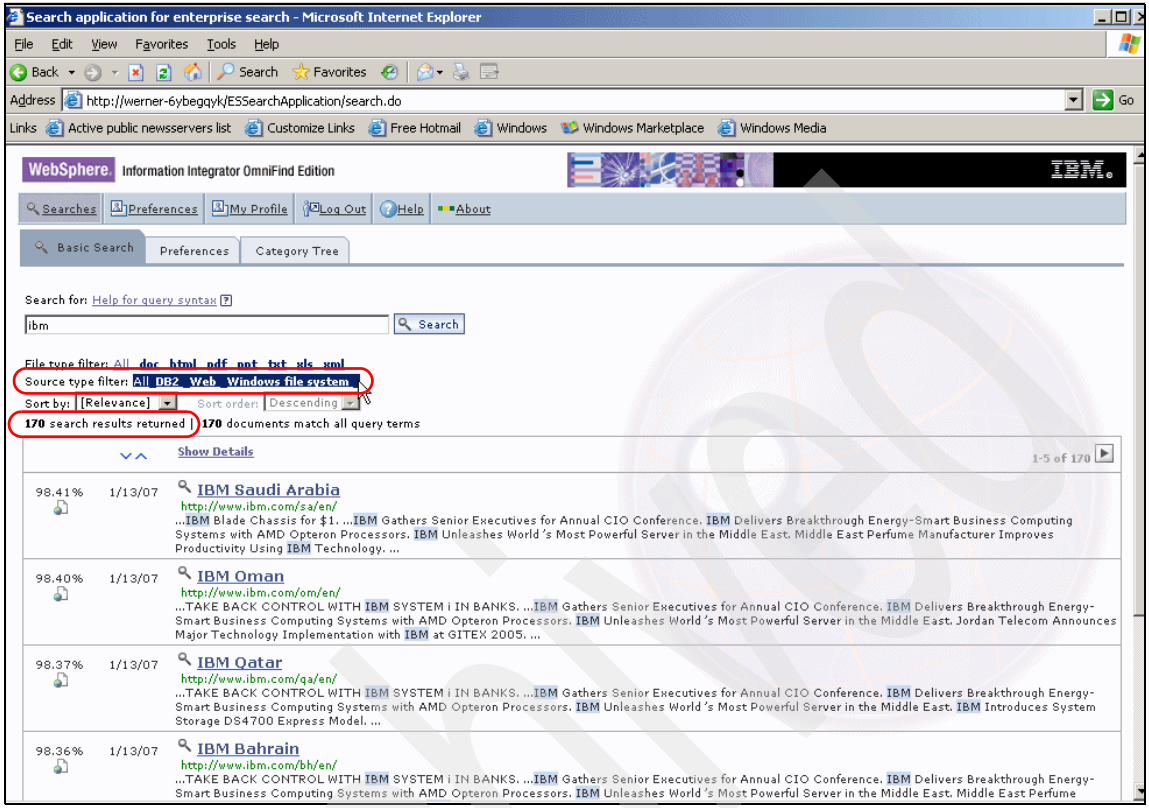

Figure E-20 Search results for "ibm" with no NNTP data source

MIGSTEP4: Cleanup unused items

In this step, we could only remove the WebSphere Application Server V5.1.1.3 and Network Deployment software, since it was not used anywhere.

Even though DB2 is no longer used for storing metadata and raw data in OmniFind V8.4, we could not remove DB2, since it is a data source for the DB_CRAWLER DB2 crawler in the Security_Collection.

MIGSTEP5: Optional: update the collections

When documents processed with an earlier version of OmniFind Enterprise Edition are indexed and searched in OmniFind V8.4:

- The search results might be adversely affected. This is especially important for collections that use the n-gram tokenization mechanism, but also true for collections that use standard tokenization.

The main areas of improvement include the treatment of hyphenated words, e-mail addresses, unified resource locators (URLs), and some numerical values and special characters.

- Some features introduced with OmniFind Enterprise Edition V8.4 will not function on documents that are not reprocessed. These features include the settings that can be applied in the crawler configuration on how to handle fields, such as complete match queries and sorting search results by field name.
  - For the complete match feature, mixed collections can be searched with the limitation that a complete match query returns only documents that were processed with the new settings.
  - The ability to sort search results by field name is not useful in mixed collections because only documents that are processed with the new settings have fields flagged as sortable fields. Older documents that do not have a sortable field will be represented incorrectly in the search results and be placed at the end of the list of sorted results.
  - Automatically assigning a content boost value to fields if the field is flagged as a document content field and a free text searchable field works as it does in OmniFind V8.3. For new documents and documents that are reprocessed, fields that are flagged as document content and free text searchable can also be displayed in the document summary area of the search results.

Note: However, a collection that includes a mixture of documents processed with the new settings and indexed documents that are not yet reprocessed will behave correctly for the functionality that was available with the previous version of OmniFind Enterprise Edition.

Therefore, to fully leverage improvements in character normalization and text tokenization, you should re-crawl, re-parse, and re-index all documents in your collections after you upgrade to OmniFind Enterprise Edition V8.4.

Note: For our test environment, we chose *not* to re-crawl, re-parse, and re-index the collections, since our emphasis was on the migration messages generated.

F

Troubleshooting aids

In this appendix, we describes the main troubleshooting tools available with IBM OmniFind Enterprise Edition and provides usage examples of them where appropriate. It does not describe a troubleshooting methodology or a procedure to address commonly encountered problems.

Introduction

IBM OmniFind Enterprise Edition provides a number of facilities to troubleshoot problems occurring in the OmniFind environment. It does not describe a troubleshooting methodology or a procedure to address commonly encountered problems. The troubleshooting facilities may be broadly categorized as follows and are described briefly in the following sections:

- Log files
- Configuration files
- Commands
- Miscellaneous

Important: A lot of the information gathered from these facilities is aimed at the sophisticated enterprise search administrator with a deep understanding of the IBM OmniFind Enterprise Edition product, and for IBM development to debug problem situations. You are strongly encouraged to use the facilities described here with great care, and preferably by consulting with IBM support.

Log files

OmniFind writes a number of log files that record errors, warnings, and informational messages, as well as audit information about process flow among the various software components. These log files contain information at the system level as well as a collection level, and reside in one or more servers of a multi-server OmniFind configuration.

The two broad types of logs are error logs and audit logs. Most of these logs are described here.

These logs may be formatted and viewed using the esviewlog.bat or esviewlog.sh scripts.

Important: The OmniFind administration console GUI simplifies the analysis of information generated in the various logs by displaying consolidated related information from multiple logs.

Error logs

Table F-1 on page 563 lists the names of the various error logs, their scope (whether at the system level or collection specific), and a brief description of their

contents, while Table F-2 on page 563 lists the names of the dropped documents logs.

