Implementing IBM Content Manager OnDemand Solutions with Case Studies

- Product philosophy and history
- Platform specific implementation guidelines
- Multiple case studies for various platforms

Wei-Dong (Jackie) Zhu
Carol Allen
Terry Brown
James Ilardi
Dewey Jackson
Hassan A Shazly
Edward E Stonesifer
Vanessa T Stonesifer

Redbooks

ibm.com/redbooks

International Technical Support Organization

Implementing IBM Content Manager OnDemand Solutions with Case Studies

December 2007

SG24-7511-00

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

First Edition (December 2007)

This edition applies to Version 8 Release 4 of IBM DB2 Content Manager OnDemand for Multiplatforms (program number 5724-J33), Version 7 Release 1 of IBM DB2 Content Manager OnDemand for z/OS and OS/390 (Program Number 5655–H39), Version 5 Release 4 of IBM DB2 Content Manager OnDemand for i5/OS (Product number 5722-RD1).

Contents

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®
AS/400®
CICS®
DB2®
DS8000®
eServer™
i5/OS®
IBM®
IMS™
Integrated Language Environment®
iSeries®
Language Environment®
Lotus®
MVS™
OS/390®
OS/400®
Parallel Sysplex®
pSeries®
RACF®
Redbooks®
Redbooks (logo) ®
System i®
System Storage®
Tivoli®
VTAM®
WebSphere®
xSeries®
z/OS®
zSeries®

The following terms are trademarks of other companies:

Adobe, PostScript, the Adobe logo, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

NetApp, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

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

NetApp, and the Network Appliance logo are trademarks or registered trademarks of Network Appliance, Inc. in the U.S. and other countries.

Adobe Reader, PostScript, Distiller, Adobe Type Manager, 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.

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

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 will help you implement an IBM Content Manager OnDemand solution from the beginning to the end. We discuss various stages of the Content Manager OnDemand solution implementation, including planning, software installation and configuration, design, application setup and verification, functional testing, performance testing, training, and finally deployment into production.

The book is intended to provide end-to-end implementation guidelines to audiences who already have general Content Manager OnDemand product knowledge. To really help you understand the implementation process, we provide case studies drawn from real-life implementations. We cover all platforms of the Content Manager OnDemand products, which include:

- IBM Content Manager OnDemand for Multiplatforms
- IBM Content Manager OnDemand for i5/OS®
- IBM Content Manager OnDemand for z/OS®

Although all three products have a common code base and use the same Content Manager OnDemand Windows Client and Administration Client, there are architectural differences among the various platforms that Content Manager OnDemand supports. When you install and implement a Content Manager OnDemand solution, customize the implementation according to the hardware and software platform and your individual requirements. We had planned to have a chapter in this book devoted to the implementation steps common to all platforms. But as we worked on this publication, we decided that it would be easier for a reader to refer to a single chapter with implementation guidelines and suggestions for a specific platform. That is why we have separate chapters for implementing Content Manager OnDemand on Multiplatforms, on IBM System i®, and on IBM System z/OS. Some of the information will be repeated, but you will have only one chapter to go to for your environment and another chapter to read on the case studies related to your particular platform.

We suggest that after you read Chapter 1, “Introduction to IBM Content Manager OnDemand” on page 3, you read the implementation chapter and the case studies specific for your platform. Every platform covers information from a slightly different angle, especially for the case studies. If you have time, we recommend that you read the chapters related to other platforms to gain further insights about the product implementation.

We learned a great deal meeting and working with our counterparts who have been working on the same product but on different platforms during this publication project. We shared many implementation success stories and problems, and discussed and learned hints and tips from each other's experiences. We put this book together to share with you our collective knowledge and experiences from all these years. Hopefully, you will learn from this book and gain the real knowledge required to implement a Content Manager OnDemand solution.

As Content Manager OnDemand advocates, it was important for us to remember where the product has come from and has gone through over the years. We decided to include a chapter about the history of Content Manager OnDemand at the beginning of the book.

The team that wrote this book

This book was produced by a team of specialists from around the world working at the Software Development Lab in Boulder, Colorado.

Wei-Dong (Jackie) Zhu is an Enterprise Content Management, Risk, and Discovery Project Leader with the ITSO in San Jose, California. She has more than 10 years of software development experience in accounting, image workflow processing, and digital media distribution (DMD). Her development work with one of the DMD solutions contributed to a first time ever win for IBM of a technical Emmy award in 2005. Jackie holds a Master of Science degree in Computer Science from the University of the Southern California. Jackie joined IBM in 1996. She is a Certified Solution Designer for IBM Content Manager and has managed and lead the production of many Enterprise Content Management, Risk, and Discovery IBM Redbooks publications.

Carol Allen is a Certified Solution Expert for Content Manager OnDemand for i5/OS who specializes in implementations, training, and migration services for IBM customers. Before starting her own consulting business, she worked for IBM for 24 years, concentrating since 1992 on IBM eServer™ iSeries® and AS/400® document archive solutions. She has extensive experience in OnDemand customer implementations and consulting, along with writing and teaching technical classes around the world. Carol holds a Bachelor of Arts degree in psychology from Emory University and a Masters degree in Counseling from Georgia State University. She can be reached by e-mail at `carol@carolallenconsulting.com`.

Terry Brown is a Content Manager OnDemand for i5/OS consultant who specializes in implementation, migration, and training services for IBM customers. Before retiring from IBM in 2000, he spent 31 years with IBM as a software developer working on S/38, AS/400, and iSeries licensed program products. Starting in 1993, he was a developer on the R/DARS and OnDemand products and was responsible for interfacing the OnDemand Common Server to the OnDemand Spool File Archive data base and later on porting the OnDemand Common Server to the iSeries. After retiring from IBM, he has continued working with OnDemand doing implementations, migrations, and training in the United States, Canada, Jamaica, Barbados, and China (Hong Kong S.A.R.). Terry holds a Bachelor of Science degree in Electrical Engineering from Michigan State University. He can be reached by e-mail at `terry@carolallenconsulting.com`.

James Ilardi (Jim) is a Certified Solutions Expert for IBM Content Manager OnDemand Multiplatform. Jim has worked in the technology sector for 25 years, joining IBM in 1995. Jim has designed and installed numerous Content Manager OnDemand systems for some of IBM's largest customers worldwide while he was a member of the Content Management Lab Services team. Jim is also a Certified Solution Designer for IBM Content Manager. Jim has presented at numerous IBM events, including the Content Manager Technical Conferences discussing Content Manager OnDemand backup and recovery strategies.

Dewey Jackson is an Advisory Software Engineer working with the Business Partner Technical Enablement Team for Content Management in North America. He joined IBM in 1996 and has had multiple roles with the OnDemand product organization since then. He has 20 years of experience in application development, UNIX systems administration, and database administration. Dewey received a Bachelor of Science degree in Engineering Design and an Master of Business Administration (M.B.A.) in Management Science from the University of Colorado. He is certified in a number of Content Management products, including Content Manager, Content Manager OnDemand, and Filenet Content Manager. He has authored several white papers and has created eLearning modules for two IBM CommonStore products.

Hassan A Shazly is a Senior Software Engineer with Enterprise Content Management OnDemand. He has over 30 years of software management and development experience in various business and scientific applications. He has been instrumental in the design, development, and product testing of the OnDemand Content Management system. Hassan holds a Ph.D. in Remote Sensing and Image Processing from the University of South Carolina. Hassan joined IBM in 1996. He is both OnDemand and e-Business IBM Certified. He has over 20 publications and has presented at multiple technical conferences.

Edward E Stonesifer is an Executive IT Specialist with the Mid-Atlantic Enterprise Content Management Team in the USA. Ed has 26 years of experience in Information Technology and 18 years of experience with IBM Content Manager OnDemand specializing in the IBM eserver zSeries® platform using OAM, DB2®, and OnDemand z/OS. Ed has been instrumental in providing input to the product development team for enhancements to meet the business requirements for all IBM Content Manager OnDemand z/OS customers. Ed had presented at numerous technical conferences and has authored technical white papers addressing IBM Content Manager OnDemand z/OS.

Vanessa T Stonesifer is a Consulting Services Specialist of IBM Content Manager OnDemand for z/OS at IBM in the USA. Vanessa has 24 years of experience in Information Technology and 13 years of experience with Content Manager OnDemand for z/OS. Vanessa is the Technical Team Lead for Content Management Lab Services on the zSeries platform with more than eight years of experience in providing implementations and migrations for Content Manager OnDemand for z/OS. Her current areas of expertise include OAM, DB2, and Content Manager OnDemand for z/OS. Vanessa has presented at multiple technical conferences and Content Manger OnDemand User Groups and has co-authored a Content Manager OnDemand for z/OS IBM Redbooks publication.

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

Emma Jacobs
Deanna Polms
International Technical Support Organization, San Jose Center

Darrell Bryant
Gregory Felderman
Scott J Martin
Sandi Pond
IBM Software Development Group

Become a published author

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

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

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

`ibm.com/redbooks/residencies.html`

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications 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

Part 1

Implementation guidelines

For chapters in this part, we provide an introduction to IBM Content Manager OnDemand and discuss implementation guidelines based on each platform.

We cover the following topics:

- Introduction to IBM Content Manager OnDemand
- Content Manager OnDemand for Multiplatforms implementation guidelines
- Content Manager OnDemand for i5/OS implementation guidelines
- Content Manager OnDemand for z/OS implementation guidelines

1

Introduction to IBM Content Manager OnDemand

This chapter provides a brief overview of IBM Content Manager OnDemand (Content Manager OnDemand), its product philosophy, and its evolution. It provides answers to the questions: What is Content Manager OnDemand? What is content? What is content management? How did Content Manager OnDemand evolve into the product that it is today?

We cover the following topics:

- IBM Content Manager OnDemand product philosophy
- Content Management OnDemand history

1.1 IBM Content Manager OnDemand product philosophy

To compete in today's global business environment, businesses need to increase both the efficiency and effectiveness of their operations. Conflicting business requirements, such as increasing productivity while reducing costs, and increasing personalization yet at the same time expanding to larger customer bases can only be achieved through more streamlined and coordinated processes. IBM Content Manager OnDemand (Content Manager OnDemand) helps address these issues by securely storing information and managing its delivery on demand, whenever needed, wherever needed. It is part of the enterprise content manager portfolio of IBM products and solutions that are designed to help organizations manage their unstructured data.

Historically, organizations have developed applications using databases, such as DB2 to build applications that manage their data. The database applications have been highly successful in their implementations. The data that is stored in some kind of information (such as database) management system is considered structured data.

In the 1980s, especially after the introduction of the personal computer, vast amounts of digital data started to be generated. This includes data such as reports, presentations, statements, bills, invoices, letters, spreadsheets, documents, e-mails, Web content, and digital audio and video. This data is usually not indexed or stored in any formal information management system and is considered unstructured data. It is estimated that 80 to 85 percent of all corporate information is unstructured, and with the growth of the internet and corporate intranets, the volume and heterogeneity of this information has increased prodigiously.[1]

The unstructured data is referred to as *content*. Content usually belongs to one of the following four categories:

- *Operational content*, such as statements, invoices, scanned images, and faxes
- *Rich media*, such as video and audio files
- *Web content*, such as HTML, graphic files, and business content
- *Workgroup documents*, such as word processing, spread sheets, presentations, and e-mails

[1] Preface, IBM System Journal Volume 43, Number 3, 2004

The IBM enterprise content management portfolio provides a unified framework of products and solutions for managing all forms of content. Content Manager OnDemand is the product that specializes in managing computer-generated report data, such as statements, bills, and invoices.

Content Manager OnDemand is a robust report management system that enables you to:

- *Capture*: Captures various data types from various sources through a batch capture system or interactively through custom built interfaces.
- *Store*: Stores data for immediate retrieval.
- *Search*: Indexes data so that users can easily find the information.
- *Integrate*: Provides access through federated searches to other IBM enterprise content management data and third party products.
- *View*: Supports multiple viewers for different data types, thus providing fast access for browsing and printing the retrieved data.
- *Distribute*: Distributes data to selected users based on defined schedules.
- *Manage*: Expires or archives data based on defined policies.
- *Archive*: Provides data archives online, near-line, or off-line, enabling rapid archiving of data to the storage system.
- *Control*: Controls system and data access, allowing only authorized users to access specified data.

In summary, Content Manager OnDemand enables you to gain control of your information by providing access to your business' data, as needed, regardless of the size of the business or the hardware platform. Content Manager OnDemand improves your organization's bottom line by helping you become more efficient and responsive.

1.2 Content Management OnDemand history

Content Manager OnDemand is an industry leading product that did not come into existence overnight. Its roots originated in the mid-1980s and has since then added features and functions with an ever-expanding customer base. Today, Content Manager OnDemand is considered to be a mission-critical product in many businesses ranging from small businesses to the Fortune 500 enterprises.

Perhaps the most important fact to know about the evolution of various IBM solutions into the current Content Manager OnDemand product is that the product has been modified and enhanced over the years based on direct requests from IBM customers. The success of the product has been based on fulfilling customer requirements for the document archiving needs of their businesses.

Historically, Content Manager OnDemand evolved through several phases:

1. Specialized solutions, which solve specific problems for specific customers on specific platforms.
2. Generalized solutions, which solve general problems for specific customers on specific platforms.
3. Licensed program products, which solve general problems for all customers on specific platforms.
4. Product integration, which solves general problems for customers on all platforms.
5. Cross-product integration, which integrates with other products to solve general problems on all platforms.

Specialized solutions

In the mid to late 1980s, specialized solutions were developed to address specific customer problems. These solutions were customized on a per customer basis and were environment- and system-dependent. These solutions worked very well once they were implemented, but they lacked portability to other hardware platforms.

Two early and highly successful IBM mainframe products in the specialized solution phase were:

- Report Management and Distribution System (RMDS)
- Item Access™ Facility (IAFC)

Report Management and Distribution System (RMDS)

RMDS started as a services offering in 1985. It later became a program product and is still supported by IBM.

RMDS was created to allow capturing of report data from the JES Spool, storing it in the RMDS report library, and subsequently permitting users to view the captured data using various 3270 based viewers. A stand-alone VTAM® based viewer was the most popular viewer with a CICS® viewer coming in as a close second.

RMDS also permitted indexes to be created over captured report data. These indexes could subsequently be used to rapidly position desired report pages while viewing or printing a stored report. The indexes could also be used to restrict access to subsets of the report to selected users or groups.

RMDS also includes a report distribution facility that automatically creates a report print bundle containing specified reports for a particular recipient.

Early releases of RDMS had the following functions:

- Processing line data
- Storing metadata in VSAM key sequenced data sets (KSDS)
- Storing report data in VSAM KSDS or QSAM data sets
- A proprietary data retention management system

With the release of V2.1 of RDMS came additional functions, capabilities, and changes:

- Support for AFP and MODCA data in addition to line data
- VSAM KSDSs were replaced by DB2 tables
- The ability to store report objects as OAM objects (and in RMDS V2.3, to VSAM Linear Data Sets)

Over the years, the RMDS customer base stabilized, with customers migrating their applications to the newer Content Manager OnDemand for z/OS.

Item Access Facility (IAFC)

IAFC started as a services offering in 1990. It was adopted by a handful of customers in the United States but many more in Europe.

IAFC was designed to provide support for high volumes of coded data objects on optical media or low cost DASD (such as the 3390-9). The main goal was the storage of print data for archiving. The stored data is indexed for online access and selective printing.

IAFC was withdrawn in September 2001.

Support for IAFC stored data was merged into Content Manager OnDemand V2.1 and was available as an optional V2.1 feature. This feature was known as IAFC Data Compatibility.

The IAFC Data Compatibility feature supports the searching, retrieval, viewing, and printing of IAFC objects from Content Manager OnDemand for OS/390®. This feature enables an IAFC customer to gain the additional features of Content Manager OnDemand for OS/390 while supporting data access to the IAFC archive.

The IAFC Data Compatibility feature was later included in Content Manager OnDemand for z/OS starting with V7.1.

Generalized solutions

In the early 1990s, generalized solutions replaced the previously developed specialized solutions. The code included all the functions available in the previous offerings but was less dependent on the specific environment in which it was executed. The code on each platform was customized based on requests from IBM customers for that platform.

The services offering were:

- 1991: Report/Data Archive and Retrieval system (R/DARS) for OS/390
- 1993: Report/Data Archive and Retrieval System (R/DARS) for AS/400
- 1993: OnDemand for Multiplatforms

Licensed program products

The service offerings worked well. However, customers requested that IBM convert them to program products. The reason was that OnDemand had become mission-critical to customers' business operations and customers wanted the long term assurances and commitments that come with IBM in support of a program product. So, the three services offerings became three program products with full support from IBM.

Three new program products were:

- 1995: RDARS on AS/400
- 1995: OnDemand V2 Multiplatforms
- 1997: OnDemand V2 for OS390

Product integration

From a business perspective, maintaining three different platform-specific code bases that provided nearly equivalent functionality did not make sense to IBM or to customers. IBM needed to develop duplicate platform-specific code and customers were locked in to the specific platform that they first started with. So a product integration effort was initiated that resulted in the first code release that supported xSeries®, pSeries®, iSeries, and zSeries systems.

Platform differences could not be totally ignored, so a common server code base was established. This common code base included the client interface and the core server functions. Platform specific extensions to the common code base were maintained where it made sense. For example, the OAM archive manager is unique to the z/OS platform, so that code is also unique to the z/OS platform.

After product integration, the new products announced in 2001 were:

- Content Manager OnDemand for Multiplatforms (Version 7)
- Content Manager OnDemand for OS/390 (Version 7)
- Content Manager OnDemand for AS/400

On the AS/400 (later called the iSeries and then the System i), the version of the Content Manager OnDemand licensed program product is the same version as the operating system. The naming of Content Manager OnDemand server version, however, is the same as with the other hardware platforms.

Product integration comes with many benefits. The common code base, resulting from product integration, meant less time and effort spent developing and testing the common portion of the code (it needed to be done one time instead of three). This enables developers to spend more time addressing other issues such as adding more functions to the product. In addition, more time was spent on addressing scalability, high availability, and performance issues.

The common client interface meant that there would be only one set of common clients that access any of the Content Manager OnDemand servers. It also implied the *possibility* and *ability* to import and export metadata (administrative definitions for applications, application groups, and folders) between platforms. Furthermore, it led to the ability to place the library server and object servers on the platform of choice based on customer requirements.

Cross-product integration

As the market became more established and complex, customers requested cross-product integration and additional functions. During the early 2000s, IBM focused on integrating Content Manager OnDemand with other IBM enterprise content management products as well as with third-party vendors.

Content Manager OnDemand V8.4 enhancements include new Web clients with additional features, including 64-bit addressing, more indexing options, and more diagnostic capabilities. The planned dates for the release of this server level are:

- 2007: Content Manager OnDemand for Multiplatforms
- 2007: Content Manager OnDemand for z/OS
- 2008: Content Manager OnDemand for i5/OS

The focus throughout the Content Manager OnDemand evolution has been to create a high quality product that satisfies customer needs and meets their expectations. This will remain the focus as the product continues to evolve.

2

IBM Content Manager OnDemand for Multiplatforms implementation guidelines

This chapter provides guidelines for implementing an IBM Content Manager OnDemand for Multiplatforms solution.

We cover the following topics:

- Identify project resources
- Content Manager OnDemand server sizing
- Installation and configuration
- Design
- Application setup and verification
- Functional testing
- Performance testing and considerations
- Training
- Deployment into production

2.1 Identify project resources

IBM Content Manager OnDemand for Multiplatform (Content Manager OnDemand) implementations tend to be installed on new hardware, as opposed to being installed on an existing system. In the majority of cases, the new hardware is chosen because the organization is already familiar with the platform (hardware and software) and little additional training (if any) is required to support the application infrastructure. In other cases, the new hardware represents a departure from previous experience and either existing employees are trained to support the new equipment or new employees are hired. Depending on the scope of the implementation, anywhere from one to a dozen (or more) people could be involved in the project.

In order to determine which employees should participate, we look at the tasks involved in administering and maintaining a Content Manager OnDemand system. Any Content Manager OnDemand implementation requires a Content Manager OnDemand administrator, although this position may not require a full-time, dedicated (to Content Manager OnDemand) employee. However, someone in the organization must be able to oversee and coordinate the tasks listed in Table 2-1.

Table 2-1 Content Manager OnDemand tasks and responsibilities

Task	Person/Department responsible
Installing/upgrading hardware	Operations/Hardware
Installing/maintaining software	Operations/Software
Defining/labeling storage volumes	Operations/Storage
Database monitoring	Operations/Database administrator (DBA)
Cache monitoring	Operations/System administrator (SA)
Archive storage monitoring	Operations/Storage
Scheduling daily maintenance	Operations/SA/DBA/Storage
Defining new applications	Content Manager OnDemand administrator and LOB Manager/Users
Defining storage sets	Content Manager OnDemand administrator and Operations/Storage
Defining printers	Content Manager OnDemand administrator and Operations/Printers

Task	Person/Department responsible
Defining Content Manager OnDemand users and groups	Content Manager OnDemand administrator and Operations/Security
Loading reports	Operations/SA
Managing backups	Operations/SA/DBA
Monitoring server performance and tuning	Operations/SA
Solving server/network/application problems	Operations/SA/DBA/Network administrator
Answering user questions	Content Manager OnDemand administrator and help desk
Establishing security/audit policies	Operations/Security
Establish Disaster and Recovery plan	Content Manager OnDemand administrator/ Operations/LOB Manager/Legal department

Note that even if the Content Manager OnDemand administrator were to be responsible for all IT operations shown in the table, this person still needs to work with the Line-Of-Business or department manager and the users in order to define the various applications that Content Manager OnDemand is used for. In general, application definitions include deciding which criteria to be used to search for documents and how long the documents are retained.

2.2 Content Manager OnDemand server sizing

In this section, we discuss the factors to be considered when sizing the Content Manager OnDemand server:

- Architecture and platform
- CPUs
- Memory
- Disks

All server sizing efforts should focus on the first two or three years of the Content Manager OnDemand implementation.There are two primary reasons for this:

- Hardware gets cheaper and faster as time passes.
- Growth and usage rates are difficult to estimate in the long term.

2.2.1 Architecture and platform

The basic Content Manager OnDemand configuration is to install the library server and object server on the same machine. However, you should consider the benefit of moving the object server(s) to another machine, which might be on a different platform. The main advantage in doing this is to move the load process, a very CPU intensive activity (as well as memory and I/O intensive), off of the library server, If the library server is only used for retrievals for part of the day, and the load processing can be accomplished during the remainder of the day (also taking into consideration daily maintenance such as database backups), then a single machine can host both the library server and object server (Figure 2-1).

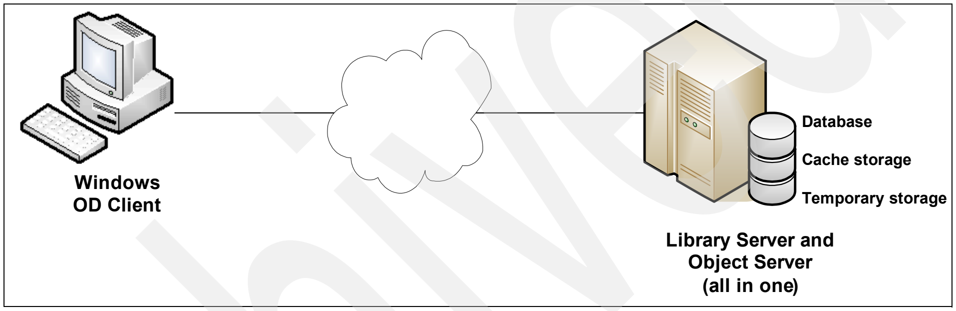


Figure 2-1 Standard library server and object server

If the daily load requirement is substantial (that is, the loading cannot be accomplished during off hours), or business requirements dictate immediate data availability, a separate object server is highly recommended. Multiple object servers are also recommended for a geographically distributed system when the documents need to be located near the users who most frequently access them (see Figure 2-2 on page 15).

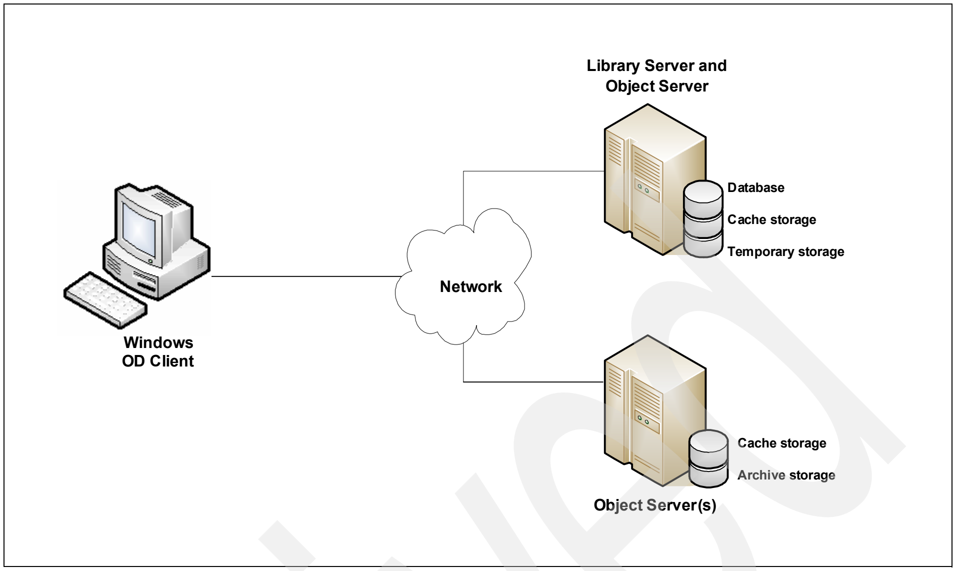


Figure 2-2 Distributed library server and object server

An additional benefit of separating the library server and object server is in reducing the time required for daily maintenance. In the scenario mentioned earlier for the geographically distributed system, the database can be backed up on the library server at the same time **`arsmaint`** runs on the object server.

2.2.2 CPUs

The Central Processing Unit (CPU) performs all of the logic embedded within a Content Manager OnDemand installation. There are three general areas of activity that involve CPUs:

- Loading
- Search and arssockd processes
- Retrieval and arsobjd processes

Loading

Loading is probably the most CPU intensive task. It is dependent on the type of loading done, and whether the data is indexed at load time or is pre-indexed. There are two indexers included with the multiplatform product: the PDF indexer and the ACIF indexer (for AFP and line data). The generic indexer is somewhat of a misnomer, since the indexing is performed outside of Content Manager OnDemand.