Note: The error logs are always created in the directory $ES_NODE_ROOT[a]/logs, while the dropped documents logs are created in the directory $ES_NODE_ROOT/data/<collection id>. Both these logs are created on the Indexer server. If a non-indexer node is unable to send error messages to the centralized logger on the indexer node, it will log locally.

a. Enterprise search data directory

Table F-1 . Error logs

Name	Scope	Description
system_yyyymmdd.log	System	Contains errors and warnings from system sessions[a], such as controller, searchmanager, and configmanager
<collectionid>_yyyymmdd.log	Collection	Contains errors and warnings from collection specific sessions, such as runtime and parser driver
WC_yyyymmdd.log	Collection	Contains errors and warnings from the Web crawler
CCLServer_yyyymmdd.log	System	Contains errors and warnings from the CCL server
ccl<#>.log	System	Contains messages from the CCL, which is much more detailed than that recorded in the CCLServer log
cmdline_yyyymmdd.log	System	Contains errors and warnings from all the commands issued from the command line
ESSearchServer.<#>.log	System	Contains errors and warnings from the WebApp modules
backup.log	System	Contains messages about backup/restore operations
migration_yyyymmdd.log	System	Contains migration errors
derby.log	System	Contains Cloudscape errors
ESSearchApplication.0.log	System	Contains error messages from ESSearchApplication
CCLSearchServer.<#>.log	System	Contains errors and warnings from ESSearchServer

a. Another term for component

Table F-2 Dropped documents logs

Name	Scope	Description
dropped_doc_pd_yyyymmdd.log	Collection	Contains documents dropped by parser
dropped_doc_in_yyyymmdd.log	Collection	Contains documents dropped by indexer

Audit logs

Table F-3 lists the names of the various system level audit logs, while Table F-4 on page 565 lists the names of the various collection specific audit logs recorded by OmniFind. The audit logs contain audit information about the sessions from which they are logging. The information therein typically reflects the state of the session, as well as the events that occur through the lifetime of the session.

Attention: Audit log messages are written at the discretion of the developer; therefore, there are no strict guidelines about the information that is logged. *These logs are most useful for OmniFind developers in IBM to reconstruct the state of the session in case of a problem.*

Note: The audit logs are always created in the directory $ES_NODE_ROOT/logs/audit on the local server (crawler, indexer, and search) where the session is running

Table F-3 System level audit logs

Name
parserservice_audit_yyyymmdd.log
resource.<node id>_audit_yyyymmdd.log
scheduler_audit_yyyymmdd.log
searchmanager.<node id>_audit_yyyymmdd.log
TraceDaemon_audit_yyyymmdd.log
utilities.<node id>_audit_yyyymmdd.log
monitor_audit_yyyymmdd.log
discovery_audit_yyyymmdd.log
datalistener_audit_yyyymmdd.log
WC_yyyymmdd.log
customcommunication_audit_yyyymmdd.log
configmanager_audit_yyyymmdd.log
controller_audit_yyyymmdd.log
cmdline_audit_yyyymmdd.log

Name
CCLServer_audit_yyyymmdd.log
ESSearchServer_audit_yyyymmdd.log

Table F-4 Collection specific audit logs

Name
<collection id>.indexcopy.<node id>_audit_yyyymmdd.log
<collection id>.indexer_audit_yyyymmdd.log
<collection id>.indexer.delta_audit_yyyymmdd.log
<collection id>_OmniFindQueryLog_yyyymmdd.log
<collection id>.parserdriver_audit_yyyymmdd.log
<collection id>.runtime.<node id>_audit_yyyymmdd.log
<collection id>.stellent_audit_yyyymmdd.log
<collection id>.<crawler id>_audit_yyyymmdd.log

Configuration files

OmniFind has a large number of configuration files (system level and collection specific) that drive the operation of the various OmniFind components. These files reside on the appropriate server where the components are. Each of these configuration files has a number of parameters, not all of which are documented.

- The system level configuration files are set during installation.
- The collection level configuration files are only created when a collection is created, and can be set during collection creation. Some properties can be modified later from the administration console GUI.

These configuration files may also be edited directly using a text editor, such as Notepad.

Note: The es.cfg configuration file is one of special significance that is located in the $ES_NODE_ROOT/nodeinfo directory. It contains critical configuration information for accessing the OmniFInd system, such as the install enterprise search administrator user ID and encrypted password with no global security enabled, and the authentication information when WebSphere Application Server global security is enabled.

Attention: You are strongly advised to consult with IBM support personnel before making changes to undocumented parameters in these configuration files.

Table F-5 lists some of the key system level configuration files, with a brief description of their function, while Table F-6 on page 567 lists some of the key collection specific configuration files.

Note: All configuration files are located in the Indexer server under the $ES_NODE_ROOT/master_config directory and its sub-directories.

- System level configuration files are located directly under the $ES_NODE_ROOT/master_config directory.
- Collection specific configuration files are located in the $ES_NODE_ROOT/master_config/<collection id>.<session_type>.[<node id>][a]/ directory.

a. The square brackets indicate that the node ID is optional

Table F-5 Key system level configuration files

Name	Description
nodes.ini	Information about all the nodes in the system
Services.ini	System session configuration information
collections.ini	Collections information
<collection id>_config.ini	Session configuration information for a collection
uima.xml	Uploaded UIMA annotators and collection associations
SynonymConfiguration.xml	Uploaded synonym dictionaries and collection associations
StopWordDictionaryConfiguration.xml	Uploaded stop word dictionaries and collection associations
BoostingWordDictionary Configuration.xml	Uploaded boost word dictionaries and collection associations

Table F-6 Key collection specific configuration files

Name	Description
<collection id>.<crawler_id>	Crawler configuration
<collection id>.parserdriver	Parser configuration
<collection id>.runtime.nodeN	Runtime/search configuration
<collection id>.indexer	Main index build configuration
<collection id>.indexer.delta	Delta index build configuration
<collection id>.stellent	Stellent file parser configuration
<collection id>.casprocessor	Optional custom analysis text annotator configuration
ccl.properties	CCL settings
collection.properties	Collection settings
runtime_generic.properties	Search runtime settings
sso.properties	Single sign-on settings
appids.properties	Search applications defined
parserTypes.cfg	Defines rules for mapping file extensions or MIME types to parser types.

Commands

Table F-7 lists some of the key commands available with a brief description of their function and examples where appropriate. As mentioned earlier, the cmdline_yyyymmdd.log files in the $ES_NODE_ROOT/logs directory has information about the commands executed in the OmniFind environment.

Table F-7 Troubleshooting commands

Name	Description
esadmin check	Lists all the active processes, as shown in Figure F-1 on page 569.
esadmin rds	Reads RDS information. Use **esadmin rds help** for details on this command, as shown in Figure F-2 on page 569

Name	Description
esadmin report	There are many options on this command, including sessions, nodes, collections, and interfaces. Use **esadmin report help** for details on this command, as shown in Figure F-3 on page 570.
esadmin session discovery discover -api <apiname> <options>	There are many options to this command, as shown in Example F-1 on page 570.
dumpstore	Prints the contents of the RDS and Trevistore. Figure F-4 on page 573 shows the output of **dumpstore help**. It does not print the contents of the index. Example F-2 on page 573 shows the output of RDS where the crawled data includes Notes Domino (native ACLs), DB2 Content Manager (native ACLs), and DB2 (no security). The key elements are SecurityACLs and NativeACLs. You will note that SecurityACLs is PUBLIC="NO" and NativeACLs has the appropriate access control list information for Notes Domino and DB2 Content Manager. For DB2 however, SecurityACLs is PUBLIC="YES" and there is no NativeACLs information because security is not enabled for DB2. Example F-3 on page 576 shows the output of trevistore for the corresponding RDS in Example F-2 on page 573. It shows the Security information for Notes Domino and DB2 Content Manager, but is blank for DB2. Example F-4 on page 577 shows the output of RDS where the crawled data using the JDBC crawler is DB2 with security tokens. You will note that SecurityACLs is PUBLIC="NO"with the token PERSONAL. Example F-5 on page 577 shows the output of trevistore for the corresponding RDS in Example F-4 on page 577. It shows the security token (PERSONAL) in the field esfield.security_acl.