Content Manager OnDemand is highly scalable, in large part due to the object server feature. Object servers can be used not only to store data, but to load data as well. A designated object server can be set up to not store any data, but only serve as an additional CPU resource to load data.

Data can also be indexed by the system that generates the print stream. In some implementations, a mainframe does not only generate a line data or AFP print stream, but it also uses the ACIF indexer to generate the index file. Or, virtually any platform can generate a generic index file and build the input file (this is very common for batch scanning systems).

Search and arssockd processes

The search function is the only task that must occur on the library server. Most CPU activity is database related, and involves executing SQL queries (select statements), inserts, updates, or deletes. The arssockd process controls a Content Manager OnDemand instance running on the library server and is the middleman between the hardware, the operating system, and the database process.

There are other Content Manager OnDemand related processes that run on the library server, but they are generally used only for database operations (such as startup, shutdown, database creation, and database maintenance) that do not run when users are using the system.

Retrieval and arsobjd processes

Document retrieval is performed by the object server. Usually just an I/O operation (the arsobjd process controls a Content Manager OnDemand instance on an object server), it may involve additional CPU load if data conversion is performed.

All Content Manager OnDemand for Multiplatform implementations should have a *minimum* of two CPUs for small systems, and four to eight CPUs for any larger system. For two-CPU servers, separate the database from the Content Manager OnDemand processes and operating system. For four- or more-CPU servers, use the additional CPUs for the arsload processes.

2.2.3 Memory

All Content Manager OnDemand multiplatform implementations should have a minimum of 8 GB of memory for the Content Manager OnDemand, database, and operating system software. If there are a significant number of users, add 8 MB of memory per user. For example, 500 users requires 4 GB of additional memory (that is 500 users * 8 MB = 4 GB).

2.2.4 Disks

For best performance, all Content Manager OnDemand implementations should include at least four physical disk drives:

- One for the operating system and paging space
- One for the database
- One for Content Manager OnDemand cache
- One for temporary work space and logging

Most standard hardware configurations include a single large capacity disk, but four (or more) smaller capacity disks can greatly improve performance. The use of RAID or SAN technology can improve performance, but care should be taken when allocating Content Manager OnDemand file systems across these disks to prevent too much I/O on a singe path.

Long term object storage can utilize removable media, such as optical or tape, if older data is infrequently accessed. Otherwise, large capacity but slower disks (Storage Area Network or CAS Based Storage) should be used. This type of storage can also meet the legal requirements necessary for archive data. Check with your legal department when planning this type of storage.

2.3 Installation and configuration

In this section, we discuss the following installation and configuration related tasks:

1. Environment preparation.
2. Add and modify an operating system user.
3. Create Content Manager OnDemand instance.
4. Create migration policies.
5. Configure storage management.
6. Verifying the installation.
7. Implementation best practices.

The complete and detailed installation procedure can be found in the *IBM Content Manager OnDemand for Multiplatforms Installation and Configuration Guide,* SC18-9232.

2.3.1 Environment preparation

Before installing Content Manager OnDemand, you (or someone identified in 2.1, “Identify project resources” on page 12) should be familiar with:

- The proposed and the existing network architecture.
- The operating system of the machine on which Content Manager OnDemand will be installed.
- The database management product which Content Manager OnDemand will use.
- The storage system where documents will be archived (if appropriate).
- The operational requirements for the Content Manager OnDemand system, both short term (daily maintenance such as database backups) and long term (disaster recovery procedures).

Prerequisites for the Content Manager OnDemand product are version specific and can be found in the following locations:

- The product install and configuration guide (platform specific)
- The Content Manager Ondemand V8.4 Information Center, found at:

  http://publib.boulder.ibm.com/infocenter/cmod/v8r4m0//index.jsp

You can also contact the IBM Support Center for the latest maintenance levels of DB2, Content Manager OnDemand, and optionally, Tivoli® Storage Manager, and IBM Infoprint Manager (Infoprint).

Network architecture

Content Manager OnDemand is a client/server application that requires a network running the TCP/IP communications protocol. The network maybe wired or wireless, but it must be capable of high transfer rates; otherwise overloaded networks reflect badly on client/server applications. Network I/O is the resource that most affects the performance of TCP/IP.

Content Manager OnDemand has various features that address the problem of delivering large quantities of information from a server to clients over a network, specifically data compression and large-object support. Data compression will reduce the amount of network traffic, but only if the data is able to be compressed (TIFF, GIF, and JPEG image data do not compress well) and the Windows Client is utilized for document viewing (the client performs the decompression). Large-object support reduces network traffic by only transmitting those sections of the document to be viewed, rather than the entire document.

The data type can affect network traffic in other ways also. For example, one of the advantages of PDF documents is that resources are part of the document and are included once at the beginning. However, if the indexing operation divides the PDF document into many smaller documents (chapters or pages), the resources must be included for each subdocument and the number of bytes stored and transmitted is increased many times over. For small PDF documents (one to five pages), the resources can account for the majority of the document size, especially if custom (other than the base 14) fonts are used.

> **Note:** Base 14 fonts are specific common Type 1 fonts installed as a part of the Adobe Acrobat installation and include:
>
> - Times (v3) or Times New Roman PS MT (v4.x): four versions (regular, bold, italic, bold italic)
> - Helvetica (v3) or Arial MT (v4.x): four versions
> - Courier: four versions
> - Symbol
> - Zapf Dingbats

Similarly, network I/O can be minimized by designing applications properly. For example, limiting the size of the hit-list reduces network traffic by limiting response size. Applications should be designed to discourage wild-card searches, which necessitates complete table scans and can produce very large hit-lists. Or consider a call center application where 90% of document views are of the previous month's statement; a good design would include a date default of the previous 30 days, an index on the account number and setting the Windows Client "AutoView" option (on the Options menu) to **Single Document**. Not only would the hit-list seldom be displayed, but the user would not have to select the document for viewing.

There are additional techniques discussed in "Storage system" on page 20 that reduce storage requirements *and* network (transmission) requirements as well.

Operating system

Content Manager OnDemand for Multiplatforms can be installed on many different operating systems. In order to obtain the latest operating system software updates, go to the appropriate Web site for your operating system:

- AIX®

  http://www14.software.ibm.com/webapp/set2/sas/f/genunix3/aixfixes.html

- HP-UX

  http://www1.itrc.hp.com/service/index.html

- Linux

  Red Hat support can be found at:

  https://www.redhat.com/apps/support/

  SUSE support can be found at:

  http://www.novell.com/support/supportcentral/supportcentral.do?id=m1

- Solaris

  http://sunsolve.sun.com/pub-cgi/show.pl?target=patchpage

- Windows (you must use Internet Explorer v5 or later):

  http://update.microsoft.com/windowsupdate

Database

Content Manager OnDemand supports three different database management products, although the choices are platform dependent. While not typically installed in this manner, different Content Manager OnDemand instances can use different database products. The database product(s) *must* be installed on the library server.

- DB2 (AIX, HP-UX, Solaris, Linux, Linux z/OS, and Windows)

  You can obtain the latest service update from IBM service at:

  ftp://service.software.ibm.com/ps/products/db2/fixes

- Oracle (AIX, HP-UX, Solaris, and Windows)

  Oracle requires additional configuration steps after installation in order to work with Content Manager OnDemand. Contact Oracle for information about the latest maintenance level of Oracle:

  http://www.oracle.com/support/index.html

- SQL Server (Windows only)

  If you are using SQL Server instead of DB2, contact Microsoft for information about the latest Service Packs for SQL Server:

  http://www.microsoft.com/sql/support/default.mspx

Storage system

The most popular storage system for Content Manager OnDemand for Multiplatforms is Tivoli Storage Manager (TSM). Software support can be found at:

http://www.ibm.com/developerworks/linux/linux390/index.html

TSM is not required for Content Manager OnDemand, and some customers use Storage Area Network (SAN) or Network-Attached Storage (NAS) instead. The primary difference between the two alternatives is that SAN utilizes a high-speed subnetwork and therefore delivers better performance.

It is not the intent in this section to describe the benefits of the various storage systems, or the operation of them, but rather how the Content Manager OnDemand implementation can be optimized for any type of storage system. In this respect, the storage system is similar to the network architecture, because we are interested in minimizing the number of bytes stored.

Probably the most apparent method of minimizing storage requirements is to consider each type of compression available with Content Manager OnDemand and use representative sample data to evaluate each compression type with the **`arsadmin compress`** command. There are four types of compression: OD77, OD77Lite, LZW16 and LZW12. Refer to Table 2-2 for an example of the compression ratio (original size / compressed size) for different data types and compression methods. Note that the default method (OD77) is probably the best method to use and that TIFF data should not be compressed. There are two compression methods that are frequently confused with one another: Disable and None. *Disable* was designed for data that is already compressed, such as PDF or compressed TIFF, and the documents are uncompressed by the viewer on the client, for example, Adobe Reader. *None* means Content Manager OnDemand will not compress the input data when loading takes place, but the document will be compressed for transmission when viewed (Windows Client only).

Table 2-2 Compression ratios for various data types

Compression type	Line data	AFP data	PDF data	TIFF data
OD77	7.72	8.59	2.24	1.04
OD77Lite	5.53	4.90	1.94	1.03
LZW12	4.33	1.84	0.92	0.78
LZW16	7.14	3.58	1.13	0.80
Average	6.18	4.73	1.56	0.91

There are less apparent factors that contribute to minimizing storage requirements. For example, when loading AFP data, you should probably never save font information (RESTYPE=FONT or RESTYPE=ALL) unless you will be printing the documents using server print and the fonts are not available on the printer. If AFP documents use a limited number of custom fonts in addition to standard fonts, consider writing a resource user exit to include only the *custom* fonts in the resource file. If the AFP input files are fully composed, you can write an output exit to delete TLE and NOP structured fields (typically used for indexing information) from the output file.

Depending on the number of documents stored, an application should be evaluated using different data type formats. For example, consider an existing application that creates AFP documents. The Content Manager OnDemand application includes a business requirement for customers to be able to view statements through the internet, but the company does not want to impose a requirement for the customer to download and install the AFP plug-in in order to view their statement. There are several options to consider at this point, all of which involve data conversion (AFP to HTML or AFP to PDF). The data conversion can be performed either during loading or as part of the retrieval process, and the choice involves several trade-offs. If the conversion is performed during the load process, the number of bytes stored may increase, or the compression ratio may be lower and both storage and transmission costs are higher. If the conversion is performed at retrieval time, a document may be converted several times during the life of the document, and the performance cost is moved from the object server to the mid-tier Web application server.

Operational requirements

Corporate data is extremely important for any company to function. A backup and recovery strategy of the data should be a part of an overall data management plan. Your system might fail due to many causes. They include human errors, hardware failures, transaction failures, and disasters. Data backup and recovery is so important that another IBM Redbooks publication was written to address the issues involved and provide recommended actions to take: *Content Manager OnDemand Backup, Recovery, and High Availability*, SG24-6444.

Because data backup and recovery is of the utmost importance, this section will not attempt to address the problem and we recommend you read the aforementioned IBM Redbooks publication, which can be downloaded from:

http://www.redbooks.ibm.com/abstracts/sg246444.html?Open

The *Introduction and Planning Guide* for your library server platform also contains a chapter on backup and recovery that details the files and file systems that must be backed up to ensure data protection.

Aside from data protection, other operational requirements also exist. System performance and tuning is an ongoing activity that generally will require daily attention. These tasks include (and are not limited to):

- Disk space usage
- CPU usage

- Page file usage
- Network usage

Once again, we recommend reading another IBM Redbooks publication: *Content Manager OnDemand Guide*, SG24-6915. You can download it at:

http://www.redbooks.ibm.com/abstracts/sg246915.html?Open

2.3.2 Add and modify an operating system user

Here we discuss how to add and modify an operating system user for the Content Manager OnDemand system.

AIX/UNIX

The standard installation procedure assumes that the *root* user will be the Content Manager OnDemand instance owner, be the owner for the cache file system, and run all Content Manager OnDemand processes. Most IT organizations do not want the root user running applications because of the inherent power of the root user. We strongly advise you to create a special user (typically odadmin) for the Content Manager OnDemand system. If you have decided to use a distributed library server and object server configuration, the odadmin user should be created on the object server machines as well. Make sure the odadmin user is a member of the database administration group.

If your configuration includes multiple Content Manager OnDemand instances, you should create an administration user for each instance.

Windows

Each library server must have a user account that will be used to install Content Manager OnDemand software, administer the system, load data, and perform other Content Manager OnDemand functions. If you use DB2 to manage the database, the user name must meet the DB2 naming rules. The suggested user name is ODADMIN, and this user must be a member of the *local* Administrators group and have the following *local* security policy settings:

- Act as part of the operating system
- Create a token object
- Increase quotas
- Log on as a service
- Replace a process level token

After you install and configure the system, remember to add the ODADMIN user to Content Manager OnDemand. Set the User Type to System Administrator. If you change the password in Windows, remember to change it in Content Manager OnDemand and vice versa.

2.3.3 Installation and configuration summary

You need to install the database manager first, optionally TSM, and then the Content Manager OnDemand software.

Install the database manager

You must install DB2, Oracle, or Microsoft SQL Server (Windows only) on the Content Manager OnDemand library server. Use the user account you created in the previous step to install the database software.

Install TSM (optional)

Tivoli Storage Manager can be used with Content Manager OnDemand object servers to store report data on devices that are supported by Tivoli Storage Manager. Devices supported by Tivoli Storage Manager include optical libraries, tape media, and the newer CAS based WORM Disk arrays. (DR550, NetApp Snaplock, or EMC Centera are examples of this type of device.) The use of Tivoli Storage Manager is *optional* and is needed only if you want to provide long-term storage for your reports on devices other than the fixed disks attached to the object server or network. You can also use Tivoli Storage Manager facilities to maintain DB2 archived log files and backup image files. You will need the use the *IBM Tivoli Storage Manager: Quick Start* publications to install and configure Tivoli Storage Manager. HTML and PDF versions of Tivoli Storage Manager publications, including the *Quick Start*, are available at:

http://publib.boulder.ibm.com/tividd/td/tdprodlist.html

Note that TSM is not necessarily installed on the Content Manager OnDemand object server. Content Manager OnDemand uses the Tivoli Storage Manager API client to store data into the Tivoli Storage Manager server. A Tivoli Storage Manager server is managed and administered independently of Content Manager OnDemand and can exist on a separate server.

Install Content Manager OnDemand

You must install a copy of the Content Manager OnDemand software on each workstation or node that is part of the Content Manager OnDemand system (library server and object server).

AIX/UNIX only

By default, the installation is carried out in the GUI mode, therefore the X Window System support is required for the GUI install. Optionally, the install can be performed in the character based console mode. To install the Content Manager OnDemand for AIX server in the console mode, enter the following command from the directory that contains the installer:

```
./odaix -console
```

2.3.4 Create Content Manager OnDemand instance

A Content Manager OnDemand instance is a logical server environment made up of a database, a library server, and one or more object servers.

You can run multiple instances on the same workstation, with each instance configured differently:

- To have separate test and production environments
- To have databases using different code pages

Each instance has different security from other instances on the same workstation. You must define users and groups to each instance and set application group and folder permissions for users of each instance. Each instance has its own System Log. Each additional instance requires additional system resources, such as virtual storage and disk space, and more administration.

If you plan to run more than one instance on the same workstation:

- The ARS.INI file must contain one section for each instance. Each section identifies the ARS.CFG file, ARS.DBFS file, and ARS.CACHE file used by the instance.
- You must create a unique copy of the ARS.CFG file for each instance.
- You should maintain separate tablespace file systems and cache storage file systems for each instance. Create a ARS.DBFS file and ARS.CACHE file for each instance.

AIX/UNIX

The AIX/UNIX platforms use four different configuration files. An instance is defined in the ARS.INI file by naming the instance, identifying the name of the database used by the instance, and identifying the library server on which the database will be maintained. When you configure an object server, you identify its library server in the ARS.CFG file on the object server. An instance has its own tablespace file systems for the database and cache file systems. The tablespace file systems are defined in the ARS.DBFS file on the library server. The cache file systems are defined in the ARS.CACHE file on each object server. All of the servers that belong to an instance run in one and only one code page.

ARS.INI

The ARS.INI file contains information about Content Manager OnDemand instances. When you install the Content Manager OnDemand software, the ARS.INI file contains information about a default instance named archive. Most customers will use the default instance for their first or only instance of Content Manager OnDemand. Each instance in the ARS.INI file will point to the other three configuration files, which must be unique for each instance. The files can not be shared across instances.

ARS.CFG

The ARS.CFG file contains information about the instance, such as identifying the object servers that belong to the instance, the language settings for the instance, and information that is used by database, storage, and print manager programs. Some parameters in the ARS.CFG file are not used on object servers. For example, an object server does not use the license parameters, server print parameters, and database parameters.

ARS.DBFS

The ARS.DBFS file lists the file systems on the library server that can be used by the database manager to maintain index data in tablespaces. Each line in the ARS.DBFS file identifies the name of a file system that Content Manager OnDemand can use to store tablespaces and specifies the type of tablespaces created in the file system. All database file systems should be identical in size for a given instance.

ARS.CACHE

The ARS.CACHE file lists the file systems on the object server that can be used by Content Manager OnDemand for cache storage. If there are multiple file systems in the ARS.CACHE file, Content Manager OnDemand uses the file system with the greatest amount of free space to store the objects. All cache file systems should be identical in size for a given instance.

Windows

After installing software on the server, you need to configure Content Manager OnDemand to integrate the various software products and control information, building your specific Content Manager OnDemand operating environment. In general, the initial configuration of a Content Manager OnDemand system consists of:

1. Defining the server or servers
2. Defining an instance on each server
3. Specifying properties of the instance:
   - Server type and other options
   - NLS
   - Directories for Content Manager OnDemand programs to use
   - Database manager options
   - Storage manager options
4. Creating the instance
5. Installing services
6. Creating and initializing the database

After you complete the initial configuration of your system, you might need to perform advanced configuration, such as:

- Configuring services
- Configuring scheduled tasks
- Managing multiple servers from one workstation
- Defining multiple instances on one workstation

You configure servers by using the Content Manager OnDemand Configurator program.

The configurator provides online help to assist you with completing tasks. The online help contains information about the options, fields, and commands on the windows, dialog boxes, and property sheets that you see when using the configurator. To display online help, press F1 any time the configurator is active in Windows. Help is available for dialog box commands and options. The main help topic for each dialog box usually contains information about the purpose of the dialog box and the commands and options that appear on the dialog box.

If your Content Manager OnDemand system consists of more than one workstation, you must define each server and create an instance for each server. An instance owner is assigned during the process of creating the instance. By default, the user name that you use to log on to Windows is assigned as the instance owner. After an instance is created, *only* the creater-owner of the instance can update or delete the instance. When you want to create an instance, you should log on with the Content Manager OnDemand system administrator account.

After an instance is created, the following properties of the instance cannot be changed:

- Instance name
- Server type
- Language and code page
- Database instance name
- Instance owner
- Database engine
- Location of database
- Size of the database
- Location of log files
- Size of log files
- Number of log files
- First cache file system named

If you update an instance, you must stop and restart the Content Manager OnDemand library server and object server by using the configurator program or system services.

System logging facility

After successfully creating the database, initialize the system logging facility.

System migration facility

If you plan to migrate application group index data from the database to archive storage, you must initialize the system migration facility. Migration is the process by which Content Manager OnDemand moves seldom accessed indexes from the database to the storage archive.

Install the Administration Client and Windows Client

At this point, install both the Administration Client and Windows Client on a Windows machine. Unless otherwise required, install only the English language clients.

2.3.5 Configure storage management

Configuring storage management consists of creating storage sets, configuring the System Log application group, configuring the system migration application group, and, optionally, configuring TSM.

Creating storage sets

You must add storage sets to the system before you can create application groups or assign the system-defined application groups to a storage set. Depending on the storage management characteristics of the reports that you plan to store on the system, you might need to add more than one storage set.

A storage set is a named collection of storage nodes that support application groups with similar storage management characteristics. For example, a storage set can maintain data for several application groups that need to keep documents for the same length of time and store the data on the same type of media. Additional storage sets might be required if there is a requirement to segregate data for legal reasons.

A primary storage node is where Content Manager OnDemand manages reports and resources stored in an application group. A storage node identifies the object server on which Content Manager OnDemand stores the data.

If you use TSM to manage application group data in archive storage, then each storage node that writes data to TSM-managed storage must be registered as a client node in a TSM domain. The properties of the domain determine the devices that can hold the data and how long that TSM maintains the data.

A storage set can contain one or more primary and secondary storage nodes.

You do not need to install TSM software and configure it now in order to install and use it later. Set up new storage sets as though they were accessed by TSM (that is, not Cache Only) and assign the storage sets to the application groups. You will have to select **Cache Data for __ Days** when you assign the storage set to the application group, and then select the **Advanced** button. When the Advanced Storage Management window appears, you must select either **Next Cache Migration** or **After __ Days in Cache** in the Migrate Data from Cache pane. You will not be able to run the arsmaint program to migrate or expire the cache (successfully) until TSM is installed and configured. Because the documents are not being copied to TSM, you will need to back up the cache file systems to prevent loss of data. Note that it is still possible to lose data between backups, so you may want to mirror the file system on another drive in addition to backing it up.

Configure the System Log application group

You should assign the System Log application group to a storage set that specifies a client node in storage managed by Tivoli Storage Manager so that the system can maintain a permanent copy of the System Log data. You should also store System Log index data in tablespaces.

Configure the system migration application group

If you plan to migrate index data from the database to archive storage, then you must create a storage set that specifies a client node in storage managed by Tivoli Storage Manager. After you add the storage set to the system, you can assign the System Migration application group to the storage set. You should also store System Migration index data in tablespaces.

TSM configuration

When you initially install Tivoli Storage Manager, the installation procedure creates a default 17 MB database volume (db.dsm) and a default 9 MB recovery log volume (log.dsm). The database size is determined by the amount of data that you plan to store on the system. The recovery log might need to be increased depending on the current utilization. It is strongly recommended that you do not use these system created files and you create new files on a separate file system, as the default files are created in the product install directory. *IBM Content Manager OnDemand for z/OS and OS/390: Introduction and Planning Guide*, GC27–1438 provides formulas that you can use to estimate the database and recovery log sizes.

The *Content Manager OnDemand for Multiplatform Installation and Configuration Guide*, SG18-9232 contains detailed platform specific information for TSM configuration.

Storage library

When you add an optical or tape library to the system, you must define it to Tivoli Storage Manager. When you define a library to Tivoli Storage Manager, you define a device class for the library and define the library and the drives contained in the library. You also define a storage pool for the collection of storage volumes that belong to the library.

Policy domain

The Tivoli Storage Manager policy domain links data with media in a storage pool. A policy domain supports a single storage pool, which in turn supports a single library. You define a domain, a policy set, a management class, and a copygroup.

Client nodes

A client node links clients and their data with storage volumes and devices. Before Content Manager OnDemand can store data in Tivoli Storage Manager storage, you must register at least one client node. You must register at least one client node in each policy domain that will contain Content Manager OnDemand data.

Storage pool volumes

You need to perform some steps to prepare removable media for initial use by Tivoli Storage Manager. This section provides general information and examples showing how to label storage pool volumes and check them into an automated library.

System Storage Archive Manager (Optional feature)

IBM System Storage® Archive Manager is an optional feature of TSM. This software will not allow data on the storage managed by the IBM System Storage Archive Manager server to be deleted before its scheduled expiration date. Content management and archive applications can apply business policy management to control expiration of archived data at the appropriate time.

Backing up DB2 to TSM (optional)

You can use Tivoli Storage Manager to maintain DB2 archived log files and backup image files. This capability means that you do not have to manually maintain these files on disk. The tasks are optional, and are only recommended for customers who need to use Tivoli Storage Manager facilities to back up the Content Manager OnDemand database in DB2.

2.3.6 Verifying the installation

To verify your installation:

1. Reboot the Content Manager OnDemand library server and verify that the following services start up automatically:
   - arssockd processes
   - arssched process (Windows only)
   - arsload process (optional)
   - arsjesd process (optional)
2. If Content Manager OnDemand object servers exist on other machines, reboot those machines (after the library server) and verify that the object server establishes a connection with the library server.

3. Start the Content Manager OnDemand Windows Client and click **Update Servers**. Add the name of the Content Manager OnDemand library server, and fill in the appropriate Port number (0 equates to 1445, the default value). Click **Help** for more information. Click **Close** to return to the login window.
4. Select the server you just added and type `admin` (the built-in account) for the user ID, and leave the password field blank. You will be prompted to change the password; make sure you remember what the new password is. Click **Enter**.
5. Open and search the System Log folder. You should see three rows in the document list, one for the admin login, one for the user update with new password, and one row with concurrent license information.

2.3.7 Implementation best practices

There is a saying that there are two kinds of people in the world: those that have lost data and those that will lose data. Before the Content Manager OnDemand implementation begins operation in a production mode, the Content Manager OnDemand administrator should write a Disaster Recovery plan (or update an existing plan to include the Content Manager OnDemand system). This documentation should include, but is not limited to, the following information:

- How the system was installed:
  - All configuration settings
  - File system names, sizes, and permission settings
  - User IDs and passwords associated with the application
  - Where the original install media is stored
  - Who to call for support
- How the system is backed up:
  - Content Manager OnDemand documents
  - Content Manager OnDemand database (document metadata)
  - TSM database
  - Configuration files (Content Manager OnDemand, database and TSM)
- How the system is restored

In addition to the backup requirements, there are Content Manager OnDemand operations that must be performed periodically (refer to 2.9.1, “Automate system process” on page 70).

Finally, there are application design considerations that may be classed as best practices because they impact performance or storage requirements in some fashion. These include:

- The maximum rows value, which determines how many data rows will be loaded into each database table, is used for segmenting the index data and deciding when to close a database table and open a new one. The default value of 10,000,000 rows is recommended for balancing the performance of data loads and queries. However, if the expiration type is segment, use a maximum rows value such that the data will expire (the table will be deleted) in a reasonable amount of time. For example, if you load 1 million rows to an application group in one year, and the data is to be retained for seven years, the default value of 10 million rows in a data table is too large. The table will not fill up until the tenth year, and the table will not be deleted until all data in the table is seven years old, which means the table will exist for 17 years. When a wildcard search is specified and no index field is used (or segment date), a full table scan takes place. If the table contained 10 million rows, a wildcard search would be looking at nine million rows of “expired” data in the 17th year. A good rule of thumb is that no table should hold more than 10% of the data that would be loaded during the lifetime of that data when expire by segment is specified (and thus the lifetime of a table is 110% of that of the data). 10% of seven years (84 months) is 8.4 months, you can round up to nine months. Nine months of data would be 750,000 rows, so maximum rows per table would be 750,000.