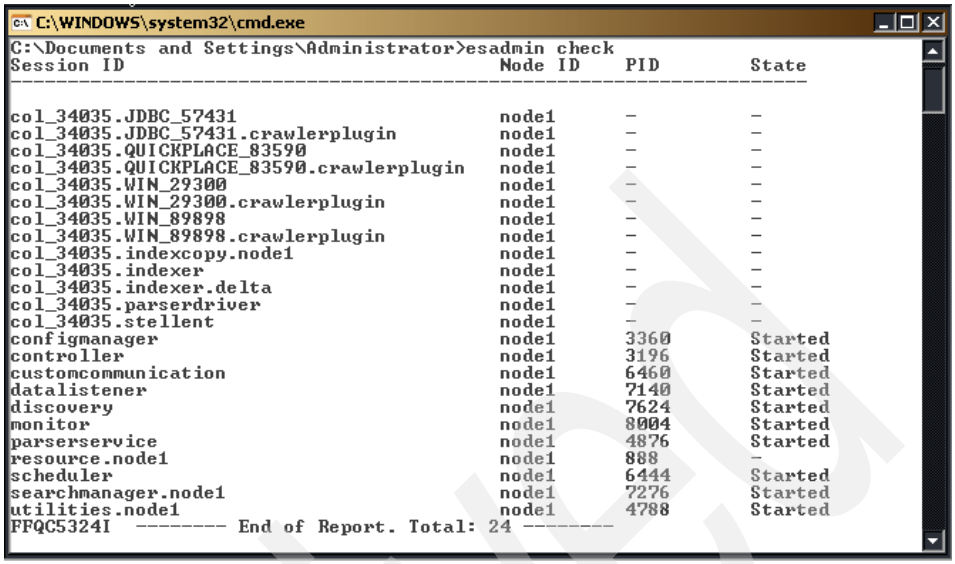

Figure F-1 esadmin check output

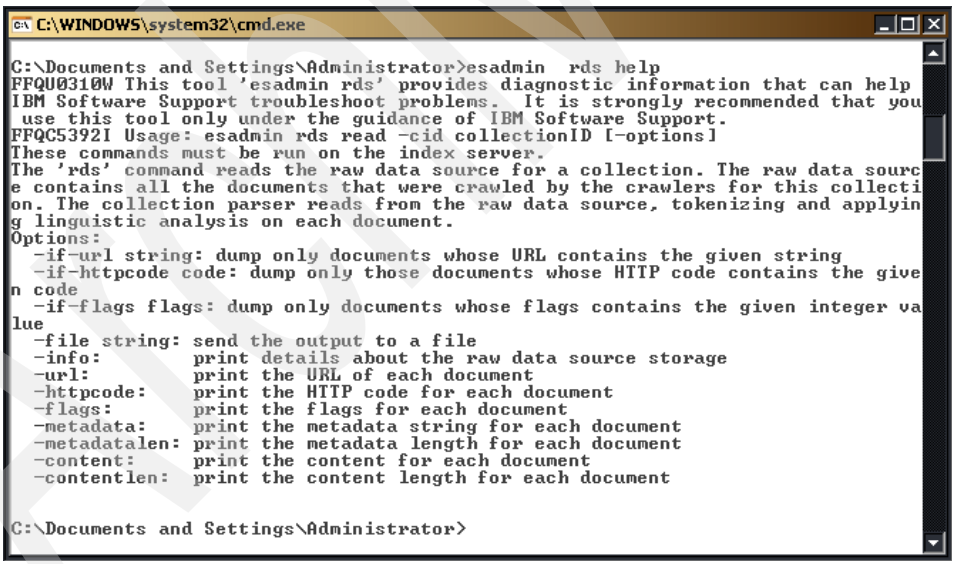

Figure F-2 esadmin rds help output

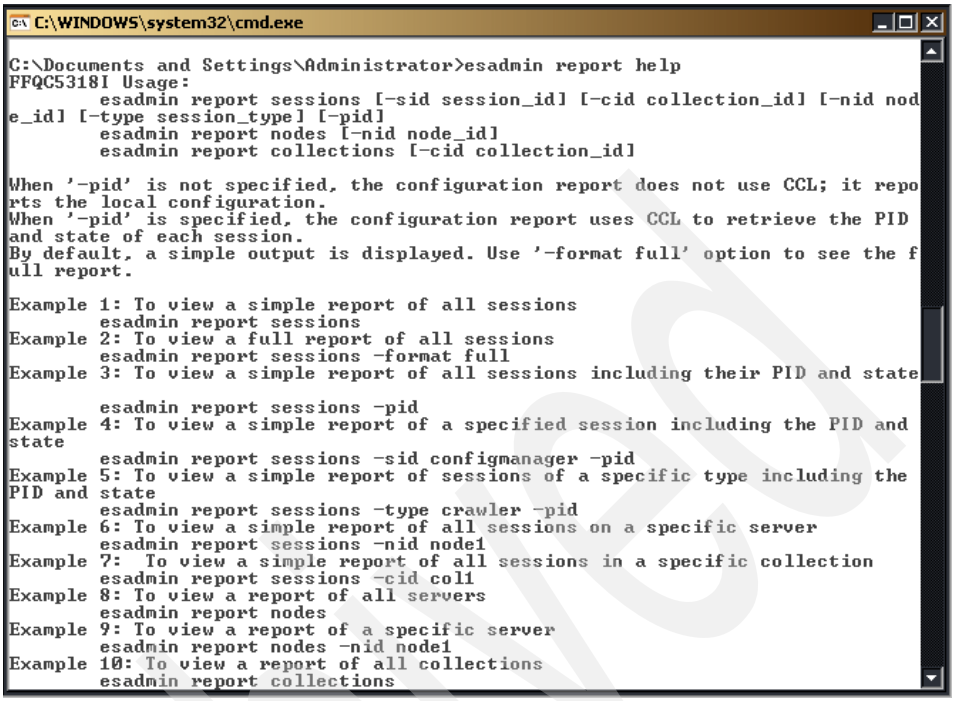

Figure F-3 esadmin report help output

Example: F-1 esadmin session discovery discover -api command options

```
esadmin session discovery discover -api <apiname> <options>
-- where apiname can be any of the following, and options can be any one associated with each apiname
-- For example,
-- esadmin session discovery discover -api getJDBCDriverClasspath -driver
--
-- List of APIs and their corresponding options
--
getDB2Databases -system -database -host -port

validateDB2User -url -user -password -driver

getDB2TargetData -url -user -password -schema -table -driver

getDB2Fields -url -user -password -schema -table -driver -qRepSchema -publishingQMap

getDB2PublishingQMaps -url -user -password -driver -schemaX -tableX

getDB2PublishingCondition -url -user -password -schema -table -driver -qRepSchema -publishingQMap

validateMQConnection -queueManager -queue -host -port -channel

getSupportedDatabases

validateDatabaseUser -url -user -password -driver -driverpath

getDatabaseTargetData -url -user -password -schema -table -driver -driverpath

getJDBCDriverClasspath -driver
```

```
getDatabaseFields  -url -user -password -schema -table -driver -driverpath

getNotesDatabases -server -protocol -id -password -database -flags @DIIOP
--
-- @DIIOP options are -iormethod -iorparam -usediiops -tcpath
--

getNotesViewsAndFolders -server -protocol -database -id -password -view -folder -flags -showhiddenviews @DIIOP

validateNotesIDFile -server -protocol <0:nrpc, 1:diiop> -id -password -filter_id -flags @DIIOP

validateNotesFields -server -database -id -password -field @DIIOP

validateNotesDatabases -server -protocol -id -password -databaseX @DIIOP

getNotesDatabaseDirectories -server -protocol -id -password -directory -flags @DIIOP

getNotesDomainName -server -protocol -id -password @DIIOP

getNotesUserInformation -server -protocol -id -password -validateuser @DIIOP

getNotesUserInformation -server -protocol -id -password -ltpatoken @DIIOP

validateNotesUser -server -protocol -id -password -validateuser -validatepassword @DIIOP

validateNotesUser -server -protocol -id -password -ltpatoken @DIIOP

getCMServers -server

getCMItemTypes -server -user -password -itemtype

getCMAttributes -server -user -password -itemtype

validateCMUserID -server -user -password

getCMUserInformation -server -user -password -validateuser

getDirectoryList -rootpath -level

getWinDirectoryList -rootpath -level -user -password

validateWinDomain -domain

getWinDomainName -rootpath

validatePublicFolder -url -user -password -trustpath -truthpassword

getPublicFolderList -url -user -password -trustpath -truthpassword -folder -level

getVBRRepositories -repository -jndi_factory -jndi_provider

getVBRRepositoriesDirect -repository

validateVBRUserID -jndi_factory -jndi_provider -repository -user -password -optional

validateVBRUserIDDirect -repository -user -password -optional

validateVBRUserID -jndi_factory -jndi_provider -repository -ltpatoken

validateVBRUserIDDirect -repository -ltpatoken

getVBRItemClasses -jndi_factory -jndi_provider -repository -user -password -optional -itemclass

getVBRItemClassesDirect -repository -itemclass

getVBRProperties -jndi_factory -jndi_provider -repository -user -password -optional -itemclass
```

```
getVBRPropertiesDirect -repository -user -password -optional -itemclass

validateVBRFolders -jndi_factory -jndi_provider -repository -user -password -optional -folder AAAA|BBBB|CCCC

validateVBRFoldersDirect -repository -user -password -optional -folder AAAA|BBBB|CCCC

getVBRDataMaps -jndi_factory -jndi_provider -repository -user -password -optional

getVBRDataMapsDirect -repository -user -password -optional

getVBRUserInformation -jndi_factory -jndi_provider -repository -user -password -optional -validateuser

getVBRUserInformationDirect -repository -user -password -optional -validateuser

getDominoDocLibraries -server -protocol -id -password -library -flags @DIIOP

getDominoDocCabinets -server -protocol -id -password -cabinet -flags @DIIOP

validateDominoDocDatabases -server -protocol -id -password -database

getDominoDocDomainName -server -id -protocol -password @DIIOP

getDominoDocUserInformation  -server -id -protocol -password -validateuser @DIIOP

getDominoDocUserInformation  -server -id -protocol -password -ltpatoken @DIIOP

validateDominoDocUser -server -id -protocol -password -validateuser -validatepassword @DIIOP


validateDominoDocUser -server -id -protocol -password -ltpatoken @DIIOP


getQuickPlacePlaces -server -protocol -id -password -place -flags @DIIOP
getQuickPlacePlaceAndRooms -server -protocol -id -password -placedirectory -room -flags @DIIOP

validateQuickPlaceDatabases -server -protocol -id -password -databaseX @DIIOP

validateLdapServer -server -port -baseDn -directoryUser -directoryPassword -trustpath -trustpassword @DIIOP

getQuickPlaceUserInformation -server -protocol -id -password -validateuser -port -baseDn -trustpath -trustpassword @DIIOP

validateQuickPlaceUser -server -protocol -id -password -validateuser -validatepassword @DIIOP

validateQuickPlaceUser -server -protocol -id -password -ltpatoken @DIIOP

validateNNTPServer -host -user -password

validateNNTPGroups -host -user -password -newsgroups

validateWPUser -url -user -password -auth_type -auth_user -auth_password -auth_startUrl -auth_formName -formKeyX
-formValueX -proxy_server -proxy_port -proxy_user -proxy_password -trustpath -trustpassword

getJDBCTargetData -url -user -password -schema -table -driver

getJDBCFields -url -user -password -schema -table -driver
```

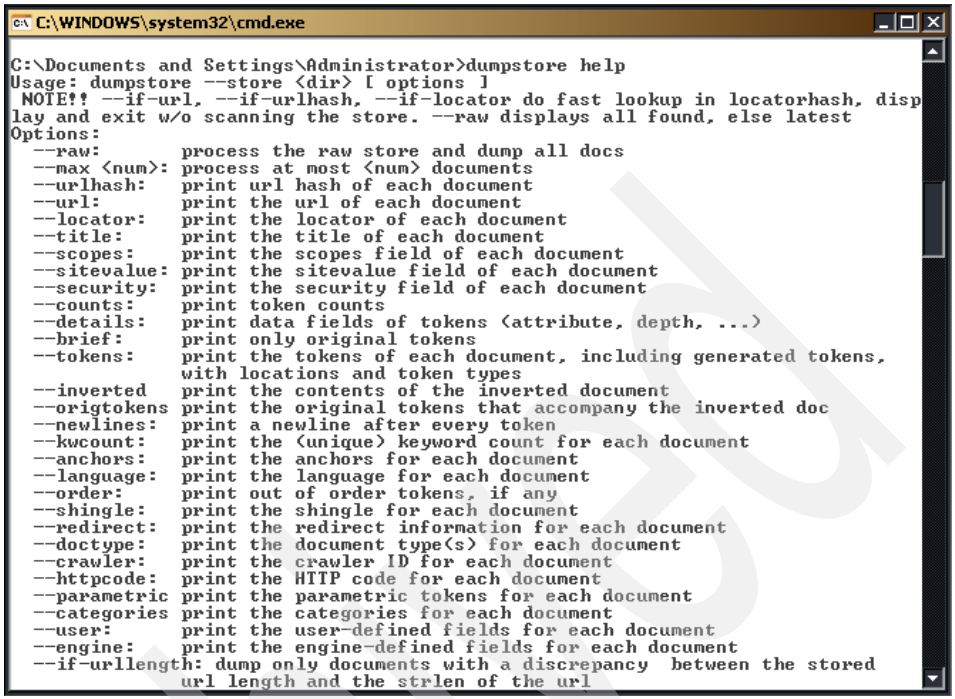

Figure F-4 dumpstore help

Example: F-2 Dumpstore RDS with Domino, DB2 Content Manager, and DB2 (no security) data

```
esadmin rds read -cid IBMCIF -metadata -url

URL: domino://kazan.itsosj.sanjose.ibm.com/8825724500710701/TrainingOfferings.nsf//11C275F9B62B87408825724500710984
Metadata: <?xml version='1.0' encoding='UTF-8'?>
<Metadata Language="en">
  <CommonMetadata Datasource="Notes" StaticScoreRef="0" CrawlerId="IBMCIF.NOTES_46149" DatasourceName="IBM Offerings"
Date="1166231707">
    <HasSeparateContent Language="en" Truncated="NO">NO</HasSeparateContent>
    <SecurityACLs Public="NO"></SecurityACLs>
    <NativeACLs>          <Impersonate>YES</Impersonate>
            <Protocol>DIIOP</Protocol>
            <Domain>ITSOQP!!kazan.itsosj.sanjose.ibm.com!!NOTES</Domain>
            <Level1>
                <Allow>
                    <Entry>-Default-</Entry>
                </Allow>
            </Level1>
            <Level2>
                <Allow>
                    <Entry>CN=Administrator/O=IBM</Entry>
                    <Entry>LocalDomainServers</Entry>
                    <Entry>-Default-</Entry>
                </Allow>
            </Level2>
</NativeACLs>
  </CommonMetadata>
  <DatasourceSpecificMetadata>
```

```
    <Field FieldName="ServerName" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">kazan.itsosj.sanjose.ibm.com</Field>
    <Field FieldName="DatabaseName" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">TrainingOfferings.nsf</Field>
    <Field FieldName="DatabaseTitle" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">Training Offerings</Field>
    <Field FieldName="CreatedDate" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="YES"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1166214880000</Field>
    <Field FieldName="ModifiedDate" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="YES"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1166231707000</Field>
    <Field FieldName="Status" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1.0</Field>
    <Field FieldName="Creator" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">CN=Administrator/O=IBM</Field>
    <Field FieldName="NotifyAfter" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">0</Field>
    <Field FieldName="Resubmit" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">0</Field>
    <Field FieldName="SubmitNow" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">0</Field>
    <Field FieldName="Body" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="YES" ExactMatch="NO" Sortable="NO"></Field>
    <Field FieldName="CurrentEditor" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO"></Field>
    <Field FieldName="ReviewWindow" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">0</Field>
    <Field FieldName="AltFrom" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">CN=Administrator/O=IBM</Field>
    <Field FieldName="CurrentUser" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">CN=Administrator/O=IBM</Field>
    <Field FieldName="Categories" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">Service &gt; Training &gt; Software
&gt;</Field>
    <Field FieldName="Form" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">Document</Field>
    <Field FieldName="_UpdatedBy" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">CN=Administrator/O=IBM</Field>
    <Field FieldName="Title" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">IBM WebSphere Portal Training</Field>
    <Field FieldName="_Revisions" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">Fri, 15 Dec 2006 14:36:02 CST</Field>
    <Field FieldName="_Revisions" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">Fri, 15 Dec 2006 14:39:41 CST</Field>
    <Field FieldName="readers" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO"></Field>
    <Field FieldName="ReviewType" Searchable="YES" FieldSearchable="NO" Metadata="NO" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1</Field>
    <Field FieldName="Protocol" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">DIIOP</Field>
    <Field FieldName="DocumentUNID" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">11C275F9B62B87408825724500710984</Field>
    <Field FieldName="View" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO"></Field>
    <Field FieldName="NoteID" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">8F6</Field>
    <Field FieldName="IsUseSSL" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">false</Field>
    <Field FieldName="Date" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="YES"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1166231707000</Field>
  </DatasourceSpecificMetadata>
</Metadata>
...................
...................
URL: cm://icmnlsdb/Support_TS/85+3+ICM8+icmnlsdb7+ICMBASE58+26+A1001001A06L15B35837E3216618+A06L15B35837E321661+13+300
Metadata: <?xml version='1.0' encoding='UTF-8'?>
<Metadata Language="en">
```

```
  <CommonMetadata Datasource="CM" StaticScoreRef="0" CrawlerId="IBMCIF.CM_89816" DatasourceName="IBM Support"
Date="1166219917">
    <HasSeparateContent ContentType="application/msword" Language="en" Truncated="NO"
Filename="E:&#x5C;SW&#x5C;Docs_and_Categorization&#x5C;DataStores&#x5C;Nile_CM&#x5C;Support_Troubleshooting(TS)&#x5C;IBM -
Support - Premium Service.doc">YES</HasSeparateContent>
    <SecurityACLs Public="NO"></SecurityACLs>
    <NativeACLs>            <Impersonate>YES</Impersonate>
            <Domain>icmnlsdb</Domain>
            <Level1>
                <Allow>
                    <Entry>ICMADMIN</Entry>
                    <Entry>UNDERWRITER1</Entry>
                    <Entry>ADJUSTER1</Entry>
                    <Entry>ESADMIN</Entry>
                    <Entry>ADJUSTER2</Entry>
                    <Entry>UNDERWRITER2</Entry>
                </Allow>
            </Level1>
</NativeACLs>
  </CommonMetadata>
  <DatasourceSpecificMetadata>
    <Field FieldName="ServerName" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">icmnlsdb</Field>
    <Field FieldName="ItemTypeName" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">Support_TS</Field>
    <Field FieldName="ObjectType" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">ICMBASE</Field>
    <Field FieldName="SemanticType" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">Base</Field>
    <Field FieldName="CreatedDate" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="YES"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1166219917515</Field>
    <Field FieldName="ModifiedDate" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="YES"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1166219917515</Field>
    <Field FieldName="VersionID" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1</Field>
    <Field FieldName="sup_status" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="YES" ExactMatch="NO" Sortable="NO">published</Field>
    <Field FieldName="Category" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="YES" ExactMatch="NO" Sortable="NO">Support &gt; Troubleshooting &gt;
</Field>
    <Field FieldName="Mimetype" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">application/msword</Field>
    <Field FieldName="OrgFileName" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO"
Sortable="NO">E:&#x5C;SW&#x5C;Docs_and_Categorization&#x5C;DataStores&#x5C;Nile_CM&#x5C;Support_Troubleshooting(TS)&#x5C;I
BM - Support - Premium Service.doc</Field>
    <Field FieldName="PartNumber" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1</Field>
    <Field FieldName="Date" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="YES"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1166219917515</Field>
  </DatasourceSpecificMetadata>
</Metadata>
..................
..................
URL: db2://SAMPLE/ADMINISTRATOR.DEPARTMENT/DEPTNO/IBM
Metadata: <?xml version='1.0' encoding='UTF-8'?>
<Metadata Language="en">
  <CommonMetadata Datasource="DB2" StaticScoreRef="0" CrawlerId="IBMCIF.DB2_99868" DatasourceName="Support Locations"
Date="1166233134">
    <HasSeparateContent Language="en" Truncated="NO">NO</HasSeparateContent>
    <SecurityACLs Public="YES"/>
  </CommonMetadata>
  <DatasourceSpecificMetadata>
    <Field FieldName="DatabaseName" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">SAMPLE</Field>
    <Field FieldName="TableName" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">ADMINISTRATOR.DEPARTMENT</Field>
```

```
    <Field FieldName="Category" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="YES" ExactMatch="NO" Sortable="NO">Support &gt; Location &gt;</Field>
    <Field FieldName="Name" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="YES" ExactMatch="NO" Sortable="NO">IBM ITSO</Field>
    <Field FieldName="Number" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">IBM</Field>
    <Field FieldName="Location" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="YES" ExactMatch="NO" Sortable="NO">San Jose        </Field>
    <Field FieldName="Telephone" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="YES" ExactMatch="NO" Sortable="NO">408-000-0000</Field>
    <Field FieldName="Date" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="YES"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1166233134308</Field>
  </DatasourceSpecificMetadata>
</Metadata>
.....
.....
```