- The date field that is used for the segment date should probably always have a type of filter. By default, a Content Manager OnDemand system table (arsseg) stores the first and last segment date for each application group data table in the system. When a search is invoked, one of the first actions is to scan the arsseg table and select all application group data table names that satisfy the application group ID (recall that folders can search more than one application group) and that the date (or date range), if specified, is between the first and last dates for the tables that meet the first criteria. Assuming that the maximum rows value was specified correctly, a very small number (one or two) of tables should have to be searched. The segment date is probably not going to be the primary search criteria, and whatever is the primary search criteria (for example, account number, social security number, or ID number) should be of type index. Specifying type index for the segment date would therefore create unnecessary overhead, and not contribute to finding the record any faster.
- We recommend that you set the Number of Database Servers parameter to support the peak number of concurrent database connections that you expect the library server to handle. A low value limits access to the database during periods of high database activity. A high value requires more system resources during periods of low database activity. The value that you choose also depends on the characteristics of the queries. For example, general queries typically keep a connection open longer than a more specific query.

- Advanced Function Presentation (AFP) data is a multi-part data type. This means that in addition to the variable data itself, there are also external resources, such as images, fonts, and logos, which are referenced by the AFP data stream. When Content Manager OnDemand stores AFP, user specified resources are also archived. When the data is viewed, the referenced resources are displayed. It is a common misconception that if fonts are collected when the data is loaded, that they are available for viewing in the Windows Client. The fact is Windows does not recognize AFP fonts. It is not possible to use these fonts even if they are sent to the client as part of the resource. The Window Client requires a mapping from AFP fonts to Adobe Type Manager or True Type fonts. Content Manager OnDemand provides these mappings for most standard fonts and the mapping files can be modified for custom fonts. In most cases, you should never specify FONT in the RESTYPE directive (an indexer parameter).

2.4 Design

Creating a design for your Content Manager OnDemand implementation requires you to take many options into consideration. These items include:

- Report selection
- Indexing requirements
- Retention
- Application group, application, and folder
- Security
- Indexers
- Exit points

2.4.1 Report selection

Planning for Content Manager OnDemand requires that you consider what reports your system will be receiving. This planning includes what documents or reports will be accessed by users and their frequency, and which reports need to be stored for just archive and compliance reasons.

Some of the basic considerations you need to analyze are:

- What types of data streams will the system need to ingest? Does Content Manager OnDemand support native indexing of these streams? If not, a preprocessor program, transform, or input exit is required. Possible data stream types include:
  - AFP
  - PDF
  - Line (ascii or ebcdic)
  - Image (.gif, .jpg, .png, .tif)
  - PC Documents (.doc, .xls)
  - Metacode or PCL (a transform will be required)
- What is the volume of data you will ingest into Content Manager OnDemand on a daily, monthly, and yearly basis?
- How large are your reports? Will you use report level indexing or will you burst apart the reports into smaller groupings? Is there a need for large object support?
- How long do these reports need to be kept for? Is there any legal archive needs for this data?
- How many users who will be accessing these reports?
- What is the visibility/importance of the report to the business?

2.4.2 Indexing requirements

Content Manager OnDemand allows you to index reports. When Content Manager OnDemand indexes a report, it extracts the index values from the body of the report and inserts these values into the Content Manager OnDemand database. Each database field that is defined in your application can then be used by a user to search and locate a particular document. These reports can consist of large transaction style reports or customer statements. Different indexing options supported by Content Manager OnDemand are:

- Document Level Indexing: This indexing is used for reports made up of multiple logical documents within a single data stream. An example of this would be documents such as statements, bills, or invoices.
- Report Level Indexing: This type of indexing is used for line data style reports that are one logical item. Indexing information can be extracted from the header or from the body of the report, such as report name and date.
- Transaction Level Indexing: This type of indexing is used for line data reports that contain sorted transaction data. It allows Content Manager OnDemand to support searching for individual values without indexing each line of a report.

It does this by taking the first and last value of a given column for a given number of pages.

2.4.3 Retention

Content Manager OnDemand retention includes three separate but related retention settings: Life of Data and Indexes, Number of days to cache data, and Archive retention. When designing your Content Manager OnDemand system, each of these must be calculated in order to allocate enough disk or Archive storage:

- Life of Data and Indexes: This value is used by Content Manager OnDemand to determine how long index data will be kept in the Content Manager OnDemand Database and how long before the arsmaint process will issue a delete for database and archive/cache storage.
- Number of days to cache data: This value is used by Content Manager OnDemand to keep a local disk copy of archived data. It is useful to support high volume retrievals of data during initial loading. It can also be used for Cache Only application groups.
- Archive retention: This value is set within TSM and manages life of data within that subsystem. Content Manager OnDemand has no control over this setting. In the case of System Storage Archive Manager (SSAM), Content Manager OnDemand can effect this function by initiating events based on the Life of Data and Indexes value. For more information regarding SSAM, refer to *IBM System Storage DR550 Setup and Implementation*, SG24-7091.

In the Application Group General tab, there is an Advanced... button that brings up a Database Information page. You will see a Records Management pane in the lower left, and if you specify the **Application uses Record Management** option, Content Manager OnDemand will not manage retention for the application group. This option was added for the Federated Records Management product feature, and should not be used otherwise. You can theoretically implement an external records management application and use this option, but records management can be very complex, and the file and database operations would have to be handled outside of Content Manager OnDemand (such as DB2 and TSM).

2.4.4 Application group, application, and folder

A Content Manager OnDemand application group is a grouping of applications that use the same indexing structure. These applications also must have the same storage and retention settings. An application group is the mechanism by which Content Manager OnDemand maintains the indexes for the documents that you have loaded into your system. In addition, the application group contains additional information, such as storage management settings and permissions of the users and groups that are allowed access to the data loaded into this application group. In creating an application group in Content Manager OnDemand, you must specify the fields that will hold the index data you extract from each document. You specify to Content Manager OnDemand whether a field will be an index or a filter and what kind of field it will be. Examples of this are String, Decimal, or Date. You also describe characteristics of the field, such as number of bytes and case. When you define an application group to Content Manager OnDemand, a separate table will be created in the database containing a column for each field you defined.

A Content Manager OnDemand application defines the layout of the report you are loading into the system. It defines characteristics of the report such as data format, carriage control, record layout, and compression settings. It is within the application definition where you will define the extraction rules that the Content Manager OnDemand Indexing program will use to extract information from the report to be loaded into the database for searching. It is also within the application where you specify the parameters needed so Content Manager OnDemand can properly deliver a report for viewing.

A Content Manager OnDemand folder defines the search and display fields that appear when a user opens the folder. These folder fields are mapped to the application group fields that contain the index values extracted from the reports. A user will use this folder to search for documents by entering values into the search fields. Content Manager OnDemand will then issue a query against one or more application group tables to resolve the search and produce a hit list.

2.4.5 Security

Security in Content Manager OnDemand implementations can be divided into two areas: authentication (determining if the user is who he claims to be) and authorization (determining the permissions that user has been granted). User authentication can be internal to Content Manager OnDemand (user ID and password), or external to Content Manager OnDemand (LDAP, for example). If external security is utilized, you must still create the user ID within Content Manager OnDemand for authorization purposes. Even if a user can gain access to Content Manager OnDemand (he can be authenticated), he may not have any permissions to do anything (no authorization). Permissions are granted on application groups and folders separately.

Application group permissions specify authorizations for documents and annotations (also separate). They include:

- View
- Add
- Delete
- Update
- Fax
- Print
- Copy

Defaults for access are view, fax, print, and copy on documents and view and add on annotations. Any of the permissions can be added or deleted if desired. Default administrative permissions include the above and add for documents and copy for annotations. Fax and print permissions cannot be specified for annotations, although if a user can view an annotation, it can be printed (annotations can be public or private).

Common folder permissions include access (the user can view the folder) and administrative (the user can view or change the folder), named queries (public and private), and maximum hits. There are other less frequently used permissions that are explained in the online help and administration guide.

We generally recommend assigning permissions to groups of users, rather than on a user by user basis. User groups usually correspond to roles within the organization, for example, accounting, sales, customer service, and human resources. Within a functional organization, you may have other roles: managers, users, reviewers, and legal. An individual may belong to one or more groups, and it is important to understand how group permissions are evaluated. Remember the following rules:

- By default, the person that created the folder, a system administrator, and an application group and folder administrator can access the folder.
- You can use the *PUBLIC name to specify default permissions for all other users. Sensitive applications should never permit *PUBLIC permission.
- You can specify permissions for specific groups and users:
  - All of the users that belong to a group that you add to a folder will obtain the permissions that you specify for the group.
  - A user that belongs to two (or more) groups that have been added to the same folder will obtain the permissions of the group that has the lowest Group ID (GID).

- The permissions that you specify for a user override all other permissions, including any default permissions (*PUBLIC) and any groups to which the user belongs and that are added to the folder.

The only rule that tends to cause problems is the one concerning users who belong to two or more groups. Problems can arise because the lowest GID criteria for evaluation cannot be overridden, and groups are assigned a GID in ascending order by default (1080001, 1080002, 1080003, and so on). The solution to this potential problem is simple if you take precautions when you create groups: the GID for a new group can be overridden when you create the group. It is strongly advised that your first group have a GID of 1080101, the second have a GID of 1080201, and so on. This allows you to create a new group with a lower GID when necessary. For example, managerial roles generally will have more permissions than user roles, which have greater permissions than reviewers. Unless you have a well planned schema for creating groups, you may find yourself granting authorizations to users on a user ID basis, rather than a group basis.

2.4.6 Indexers

Content Manager OnDemand for Multiplatforms includes two different indexing programs out-of-the-box: the AFP Conversion and Indexing Program (ACIF indexer), and the PDF indexer. A third indexer, the generic indexer, is not a true indexing program, as it only reads index values from a generic index file, and not from the input data itself. You can also purchase a license and use the optional Xenos transformation program. The Xenos transforms can be used to extract index data from input print files that contain AFP, Xerox Metacode/DJDE, or PCL data. Xenos transforms will also convert the above formats into PDF documents.

The ACIF indexer (arsacif program) lets you index a line data print file, optionally convert line data input into Advanced Function Presentation (AFP) documents to be stored on the system, and retrieve the AFP resources that are required for archiving and viewing in Content Manager OnDemand. ACIF can also be used to process input reports that contain AFP data. The ACIF indexer requires two files: the input (document) file and an index parameter file, which can be created manually or with a graphical indexer.

The PDF indexer (arspdoci program) processes PDF input files. A PDF file is a distilled version of a PostScript file, adding structure and efficiency. A PDF file can be created by Acrobat Distiller or a special printer driver program called a PDFWriter. You can automate the distilling process by configuring and running the Distiller daemon (UNIX servers) or Acrobat Distiller (Windows servers). The PDF indexer also requires two files: the PDF document file and a file with the indexing parameters. There are two ways to create the index parameter file:

- Content Manager OnDemand provides the **`arspdump`** command to help you determine the location of trigger and field string values in the input data. The **`arspdump`** command processes one or more pages of sample report data and generates an output file. The output file contains one record for each text string on a page. Each record contains the x,y coordinates for a box imposed over the text string (upper left, lower right). Based on the information in the output file, you can create the index parameter file with the appropriate field, index, and trigger statements.
- The second method for creating the PDF index parameter file is to use the Graphical User Interface. However, this method requires you to install Adobe Acrobat on the workstation with the Administration Client. With this method, you can define indexing information in a visual environment. You begin by opening a sample input file with the GUI. You can run the graphical indexer from the report wizard or by choosing the sample data option from the Indexing Information page of the application. After you open an input file in the GUI, you define triggers, fields, and indexes, which are saved in an index parameter file.

In addition to the indexers discussed above, you can also index documents on another system and then archive the documents on Content Manager OnDemand for Multiplatforms. For example, you can use the OS/390 indexer (AFP or Line data only) on OS/390 or z/OS platforms or the OS/400® indexer (AFP, Line, or SCS formats only) on the System i platform. These indexers are not shipped on the Content Manager OnDemand for Multiplatforms media, but are available without charge to Content Manager OnDemand licensees.

2.4.7 Exit points

Content Manager OnDemand for Multiplatform offers several different types of exit points that can be used alone or in conjunction with others. Some are Content Manager OnDemand application specific (although they may be used by more than one application), while others are more general in function (for example, security, logging, faxing, or storage). Exit points include:

- ACIF indexer user exits
- Server user exits

- Postprocessor command
- Download user exit
- System Log user exit

ACIF indexer user exits

A user exit is a point during ACIF processing that enables you to run a user-written program and return control of processing to ACIF after your user-written program ends. ACIF provides data at each exit that can serve as input to the user-written program. Note that user exits can only be used with the ACIF indexer, and no equivalent functionality exists for the PDF or generic indexers. These user exits must be written in the C or C++ programming language and compiled in order to be used.

There are four types of user exits:

- Input
- Index
- Output
- Resource

The *input exit* enables you to add, delete, or modify records in the input file. You can also use the exit to insert indexing information. This exit is called after each record is read from the input file. This exit can request that the record be discarded, processed, or processed and control returned to the exit for the next input record. The largest record that can be processed is 32756 bytes.

The *index exit* allows you to modify or ignore the records that ACIF writes in the index object file. This exit receives control before a record is written to the index object file and cannot be used to insert additional index information.

The *output exit* allows you to modify or ignore the records ACIF writes into the output document file. This exit receives control before a record is written to the output document file. Like the input exit, the largest record that the output exit can process is 32752 bytes. This exit is not called when ACIF is processing resources.

The *resource exit* allows you to filter resources from being included in the resource file. If you want to exclude a specific type of resource (for example, an overlay), you can control this with the RESTYPE parameter. This exit is useful in controlling resources at the file name level. This exit receives control before a resource is read from a directory. This exit program can request that the resource be processed or ignored (skipped), but it cannot substitute another resource name in place of the requested one.

Server user exits

There are nine types of server user exits:

- Client retrieval preview user exit
- Fax options exit
- Permission exit
- Print exit
- Report specification archive definition exit
- Security user exit
- Storage management external cache exit
- System Log user exit
- Tablespace creation exit

Client retrieval preview user exit

The *client retrieval preview user exit* allows for the modification of document data prior to the data being displayed for the client. The retrieval preview user exit allows you to run a user-written program to process documents that belong to a specified application. The user-written program is activated by selecting the **Use Preview Exit** option on the Miscellaneous Options page of an application. When the option is selected, the user-written program will be called any time that a request is made to retrieve a document. Any information that is specified in the Parameters field will be passed to the user-written program.

The retrieval preview exit can be used to add, remove, or reformat data before the document is presented to the client. For example:

- Remove pages from the document, such as banner pages, title pages, or all pages but the summary page.
- Remove specific words, columns of data, or other information from the document. That is, omit (white out) sensitive information such as salaries, social security numbers, and birth dates.
- Add information to the document, for example, a summary page, data analysis information, and Confidential or Copy statements (watermarks).
- Reformat data contained in the document, for example, reorder the columns of data.

The retrieval preview user exit is not called for all document retrievals. In particular, the user exit is not called for functions that use the Bulk Retrieval method of retrieving documents or for server printing. For example, running the **`arsdoc get`** function without specifying the `-n` parameter performs a bulk retrieval, and documents retrieved will not be passed to the client preview exit.

The retrieval user exit point may be enabled for more than one application. However, all applications must be processed by the same user-written program (only one user-written program is supported). The system passes the name of the application that is associated with the document to the user-written program. The user-written program can perform processing based on the application or it can perform the same processing for all documents regardless of the application. The retrieval preview exit must be written in the C language, compiled, and placed in the <OnDemand_install>/bin/exits subdirectory in order to be used. This user exit must be named arsuprep.

When modifying the data, the format and type of the data *must not be changed*; only the content may be changed. When the modified data is viewed by the Windows Client, the format of the data and the data type that is defined in the application on the View Information page will be used to display the data. If the format or data type has changed, the document will not view properly.

Fax options exit

The *fax options exit* is for specialized applications and is normally not used. When a user chooses to fax a document, the fax options exit can help to pre-fill the fax options accordingly. Depending on the code, information can be pre-filled according to the document being opened. Faxing a document is a server print function (not local print). This user exit must be written in the C language and compiled, and it must be named arsufax. The fax options exit generally requires modifications to the arsprt user exit also. To enable the fax options exit, place the compiled exit program arsufax in the bin/exits directory.

Permission exit

The *permission exit* is more complex than most user exits. It is used to customize permission in a more flexible way than the standard Content Manager OnDemand Administration Client can provide. This exit is called during login if the permission exit is turned on for folder and application groups. It is also called during a search when the permission exit is turned on for an SQL query string or document.

The input to the exit program is the user ID and the information from the structure field ArcCSXitPermExit. The output is the access values of the different actions. The access values of the first two actions determine whether the user has the right to access the folder and application group during logon. This exit program can also change the SQL query and the SQL query restriction for the application group in action 4. Finally, the access value of action 3 determines the permission to retrieve the document into the hit list.

The output of the program is in a structure of ArcCSXitPermExit, with the final access values and SQL queries. The permission exit overrides the permission defined on the Content Manager OnDemand Administration Client. It can be used for such occasions as when you want specific users or groups to view certain financial reports only during a certain time of the year, but you do not want to change the permission from the Administration Client.

The permission exit can be activated by specifying the respective variables in the ARS.INI file with the arsuperm exit program residing in the bin/exits directory of the Content Manager OnDemand installation root. The ARS.INI file is found in the config directory of the Content Manager OnDemand installation root. For AIX, the ARS.INI file to be modified is in the /usr/lpp/ars/config/ars.ini directory. You set the following variables to activate the different permissions in the ARS.INI file:

- To activate the folder or the application permission, set:

  ```
  SRVR_FLAGS_FOLDER_APPLGRP_EXIT=1
  ```

- To activate the SQL query exit, set:

  ```
  SRVR_FLAGS_SQL_QUERY_EXIT=1
  ```

- To activate the document permission exit, set:

  ```
  SRVR_FLAGS_DOCUMENT_EXIT=1
  ```

Print exit

There are two ways to print a document stored in Content Manager OnDemand: local printing, through a LAN attached PC printer, or server printing, through a printer managed by the print manager installed on the Content Manager OnDemand server machine. The *print exit* for Content Manager OnDemand for Multiplatforms can only be used for documents that are printed through a server printer.

The print exit for Content Manager OnDemand for Multiplatforms is the arsprt file, which resides in the <OnDemand_install>/bin directory. It is a batch file (arsprt.bat) on the Windows platform, and a shell script on the UNIX platform (arsprt). The print exit can be modified for many different uses, for example, to keep track of department printing expenses (based on the user ID or application group name), or even to keep a running count of the number of times a particular document is printed (using **`arsdoc update`** to change index values).

Report specifications archive definition exit

The *report specifications archive definition exit* allows an installation to modify some of the parameters used by Content Manager OnDemand when document data is being captured (loaded) by the arsload program. The following parameters can be modified:

- The application group name
- The application name
- The name of the object server to be used for data storage
- The name of the storage node to be used for data storage
- The indexer parameters set
- The input file control character type, logical record length, and record format

The user exit is named arsuupdt and must be written in the C programming language. It is provided in both source and executable forms, with the source being provided to help users to understand how the exit is used. If, however, you choose to modify arsuupdt, it must be compiled with a supported C/C++ compiler. The arsuupdt program must reside in the <OnDemand_install>/bin/exits directory. To call this exit, you must specify the -E parameter when you run the arsload program.

Security user exit

Content Manager OnDemand provides a user exit that allows you to implement your own user exit program to identify and authenticate users that log on to the system. You can use the *security user exit* to authenticate a user's password by some means other than the way that is built in to Content Manager OnDemand. For example, you may want to deny access to the system after three incorrect logon attempts are made by a user; you may want to enforce some sort of password uniqueness. You can also use the security user exit to allow users that are not already in the Content Manager OnDemand user database to access the system. The user security exit is enabled by changing a parameter in the ARS.INI file:

```
SRVR_FLAGS_SECURITY_EXIT=1 (0 to disable)
```

Once enabled, the user exit program processes *all* logons to the system. The user exit must be named arsusec and must reside in the bin subdirectory of the Content Manager OnDemand installation. Note that all Content Manager OnDemand instances will use the same security exit if enabled.

Storage management external cache exit

The *storage management external cache exit* is used to retrieve data from external storage. Depending on your programs, the external cache can be just a file from a directory or you can interface with other software to retrieve documents from other applications. This exit is for specialized applications and is normally not used.

To use this exit, you must select **External Cache** when the application group is created. When the user retrieves the document from Content Manager OnDemand based on the indexes, the exit is activated to pull the document from respective location. This exit is only activated when a user retrieves a document data that is stored in external cache. The `smextcac exit` program should be placed in the <OnDemand_install>/bin/exits directory.

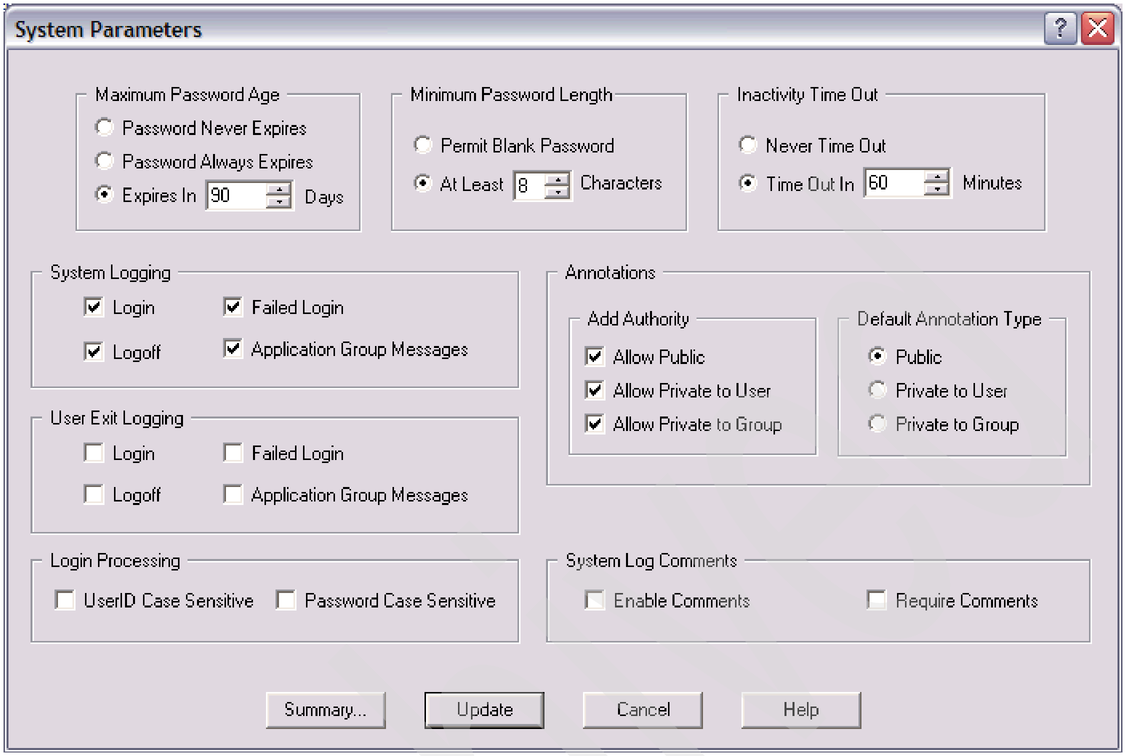
System Log user exit

When you enable logging for system, user, and application group events, Content Manager OnDemand sends a copy of each message that is generated by the system to the System Log user exit program. The *System Log user exit* program that is provided by IBM does not perform any functions. However, you can replace the program that is provided by IBM with a user-written program that does user-defined processing. The user-written program can process the messages in any way that you want. For example, you could provide a user-written program to check for certain message numbers or severity, and take whatever action you deem appropriate. The standard System Log user exit program is named arslog. The System Log user exit program on UNIX servers is named arslog (a UNIX shell script); the System Log user exit program on Windows servers is named ARSLOG.BAT (a batch file). Note that the name of the System Log user exit cannot be changed. The program must reside in the <OnDemand_install>/bin directory. Typical uses for the System Log user exit include:

- Notifying the Content Manager OnDemand administrator or LOB manager of failed loads
- Tracking failed login attempts
- Tracking successful loads for billing purposes
- Monitoring the number of users logged on
- Tracking retrievals

You must perform the following steps to configure the system to send the messages to the System Log user exit:

- Enable Content Manager OnDemand to generate system messages and specify the types of messages generated by selecting the appropriate options (System Logging pane) in the System Parameters dialog box. The System Parameters dialog box (see Figure 2-3 on page 47) is reached by right-clicking the Content Manager OnDemand server name in the System Administration Client and selecting **System Parameters**.



System Parameters

Maximum Password Age
○ Password Never Expires
○ Password Always Expires
● Expires In 90 Days

Minimum Password Length
○ Permit Blank Password
● At Least 8 Characters

Inactivity Time Out
○ Never Time Out
● Time Out In 60 Minutes

System Logging
☑ Login ☑ Failed Login
☑ Logoff ☑ Application Group Messages

User Exit Logging
□ Login □ Failed Login
□ Logoff □ Application Group Messages

Login Processing
□ UserID Case Sensitive □ Password Case Sensitive

Annotations

Add Authority
☑ Allow Public
☑ Allow Private to User
☑ Allow Private to Group

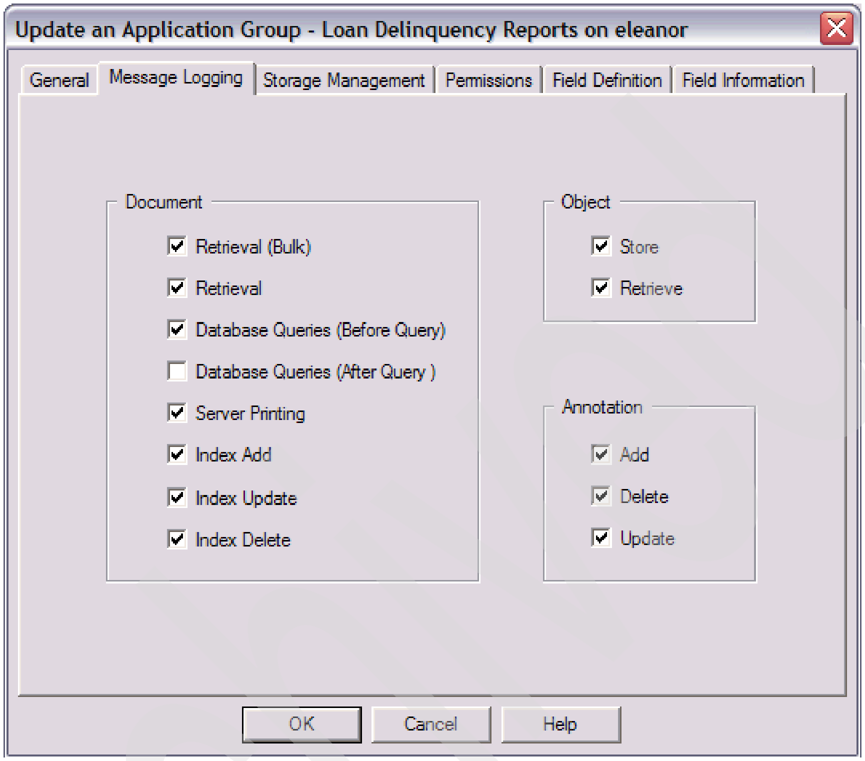
Default Annotation Type
● Public
○ Private to User
○ Private to Group

System Log Comments
□ Enable Comments □ Require Comments

Figure 2-3 System Parameters dialog box

- Enable Content Manager OnDemand to generate application group messages by selecting the appropriate option (User Exit Logging pane) in the System Parameters dialog box.

- Specifying the types of application group messages generated by selecting options in the Message Logging page in application groups (see Figure 2-4).



Document

☑ Retrieval (Bulk)
☑ Retrieval
☑ Database Queries (Before Query)
□ Database Queries (After Query)
☑ Server Printing
☑ Index Add
☑ Index Update
☑ Index Delete

Object

☑ Store
☑ Retrieve

Annotation

☑ Add
☑ Delete
☑ Update

Figure 2-4 Application Group Message Logging dialog box

After you have completed these steps, Content Manager OnDemand automatically saves the messages in the System Log and sends the messages to the System Log user exit.

Tablespace creation exit

The Content Manager OnDemand *tablespace creation exit* allows an installation to take action when Content Manager OnDemand creates a tablespace, table, or index tables that will be used to store application index data. This exit is not called for the Content Manager OnDemand system tables. This exit is enabled by setting a parameter in the ARS.CFG file:

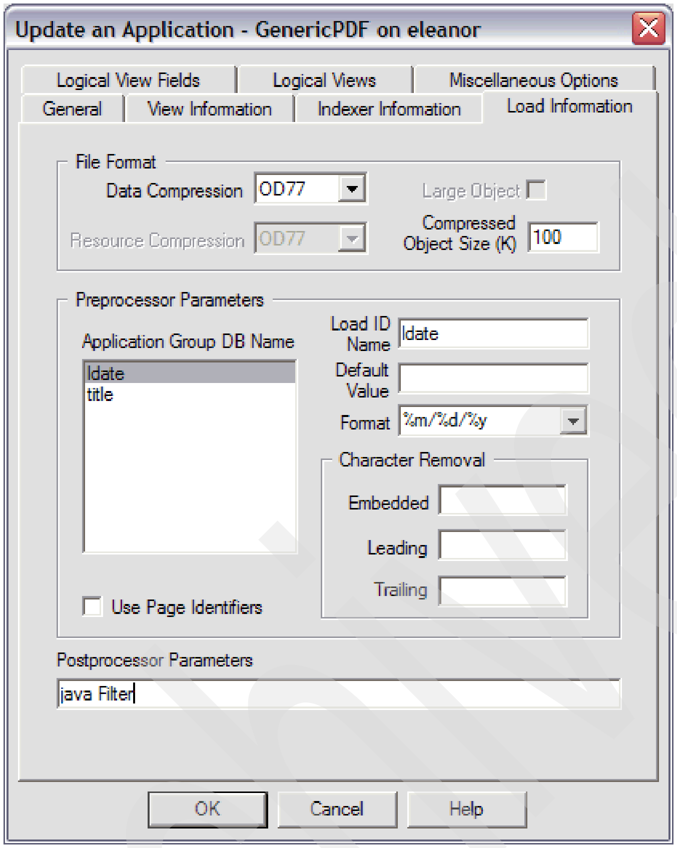
```
ARS_DB_TABLESPACE_USEREXIT=absoulte path name
```

You have the option of specifying the SQL to create the table or index on that table within the tablespace.

Postprocessor command

When you define an application, one of the tabs is named Load Information. There is a text input box on this tab that allows you to specify Postprocessor Parameters. This field is where you can specify an operating system command or user-defined program that Content Manager OnDemand runs against the index file before loading the index records into the database. For example, on a UNIX system, you could specify the string “`/bin/sort | uniq`” in order to remove duplicate index rows before loading.

Another example would be to write a Java program to process a date field that may or may not contain slashes (date formats must be consistent; all date fields must contain slashes or not contain slashes). Note that this particular function can be performed by an index user exit. However, there may be other circumstances that preclude that possibility. One reason would be that the ACIF indexer is not used for that particular application (the PDF indexer and generic indexer do not allow for user exits). A second reason is that ACIF indexer user exits must be written in the C or C++ programming language and compiled. If a programmer or a compiler program is not available, an index user exit is not an option (see Figure 2-5 on page 50).



File Format

Data Compression: OD77

Large Object □

Resource Compression: OD77

Compressed Object Size (K): 100

Preprocessor Parameters

Application Group DB Name:
ldate
title

Load ID Name: ldate

Default Value:

Format: %m/%d/%y

Character Removal

Embedded:

Leading:

Trailing:

□ Use Page Identifiers

Postprocessor Parameters: java Filter

Figure 2-5 Postprocessor command invoking a Java program

Download user exit

The *download user exit* is called by the arsjesd program, which receives the data sets into file systems on the server. If you start the arsjesd program with the -x parameter, the arsjesd program invokes the specified user-written program, which can be any user-written program. For example, you can provide a user-written program that parses the additional job information transmitted by the corresponding program on the mainframe. By using the download user exit program, you can configure the system so that each file that is transmitted to the server is automatically processed and loaded into the correct application group and application.

2.5 Application setup and verification

In this section, we discuss the following application setup and verification tasks:

- Report setup
- Permissions setup
- Load, retrieve, and print verification
- User acceptance

2.5.1 Report setup

A Content Manager OnDemand report is composed of four items:

- Application group
- Application
- Folder
- Storage set

You can only assign a storage set to an application group once (and you cannot update the assignment), so some thought must go into creating storage sets. A storage set is a named collection of storage nodes that support application groups with similar storage management characteristics (retention and type of media used). Note that the data retention values are application group specific, and do not appear in the storage set definition. All implementations should probably include cache storage, so this storage set should be created first. For Content Manager OnDemand for Multiplatform, the only other *type* of storage set will be Tivoli Storage Manager (access method TSM). While Content Manager OnDemand for Multiplatform calls out TSM as a prerequisite, you can set up storage sets that will use TSM in the future without installing TSM initially. However, all data will have to be stored in cache until TSM is installed and configured, and the data must be protected in the meantime (disk mirroring or backups).

> **Tip:** Name the storage set so that the media and retention is obvious, for example, 7YR_TAPE or FOREVER_OPTICAL. This will help prevent assigning storage sets to application groups that have different retention requirements. You would not want to store data that is only kept for two years on optical platters that cannot be erased and reused.

You can think of an application group as the definition of the metadata to be kept for a report or group of reports (hence the name). The application group also specifies the data retention, so all reports that belong to an application group *must* have identical retention requirements. An application group does not specify a data format, so you may have several different report data types that belong to the same application group.

An application can be thought of as how the report is loaded into Content Manager OnDemand. Each application definition specifies the report data format, the indexer to be used, the indexer parameters for the ACIF and PDF indexers, and logical view information for all data types other than PDF and user defined (Excel or Lotus® spreadsheets, Word documents, Autocad, and any format that invokes an external application/viewer). Any group of reports that will use the same metadata (database fields) and have the same retention can belong to the same application group. However, in order to assign multiple applications to an application group, you must define an Application ID field when you create the application group.

> **Tip:** You should add an Application ID field to all application groups. This is a string field and is typically only one or two characters in length, so the overhead is minimal. This will allow you to add applications to existing application groups in the future, rather than creating new application groups. The benefit to this is that the metadata for similar reports is stored in the same database table, and search performance will be faster (fewer tables to search).

A folder represents a Content Manager OnDemand user's view of the archived data. A folder can be assigned to multiple application groups even though all search fields that appear in a folder are not mapped to all application groups. For example, consider a folder for a call center application. The folder search fields might include a date field, a customer ID field, the customer name field, customer address information (street, city, state, zip) and a application ID field (a drop-down menu). The folder might be assigned to two application groups: customer billing and correspondence. The correspondence application group archives letters that are received from customers and scanned. Correspondence may not include the customer ID number, so it must be searched by name or address.

There are three commonly used methods of setting up a report in Content Manager OnDemand:

- Report wizard
- Manually
- Copying an existing component

Report Wizard

The Report Wizard assists you in adding reports to the system. The Report Wizard is started by clicking the **Report Wizard** icon (document with a wand and stars) on the System Administration Client toolbar. The Report Wizard helps you add a report to the system by asking questions, which allows you to progress in an organized manner toward completing an application group, application, and folder. When you start the report wizard, you will answer questions that appear in the window. You can move forward or backward through the windows to change answers. There is also a finish button that will move you to the final window (Report Wizard will make all the remaining decisions for you).

You can use the Report Wizard to add an application group, application, and folder for a report. These actions include defining indexing information, defining database and folder fields, configuring data and storage management, specifying whether the application group can contain more than one application, and naming the application group, application, and folder.

You can also use the Report Wizard to add an application to an existing application group. This action includes defining indexing information, specifying storage information, and identifying the application within the application group. To add an application to an application group, the application group must have a database field to hold the values that uniquely identify an application within the application group.

> **Important**: The Report Wizard will only process files that contain line data or PDF data. PDF data requires that you have Adobe Acrobat installed on your workstation. You must also have a sample file available in order to proceed.

Manually

You can always right-click the component icon in the left (navigation) pane of the System Administration Client and choose the **Add New Application Group/Application/Folder** option. You should start with the General tab and then move to the right and up (applications have two rows of tabs). Not all tabs may be applicable for a given report, and some choices can be deferred or changed later. There is a Help button on every tab, and the user is advised to click that button and read the online documentation whenever in doubt as to what a particular option is. You always begin with a New Application Group (unless adding an application to an existing application group), then create a New Application, and then create a New Folder (or modify an existing folder). You can always modify Permissions, so that tab can always be skipped when creating a new Application Group or Folder (applications do not have a Permissions tab).

Copying an existing component

In many cases, you will find the easiest method to add a new report is to copy (right-click an existing application group, application, or folder) a similar component and make changes manually. A window with the properties for the component you copied will appear (the same window that appears when you choose **Add New...**), and you need only change the properties that are different (Name at a minimum).

2.5.2 Permissions setup

Permissions are the means by which Content Manager OnDemand determines who can open folders and search for documents stored in application groups. Content Manager OnDemand also uses permissions to determine who can maintain folders, application groups, and other objects with the administrative client. By default, only the person that adds the folder, an application group/folder administrator, or a system administrator can open and maintain the folder. By default, only the person that adds the application group, an application group/folder administrator, or a system administrator can access data stored in the application group or maintain the application group. Note that any other user will require Document View (Access) permission to the application group and Access Authority to the folder in order to search for and retrieve documents stored in the application group. Also note that while *PUBLIC always appears in the Defined (application group permissions) and Selected (folder permissions) panes on the Permissions tab, *PUBLIC does not have any permissions by default (click ***PUBLIC** to view the permissions).

You can grant permissions as you set up the application groups or go back later and authorize users. Users are authorized to application groups and folders as individual users, as members of user groups, or as part of *PUBLIC. Do not grant permissions to a folder until you are ready for the users to access that folder, which should be after you have tested retrieving documents from that folder. When users log on to Content Manager OnDemand, they see a list of folders they are allowed to access.

Tip: Assign permissions on a user group basis. Think of a user group as a role, and assign all users who perform that role to the user group. When you set up user groups, do not use the default Group ID (GID). The default GID begins at 1080001, but permissions are based on the lowest GID if a user belongs to more than one group. By specifying a higher number for the GID, and creating GIDS as a multiple of 100, you can always create a new group with a lower GID if necessary.

If you have several different applications within an application group, they will all be included in the folder for the application group by default. If you want to restrict access to certain applications, highlight the application group in the Selected pane, click the **Applications...** button on the folder's General tab, and remove applications as necessary.

2.5.3 Load, retrieve, and print verification

Once you set up a report definition (application group, application, and folder), the next step is to load and retrieve a sample report. The first time you load a report, you should probably use the **arsload** command (in a Content Manager OnDemand command window on the Windows platform, or from a command prompt on the UNIX platform):

```
arsload -g <app_grp> -inv -u <user_ID> -p <passwd> <file_name>
```

If the report belongs to an application group with multiple applications, you will also need to specify the application (-a <app>). The -inv options are as follows:

- -i: Indexing only (report is not loaded).
- -n: Do not delete report (input) file.
- -v: Verbose mode.

Note that all options that do not include an argument can be combined together with a single hyphen. If this is successful, you can attempt to load the report file:

```
arsload -g <app_grp> -fnv -u <user_ID> -p <passwd> <file_name>
```

The -f option tells **arsload** to unload the report file if the load fails for any reason: If the database manager step fails, then Content Manager OnDemand should remove any index data that was added to the database, and if the storage manager step fails, then Content Manager OnDemand should remove any storage objects that were copied to storage volumes.

If the indexing or load fails for any reason, the verbose (-v) option will give you an error number that can be used to reference a possible solution in the manual *IBM Content Manager OnDemand: Messages and Codes*, SC27-1379.

Assuming the report loads successfully, the next step is to search for and view one more documents in the report with the Content Manager OnDemand Windows Client. You should check several items when the folder opens:

- Is the default date range correct?
- Are the search criteria fields labels correct and in the correct order?
- Are the default search operators correct?

- Are the wild card options correct for the Like search operator?
- Are the document list fields in the desired order (not all fields need to be displayed)?
- Are any “required” search fields (asterisk before the search field label) correct?

All of the above (and more) can be found in the Field Information tab of the folder. After you choose a document to view, check these items:

- Is the document orientation correct? (Orientation option on View Information tab of application)
- Is the document page visible in its entirety? (Zoom option on Logical Views tab of application)
- Is the background color correct? (Background Color on Logical Views tab of application)

If any of the search criteria fields or the document view is incorrect, you can make the change in the System Administration Client, close the folder in the Windows Client, and press the F5 button when you see the Open a Folder window (F5 will refresh the folder/application/application group information that the client displays).

2.5.4 User acceptance

After you test the reports (alpha testing), it is time to demonstrate your work to the parties that helped you define the reports. If that happened to be department or LOB managers, this is probably a good time to involve the actual users. If you work in an informal environment, you may be able to show the users how to access Content Manager OnDemand to see their reports and then install the Content Manager OnDemand Windows Client on their PCs. After they have the opportunity to use Content Manager OnDemand for a day or two, meet with them again and answer any questions they might have. Be prepared to make some changes to their reports, as it was probably more difficult for them to express their needs prior to working with their reports in Content Manager OnDemand. Remember, they are the people who will use the reports, and you want them to be satisfied. Your patience and attention to detail at this time will make future report definitions easier to create.

If significant changes are made after the second meeting, allow the users to play with Content Manager OnDemand for another day or two and set up a third meeting. By this time they should be familiar with the Content Manager OnDemand vocabulary and nomenclature, and you will have a better understanding of their business requirements. In software development, there is a concept referred to as *Build one to throw away.* You should not be surprised if the final version of their reports bear little resemblance to the initial design.

2.6 Functional testing

In this section, we discuss functional testing for the following functions:

- Load daemons
- Expiration processing
- Startup and shutdown processing
- Retrieval processing
- Printing
- Custom scripts
- Backup and recovery

2.6.1 Load daemons

The arsload program is used to process the input files that you want to load into the system. The arsload program determines if the input data needs to be indexed, and if so, calls the appropriate indexing program (based on the application definition). The arsload program calls the storage manager programs to load report data on storage volumes and the database manager to update the Content Manager OnDemand database with the index information that was extracted from or specified for the input file (generic indexer). The arsload program saves processing messages in the System Log. You can open the System Log folder and list the messages that were generated when an input file was processed. The relevant message numbers are:

- 86: Rows loaded (no view document)
- 87: Successful load (with view document)
- 88: Failed load (with view document)

You typically configure the arsload program to run as a daemon (UNIX servers) or service (Windows servers) to periodically check specified file systems for input files to process. You can specify multiple directories to monitor for input files (-d parameter) for a single arsload program, or you can run multiple UNIX daemons or Windows services. If you have extremely large input files for a particular application, you should consider a separate arsload program just for that application. The logic for this is that a single arsload program will sequentially process files in each input directory before it cycles back through to the first input directory. A large input file may delay loading smaller files with equal or greater priority. You may also want to consider setting up Content Manager OnDemand object servers on other computers for the express purpose of loading. Indexing files can be very CPU intensive and a limited time window for loading files into Content Manager OnDemand may over-utilize a single library server/object server implementation.

Another performance consideration for indexing is to make sure that the -c parameter (the file system in which Content Manager OnDemand temporarily stores data created by the indexing program) specifies a different file system than the file system that is specified with the -d parameter. If not specified, the temporary working directory (-c) defaults to the directory in which the arsload program was invoked. When arsload runs as a UNIX daemon, the default location is typically an operating system file system, which may cause failed loads due to inadequate space (the default Windows service location is the /arstmp directory). It is therefore highly recommended that you always include the -c parameter and specify a file system on a different physical hard drive than the monitored input directory (-d parameter).

When arsload runs as a UNIX daemon or Windows service, the input files must conform to a naming convention. The file name extension must be .ard or .pdf, and the file name itself is used to designate the application group (and application if necessary) that the file will be loaded into. Files downloaded through the arsjesd program will already have the correct file name schema:

```
MVS.JOBNAME.DATASET.FORM.YYYYDDD.HHMMSST.ARD
```

By default, the FORM field is used for the application group name. Another field can be used if you include the -G parameter and specify which field to use (MVS™, JOBNAME, DATASET, or FORM). If the application must be specified, you need to include the -A parameter and specify which field to use (MVS, JOBNAME, DATASET, or FORM; there is no default).

If some other method is used to transmit the input files to the object server, it will be necessary to name the files correctly on the sending system or rename the files on the receiving system. Renaming files on a UNIX system is usually accomplished with a shell script, and a batch file on the Windows platform. Other options include compiled programs (C, C++, or VB), Java programs, or Windows PowerShell.

2.6.2 Expiration processing

Expiration processing can be tested under controlled conditions. The arsmaint program allows you to specify a date other than the current date and thereby test database and cache migration/expiration using actual data retention values. This option must be used with extreme caution, as it is very easy to expire more data than you intended if you do not fully understand how the arsmaint program options function. The command you use to test expiration processing will look like this:

```
arsmaint [ -c | -d | -e | -i | -m ] -g <app_grp> -I <inst_name> -x 0 -n
0 -t <internal date> -u <user_ID> -p <passwd>
```

Note that the first five options are shown as exclusive options to be performed by themselves. In actuality, the options can be combined, but you probably want to test them separately. You will most likely only test the -c, -d and -m options. The arsmaint program options are as follows:

- -c: Expire files from cache storage.
- -d: Expire indexes from the Content Manager OnDemand database.
- -e: Migrate index data from the database to archive storage.

Important: Before you can migrate index data, the index tables must be closed. If the Database Organization for the application group is set to Single Load per table, the index table is closed when the report is loaded. Otherwise, if the Database Organization is Multiple Loads per table, the index table is closed when the Maximum Rows value is reached. To close a table to loading before the Maximum Rows value is reached, use the arstblsp program with the -a1 parameter.

- -i: Expire imported index data from the database.

Note: An administrator must import index data that was previously migrated to archive storage back into the database to satisfy a query. After maintaining the imported index data for the number of days specified in the Length of Time to Keep Imported Migrated Indexes (Storage Management tab in the application group), the data is eligible to be removed from the database.

- -m: Migrate files from cache storage to archive storage.
- -g: The name of the application group to process. Unless you specify this parameter, the arsmaint program performs the indicated operation for all of the application groups defined on the library server.
- -I: The name of the Content Manager OnDemand instance to process. By default, arsmaint uses the ARCHIVE instance.

Tip: Ideally, you should test expiration on a Content Manager OnDemand instance other than the production instance (ARCHIVE or whatever your production instance is named).

- -x: Specifies the high expiration threshold percentage for cache storage file systems. This value determines when the arsmaint program begins expiring files from cache storage file systems (only used for the -c option). Note that the specified percentage is 0. The default value is 80, and you would most likely wonder why nothing was expired from cache storage if you do not specify 0.
- -n: Specifies the low expiration threshold percentage for cache storage file systems. This value determines when the arsmaint program stops expiring files from cache storage file systems (only used for the -c option). Note that the specified percentage is 0. The default value is 80, and you would most likely wonder why nothing was expired from cache storage if you do not specify 0.
- -t: Specifies that you want the arsmaint program to process the database and cache storage by using a date other than the current system date (the default value). The value that you specify must be a valid Content Manager OnDemand *internal date* value (an integer number, not the typical mm/dd/yy format). You can use the arsdate program to display the *internal date* value for a given date string, for example, arsdate mm/dd/yy.
- -u: Specifies a Content Manager OnDemand user that has administrator permission for the application groups to be processed. If you specify the -g parameter, the user must have permission to delete documents from the application groups.

Tip: You may want to create a special user that will only be used for expiration testing, and then give that user permission to delete documents from the application group being tested. Once testing is complete, you should delete that user's permission to delete documents from the application group.

- -p: Specifies the password for the Content Manager OnDemand user ID that is identified with the -u parameter.

When expiration testing is performed, you should test at least two dates: the day before the index/object should expire (to verify data is retained the correct length of time), and the day the expiration should take place (to verify data is expired correctly). You should only work with test data, and never when production data/reports have been loaded into the application group. Using a test Content Manager OnDemand instance and a special user ID is highly recommended.

2.6.3 Startup and shutdown processing

This section describes how to use operating system facilities to automatically start or schedule instance operations. You can automatically start these instance operations whenever the system is started:

- Start the database on the library server.
- Start the instance on the library server.
- Start the instance on an object server.
- Start the data loading programs.

You can also schedule these instance operations to begin automatically on a regular schedule:

- Schedule application group maintenance on the library server.
- Schedule application group maintenance on an object server.
- Schedule system table maintenance.
- Schedule a backup of the Content Manager OnDemand database.
- Schedule a backup of the Tivoli Storage Manager database.

UNIX

In UNIX environment, you can set up startup, schedule, and shutdown programs.

Startup

Programs that start automatically when the UNIX server is started (system services) utilize the init facility. On AIX, you create entries in the /etc/inittab file with the **mkitab** command or simply edit the file with a text editor. HPUX also uses the /etc/inittab file, as does Linux. On Solaris, the files are named differently (with different directories for different run levels), but the concept is the same (/etc/rc3.d is the multiuser run level directory). These directories contain scripts that Start (script name begins with S) or Kill (script name begins with a K) system services.

You can use the arsdb program to start the database on the library server. The following example shows an INIT record to automatically start the database when the operating system is initialized on the library server:

```
ars2:2:wait:su - archive "-c /usr/lpp/ars/bin/arsdb -gkv" >>
/tmp/arsdb.log 2>&1
```

The following example shows an INIT record that automatically starts the instance named archive when the operating system is initialized on the library server:

```
ars3:2:once:/usr/lpp/ars/bin/arssockd archive
```

The following example shows an INIT record that automatically starts the instance named archive when the operating system is initialized on an object server:

```
ars4:2:once:/usr/lpp/ars/bin/arsobjd archive
```

The following example shows an INIT record that automatically starts the arsjesd program during operating system initialization:

```
ars5:2:once:/usr/lpp/ars/bin/arsjesd -p 6001 -d /arsacif/acif1 -d
/arsacif/acif2 -d /arsacif/acif3 >> /tmp/arsjesd.log 2>&1
```

The following example shows an INIT record that automatically starts the arsload program for the instance named archive during operating system initialization:

```
ars6:2:once:/usr/lpp/ars/bin/arsload -v -c /arsacif/acif4 -d
/arsacif/acif1 -d /arsacif/acif2 -d /arsacif/acif3 -I archive
```

Scheduled

System services that are scheduled activities utilize the cron facility.

The following is an example of a CRON record that automatically starts the arsmaint program every day at 4 am for the instance named archive. The arsmaint program will maintain application group data in cache storage, including copying report data to archive storage. This format of the command is typically used for an object server with Tivoli Storage Manager on some other workstation than the library server:

```
00 4 * * * /usr/lpp/ars/bin/arsmaint -cmsv
```

On a library server, database maintenance operations would be included and the command would look like:

```
00 4 * * * /usr/lpp/ars/bin/arsmaint -cdeimrsv
```

The following is an example of a cron record that automatically starts the arsdb program to maintain the Content Manager OnDemand system tables for the instance named archive. The arsdb program will run twice a month, on the 7th and 14th of each month, beginning at 5 am:

```
00 5 7,14 * * /usr/lpp/ars/bin/arsdb -mv -I archive >> /tmp/arsdb.log
2>&1
```

The following is an example of a cron record that automatically starts the arsdb program to create a full online backup image of the Content Manager OnDemand database for the instance named archive every day beginning at 5:30 am. The backup image is written to a tape in the device /dev/rmt0. A tape must be mounted in the device before the arsdb program begins:

```
30 5 * * * /usr/lpp/ars/bin/arsdb -v -z /dev/rmt0 -I archive >>
/tmp/arsdb.log 2>&1
```

The following is an example of a cron record that automatically starts the ars_adsm program to create a backup copy of the Tivoli Storage Manager database at 5:30 am each day:

```
30 5 * * * /usr/lpp/ars/bin/ars_adsm -dv >> /tmp/ars_adsm.log 2>&1
```

Shutdown

The Solaris operating system provides a facility for stopping system services (K scripts in the various run level directories like /etc/rc3.d). All of the other UNIX platforms use a shutdown script that can be modified to call a Content Manager OnDemand shutdown script. The most important aspect of the shutdown script is the order in which services are stopped. The recommended order is:

- arsjesd
- arsload
- arssockd
- database

A sample shutdown script is shown in Example 2-1. Note that this script only guarantees a graceful shutdown of the Content Manager OnDemand process and DB2 database. The arsload and arsjesd programs do not have a command-line option to stop the process and therefore must be terminated with the UNIX **kill** command. You should verify that files are not being downloaded (arsjesd) or ingested (arsload) when you shut down Content Manager OnDemand and DB2. This can be done in a shell script, but it is beyond our discussion here.