Example: F-3 Dumpstore trevistore with Domino, DB2 Content Manager, and DB2 (no security) data

```
DumpStore --store /var/es/data/IBMCIF/indexbuild/0/trevistore --security --url

URL: domino://kazan.itsosj.sanjose.ibm.com/8825724500710701/TrainingOfferings.nsf//11C275F9B62B87408825724500710984
  Security: protocol=DIIOP domainname=ITSOQP!!kazan.itsosj.sanjose.ibm.com!!NOTES impersonation=true
  URL:
domino://kazan.itsosj.sanjose.ibm.com/8825724500710701/TrainingOfferings.nsf//11C275F9B62B87408825724500710984?AttNo=0&Att
.............
.............
URL: cm://icmnlsdb/Support_Pubs/85+3+ICM8+icmnlsdb7+ICMBASE58+26+A1001001A06L15B35700H8169718+A06L15B35700H816971+13+300
  Security: domainname=icmnlsdb impersonation=true
.............
...........
URL: db2://SAMPLE/ADMINISTRATOR.DEPARTMENT/DEPTNO/C01
  Security:

...........
...........
```

Example: F-4 Dumpstore RDS with DB2 (security tokens) data

```
..........
..........
URL: jdbc://jdbc%3Adb2%3A%2F%2Fnile.itsosj.sanjose.ibm.com%3A50000%2FNWINSUR/ADMINISTRATOR.CUST_VIEW/CUSTOMERID/1002
Metadata: <?xml version='1.0' encoding='UTF-8'?>
<Metadata Language="en">
  <CommonMetadata Datasource="Database" StaticScoreRef="0" CrawlerId="col_34035.JDBC_57431" DatasourceName="DB2_JDBC"
Date="1168297627">
    <HasSeparateContent Language="en" Truncated="NO">NO</HasSeparateContent>
    <SecurityACLs Public="NO">PERSONAL</SecurityACLs>
  </CommonMetadata>
  <DatasourceSpecificMetadata>
    <Field FieldName="DatabaseName" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO"
Sortable="NO">jdbc:db2://nile.itsosj.sanjose.ibm.com:50000/NWINSUR</Field>
    <Field FieldName="TableName" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">ADMINISTRATOR.CUST_VIEW</Field>
    <Field FieldName="CUSTOMERID" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="YES"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1002</Field>
    <Field FieldName="FIRSTNAME" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="YES" ExactMatch="NO" Sortable="NO">Chandler</Field>
    <Field FieldName="LASTNAME" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="YES" ExactMatch="NO" Sortable="NO">Youk</Field>
    <Field FieldName="POLICYID" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="YES"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">234</Field>
    <Field FieldName="ENDDATE" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="YES"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">12021120000000</Field>
    <Field FieldName="TYPENAME" Searchable="YES" FieldSearchable="YES" Metadata="YES" ParametricSearch="NO"
ResolveConflict="MetadataPreferred" IsContent="YES" ExactMatch="NO" Sortable="NO">PERSONAL</Field>
    <Field FieldName="Date" Searchable="NO" FieldSearchable="YES" Metadata="YES" ParametricSearch="YES"
ResolveConflict="MetadataPreferred" IsContent="NO" ExactMatch="NO" Sortable="NO">1168297627625</Field>
  </DatasourceSpecificMetadata>
</Metadata>
..........
..........
```

Example: F-5 Dumpstore trevistore with DB2 (security tokens) data

```
.............
.............
  Inverted Document:    Inverted Vector [PLAIN]:
      234: @59
      1002: @58
      youk: @103
      ibm:url: @30
      12021120000000: @60
      nile:databasename: @49
      chandler: @82
      sanjose:databasename: @51
      chandler\01: @82
      com%3a50000%2fnwinsur:url: @31
      com\01:databasename: @53
      cust_view:tablename: @57
      cust:url: @33
      itsosj.sanjose.ibm.com%3a50000%2fnwinsur:site: @44
      chandler:firstname: @82
      234:policyid: @59
      com%3a50000%2fnwinsur:site: @41
      personal:typename: @124
      nwinsur:databasename: @55
      view:url: @34
      com:databasename: @53
      administrator\01:tablename: @56
      jdbc:databasename: @47
      ibm.com%3a50000%2fnwinsur:site: @42
```

```
      database:esfield.datasource: @1
      itsosj:databasename: @50
      db2\01:databasename: @48
      1168297627:esfield.data: @2
      1168297627625:date: @61
      youk:lastname: @103
      jdbc%3adb2%3a%2f%2fnile.itsosj.sanjose.ibm.com%3a50000%2fnwinsur:site: @45
      personal\01:typename: @124
      ibm\01:databasename: @52
      db2:databasename: @48
      en:esfield.language: @5
      personal\01: @124
      1002:customerid: @58
      ibm:databasename: @52
      col_34035.jdbc_57431:esfield.crawler_id: @0
      sanjose:url: @29
      customerid:url: @35
      administrator:url: @32
      db2_jdbc:esfield.datasource_name: @46
      chandler\01:firstname: @82
      jdbc%3adb2%3a%2f%2fnile:url: @27
      en:language: @4
      itsosj:url: @28
      50000:databasename: @54
      text/plain:doctype: @6
      sanjose.ibm.com%3a50000%2fnwinsur:site: @43
      1002:url: @36
      PERSONAL:esfield.security_acl: @3
      personal: @124
      administrator:tablename: @56
      1202112000000:enddate: @60
      nile\01:databasename: @49
..............
..............
```

Miscellaneous

This section includes some miscellaneous troubleshooting information, such as the content location of Cloudscape data, collection data, the build level of the OmniFind installation, and the administration console GUI monitoring facility.

Cloudscape data

The Cloudscape database for crawlers, data listener, and IMC user credentials are stored on the Crawler server in the $ES_NODE_ROOT/data/cloudscape directory and sub-directories as follows:

- Crawled document metadata is stored in $ES_NODE_ROOT/data/cloudscape/OmniFind_crawlers/<collection id>.<crawler id>.
- Add/remove URIs collection data is stored in $ES_NODE_ROOT/data/cloudscape/OmniFind_datalistener/.
- IMC credentials user profile data is stored in $ES_NODE_ROOT/data/cloudscape/OmniFind_imc/.

Collection data

All collection related data, such as index, raw data store, and synonym dictionaries, is stored in the $ES_NODE_ROOT/data directory and sub-directories as follows:

- Main and delta index data is stored in the Indexer and Search servers in the $ES_NODE_ROOT/data/<collection id>/indexbuild/0|1/ directory. build_info.ini specifies which directory is the current active index.
- Crawled documents raw data store is stored in the Indexer server in the $ES_NODE_ROOT/data/<collection id>/rds directory.
- Internal stellent session data is stored in the Indexer server in the $ES_NODE_ROOT/data/<collection id>/stellent directory.
- Web crawler cookie data is stored in the Crawler server in the $ES_NODE_ROOT/data/<collection id>.<crawler id> directory.
- Uploaded synonym, boost word, and stop word dictionaries are stored in the Indexer and Search servers in the $ES_NODE_ROOT/data/<collection id>/custom_dictionary directory.