Example 2-1 Sample shutdown script

```
#!/usr/bin/ksh
INSTANCE_NAME=archive
LOG=/tmp/stop_ondemand`date +"%m%d%y_%I%M"`
{
echo "Stopping Content Manager OnDemand"
/usr/lpp/ars/bin/arssockd stop ${INSTANCE_NAME}
echo "Disconnecting database applications"
su - ${INSTANCE_NAME} -c "db2 force applications all"
sleep 10
echo "Stopping database manager"
/usr/lpp/ars/bin/arsdb -I ${INSTANCE_NAME} -hv
RC=$?
if [ "$RC" -eq 0 ] ; then
echo "database manager stopped"
else
echo "forcing database manager shutdown"
su - ${INSTANCE_NAME} -c db2stop force
fi
} >$LOG 2>$1
exit 0
```

Windows

In a Windows environment, you can set up startup, schedule, and shutdown programs.

Startup

Any Windows service can be started automatically by selecting **Start** → **Control Panel** → **Administrative Tools** → **Services**. Select the service and right-click it, select **Properties**, and change **Startup type** on the General tab to automatic (manual and disable are the other choices).

Scheduled

The Content Manager OnDemand Configurator program (started by selecting **Start** → **All Programs** → **IBM Content Manager OnDemand for Windows** → **Configurator**) allows you to schedule three common tasks:

- ApplGroup Data Maintenance (arsmaint)
- System table Maintenance (arsdb)
- Content Manager OnDemand Database Backup (arsdb)

You can schedule other tasks with the Windows Scheduled Task facility (by selecting **Start** → **All Programs** → **Accessories** → **System Tools** → **Scheduled Tasks**).

Shutdown

Unlike UNIX, there are no out-of-the-box automatic methods to explicitly shut down processes when you shut down or restart a Windows server (you must use the Windows Services tool to manually stop services), but Windows will attempt to gracefully shut down services. It is possible to write batch files (using the `net start` and `net stop` commands) or an application (in C or Visual Basic) to start and stop services.

2.6.4 Retrieval processing

Log on to the Content Manager OnDemand Windows Client or browser interface and make sure that you can find documents within reports. Test security and query permissions.

2.6.5 Printing

Test printing documents to local printers and to host printers. Use the Content Manager OnDemand Administration Client to add and authorize server printers.

2.6.6 Custom scripts

If you use any exit points (refer to 2.4.7, “Exit points” on page 40), be sure to test them. If you use Kofax Ascent Capture to scan documents and release them to a directory that is monitored by Content Manager OnDemand, be sure to test the entire process.

2.6.7 Backup and recovery

Creating a backup and recovery plan that is suitable for your environment and meets your corporate requirements is only the first step. Because a backup and recovery plan is vital to business continuity, you should actually attempt a recovery scenario using different equipment to verify that your backup and recovery plan and document is complete. Do not wait until disaster strikes to find out if your recovery plan will work.

2.7 Performance testing and considerations

Once functional testing is complete and we feel comfortable that the system is doing what we expect it to do, we need to performance test the system. The goal of performance testing is to ensure that the system does what we expect it to do under load within specified time constraints.

In this section, we discuss the following performance testing and considerations:

- Develop test cases based on requirements
- Load testing
- Search and retrieval testing
- Tuning considerations based on test results

2.7.1 Develop test cases based on requirements

The developed test cases should:

- Reflect the loads that we expect on the system.
- Reflect the document data types and sizes.
- Reflect realistic Content Manager OnDemand DB2 table sizes.
- Reflect the production system environment.

The goal is to run a set of tests that will either:

- Produce the same results that we expect to get in production. This can only happen if the test environment is identical to the production environment, which is very difficult to do.

  or

- Produce results that will allow us to extrapolate and predict what our results will be like in the production environment. This is the more realistic goal, and at worst, will provide a baseline to compare performance on the production system with.

2.7.2 Load testing

When loading reports into Content Manager OnDemand, there are two main activities:

- Indexing, which can be very CPU and memory intensive
- Loading, which is more I/O intensive (and CPU intensive if compression is used)

In general, the most important measure for load testing is time to load. Many Content Manager OnDemand implementations only load data off-hours during a so called load window. At some point, the amount of data to be loaded will exceed the time available for the load window. Load testing will help identify this problem before it occurs and can help evaluate alternatives, such as:

- Parallel loads: Multiple load jobs will theoretically increase overall efficiency by increasing CPU utilization and I/O throughput.
- Adding object servers: Multiple object servers effectively add CPU resources and only minimally increase the I/O on the library server.

For those implementations that load data around the clock, CPU activity, memory requirements, and I/O may be the most important considerations. In this case, you may want to consider running the arsload process(es) at a lower priority so that document searching and retrieval is less affected.

2.7.3 Search and retrieval testing

Search and retrieval testing should also be run in an environment as reflective as possible of the production environment. Automated testing tools and scripts can be used to mimic the expected user work load. Care should be taken to simulate both the numbers of users and the behaviors of these users (what types of documents would they retrieve, how often, how many indexes, the type of search, and so on). It is especially important to search for and retrieve different documents, as many operating systems include cache memory or disk caches that will give false performance measurements if the same document is constantly retrieved.

This type of testing is especially valuable in terms of tuning the database and operating system parameters for optimal performance. The *Content Manager OnDemand Guide*, SG24-6915, provides an overview of several DB2 parameters that can easily be adjusted to improve response time (for example, buffer pool size) and many more IBM Redbooks publications address DB2 tuning at a deeper technical level. A search for "DB2 tuning" at the IBM Redbooks site:

http://www.redbooks.ibm.com/

came back with 97 different references, some of which are product specific but many that are more generic in nature. Searching for "aix tuning" resulted in 103 references, so there is an abundance of information available to help you in this task.

2.7.4 Tuning considerations based on test results

The first set of performance tests will produce a set of baseline results. If these results are acceptable, then production deployment is the next step. Otherwise, you will need to determine what performance bottlenecks exist and how they can be minimized. There are potentially many areas that require monitoring and tuning, a full discussion of which is beyond the scope of this book. These areas include:

- OS parameters
- Database configuration
- Network infrastructure
- Hardware (CPU, memory, and disk drives)
- Content Manager OnDemand report design issues

See 5.2, "Application service provider case study" on page 229 for a case study that involves performance testing.

2.8 Training

In this section, we discuss the following training:

- Administrator training
- User training

There is classroom training available from IBM for Content Manager OnDemand administrators. The training path can be found at:

```
http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageT
ype=page&c=a0000419
```

Click **Content Manager OnDemand Administration**.

User training is not available from IBM, primarily because the administrator should be able to explain the client to the users in a very short period of time. The client was developed to be as intuitive as possible and that, combined with extensive online help, makes for a short learning curve. Users cannot cause loss of data, but they can cause performance problems if they are not aware of the implications of their actions (for example, wildcard searches, text searches, and searching without specifying a date).

2.8.1 Administrator training

We recommend that at least two people get system administrator training in Content Manager OnDemand. This training should include all features of the Administration Client in addition to learning about TSM, the back-end database, how to start and end the server, and other tasks. Any users who will be working with the Administration Client to authorize other users and user groups or set up report definitions should also be trained in those features.

2.8.2 User training

We suggest that you demonstrate the basic functions of Content Manager OnDemand to the users, then create a User's Guide that contains graphics or screen captures of how to access sample reports on your system. You may choose to have training sessions where the users go through hands-on exercises when you are there to assist them and answer questions. Whether they use the Content Manager OnDemand Windows Client or a Web browser interface to Content Manager OnDemand, all users need to know how to log on to Content Manager OnDemand, find their documents and view them, and log off. There are many other features that they may also use (not all are available in a browser session). Here are few features you can show them:

- Toolbar icons.
- View menu options.
- Options menu.
- Changing search operators.
- Wildcard search on string fields (prepend and append).
- Search by specific date range, or t-5y for today minus 5 years.
- Sort a hit list.
- Annotations (view and add).
- Print to local and, optionally, server printer.
- Change background color and zoom and save as a logical view.
- Locking headers on a report listing (if applicable).
- Find a character string within a document.
- Text search (server based).
- Move or omit columns in a report listing (if applicable).
- E-mail a document.
- Create and save a named query.
- Copy pages from a document to a file.

Tip: Train users no more than one week prior to when they begin using the system.

2.9 Deployment into production

After you have installed and configured your Content Manager OnDemand for Multiplatform environment, set up and tested some report definitions, worked with and trained the users, it is time to put the system into production. Make sure that you complete these steps:

- Modify your business applications to send print streams to output queues that are monitored by Content Manager OnDemand.

- Install and configure the Content Manager OnDemand Windows Client or browser interface for users.
- Grant permissions to user groups (and individual users if necessary) to the appropriate application groups and folders.

In this section, we discuss the following activities:

- Automate system process
- System documentation
- System monitoring

2.9.1 Automate system process

There are certain Content Manager OnDemand operations that must be performed periodically to maintain performance levels and ensure business rules are met (for example, documents must be deleted after some length of time). You can schedule these (instance specific) operations to begin automatically on a regular schedule with the Content Manager OnDemand Configurator (Windows) or cron facility (UNIX):

- Schedule application group maintenance on the library server.
  - Expire metadata from the Content Manager OnDemand database (**`arsmaint -d`**).
  - Run database statistics to optimize index data (**`arsmaint -r`**).
- Schedule application group maintenance on an object server
  - Migrate documents (**`arsmaint -m`**) from the cache file system to the storage system.
  - Expire documents from the cache file system (**`arsmaint -c`**) and storage system (TSM operation)

There are other options to the arsmaint program that can be used to set cache file system thresholds, validate cache storage, migrate metadata, expire imported metadata, and generate reports on cache file systems.

Database and TSM backups may also be automated as well. Refer to 2.6.3, “Startup and shutdown processing” on page 61 (Scheduled) for examples of database backups on the UNIX platform. Off-line database backups are slightly more complex (the database must be shut down), but the process can be automated with a shell script that is scheduled as a cron job.

2.9.2 System documentation

You may recall from 2.3.7, “Implementation best practices” on page 32 that a backup and recovery plan and document was highly recommended (if not an outright requirement) before the Content Manager OnDemand system went into production. Your Content Manager OnDemand system will change over time and you must review and update the backup and recovery plan and document as changes occur. Examples of these include:

- Software updates
- System password changes
- Changes to system security
- New reports, user groups, and printers
- New user exits
- New file systems
- Changes to configuration files
- Changes to system parameters for improved performance
- Performance reports
- Hardware and software failure notes

2.9.3 System monitoring

You need to monitor the Content Manager OnDemand system, Tivoli Storage Manager, and your operating system.

Content Manager OnDemand

Content Manager OnDemand provides the ability to log many different items, errors, and actions for application groups, users, and server events. Every application group has some message logging turned on by default, and the number of default items that are logged depends on the level of the administrative client in use. To reduce the amount of unwanted logging information, for any new application group, verify that only those items needing to be logged are checked in the Message Logging tab. Conversely, at times it may be necessary to turn on additional logging to help troubleshoot problems.

During normal processing, the Content Manager OnDemand programs generate messages. Content Manager OnDemand saves the messages in the System Log and sends a copy of each message to the System Log user exit point. Content Manager OnDemand assigns a severity to each message. For Windows servers, messages that are assigned a severity of alert or error are also sent to the Event Log. When a user runs a query that requires a table of index data that has been migrated to archive storage, Content Manager OnDemand sends a message to /dev/console (UNIX servers) or the Event Log (Windows servers). Content Manager OnDemand provides the system logging facility and a message reference to help you identify and resolve any alerts and errors that you may receive. You can open the System Log folder to display the messages that are saved in the System Log. *IBM Content Manager OnDemand: Messages and Codes*, SC27-1379 is the message reference and contains troubleshooting information for the messages that may be generated by the Content Manager OnDemand programs.

Tivoli Storage Manager

Tivoli Storage Manager provides you with ways to monitor processes and messages:

- Use the console mode from an administrative client to monitor processes and messages:

  ```
  dsmadmc -consolemode
  ```

  While the system is running in console mode, you cannot enter any administrative commands from the client session. You can, however, start another administrative client session for entering commands.

- Specify the OUTFILE option to write all terminal output to a file. For example:

  ```
  dsmadmc -consolemode -outfile=adsm.out
  ```

- From the command-line interface, query the activity log for status information and possible error messages:

  ```
  query actlog
  ```

Refer to the Tivoli Storage Manager documentation for more information about managing client sessions.

UNIX

Table 2-3 shows some of the common UNIX commands that can help you monitor the library/object server or help troubleshoot problems. Note that some commands are platform specific. In addition to these commands, there are a multitude of third-party software products whose sole purpose is to simplify system and performance monitoring.

Table 2-3 Common UNIX commands for monitoring system performance

Command	Used for
df (disk free)	Shows file system free space.
du (disk usage)	Shows file system usage.
errpt (AIX only)	Reports system errors (hardware and software).

Command	Used for
`iostat`	Shows I/O activity.
`lsps` (AIX only)	Shows paging space usage.
`mpstat`	Collects and displays performance statistics for all logical CPUs in the system.
`netstat` (network status)	Shows network status.
`ping`	Testing network connections.
`ps` (process status)	Shows process status.
`sar` (system activity report)	Collects, reports, or saves system activity information.
`svmon` (AIX only)	Captures and analyzes a snapshot of virtual memory.
`topas` (AIX only)	Reports selected local system performance statistics.
`traceroute`	Printing the route packets take to a host.
`vmstat`	Shows virtual memory statistics.

Windows

You can use Performance Monitor and Task Manager to monitor your Windows environment.

Performance Monitor

For Windows servers, Performance Monitor is the tool most often used to monitor server performance. It can be opened by navigating to the performance icon in the administrative tools folder in the control panel, from the Start menu or by typing perfmon.msc in the run box. Performance Monitor performs data collection and analysis. Performance Monitor uses objects and counters to associate statistical information with monitored components. For Content Manager OnDemand server analysis, we recommend that you collect information about the following objects:

- System
- Processor
- Memory
- Logical disk

- Physical disk (if using RAID)
- Server
- Cache
- Network adapter
- Database (DB2, Oracle, and SQL Server provide Performance Monitor objects and counters.)

Task Manager

Task Manager is another tool that provides information about programs and processes running on your computer. It displays the most commonly used performance measures for processes (CPU and memory usage), computer performance (CPU and page file usage), and networking (adapter network utilization). There are four tabs in the Windows Task Manager window:

- The Applications tab shows the status of the programs running on your computer.
- The Processes tab shows information about the processes running on your computer.
- The Performance tab displays a dynamic overview of your computer's performance.
- The Networking tab displays a graphical representation of network activity.

3

IBM Content Manager OnDemand for i5/OS implementation guidelines

This chapter provides guidelines for implementing an IBM Content Manager OnDemand for i5/OS solution.

We cover the following topics:

- Identify project resources
- OnDemand server sizing
- Installation and configuration
- Design
- Application setup and verification
- Functional testing
- Performance testing and considerations
- Training
- Deployment into production
- Lessons learned

3.1 Introduction

Any IBM System i customer who creates reports and wants to save those reports for easy retrieval can benefit from IBM Content Manager OnDemand (OnDemand). And any customer who wants to scan and archive documents or save electronic files for easy retrieval can also benefit from OnDemand. OnDemand for i5/OS is used worldwide by all kinds of businesses and companies in various fields, such as health care, insurance, manufacturing, distribution, utilities, retail, school systems, and government agencies. Instead of searching within spooled files saved in output queues or looking for printed information in file cabinets, users can access archived documents and spooled files from their computers connected to the System i.

OnDemand for i5/OS is an excellent archival and retrieval solution for businesses with few users and for businesses with thousands of users. To have a successful implementation with satisfied users, it is important to spend time up front in planning and designing, from the initial requirements gathering to the final deployment of OnDemand for i5/OS to the users.

This chapter is intended to be a supplement to IBM product documentation and publications. We give you our suggestions and general guidelines for implementing OnDemand for i5/OS, and refer you to specific publications for more details. We have several references to the publication *IBM Content Manager OnDemand for i5/OS Common Server Planing and Installation Guide,* SC27-1158, which we refer to as the *Planning and Installation Guide*.

3.2 Identify project resources

We suggest that you select at least two people to be system administrators. You also need to work with System i system administrators and operators to schedule OnDemand jobs, modify backup procedures, and manage system storage. Be sure to get some users involved in the project so you understand how they would like to access the archived data.

Review Chapter 2, “Preparing for OnDemand”, in the *Planning and Installation Guide.* The chapter provides a good summary of the tasks required to administer OnDemand and helps you assign project responsibilities.

3.3 OnDemand server sizing

Whether you have five or 5000 OnDemand users, the questions to ask for the OnDemand server sizing are the same:

- Which files (spooled, scanned, or PC files) do you want to archive?
- What are the sizes of the files and how often are they created?
- How long do you want to keep the files?
- Will you be archiving PDF files?
- Will you be adding users to your System i?
- What type of storage do you plan to use for your archived documents?

In our experience, customers rarely need to upgrade the System i when they install OnDemand, but it can be necessary in some environments. If your system is already too busy, any new application could cause performance problems. Or if you are already using a high percentage of disk storage, you may need to add capacity when you start archiving files.

For detailed guidelines on sizing, refer to Chapter 4, “Hardware and Software” and Chapter 7 “Storage Requirements”, in the *Planning and Installation Guide*.

3.4 Installation and configuration

Here are the steps that you can follow to install and configure OnDemand for i5/OS:

1. Install software.
2. Create an instance.
3. Set up the OnDemand administrator.
4. Create a migration policy using System i Navigator.

The chapters that we reference in this section are from the *Planning and Installation Guide*.

3.4.1 Installing software

To install the necessary software:

1. Print and review the Read This First document from the OnDemand support Web site:

   http://www.ibm.com/software/data/ondemand/400/support.html

2. Install the required pre-requisite software. See Chapter 4, “Hardware and Software”, of the *Planning and Installation Guide*.
3. Install the OnDemand server and client software. See Chapter 11, “Installing OnDemand Server Software”, of the *Planning and Installation Guide*. You need to do a custom installation of the client software in order to install the Administration Client. If you select **Typical** when you install the software, the English Windows Client will be installed. To add other features, just install the client software again and modify the installation to add the Administration Client or other languages or features.
4. Order, load, and apply PTFs for the software you installed. You can find a list of OnDemand PTFs at:

   http://www.ibm.com/software/data/ondemand/400/support.html

5. Use the WRKLICINF command to specify the number of authorized users.
6. Install the OnDemand Archive plug-in for System i Access for Windows on any PCs where the OnDemand Administration Client will be used.
7. Install and configure the appropriate software and hardware if you plan to use a browser to access OnDemand, such as IBM Web Interface for Content Management (WEBi) or the OnDemand Web Enablement Kit (ODWEK). Refer to the OnDemand for i5/OS Web site to find more information about these products:

   http://www.ibm.com/software/data/ondemand/400

8. Install and configure Tivoli Storage Manager (TSM) if you plan to use TSM as an additional storage manager. For more information about how to configure OnDemand to work with TSM, go to the OnDemand support Web site and search on TSM.

3.4.2 Creating an instance

To create an instance:

1. Determine the locale to use for your OnDemand instance. See Chapter 13, “Defining a locale,” of the *Planning and Installation Guide.*

2. Create a production instance. We prefer to name the instance QUSROND because that is the default instance name in the commands (for example, ADDRPTOND, RMVRPTOND, and STRMONOND). Refer to Chapter 12, “Creating an Instance”, of the *IBM Content Manager OnDemand for i5/OS Common Server Planning and Installation Guide*, SC27-1158.

> **Note:** From our experience with Version 5 of the OnDemand licensed product, the installation process partially creates an instance that you should delete. To delete the partially created instance from the installation, delete the following objects:
>
> - The QUSROND library
> - The IFS directory /QIBM/UserData/OnDemand/QUSROND
> - The authorization list QUSROND
> - The user profile QUSROND.
>
> See Chapter 12, “Creating an instance”, in the *Planning and Installation Guide.*

3. Start the server for the instance. You can use the command STRTCPSVR *ONDMD if the value for ARS_AUTOSTART_INSTANCE=1 in the ars.cfg file is in the IFS directory /QIBM/UserData/OnDemand/*instancename*. Or you can use the command CALL QRDARS/QRLMCTL *STRTCPSVRinstancename.

3.4.3 Setting up the Administration Client

To set up the OnDemand Administration Client:

1. Change the i5/OS user profiles for OnDemand administrators. Also change the user profiles for all users who will archive reports and documents:

   – Add group or supplemental profiles QONDADM, QRDARSADM, and QRDARS400.

   – Change the Locale Job Attributes and Locale parameters for the administrator profiles.

   For example:

   ```
   CHGUSRPRF USRPRF(ABC) GRPPRF(QONDADM) SUPGRPPRF(QRDARSADM QRDARS400)
   SETJOBATR(*CCSID *DATFMT *DATSEP *DECFMT *SRTSEQ *TIMSEP)
   LOCALE('/QSYS.LIB/DE_DE_E.LOCALE')
   ```

   The DE_DE_E locale is used for German with European support. For this parameter, specify the locale for your system that you determined in 3.4.2, “Creating an instance” on page 78. For example, US English is EN_US.

2. Log on to the OnDemand Administration Client with user ID QONDADM, password QONDADM1. The first time you log on, you will be prompted to change the password.
3. Add at least one user as a system administrator. Then log off and log on again with this user ID. We recommend having administrators log on with their own IDs instead of logging on as QONDADM.

3.4.4 Creating a migration policy using System i Navigator

Open the OnDemand Archive plug-in and create a disk pool or an optical storage group so that you can have storage levels in the policy. Then create a migration policy with at least one storage level. Refer to *Content Manager OnDemand Guide*, SG24-6915 for details and suggestions. Read the Archive Storage Manager for System i section in Chapter 5, “Storage Management.” To retrieve the book, go to:

http://www.redbooks.ibm.com/abstracts/sg246915.html?Open

After you have created a storage level and a migration policy, go back to the OnDemand Administration Client and select storage sets. You should see a storage set with the same name as the migration policy you created (you may need to press F5 to refresh the display).

> **Note:** You can use the OnDemand Administration Client to display a storage set associated with a migration policy. To update or delete a storage set, you must use the System i Navigator OnDemand plug-in and work with the migration policy for that storage set.

Create additional instances as needed. For example, you may want to create a test instance called ONDTEST. Each instance will need a separate TCP/IP port number. If you want the instance to be automatically started with the command STRTCPSVR *ONDMD, be sure to change the ARS_AUTOSTART_INSTANCE parameter in the ars.cfg file. Refer to Chapter 12, “Creating an Instance”, in the reference guide.

When you have completed these steps, you should be able to log on to the OnDemand Windows Client and view some entries in the System Log. When you type the command WRKACTJOB SBS(QSYSWRK), you will see an entry for your instance server job. Now you are ready to start designing *your* OnDemand environment!

3.5 Design

During this phase of the implementation, you decide which reports to archive, how long, and where you keep the reports in OnDemand, and what search fields you use to find documents within those reports.

In this section, we discuss the following design tasks:

- Report selection
- Indexing requirements
- Retention
- Application group, application, and folder
- Security
- Document Audit Facility
- Indexers
- Exit points

3.5.1 Report selection

We have found that when users stop looking for computer printouts in file cabinets and desk drawers and sometimes trash cans, and instead start accessing their reports in OnDemand, they start thinking of more and more reports they would like to have in OnDemand. And when one group starts using OnDemand, often other users want to be included in the project too. So keep in mind that you will not be able to make *all* the report decisions before you begin the project. Your OnDemand environment will change later. But you have to start somewhere.

Some businesses decide to implement OnDemand in only one area (for example, Accounts Payable) and add other areas of the business over time. Other businesses begin with a few reports from each area or department, then add more reports based on user requests. And some businesses set up all their reports in OnDemand before going into production with the users. So first you need to choose which of these approaches to follow when you select the reports. Was the decision to implement OnDemand based on the needs of a particular department, or will you begin with a company-wide implementation and let users access more and more reports in OnDemand over time? We suggest that you start by talking with the users and making a list of each report that you will load into OnDemand. Be sure to specify who will access each report.

3.5.2 Indexing requirements

Let us assume that you have a list of 50 reports from the various user departments. Your next step is to decide how to index each report. In this book, we assume that you have had some training in OnDemand, but the users may have never seen a demonstration of the product. If you work with users who are new to OnDemand, we suggest that you first set up a report yourself and load it into OnDemand. Then show it to the users and ask for their comments and suggestions. After the users have learned some easy and efficient ways to access reports in OnDemand, they will be able to tell you how they would like to have their reports defined.

For example, you decide to index accounts payable checks by check number, check date, vendor number, and vendor name. The users probably are very pleased that they can find a check by any of these search criteria. Perhaps these would have been the indexes they selected themselves. Or maybe they tell you that they also need to be able to search for checks by the check amount, for example, all checks over $500. So you go back and add check amount, and the users know that they are part of the decision-making process.