Build level of OmniFind

The bldinfo.txt file in the $ES_NODE_ROOT/nodeinfo directory contains information about the level of an existing OmniFind installation, as shown in Example F-6.

Example: F-6 bldinfo.txt file contents

```
Release:    esrchr4
Level:      846
VRCF:       8.4.0.100
Time:       Sat Nov 11 10:52:04 PST 2006
bld_mode:   x86_nt_5_msvc7
Host:       wp31
OS:         Windows 2000
bld_type:   oe
```

The Level parameter should be interpreted as follows:

- OmniFind V8.2 GA is build #300.
- OmniFind V8.2.1 (Fix Pack 1) is build #352.
- OmniFind V8.2.2 (Fix Pack 2) is build #266.
- OmniFind V8.3 GA is build #547.
- OmniFind V8.3 Fix Pack 1 (hotfix 13) is build #561.
- OmniFind V8.3 Fix Pack 2 (hotfix 18) is build #569.
- OmniFind V8.4 is Release esrchr4 and build #846.

Administration console GUI monitoring facility

Some of the information recorded in the various logs produced during crawling, parsing, indexing, and searching can be viewed from the administration console GUI, as shown in Figure F-5 through Figure F-6 on page 581.

From the Collections view in Monitor mode for the NWINSURANCE collection under the Log tab, select the log to view in the Log file box, the Filters of interest, and click **View log**, as shown in Figure F-5.

Figure F-6 on page 581 shows the summary messages. Click **Details** to view further information about an individual message, as shown in Figure F-7 on page 582.