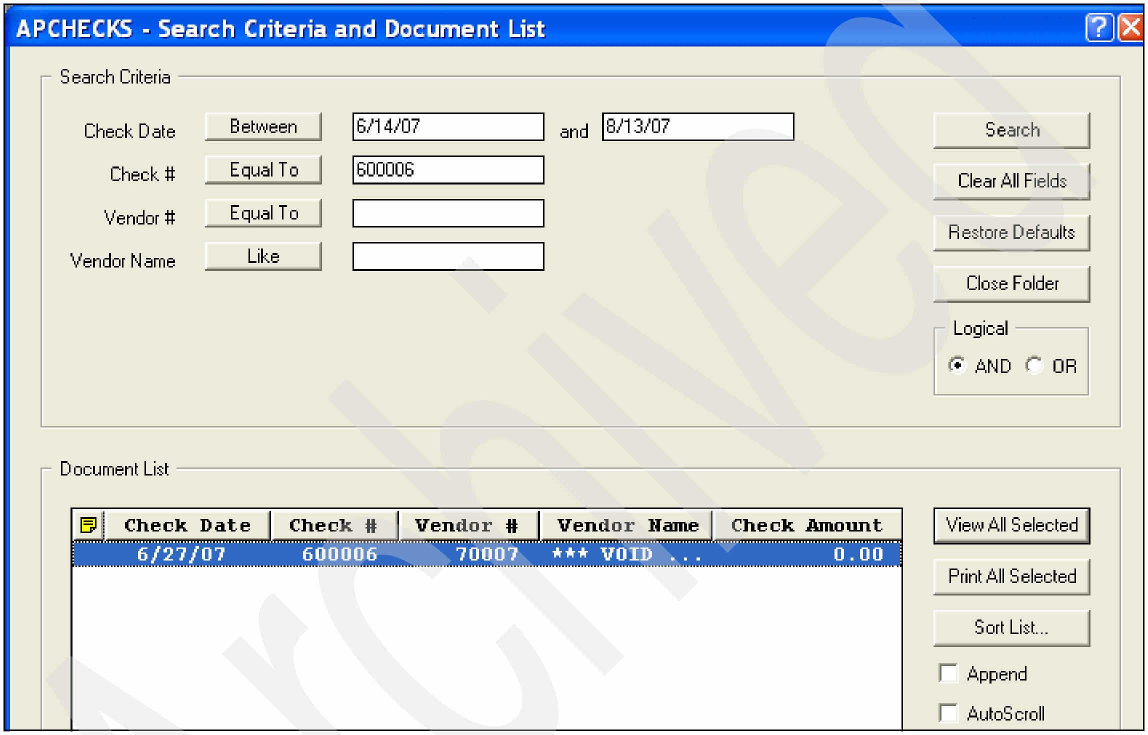
As another example, you may have a list of fixed assets or a trial balance report or some other type of management report. You define this report into OnDemand and use only the report date as an index. The users can view the entire report as a single document, go to the last page to see the totals, or search for a particular field within the report. But if you had started by asking the users how they would like to access this report, they may have said “by total” or “by part number” or another field, because they did not know how to use OnDemand to search for information within a document. After the users have seen some demonstrations of accessing their reports in OnDemand, they will be able to tell you how they would like to find their reports and documents.

The questions to ask for each report are:

- How do you want to find a specific document within the report? In other words, what are the search fields?
- Should the search fields be filters or indexes?
- When the value of a search field changes within the report, should this cause a break to a new document?

A field should be an *index* if it can be used by itself to uniquely identify and search for a document. When you create an index, OnDemand creates an access path so that the search is fast. When you search on a field that is defined as a *filter*, OnDemand searches the application group database records sequentially. We recommend making a field a filter if you do not have to search on the field, or if you only search with this field in addition to another field that is defined as an index.

In the accounts payable checks example, the users prefer to search checks by check number, check date, vendor number, or vendor name, and see the check amount displayed in the hit list, as you can see in Figure 3-1. In this example, you set Check Amount to be a filter instead of an index, which is the default when you use the graphical indexer.



APCHECKS - Search Criteria and Document List

Search Criteria

Check Date	Between	6/14/07	and	8/13/07
Check #	Equal To	600006		
Vendor #	Equal To			
Vendor Name	Like			

Logical
● AND ○ OR

Document List

Check Date	Check #	Vendor #	Vendor Name	Check Amount
6/27/07	600006	70007	*** VOID ...	0.00

□ Append
□ AutoScroll

Figure 3-1 Folder search

As another example, the users want to search by check amount, but not as a stand-alone search. In other words, they may search for all checks over a certain amount for a particular vendor number within a date range, In that case, Check Amount would also be a filter. But if it would be typical for a user to search only for a specific check amount, you would make that field an index so that an access path could be built for that field and the search would be faster.

OnDemand has been enhanced beginning with server level 7.1.2.8 so that you can change a field from an index to a filter or from a filter to an index after the application group has been created and reports have been archived. So if you make the wrong choice, you can change it later.

The graphical indexer uses the default of BREAK=YES for each index. Whenever the value of any index changes, there is a break to a new document. In our example here, you really need to break to a new document (that is, to a new check) only when the check number changes. So you can change the indexer parameter for the other indexes to BREAK=NO. This parameter can be changed in the graphical indexer, as you see in Figure 3-2, or in the indexer parameters, as you see in Figure 3-3.

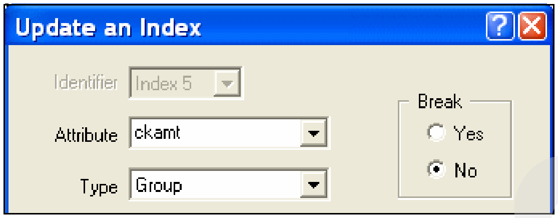
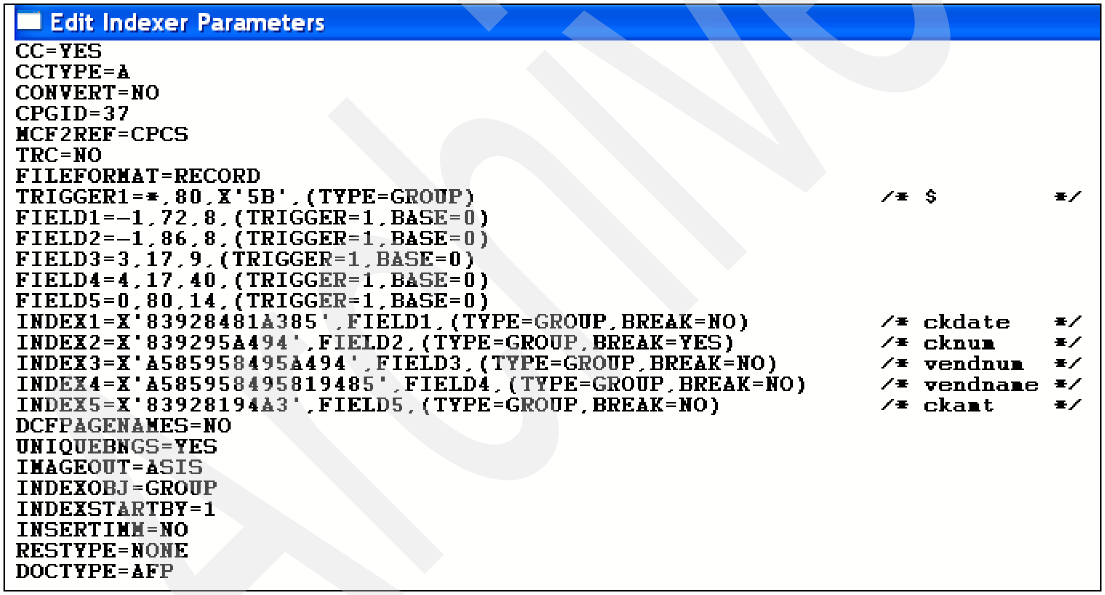


Figure 3-2 Update an index: Break=No



```
CC=YES
CCTYPE=A
CONVERT=NO
CPGID=37
MCF2REF=CPCS
TRC=NO
FILEFORMAT=RECORD
TRIGGER1=*,80,X'5B',(TYPE=GROUP)                             /* $        */
FIELD1=-1,72,8,(TRIGGER=1,BASE=0)
FIELD2=-1,86,8,(TRIGGER=1,BASE=0)
FIELD3=3,17,9,(TRIGGER=1,BASE=0)
FIELD4=4,17,40,(TRIGGER=1,BASE=0)
FIELD5=0,80,14,(TRIGGER=1,BASE=0)
INDEX1=X'83928481A385',FIELD1,(TYPE=GROUP,BREAK=NO)          /* ckdate   */
INDEX2=X'839295A494',FIELD2,(TYPE=GROUP,BREAK=YES)           /* cknum    */
INDEX3=X'A585958495A494',FIELD3,(TYPE=GROUP,BREAK=NO)        /* vendnum  */
INDEX4=X'A585958495819485',FIELD4,(TYPE=GROUP,BREAK=NO)      /* vendname */
INDEX5=X'83928194A3',FIELD5,(TYPE=GROUP,BREAK=NO)            /* ckamt    */
DCFPAGENAMES=NO
UNIQUEBNGS=YES
IMAGEOUT=ASIS
INDEXOBJ=GROUP
INDEXSTARTBY=1
INSERTIMM=NO
RESTYPE=NONE
DOCTYPE=AFP
```

Figure 3-3 Indexer parms break=no

3.5.3 Retention

How long and where do you want to keep your reports? If all your reports will be kept in the same location (for example, on System i disk storage) and for approximately the same length of time, you can use a single migration policy (storage set) for all your reports. If you plan to use a System i auxiliary storage pool (ASP), open System i Navigator and use the OnDemand Archive plug-in to create a disk pool and a migration policy with this disk pool as the only storage level. For example, create disk pool 1 if you plan to use the System ASP. Specify **No maximum** for the duration at that level. You can use this storage set (migration policy) for many application groups, although different application groups may specify different values for the Life of Data and Indexes. Remember that a report will be deleted (expired) when the number of days in the Life of Data and Indexes field has been reached, or when an Expire level in the migration policy is reached. It is a good idea *not* to have an Expire level in the policy, so you can control the expiration for each application group.

There is a change in our recommendations regarding migration policies. In the past, we recommended creating a single migration policy for all reports, whether the report would be on disk for 90 days or for seven years. There is a problem with this approach. If you take the migration policy default and use aggregation, OnDemand aggregates (or joins together) different reports so that you have a few big objects instead of a lot of small ones. This is an efficient way to move data from one storage level to another, and backups are faster when there are fewer objects in a single IFS directory. However, OnDemand does not delete an aggregated object until it is time to delete the entire aggregate. A report's database records (the indexes) will be deleted, but not the data itself. Now, we recommend that you have separate migration policies for different retention periods. You do not need to have one for reports that need to be kept for 30 days and another for reports to be kept for 60 days, but you probably do not want to mix reports with very different retention periods.

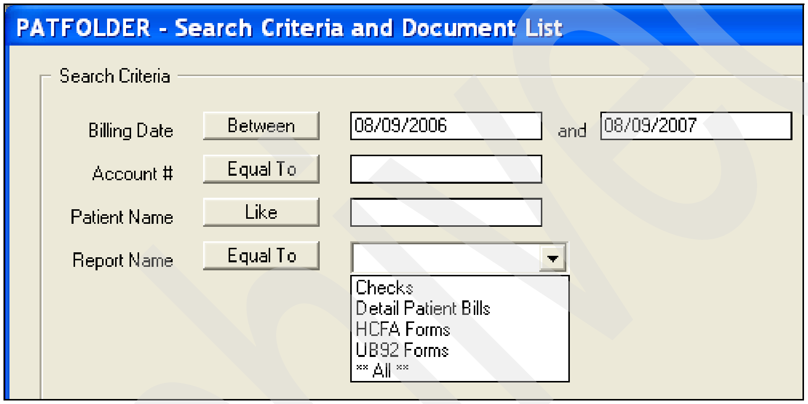
What about the System Log? The default is that it uses the Cache storage set, where the data remains for 10,000 days. You can change the System Log application group to use a migration policy you create. We suggest that you create a policy specifically for the System Log. Be sure to use the default of aggregation to group together all the small documents.

Remember that you cannot change the storage group that an application group uses, but you *can* change the Life of Data and Indexes. So if your company needs to change how long they keep a particular report, that is easy to do.

3.5.4 Application group, application, and folder

You should now have a list of the reports you want to archive into OnDemand, along with what indexes to use to search for documents within these reports, and how long and where you want to keep them. The next step is how to organize the reports into applications, application groups, and folders.

If you have multiple applications that use the same indexes and have the same storage retention, then you may decide to group these applications together within a single application group. That way all the indexes are in the same database file, and a user can easily and efficiently retrieve a list of documents from different applications within a single search in OnDemand. For example, a hospital may have several different types of reports that are each divided into individual documents indexed by billing date, account number, and patient name. Create a folder called PATFOLDER with the description, Patient Information, that contains a single application group called PATINFO. The application group contains several similar applications. A user opens the folder and searches across multiple applications, instead of doing several different searches on separate folders. See Figure 3-4.



PATFOLDER - Search Criteria and Document List

Search Criteria

Billing Date Between 08/09/2006 and 08/09/2007

Account # Equal To

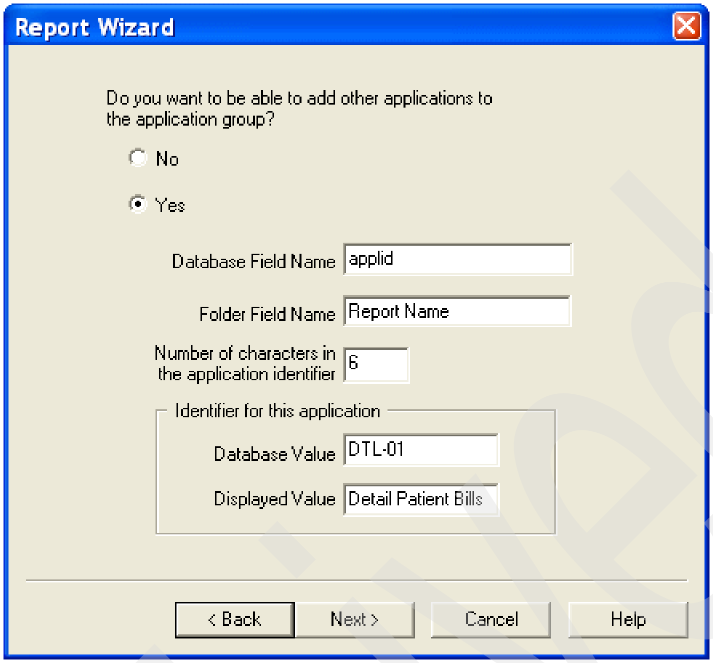
Patient Name Like

Report Name Equal To

Checks
Detail Patient Bills
HCFA Forms
UB92 Forms
** All **

Figure 3-4   PATFOLDER Search

Remember that if you want to be able to have more than one application in an application group, you must include an application ID field. You can also search on the application ID field in the folder. In this example, the internal value for the application ID for the DTLBILLS application is DTL-01, meaning version 01 of that report. The mapped field is “Detail Patient Bills” and that is what the user sees. See the example in Figure 3-5 on page 87.



Do you want to be able to add other applications to the application group?

○ No

● Yes

Database Field Name: applid

Folder Field Name: Report Name

Number of characters in the application identifier: 6

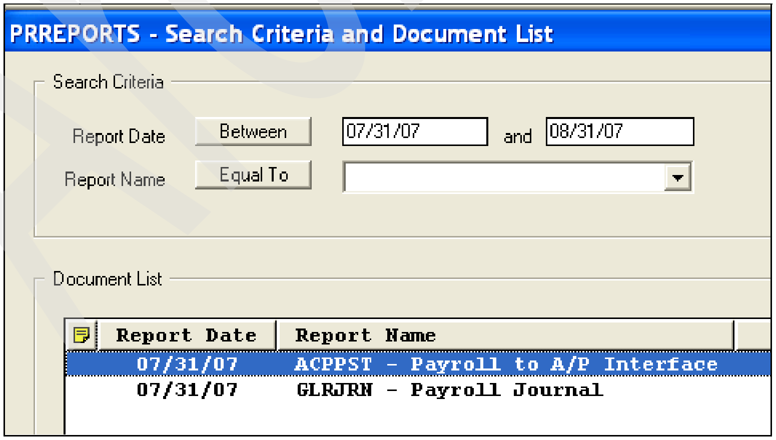
Identifier for this application

Database Value: DTL-01

Displayed Value: Detail Patient Bills

Figure 3-5 Report wizard - Application ID

Some businesses have reports that do not need to be divided into separate documents. For example, you have several monthly payroll reports that you want to access in a single folder. Users open the PRREPORTS folder and search for all reports created during the last month. From the hit list, you can open each report in its entirety. See Figure 3-6.



Search Criteria

Report Date: Between 07/31/07 and 08/31/07

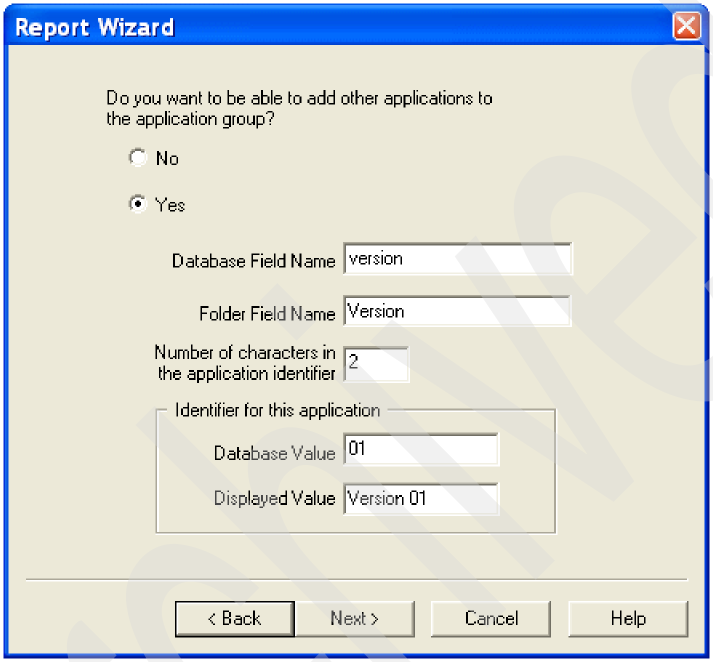
Report Name: Equal To

Document List

Report Date	Report Name
07/31/07	ACPPST - Payroll to A/P Interface
07/31/07	GLRJRN - Payroll Journal

Figure 3-6 Payroll reports: Search criteria and document list

Even if every one of your application groups contains only a single application, you still need to be sure to include an application ID field. That way you are prepared for future changes to the layout of the report. If the location of the indexes changes, you can create a new version of the application. When you use the Report Wizard to create the application group, application, and folder, be sure to answer *Yes* to the question about adding other applications to the application group, as you can see in Figure 3-7.


Report Wizard

Do you want to be able to add other applications to the application group?

○ No

● Yes

Database Field Name: version

Folder Field Name: Version

Number of characters in the application identifier: 2

Identifier for this application

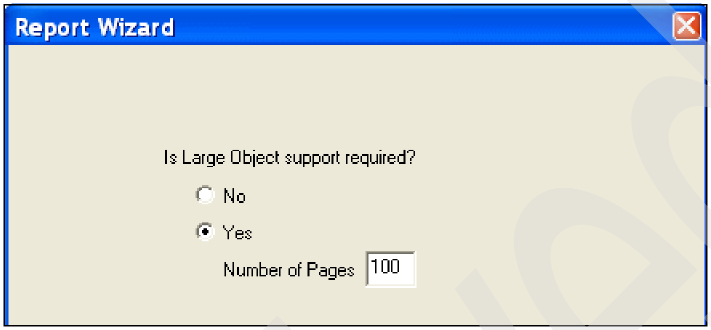
Database Value: 01

Displayed Value: Version 01

Figure 3-7 Report wizard: Application version

Naming your applications and application groups is a very important part of the design process. We always use the output queue monitor program to automate loading reports into OnDemand. When you use the STRMONOND command to monitor an output queue, each spooled file is checked to see which application and application group to use to load the report. The command looks for a match in one of the attributes of the spooled file: name, user data, form type, Job Name, user defined data, or user defined option. Typically the spooled file has a unique value for one of these attributes. For example, the accounts payable checks may have APCHECK as the user data attribute, so you name the application and application group to APCHECK. That way you do not have to change your business application software to match OnDemand. For application groups that contain multiple applications, you probably need to change at least one of the spooled file attributes so that you can match both the application group and the application names.

If you load an entire report as a single document and the report is typically more than 300 to 500 pages, or if single documents within a report are this size, we suggest that you mark the application for the report as a large object. Then when users open a document, OnDemand will not send them the entire document, which could be very slow. Instead, the users will receive the number of pages you specify, as shown in Figure 3-8.



Is Large Object support required?

○ No

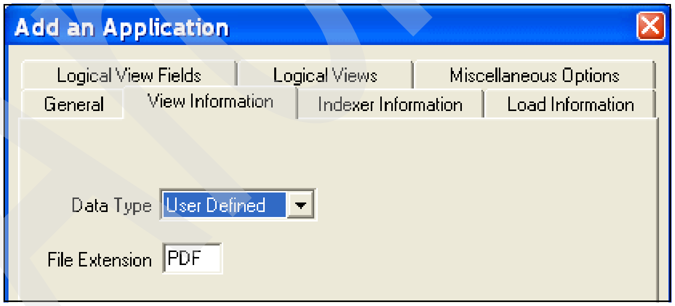
● Yes

Number of Pages 100

Figure 3-8 Report wizard: Large object

Data Type definition for viewing PDF files

Do you plan to archive PDF files? If so, do you define the application as data type *PDF* or as data type *User Defined*? If all your users have Adobe Acrobat installed on their PCs, they can view a PDF document seamlessly within the OnDemand Windows Client if you define the PDF file as data type PDF. If your users have only Acrobat Reader, then you need to define the PDF file as data type User Defined with file extension PDF, as shown in Figure 3-9. With this setting, when a user views a PDF document, Acrobat Reader is launched.



Data Type User Defined

File Extension PDF

Figure 3-9 User defined application data type

3.5.5 Security

Start security planning by asking the following questions:

- Do you want to require the users to have i5/OS user profiles, or will OnDemand manage the user IDs and passwords?
- Who will access each folder and application group?
- Can you group users together to make managing security easier?
- Do the users need to be restricted to viewing documents with certain index values?
- Who will be the OnDemand administrators?

If the users who will access OnDemand already have i5/OS user profiles, then you probably want to let them use the same ID and password. That way, when you change your password in one place (OnDemand or i5/OS), it is also changed in the other place. Remember that every OnDemand user must be added individually; you cannot just allow all users or *PUBLIC. Typically you add users through the OnDemand Administration Client. If you have many users to add, you can write a query and a Control Language (CL) program to automatically add selected users into OnDemand. You can find a program sample in the publication *OnDemand for i5/OS News and Tips from the 2005 Bulletins*. Go to the OnDemand Support Web site and search the knowledge base for bulletins:

http://www.ibm.com/software/data/ondemand/400/support

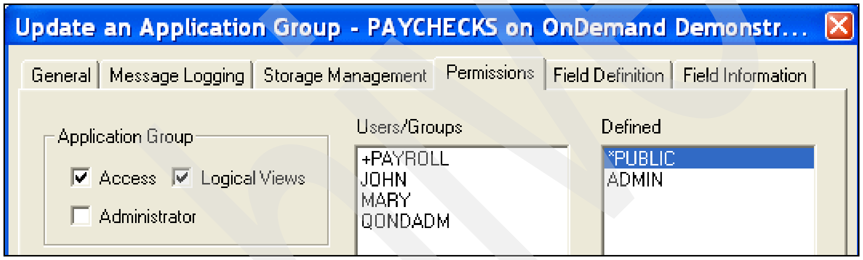
In some companies, OnDemand users are external customers who access OnDemand documents from the internet. These companies do not want to create an i5/OS user profile for each OnDemand user, so keep the security separate. Or in a browser environment, you can manage security checking in your own programs and then use the same user ID when you pass the request to OnDemand. In that case, either OnDemand security or i5/OS security would work. Even if all your users are internal employees, if they do not already have users IDs on the System i server, you may prefer to add them only into OnDemand and not create i5/OS user profiles.

The default environment is that every OnDemand user must also have an i5/OS user profile and password. To change this default, edit the IFS stream file /QIBM/UserData/OnDemand/CONFIG/ARS.INI. Change the value for SRVR_FLAGS_SECURITY_EXIT to 0 if you do not want to require i5/OS user profiles for OnDemand users.

An OnDemand administrator who will be loading reports into OnDemand will need an i5/OS user ID, as we described earlier in 3.4, “Installation and configuration” on page 77.

The next two questions can be considered together: Who needs to access which reports, and can these users be grouped together by function? The easiest way to set up security is to let every user have access to every report. Grant *PUBLIC access to every application group and folder, but maybe you need more security.

If you do not need to use query restrictions to limit users to specific documents within reports, then we recommend that you give Access or Logical Views permission to *PUBLIC for all application groups, then grant permissions to folders. With this method, you only need to manage permissions in folders. For example, you add 20 payroll reports to OnDemand, and only the payroll department should see these reports, but each user should be able to view all documents within each report. Create a user group called PAYROLL and add individual users to that group. Give Logical Views permission to *PUBLIC for each payroll application group, as shown in Figure 3-10. Then in each payroll folder, give Access permission only to the PAYROLL group, so only those users can access the documents. See Figure 3-11 on page 92.


Update an Application Group - PAYCHECKS on OnDemand Demonstr...

Application Group

☑ Access ☑ Logical Views

□ Administrator

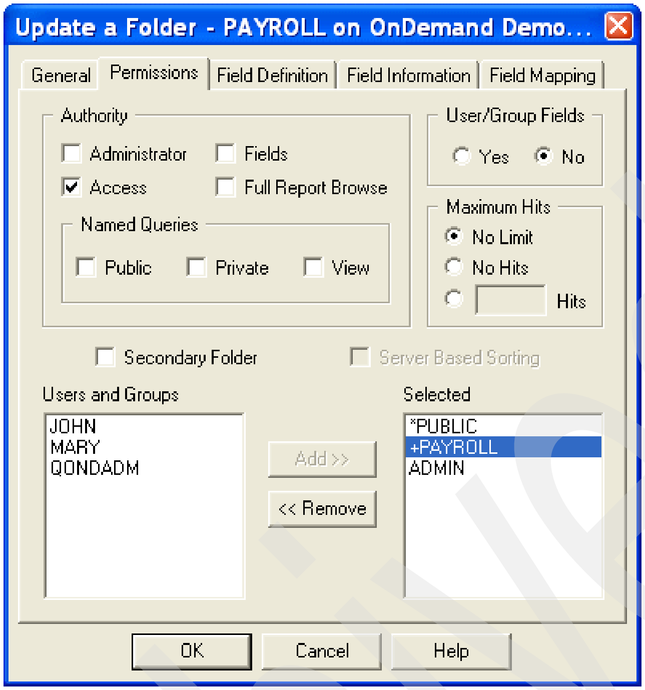
Users/Groups

+PAYROLL
JOHN
MARY
QONDADM

Defined

*PUBLIC
ADMIN

Figure 3-10 App group public permissions



Update a Folder - PAYROLL on OnDemand Demo...

General | Permissions | Field Definition | Field Information | Field Mapping

Authority
- ☐ Administrator
- ☐ Fields
- ☑ Access
- ☐ Full Report Browse

Named Queries
- ☐ Public
- ☐ Private
- ☐ View

User/Group Fields
- ○ Yes
- ● No

Maximum Hits
- ● No Limit
- ○ No Hits
- ○ Hits

☐ Secondary Folder

☐ Server Based Sorting

Users and Groups
- JOHN
- MARY
- QONDADM

Add >>

<< Remove

Selected
- *PUBLIC
- +PAYROLL
- ADMIN

OK | Cancel | Help

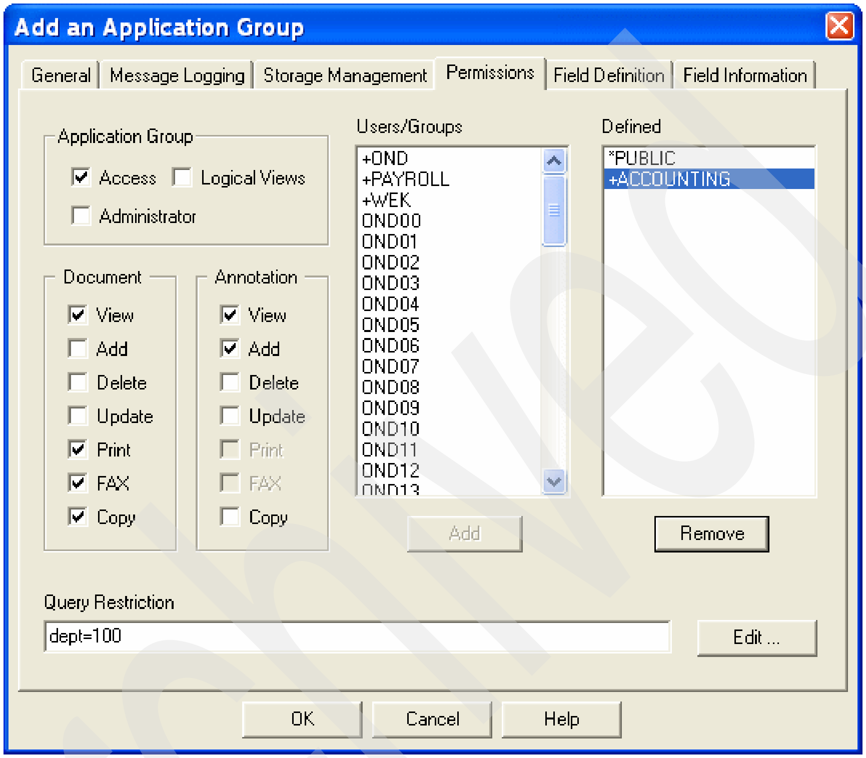
Figure 3-11 Folder group permissions

Logical Views permission within an application group allows users to make some changes to the appearance of a document they are viewing (for example, background color or zoom) and save those changes. If you do not want the users to save these changes, give them Access permission instead of Logical Views permission.

When a new person is hired into the payroll department, you can add the user ID into OnDemand and then just add the user to the PAYROLL user group. The user is automatically authorized to all the folders used by the PAYROLL group.

There is no relationship between user groups in OnDemand and i5/OS group profiles. Even if an OnDemand group has the same name as an i5/OS group user profile, the members of the group are independent from the group user profile.

You may need to restrict users or user groups so that they can access only certain documents within an application group. You can provide this level of security for your reports when you use query restrictions to limit access to specific index values. For example, you may have a report that OnDemand divides into separate documents based on a change in the value of department number (dept), which is one of the search fields. The accounting department can only access documents with department number = 100 and the purchasing department can only access documents with department number = 450. To satisfy this requirement, you can give Access permission to all user groups at the folder level, then specify restrictions on the application group, See Figure 3-12.



Application Group

☑ Access ☐ Logical Views

☐ Administrator

Document

☑ View
☐ Add
☐ Delete
☐ Update
☑ Print
☑ FAX
☑ Copy

Annotation

☑ View
☑ Add
☐ Delete
☐ Update
☐ Print
☐ FAX
☐ Copy

Users/Groups

+OND
+PAYROLL
+WEK
OND00
OND01
OND02
OND03
OND04
OND05
OND06
OND07
OND08
OND09
OND10
OND11
OND12
OND13

Defined

*PUBLIC
+ACCOUNTING

Query Restriction

dept=100

Figure 3-12 Query Restrictions setup

Who will be OnDemand administrators on your system? Many customers have only Users and System Administrators. If you have a lot of users, you may prefer to assign user administration tasks to one or more of the users so they can authorize other users to use OnDemand. Or you may have one or more representatives from each department who may be responsible for creating application groups and folders.

There are four types of users that can be defined in OnDemand:

User Users log in and query the system to retrieve documents and reports.

User administrator User administrators can add users or other user administrators to the system.

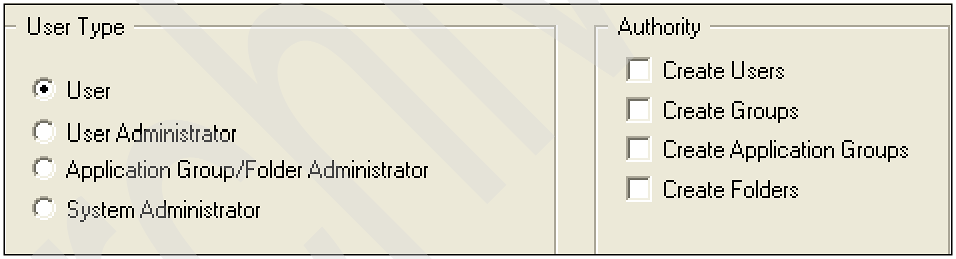
Report administrator

Report administrators define the application groups, applications, and folders in an OnDemand system. They are responsible for knowing the report and document data and for defining the indexes that are extracted from the data. Report administrators are also responsible for designing the user interface to the reports through the folder definition process and for controlling access permissions to the reports that they administer.

System administrator

System administrators have the highest level of authority in an OnDemand system. They have permissions for all system functions and can grant other users the permissions to perform various tasks. The system administrator is the only type of user who can create storage sets, define system printers, and give authority to create groups.

You can assign different user types based on job function within OnDemand, as shown in Figure 3-13.



User Type

● User

○ User Administrator

○ Application Group/Folder Administrator

○ System Administrator

Authority

□ Create Users

□ Create Groups

□ Create Application Groups

□ Create Folders

Figure 3-13 User types

We suggest one way of assigning authorities and permissions for the four user types, as shown in Table 3-1.

Table 3-1 Authority settings by user type

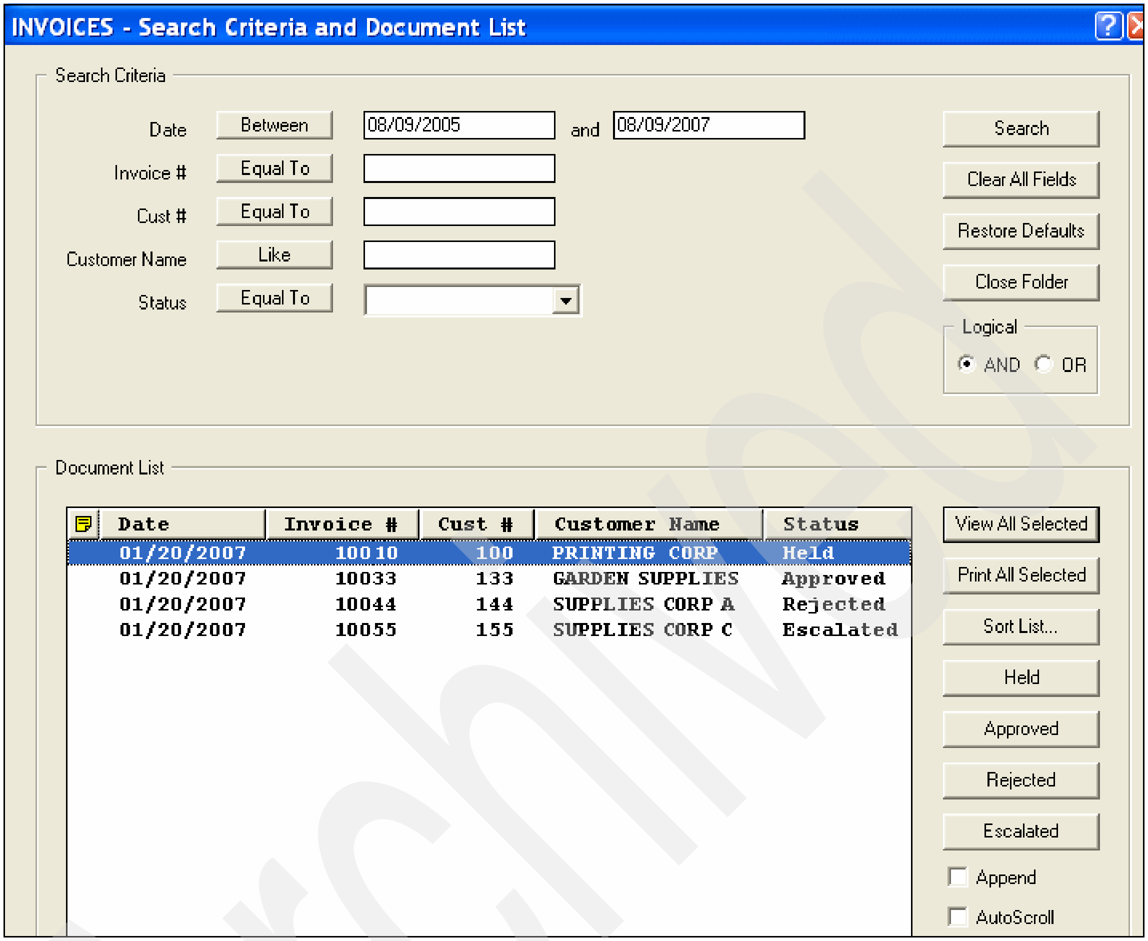
	System administrator	**Report administrator**	**User administrator**	**User**
User Authorities				
- Create Users	*		*	
- Create Groups	*		Y	
- Create Application Groups	*	*		
- Create Folders	*	*		

	System administrator	Report administrator	User administrator	User
User Permissions				
- Administrator	*		Y	
- Permissions for individual users	*			
Groups				
- Permissions for individual groups	*		O	

* Has authority and permissions because of user type.
Y Give authority/permission to this user type.
O Optional: Can be specified as needed to achieve desired security.

3.5.6 Document Audit Facility

Document Audit Facility (DAF) is a feature of OnDemand that lets users do basic approval routing through the OnDemand Windows Client. An authorized user can change a status or audit field in the report, which actually changes the value of a database field in the application group. You may want to allow certain users to click a button in the Client to approve vendor invoices, and allow other users to view only those approved invoices. See Figure 3-14 on page 96.



Date	Between	08/09/2005	and 08/09/2007
Invoice #	Equal To		
Cust #	Equal To		
Customer Name	Like		
Status	Equal To		

Date	Invoice #	Cust #	Customer Name	Status
01/20/2007	10010	100	PRINTING CORP	Held
01/20/2007	10033	133	GARDEN SUPPLIES	Approved
01/20/2007	10044	144	SUPPLIES CORP A	Rejected
01/20/2007	10055	155	SUPPLIES CORP C	Escalated

Figure 3-14 DAF auditor view

You may not need to use DAF with any of your reports, but if you do, it is better to plan ahead. You will need to create a field in the application group and in the folder to use as the status or audit field that can be modified. Beginning with server level 7.1.2.8, you can add fields to existing application groups, but not to folders. We recommend deciding whether you need to use this feature before setting up a definition for a report. To learn more about Document Audit Facility, refer to Chapter 15, “Did You Know?”, in *Content Manager OnDemand Guide*, SG24-6915.

3.5.7 Indexers

There are three different indexers that you can use with OnDemand for i5/OS: OS/400, generic, and PDF. The OS/400 indexer is the most common because you use it for all your spooled files. When you use the graphical indexer tool to select the fields you want to use in an application, OnDemand creates a list of parameters that the OS/400 indexer program reads and processes. All three OnDemand Common Server platforms (System i, z/OS, and Multiplatforms) use the same Administration Client, so the indexer parameters you create are in a similar format in all environments. They are referred to as ACIF parameters because they were designed to be used by the ACIF indexer, which is used by the other platforms. ACIF is the abbreviation for AFP (Advanced Function Presentation) Conversion and Indexing Facility.

Use the generic indexer for scanned documents and PC files. You can use Kofax Ascent Capture to scan, index, and release documents to an Integrated File System (IFS) directory on your System i server. Submit the qshell command **`arsload`** to monitor the directory, or use the command STRMONOND. For details on how to set up a scanning environment, refer to the *Kofax Ascent Capture Release Script Guide*, SC09-7602. You can find the document listed under “Other valuable resources” on the OnDemand support Web site:

`http://www.ibm.com/software/data/ondemand/400/support.html`

You can use the OnDemand Toolbox to index and archive PC files such as spreadsheets and documents. To discover more about this software, refer to Chapter 15, “Did You Know?”, in *Content Manager OnDemand Guide*, SG24-6915.

If you want to load a PDF file as a single document, you can use the OnDemand Toolbox or Kofax Ascent Capture. If you need to divide the PDF file into multiple documents based on a change in an index value, use the PDF Indexer so that you can graphically mark the locations of the indexes in the file. As we discussed in “Data Type definition for viewing PDF files” on page 89, you can define a PDF application as data type PDF or User Defined. In either case, you can use the PDF Indexer to separate the file into individually retrievable documents and extract index fields.

You can use the Report Wizard for files using the OS/400 or PDF Indexer (if the data type is PDF). For scanned documents and PC files (including PDF files you set up as User Defined data type), you need to create application groups, applications, and folders in separate steps.

3.5.8 Exit points

Exit points are rarely needed by OnDemand for i5/OS users. Most sample exit programs are not written for System i customers and the sample programs would not work on a System i. However, we discuss two situations where you might want to use exit programs: an output queue monitor exit program and a postprocessor program.

Output queue monitor exit program

If you need to change the name of an application or application group that is used to archive a spooled file, you might want to use an output queue monitor exit program. The monitor program calls the exit program if its name matches the name of an application or application group. The program passes parameters such as instance name, spooled file attributes, and job attributes, and you can use this information to decide which application and application group to use for this spooled file.

For example, a bank may have many branches and each branch has its own application within an application group. The application group is named STMTS, and the applications include a 3-character abbreviation for the branch, such as RALSTMTS, APXSTMTS, and CARSTMTS. The Job Name that creates the spooled file is always STMTS and the User Data attribute is the name of the branch (such as RALEIGH, APEX, and CARY). Write an exit program called STMTS that changes the application name based on the values for Job Name and the user data. Be sure that you compile the program into a library that is in the library list of the output queue monitor job.

You can also write and use an Integrated Language Environment® (ILE) output queue monitor exit program if you create a data area called QRLMMONQ in either the QUSRRDARS library or your OnDemand instance library. If you create the data area in QUSRRDARS, all ILE monitor exits for all instances will be called if they are found in the monitor job library list. If you create the data area in your instance library, only ILE monitor exits for that instance will be called if they are found in the monitor job library list. Create the data area using this command:

```
CRTDTAARA DTAARA(library-name/QRLMMONQ) TYPE(*CHAR) LEN(100)
```

A sample output queue monitor exit program (PGM123) is in the QSAMPLES2 source file in library QUSRRDARS.

Postprocessor program

If you want to modify the index records before OnDemand loads them into the database, you might want to use a postprocessor program. As an example, use Stream EDitor (SED) to write a UNIX shell script for a report that use account number 99999 to indicate the totals page, with no account name. You can search for 99999 and replace six contiguous blanks in the same record with the character string TOTALS:

```
/99999/s/      /TOTALS/
```

Include this line in a SED script and refer to it in the postprocessor parameters in the application.

Look in the QSAMPLES2 source file in library QUSRRDARS for samples of postprocessor programs in ILE COBOL, RPG, and C. You can also find information about postprocessors by searching on the OnDemand support Web site.

In many cases, you can make simple changes to index values without any special programming. Go to the Load information tab of an application to remove special characters from index values before they are loaded into OnDemand. For example, you can remove embedded dashes from a social security number or remove trailing spaces from a string field you have defined as a variable length field.

3.6 Application setup and verification

After you have determined which reports to archive, how you want to organize them, and who will use them, you can define the reports to OnDemand and follow a process to put them into production for the users.

In this section, we discuss the following application setup and verification tasks:

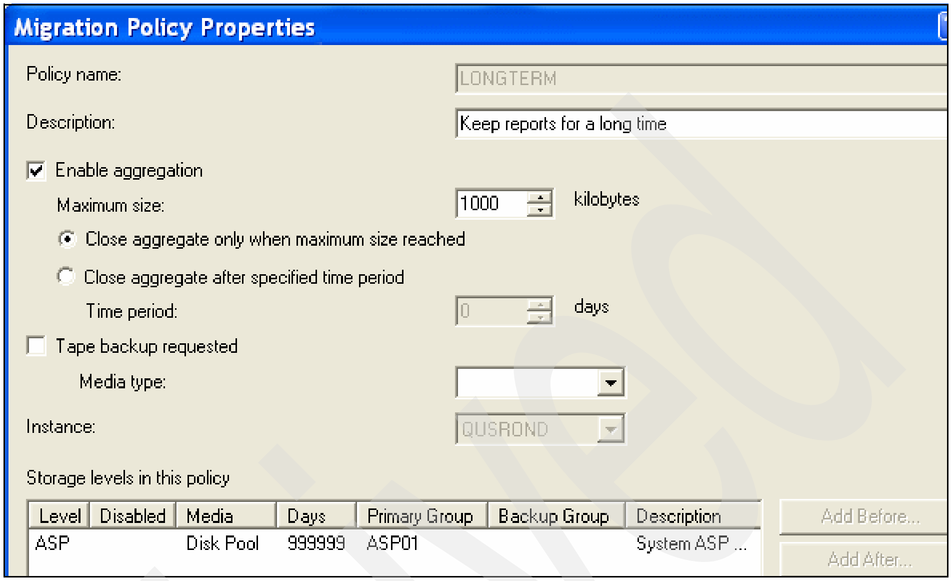
- Report setup
- Permissions setup
- Load, retrieve, and print verification
- User acceptance

3.6.1 Report setup

You can define a report to OnDemand by using the Report Wizard or by creating an application group, an application, and a folder as separate steps. We almost always use the Report Wizard to set up new reports.

As an example for a report setup, let us assume that you want to start archiving your accounts payable checks. You have determined that the checks should be kept in System i disk storage for 10 years, and they should be accessed only by the accounting group. You have already created a migration policy called

LONGTERM that includes a single storage level, which is ASP01, the system Auxiliary Storage Pool. See Figure 3-15.



Migration Policy Properties

Policy name: LONGTERM

Description: Keep reports for a long time

☑ Enable aggregation

Maximum size: 1000 kilobytes

● Close aggregate only when maximum size reached

○ Close aggregate after specified time period

Time period: 0 days

□ Tape backup requested

Media type:

Instance: QUSROND

Storage levels in this policy

Level	Disabled	Media	Days	Primary Group	Backup Group	Description
ASP		Disk Pool	999999	ASP01		System ASP ...

Figure 3-15 Migration policy properties

Locate a sample spooled file of the accounts payable checks. Talk with the user who created the spooled file to discover if you can hold the file so you can archive it into OnDemand, or if you can change the spooled file attributes to SAVE(*YES) and work with the file after it is printed. The best approach is to get sample spooled files and put them in an output queue you create just for testing with OnDemand.

Here are some steps you can follow to set up the checks in OnDemand:

1. Use the Report Wizard in the OnDemand Administration Client to create an application group, application, and folder. Follow these guidelines:

   – The first time you use the Report Wizard and select a sample spooled file, choose the location for your sample files.

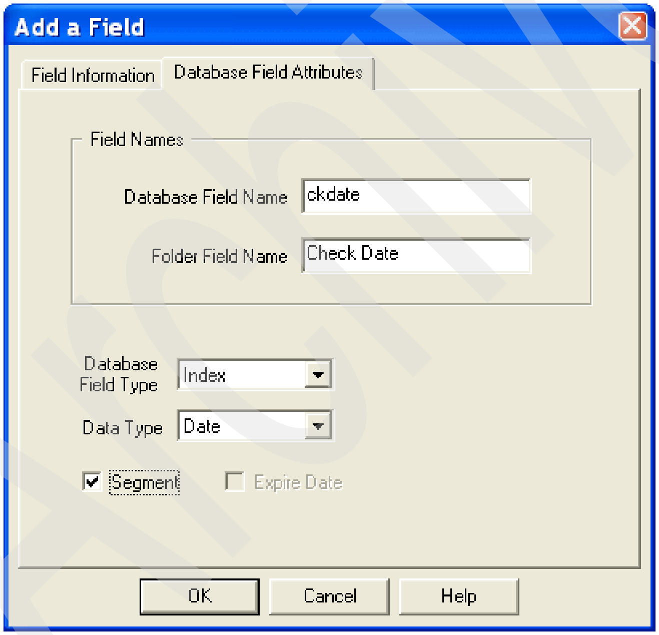
     The default directory to hold these files is C:\Program Files\IBM\OnDemand32. We recommend creating your own directory, for example, C:\Documents and Settings\Callen\My Documents\OnDemand Reports. If you ever need to un-install and re-install the OnDemand Administration Client, this directory will not be affected. After you download a spooled file to your new directory, this directory becomes the default for your sample files.

> **Note:** The folder fields will be added in the order that you add them in the Wizard. You can always change the order of the fields in the search and hit lists later, but it is easier to add the fields in the order you prefer when you create them. We like to add a date field first in all reports, so the folder views are consistent.

– Create an application ID field so that you can add other applications to the application group or keep track of different versions of the same application.

  For example, if the location of the index fields in the spooled file changes, you need a new version of the application and, therefore, an application ID field. If the accounts payable checks is the only application within the application group, you can use *version* as the name of the field.

– Be sure to make one of your fields a segment date field, as shown in Figure 3-16.



Add a Field

Field Information | Database Field Attributes

Field Names

Database Field Name: ckdate

Folder Field Name: Check Date

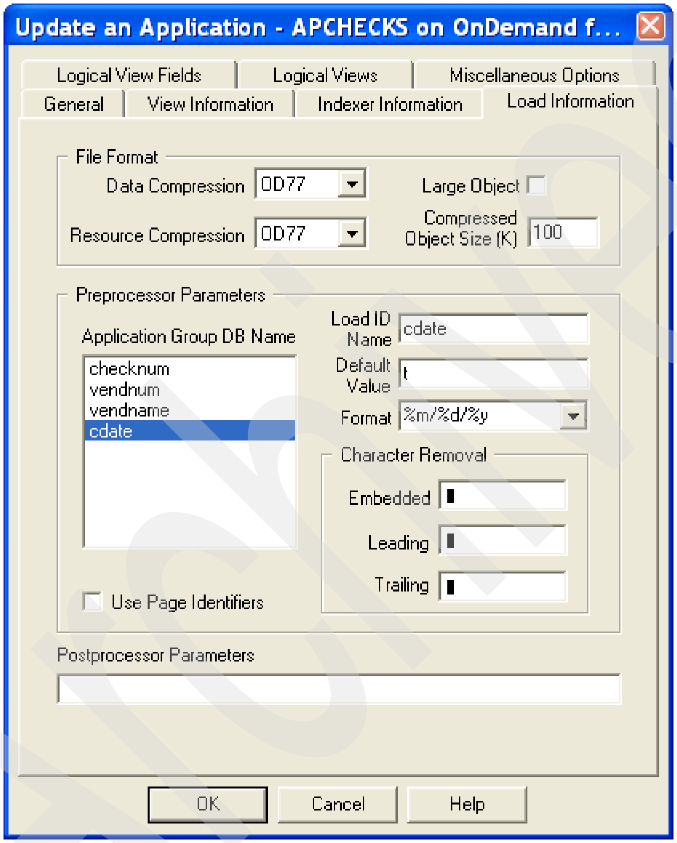
Database Field Type: Index

Data Type: Date

☑ Segment ☐ Expire Date

Figure 3-16 Segment date field

If there is no date field on the report, then use today's date (the date the report is loaded) so the users can search for a document by a date range, which improves performance. In this case, select any character string as a date, just so the Wizard creates a date field in the application group. Then after you finish using the Wizard, update the application. Remove the index and field parameters for the date in the indexer parameters and insert a default value of *t* in the Load Information tab as shown in Figure 3-17.


Update an Application - APCHECKS on OnDemand f...

File Format

Data Compression: OD77
Resource Compression: OD77
Large Object: □
Compressed Object Size (K): 100

Preprocessor Parameters

Application Group DB Name:
checknum
vendnum
vendname
cdate

Load ID Name: cdate
Default Value: t
Format: %m/%d/%y

Character Removal
Embedded:
Leading:
Trailing:

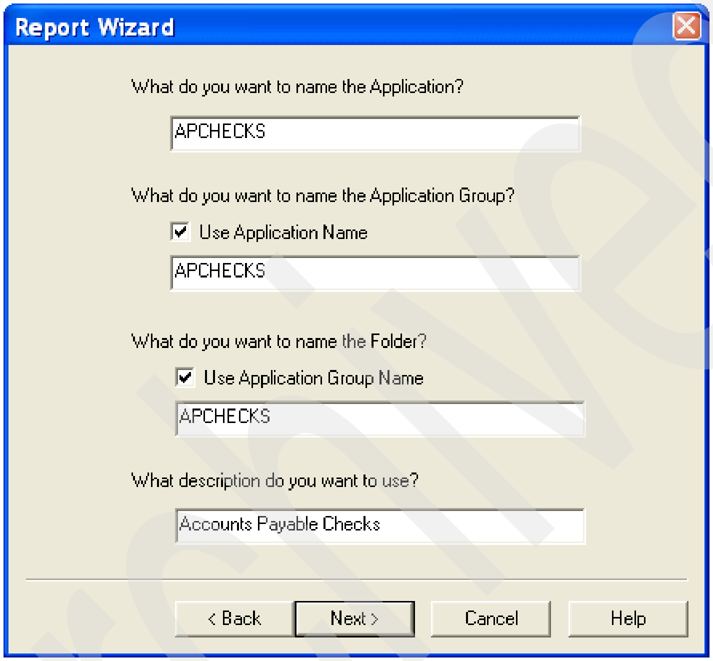
□ Use Page Identifiers

Postprocessor Parameters

Figure 3-17 Today's date in application

– Use capital letters and no spaces for the names of the application group and application.

When you automate report loading, you use an output queue monitor to archive spooled files. The monitor looks for a match in one of the spooled file attributes, so it knows which application and application group to use for that file. Most spooled file attributes are upper case, and they need to match the OnDemand names, which are case sensitive. For example, the accounts payable checks may use a printer file (spooled file name) named APCHECKS, or have APCHECKS as the user data attribute. If this is the only application in the application group, you can use APCHECKS as the name for both application and application group. You do not need to change the spooled file attributes. If the folder contains only one application group, you may choose to name the folder APCHECKS also. The user sees the short folder name with a long description. See Figure 3-18.