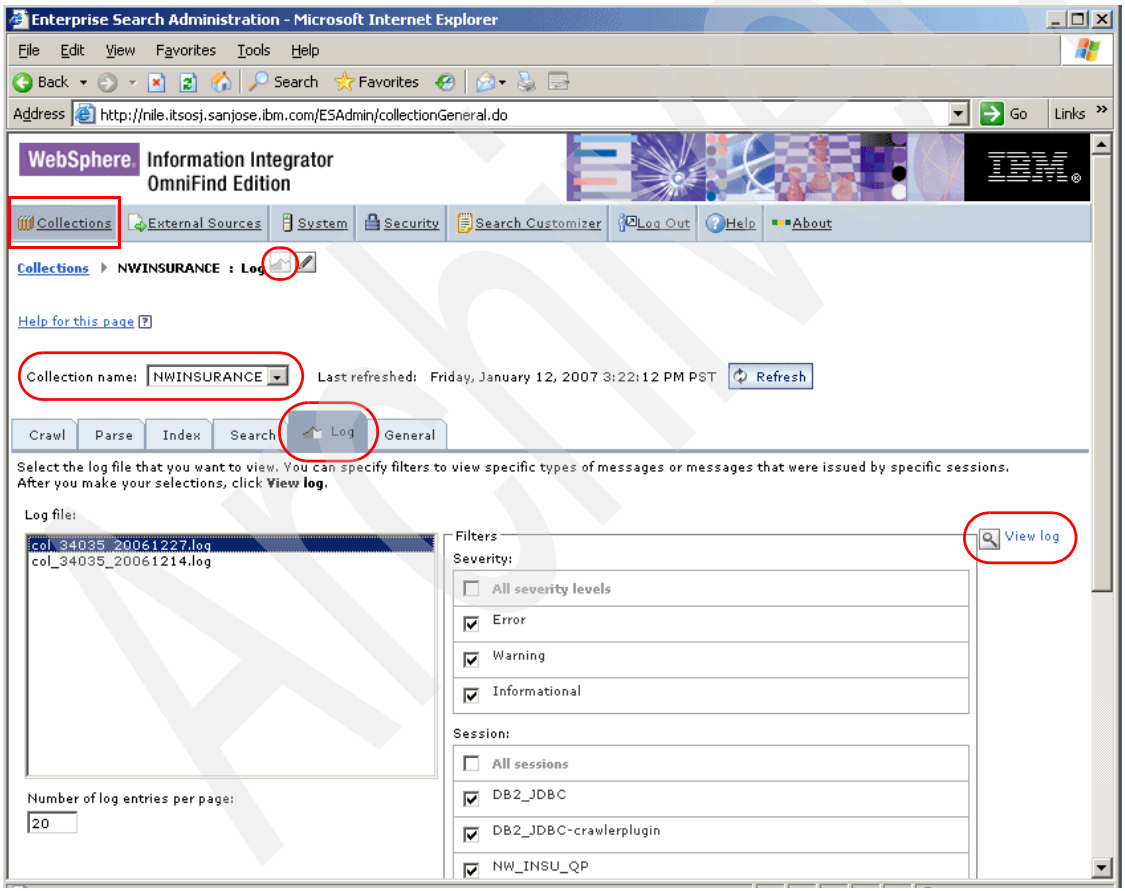

Figure F-5 View logs through the administration console GUI 1/3

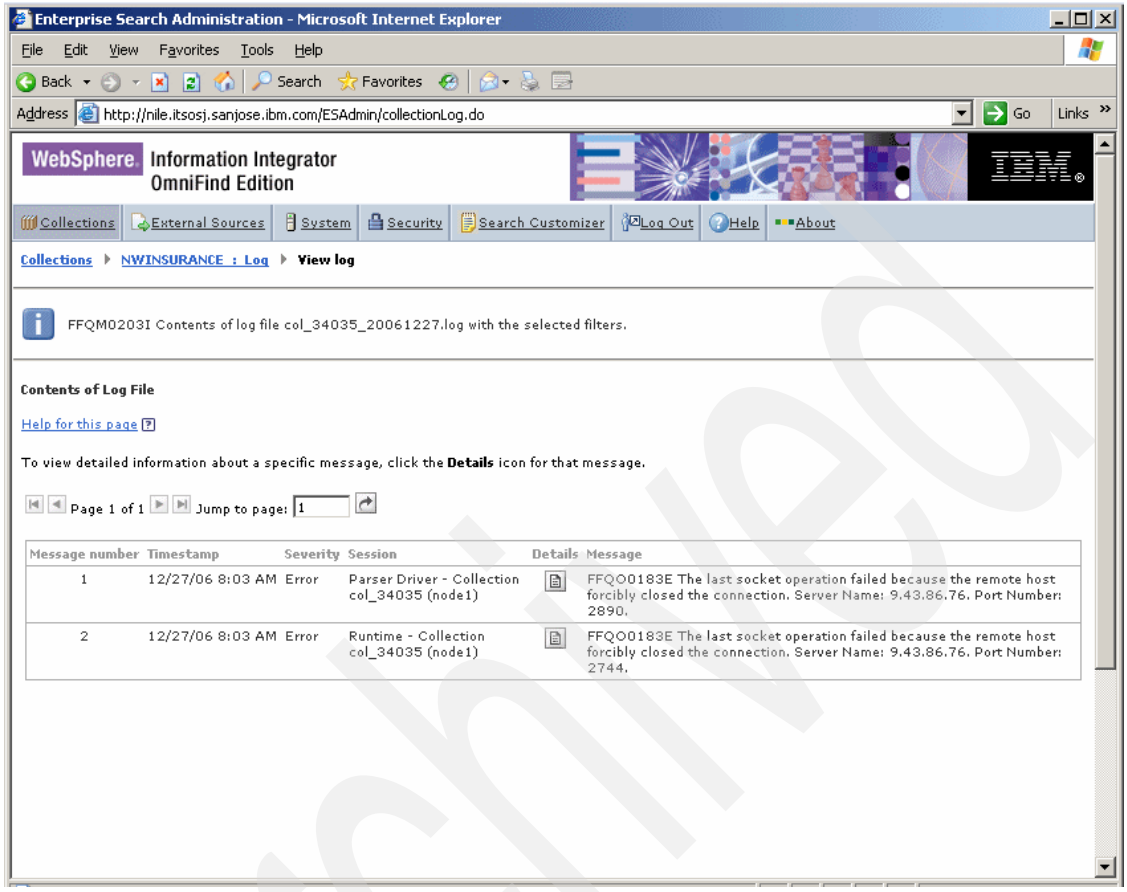

Figure F-6 View logs through the administration console GUI 2/3

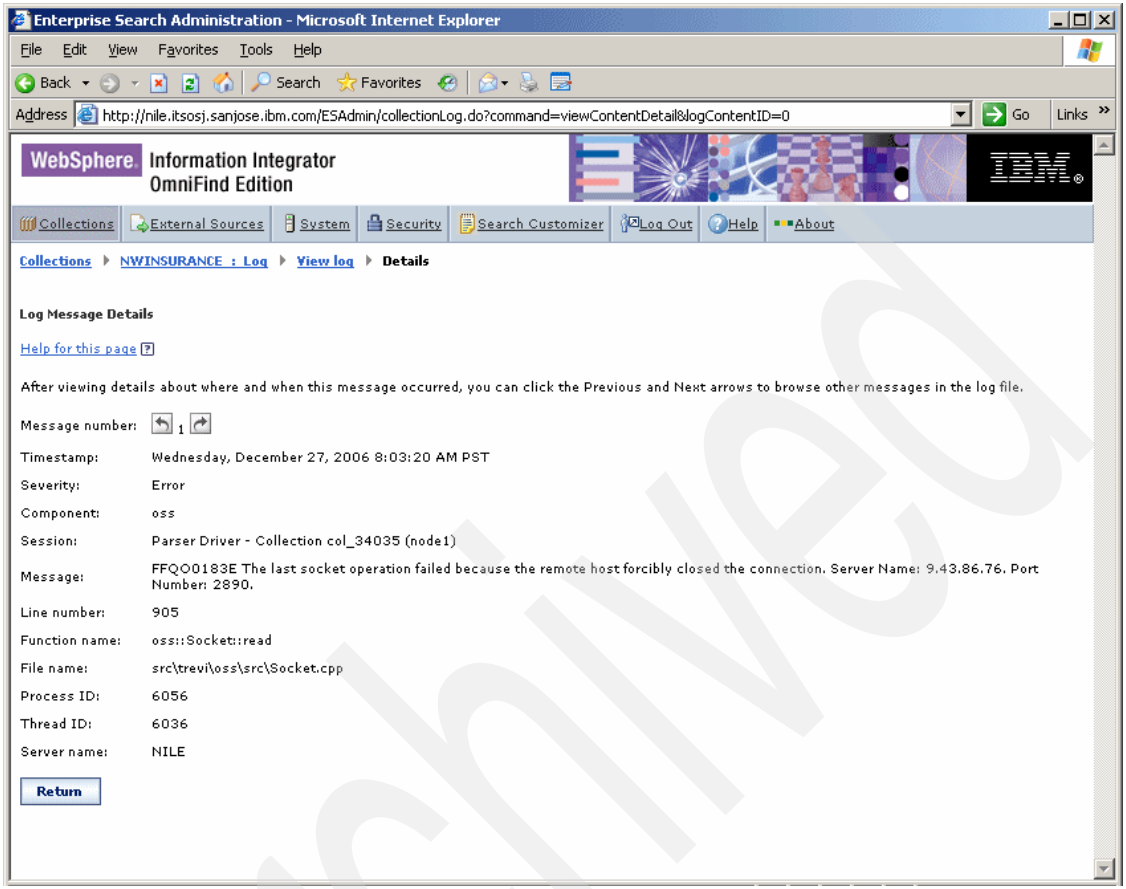

Figure F-7 View logs through the administration console GUI 3/3

G

Additional material

This book refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/SG247394

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the Redbooks form number, SG247394.

Using the Web material

The additional Web material that accompanies this book includes the following files:

File name	*Description*
SG247394.zip	Zipped code samples

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space: 2 MB
Operating System: Windows

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 586. Note that some of the documents referenced here may be available in softcopy only.

- *WebSphere Information Integrator OmniFind Edition: Fast Track Implementation*, SG24-6697

Other publications

These publications are also relevant as further information sources:

- *IBM OmniFind Enterprise Edition Version 8.4 Administering Enterprise Search*, SC18-9283
- *IBM OmniFind Enterprise Edition Version 8.4 Installation Guide for Enterprise Search*, SC18-9282
- *IBM OmniFind Enterprise Edition Version 8.4 Programming Guide and API Reference for Enterprise Search*, SC18-9284
- *IBM OmniFind Enterprise Edition Version 8.4 Text Analysis Integration*, SC18-9674
- *IBM OmniFind Enterprise Edition Version 8.4 Troubleshooting Guide and Messages References*, GC18-9673
- *IBM OmniFind Enterprise Edition Version 8.4 Plug-in for Google Desktop Search*, SC19-1003
- Technote on “OmniFind Enterprise Edition returns error FFQE0108E on the My Profile panel when using single sign-on through LTPA with WebSphere Portal Document Manager”, found at:

  `http://www-1.ibm.com/support/docview.wss?rs=0&q1=1252053&uid=swg21252053&loc=en_US&cs=utf-8&cc=us&lang=en`

Online resources

These Web sites are also relevant as further information sources:

- IBM OmniFind Enterprise Edition home page

  http://www.ibm.com/software/data/enterprise-search/omnifind-enterprise/

- Introduction to WebSphere Information Integrator OmniFind Edition tutorial

  http://www.ibm.com/developerworks/edu/dm-dw-dm-0503buehler-i.html

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

D

E

F

G

H

I

Q

R

S

T

U

V

W

IBM OmniFind Enterprise Edition Version 8.4 Configuration and Implementation Scenarios

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

IBM OmniFind Enterprise Edition Version 8.4

Configuration and Implementation Scenarios

IBM OmniFind Enterprise Edition V8.4 architecture

Security considerations

Business scenarios using single sign-on (SSO)

This IBM Redbooks publication documents the procedures for implementing IBM OmniFind Enterprise Edition Version 8.4 technology in a single-server Windows environment, two-server Linux environment, and a four-server AIX environment. Supported data sources include DB2, Windows file system, DB2 Content Manager, Lotus Domino, Lotus Quickplace, WebSphere Portal Server, Web Content Management (WCM), and Portal Document Manager (PDM). Tivoli Directory Server (TDS) is the LDAP repository used in the scenarios.

It is aimed at IT architects and search administrators who are responsible for managing IBM OmniFind Enterprise Edition on Windows 2003, Red Hat Enterprise Linux, and AIX platforms.

The book offers a step-by-step approach to implementing a single-server, two-server, and four-server IBM OmniFind Enterprise Edition environment using typical customer scenarios.

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

SG24-7394-00 ISBN 0738486345