Report Wizard

What do you want to name the Application?

APCHECKS

What do you want to name the Application Group?

☑ Use Application Name

APCHECKS

What do you want to name the Folder?

☑ Use Application Group Name

APCHECKS

What description do you want to use?

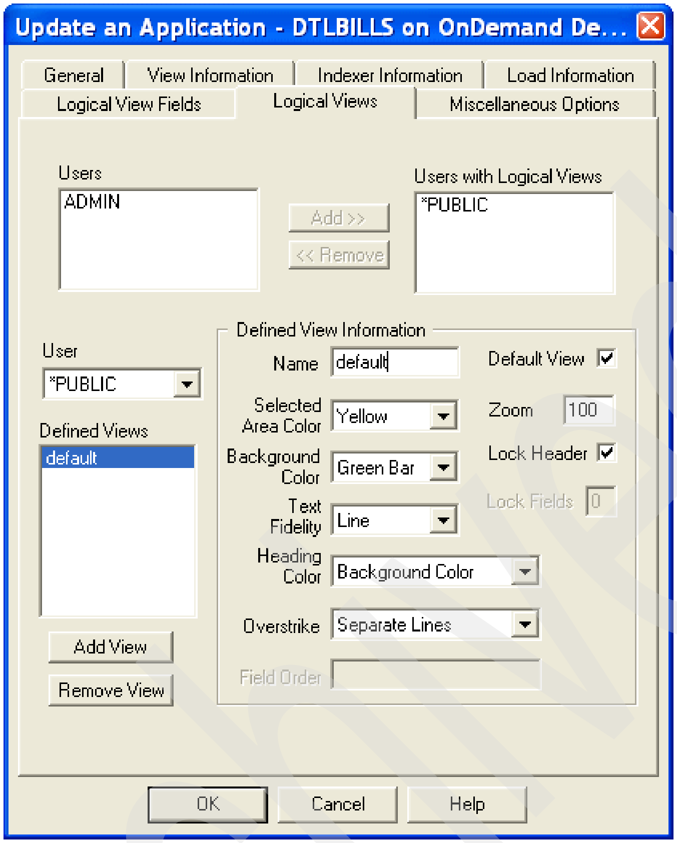
Accounts Payable Checks

Figure 3-18 Wizard name objects

2. Update the application group that you just created with the Wizard. Modify the storage management if necessary. The defaults are 90 days in Cache and a value of 2555 days (seven years) for Life of Data and Indexes. We suggest zero days in Cache if the first storage level is a disk pool. If optical storage is the first level, you may want to keep the data in Cache for a while for faster access. Change the Life of Data and Indexes if you want to keep the reports and indexes for more than or less than the default of seven years.

Note: The default value of 2555 days represents seven years of 365 days each. If you have legal requirements to keep your data for seven years, a value of 2557 days guarantees that your data does not expire before seven years, which allows you to handle the situation where there are two leap years during the seven years.

3. Modify the permissions in the application group to meet your requirements. If you are not sure who should access this report, you can leave the default of no access to anyone except administrators.
4. Update the date format in the application if necessary. The Wizard can recognize a date in the format mm/dd/yy (with or without separator characters), but if your date is in a different format, select the correct one on the Load Information tab.
5. Create a logical view for the application if desired. This step is optional; you can do this now or after reports have been loaded, or not at all. Some customers like to create a simple logical view just to lock the headings and change the background color to green bars for report listings. See Figure 3-19 on page 105.



Users: ADMIN

Users with Logical Views: *PUBLIC

User: *PUBLIC

Defined Views: default

Name: default

Default View ☑

Selected Area Color: Yellow

Zoom: 100

Background Color: Green Bar

Lock Header ☑

Text Fidelity: Line

Lock Fields: 0

Heading Color: Background Color

Overstrike: Separate Lines

Field Order:

Figure 3-19 Application logical view

You can also add logical view fields so that a user can move or hold columns in a report and do more complex searches within a report. Mark the page header, field header, and validation string from the Logical View Fields tab, then add a default view from the Logical Views tab, as you can see in Figure 3-19. You can read more about logical views in the *Quickstart Guide*, which is listed under Related Resources at the support Web site:

http://www.ibm.com/software/data/ondemand/400/support.html

You can find more information in the white paper *Create customized views for line data reports*. You can find this publication at the Web site:

http://www.ibm.com/developerworks/db2/library/techarticle/dm-0607wagner/

6. Update the folder if desired. You can change a folder field name, but not the data type. You can also change the default date search range (the default is the last 60 days), change the order of fields in the query and hit lists, grant permissions, and other tasks. Or you can leave the defaults as they are, and change the folder later if necessary when you work with the users.

What if you need to add a new application to an application group you have already created? First add a value for the application ID field in the application group. Then, from the Administration Client, create a new application in that application group.

What if you need to create three application groups and join all of them in a single folder? It still may be easier for you to use the Report Wizard to create each of them, then delete the two folders you do not need. Add the other two application groups to the remaining folder and map their fields to the folder fields.

3.6.2 Permissions setup

You can grant permissions when you set up the application groups or go back later and authorize users. Do not grant permissions to a folder until you are ready for the users to access that folder, which should be after you have tested retrieving documents from that folder. When users log on to OnDemand, they see a list of folders they are allowed to access. Users are authorized to application groups and folders as individual users, as members of user groups, or as part of *PUBLIC.

If you have several different applications within an application group, they are all included in the folder for the application group. If you need to restrict user access to some of the applications, you can create a separate folder that contains the application group but does not include all the applications. Be sure then not to give those users access to the folder that contains all the applications for the application group.

3.6.3 Load, retrieve, and print verification

After you set up a report definition (application group, application, and folder), the next step is to load and retrieve a sample report. Find a sample spooled file and load it to OnDemand with the ADDRPTOND command. You must know the spooled file name and number, the Job Name, user, and job number. If you are not using the default instance QUSROND, press F10 and page down so you can see additional parameters for the command and type in your instance name. For example:

```
ADDRPTOND APPGRP(APCHECKS) APP(APCHECKS) SPLF(QSYSPRT)
   JOB(305410/CALLEN/CHECKS) SPLNBR(3) INSTANCE(ONDTEST)
```

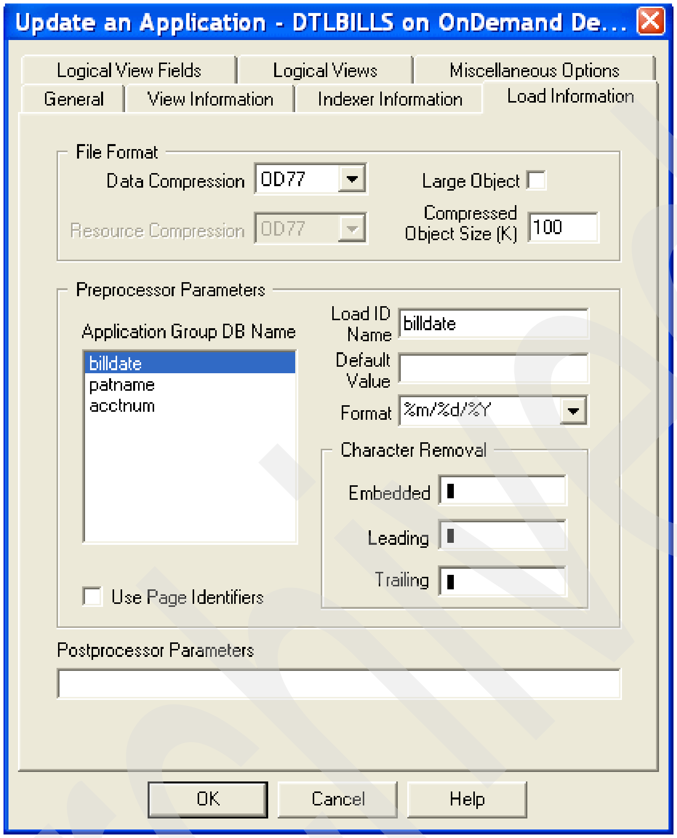
The spooled file will remain in its output queue and you will get a message that the job ended normally or abnormally.

If the spooled file does not need to remain in its output queue, you may find it easier to create a test output queue and start an OnDemand monitor program over it. That way you can move spooled files to that queue whenever you want to test archiving them. As an example, create an output queue called ONDTEST in library QUSRSYS. In the STRMONOND command, match the application group and application names to the appropriate parameters. If you plan to have a single application for each application group, and the name will be found in the User Data spooled file attribute, you can start a monitor with the command:

```
STRMONOND OUTQ(ONDTEST) APPGRPSRC(*USERDATA)
```

You can see the MONOUTQ job running in the QSYSWRK subsystem. The monitor program uses the default instance QUSROND unless you press F10 when you issue the command and change the instance parameter. Also, the default error and processed output queues will be used unless you change them. If a spooled file loads successfully, it will be moved to the ONDPROC output queue in library QUSRRDARS and there will be a message in the System Log (message number 87). If the report fails to load, it will be moved to the ONDERR output queue in library QUSRRDARS. If it fails because there is no match for the application group or application name, the spooled file will be in RDY (ready) status and there will be no entry in the System Log. If it fails because of a problem with the application definition, it will be in HLD (held) status and there will be a message in the System Log (message number 88). In our experience, the most common error message is that the date format is incorrect. If you use the Report Wizard and the segment date field is in mm/dd/yy format (with or without separator characters), the format will be automatically inserted into the format field on the Load Information panel for the application. But if the date format is not recognized, you must insert the correct format manually.

For example, if the date is in the format mm/dd/yyyy, select the format %m/%d/%Y as shown in Figure 3-20.



Update an Application - DTLBILLS on OnDemand De...

File Format

Data Compression: OD77

Large Object: □

Resource Compression: OD77

Compressed Object Size (K): 100

Preprocessor Parameters

Application Group DB Name: billdate, patname, acctnum (billdate selected)

Load ID Name: billdate

Default Value:

Format: %m/%d/%Y

Character Removal

Embedded:

Leading:

Trailing:

□ Use Page Identifiers

Postprocessor Parameters:

Figure 3-20 Application date format

If you view the message in the System Log and still do not know why the report failed to load, look at the .ind file that OnDemand created. From a 5250 session, issue the command WRKLNK / and page down in the root directory until you see the work files: an .ind file that contains the index values the indexer program found, an .out file that contains the compressed data file, and an .res file that contains AFP resources if the application is data type AFP. If you have created a home directory for your user profile, you will find the work files in that directory. Sometimes it is helpful to look at the .ind file to see the index values (if any) that OnDemand was able to extract from your sample spooled file. You may find that the locations of the triggers and fields change on some of the pages at the end of the report, and you did not see these pages when you worked with the Report Wizard.

After you have successfully loaded your sample spooled file, log on to the OnDemand Windows Client to make sure that you can find and view documents in the report. Is the date search range what you want? If not, update the folder and change the default. Are the names of the fields satisfactory? If not, change those names in the folder. Is the order of fields in the search and hit lists what you want? If not, change the order in the folder. After you make a change to a folder, press F5 to refresh the “Open a Folder” display in OnDemand.

Maybe the file loaded successfully, but it did not archive the way you expected. For example, the customer name field sometimes contains blanks. You look at the spooled file again and determine that this is because you forgot to change the record range values for the group trigger that is used to find that field. You know how to change the application, but you need to delete the archived report so you will not have bad data stored. Look in the System Log for message number 87 for that file to get the Load ID. Or, from the hit list in the OnDemand Windows Client, right-click a document to see its properties. You will see a partial Load ID, which you can highlight and copy. Issue the RMVRPTOND command and enter the Load ID as the Report ID, or the partial Load ID followed by -0-0, as shown here:

```
RMVRPTOND APPGRP(APCHECKS) RPTID('5025-3-0-230FAA-0-0')
```

If the report archived successfully and can be easily retrieved, congratulations! Now you can show the users how they can use OnDemand to access their reports.

3.6.4 User acceptance

If you work in an informal environment, you may be able to show the users how to access OnDemand to see their reports and then install the OnDemand Windows Client on their PCs. But your company may have more formal user acceptance procedures, and designated users may need to sign off on their satisfaction with the OnDemand setup.

You may want to install the OnDemand Windows Client on the users’ PCs, create a User’s Guide for them to follow, and let them work with OnDemand for a day or two. Then meet with them again to discover if they would like for you to make some changes to their reports. Remember, they are the people who will use the reports, and you want them to be satisfied.

If the users will access OnDemand through a browser, you may need to install and configure software on their PCs. Refer to the OnDemand for i5/OS support Web site for more information about software required for browser access. Then show them how to access their reports using the browser.

3.7 Functional testing

In this section, we discuss functional testing for the following areas:

- Automatically loading reports
- Storage management
- Start up and shut down processing
- Retrieval processing
- Printing
- Custom scripts
- Backup and recovery

3.7.1 Automatically loading reports

Create output queues that you will monitor to select spooled files to load into OnDemand. Test using the STRMONOND command on each output queue to make sure that you can match spooled file attributes with application and application group names, load the files into OnDemand, and delete the files or send them to another output queue for printing. If you are loading scanned documents or PC files, test them using the **`arsload`** or STRMONOND commands to monitor the appropriate IFS directories and load the files into OnDemand.

3.7.2 Storage management

First, look at your migration policies and your application groups to see what should happen and when. Then issue the WRKLNK command to see the files in the IFS directory /QIBM/UserData/OnDemand/*instancename*. Do you have files in the CACHE directory? In the ASMREQUEST directory? Run the command STRDSMOND, which calls the disk storage manager program. If you take the command default, the Archive Storage Manager (ASM) program will run when DSM completes.

If you specify aggregation in your migration policies (which we recommend in most cases), you will see files in the ASMAGGREGATION directory as they are being aggregated, and files in the storage level after objects are aggregated. For example, if you have a storage level that uses the System ASP as a disk pool, you will see objects in the ASMASP01/PRIMARY directory that OnDemand will create the first time it is needed. If you are using optical storage and you have initialized and added optical volumes to a storage group, you should see some objects written there. Issue the command GO OPTICAL and take option 1 to Work with Optical Volumes and see if the volumes are being used. If you are not using aggregation, ASM will move the files from the ASMREQUEST directory to the storage level in the policy.

OnDemand uses the segment date in a report to determine when the report should be expired. It uses the date the report was loaded to determine when to move it to storage levels within a migration policy.

Go to output queue QRDARS400 in library QRDARS to find the ASM status reports. Look in the System Log to see information about cache expiration.

If you are using TSM as a storage manager for OnDemand, log on to the TSM Administrative Client Command Line and use the query occupancy command to verify that objects are being written to the TSM server. For more details and examples of this command, refer to the OnDemand TSM documentation on the OnDemand support Web site.

3.7.3 Start up and shut down processing

Start the server with the command STRTCPSVR *ONDMD. You should see the name of the instance (for example, QUSROND) as the name of a job running in the QSYSWRK subsystem. Stop the server with the command ENDTCPSVR *ONDMD. These commands will work if you have the parameter ARS_AUTOSTART_INSTANCE=1 in the /QIBM/UserData/OnDemand/*instancename*/ars.cfg file. If the value is set to 0, you can start and end the server with the following commands:

```
CALL QRDARS/QRLMCTL *STRTCPSVRinstancename
```

or

```
CALL QRDARS/QRLMCTL *ENDTCPSVRinstancename
```

3.7.4 Retrieval processing

Log on to the OnDemand Windows Client or browser interface and make sure that you can find documents within reports. Test security and query permissions.

3.7.5 Printing

Test printing documents to local printers and to server printers. Use the Administration Client to add and authorize server printers.

3.7.6 Custom scripts

If you use any monitor output exit programs or postprocessor programs, be sure to test them. If you use Kofax Ascent Capture to scan documents and release them to an IFS directory that is monitored by OnDemand, be sure to test the entire process.

3.7.7 Backup and recovery

Do your migration policies use disk, optical, tape, Tivoli Storage Manager (TSM)? Optical volumes and tapes need to be duplicated or backed up, and documents in TSM need to be backed up. Create and test a backup/recovery plan that is suitable for your environment. In all situations, each day you should save the QUSRRDARS library, the IFS directory /QIBM/UserData/OnDemand/*, and your instance libraries (such as QUSROND).

3.8 Performance testing and considerations

We have never worked with a System i customer who had to upgrade their system when they installed OnDemand. However, if you are already experiencing high CPU or disk utilization, you should upgrade your system before implementing any significant changes. Also, most customers do not add hundreds of new users when they implement OnDemand. Typically they give existing users access to OnDemand, so there is not much additional workload on the system.

Testing OnDemand for i5/OS is usually done informally. Users access their reports and are asked to report problems with retrieval time. In our experience, accessing information online is so superior to looking for data offline that we do not hear any complaints. But if you have procedures and guidelines for new software in your business, you may need to test the performance of OnDemand. In this section, we discuss the following performance testing and considerations:

- Develop test cases based on requirements.
- Load testing.
- Retrieval testing.
- Tuning considerations based on test results.

3.8.1 Develop test cases based on requirements

What are your concerns with OnDemand performance? Do you have several large spooled files at a specific time during the month? Do you receive big PDF files from another source? Are you migrating from another system? Will documents be retrieved in a call center where retrieval response time is critical? Think about your busiest time of the month and create test cases based on that time.

3.8.2 Load testing

You may want to generate some typical spooled files and then move them to an output queue that is monitored by OnDemand. See how long it takes to process them. Look in the System Log for entries with message number 87. These entries will show you the size of the original file, the compressed size after OnDemand loads it, the load time, and the number of index records. You can also write a program that uses qshell commands to query these System Log entries and create a report of the statistics.

3.8.3 Retrieval testing

Test logging on to OnDemand to see how long that takes. Then test retrieving both large and small documents, and open an application defined as a large object if you have any of these. Test doing a find on a character string within a document. Test opening a PDF file, a PC file, and a scanned document if you have those data types.

3.8.4 Tuning considerations based on test results

If it takes a long time to see the folder list when you log on to OnDemand, it may be because your OnDemand Windows Client connection uses a Dynamic Host Configuration Protocol (DHCP) server rather than a static address. Edit the IFS stream file /qibm/userdata/ondemand/*instance_name*/ars.cfg and add this statement at the end of the file to turn off name resolution:

```
ARSSOCK_RESOLVE_CLIENT_NAME=0
```

Stop and re-start the server and see if the logon time is faster.

If you have hundreds of OnDemand users, you may need to change the number of database subservers that are used to handle database requests. Edit the ars.cfg file and change the value for ARS_NUM_DBSRVR.

OnDemand jobs run by default at priority 50 in the QSYSWRK subsystem. If you prefer to use a different subsystem or priority, change the job description QOND400 in library QRDARS.

3.9 Training

In this section, we discuss the following training:

- Administrator training
- User training

3.9.1 Administrator training

We recommend that at least two people get system administrator training in OnDemand. This training should include all features of the Administration Client in addition to learning about migration policies, output queue monitors, and how to start and stop the server. Any users who will be working with the Administration Client to authorize other users and user groups or set up report definitions should also be trained in those features.

Who will provide this training? IBM no longer offers OnDemand for i5/OS classroom education, but several Business Partners offer classroom or on-site education. You also need to keep current with new features of OnDemand and learn tips and techniques for administering and using the product. Subscribe to the OnDemand for i5/OS Bulletin, distributed from IBM every two to three months. Send a note to Darrell Bryant (dbryant@us.ibm.com) and ask him to add your name to the bulletin distribution. Also request a list of partners. After you have learned the basics of OnDemand, go back and review bulletins from previous years. Use the search word *bulletin* on the IBM Support Web site to see a list of bulletin summaries from prior years that you can download:

http://www.ibm.com/software/data/ondemand/400/support.html

3.9.2 User training

We suggest that you demonstrate the basic functions of OnDemand to the users, then create a User's Guide that contains graphics or screen captures of how to access sample reports on your system. You may choose to have training sessions where the users go through hands-on exercises when you are there to assist them and answer questions. Whether they use the OnDemand Windows Client or a Web browser interface to OnDemand, all users need to know how to log on to OnDemand, find their documents and view them, and log off. There are many other features that they may use also (not all are available in a browser session). Here are a few features you can show them:

- Different search operators, for example, a *Like* search on character string fields.
- Search by specific date range, or t-5y for today minus 5 years.
- Sort a hit list.
- Annotations.
- Print to local and, optionally, server printers.
- Change background color and zoom and save as a logical view.
- Scroll down and see the heading locked on a report listing (if applicable).
- Find a character string within a document.
- Find a character string within a document created as a large object (if applicable).
- Move columns in a report listing (if applicable).
- Stop icon at top of display to stop building the hit list.
- Other icons, such as view next page and view next item in the hit list.
- Send a document (e-mail).
- Create and save a named query.
- Copy pages from a document to a file.
- And more, depending on your setup.

> **Note:** Train the users no more than a week before they start using the system. You do not want them to forget everything before they begin using OnDemand in production!

3.10 Deployment into production

Some companies prefer to create all their report definitions in a test instance, then export these application groups, applications, and folders to a production instance. It is easy to use the OnDemand Administration Client to drag and drop these objects. Be sure that you create migration policies in the production instance and add the users and groups before you export the other objects. That way, the permissions will also be exported.

After you have installed and tailored your OnDemand for i5/OS environment, set up and tested some report definitions, and worked with and trained the users, it is time to put the system into production. Make sure that you complete these steps:

- Modify your business applications to send spooled files to output queues that are monitored by OnDemand.
- Install and configure the OnDemand Windows Client or configure the browser interface for users.
- Grant permissions for users and user groups to the appropriate application groups and folders.

In this section, we discuss the following activities associated with deployment:

- Automating the system
- System documentation
- System monitoring

3.10.1 Automating the system

We typically add entries to the i5/OS job scheduler, because this feature is available on every System i system. Type the command WRKJOBSCDE and you can schedule jobs. Decide when you want jobs to run. We recommend stopping the output queue monitors before you run DSM or ASM, and allow all storage manager jobs to complete before you do your nightly backups. If you use a disk pool storage level in a migration policy, you need to unmount it before performing your backups. Here are our suggestions for scheduled jobs:

- STRTCPSVR *ONDMD (or CALL QRDARS/QRLMCTL *STRTCPSVR followed by the name of the instance)
- ENDTCPSVR *ONDMD (or CALL QRDARS/QRLMCTL *ENDTCPSVR followed by the name of the instance)
- STRMONOND for each output queue monitor
- `arsload` (or STRMONOND) for each IFS directory to monitor
- CALL QRDARS/QRLCASMUFS PARM('*instance-name*') to unmount the disk pool prior to a backup
- STRDSMOND
- Control Language (CL) program to perform backups

Be sure to modify your backup procedures to save the OnDemand libraries and directories. If you use TSM with OnDemand data, make sure that you back up that data. If you use an optical library, use the DUPOPT or CPYOPT commands to save optical data for off-site storage. You can find more information about optical support at the Web site:

http://www.ibm.com/servers/eserver/System i5/optical/

3.10.2 System documentation

We recommend documenting the procedures to start and stop the server, start and stop the monitors, unmount the disk pool, and run backups. Even though these jobs may be in your job scheduler, there may be times that you need to run them manually.

3.10.3 System monitoring

Here are two things you need to do each day:

- Look in the System Log for entries with message number 88; this message is written if a file fails to load into OnDemand.
- Review the ASM reports, which you can find in the output queue QRDARS/QRDARS400.

You may also want to run programs that monitor performance on your system and review how the system is being utilized by OnDemand. But the best way to monitor this type of performance is to listen to user comments and to make sure that OnDemand is able to load spooled files and documents in an acceptable amount of time.

3.11 Lessons learned

We summarized the following lessons learned from many years of implementing OnDemand solutions for customers for your reference:

- When we first started implementing OnDemand, we would create a test instance that we used for new development. Then we would export the application groups and folders to the production instance after we had tested archiving and retrieving reports. We found that it was not necessary to create a test instance except with businesses that have that requirement. It is easier just to create a production instance that we name QUSROND since that name is used as the default in system commands. The default *PUBLIC permission for each folder is *no access*, so users do not see new folders in the list until we grant permissions to them.

- Document retrieval performance is better if you have only one or just a few application groups in a folder instead of grouping multiple application groups together. However, if *application group name* is one of the search fields and the users always select a single application group, retrieval performance should be fine. If the users do not want to see a long list of folders, you can mark rarely used folders as secondary folders. Or the users can create filters within the OnDemand Windows Client to show only selected folders on the list.
- For indexes that can vary considerably in length (such as employee name, customer name, and vendor name), we recommend defining them as variable length string fields. Remove trailing blanks for the field in the application and make sure that each user sets the option in the OnDemand Windows Client to auto size the document list column.
- When you use the graphical indexer to define fields for the application, the default parameter is BREAK=YES. Change the parameter to BREAK=NO for fields that should not cause a break to a new document. Often we have only one field that we define as BREAK=YES.
- When you add user groups, we suggest you start with a Group ID (GID) of 80100, then add groups in GID increments of 100. With this method, you can insert groups later if necessary for complex permission setups. Remember that if a user is in multiple groups and each group is *specifically* authorized to an application group or folder, the permissions of the group with the lowest GID is used.
- We recommend to order the search fields in a folder based on the way users are most likely to search for documents. When you open a folder, the cursor is positioned at the first search field, so this field should be the one that is most commonly used. For the date field, we suggest defining a default date range that the users do not need to modify. In addition, many users prefer to have the date as the last field in the search criteria and the first field in the document list.
- Create a home directory for the user profile that runs the output queue monitors. If a load fails, you can easily find the .ind and .out work files in that directory.
- When you integrate with Kofax Ascent Capture to scan, index, and archive documents, we recommend using the **arsload** command for archiving documents in an IFS directory instead of the STRMONOND command. The **arsload** command will not try to archive the .ind and .out files until the .ard file associated with them is sent from the PC to the IFS directory. The STRMONOND command tries to load the files as soon as an .ind file is

present, so if the .out file has not yet been received into the directory, the load will fail.

- A very common practice is to make trigger 1 the new page carriage control character. Here are two examples where making trigger 1 the character string “Page   1” is a better choice:
  - If you have a report that contains multiple documents that you want to break apart and the first page of each document always starts with Page 1, then you can use the string “Page   1” as trigger 1. This is better than using the new page carriage control character because the index values are only searched for when trigger 1 is found. If you use the carriage control character, the indexer will search on every page in the report for the indexes. If “Page   1” is used as trigger 1, then only the first page of each document is searched by the indexer.
  - If you have a multiple page report where you want to archive it as a single document, then using “Page   1” as trigger 1 is the best choice. This prevents the indexer from searching every page for the index values when you only need the values from the first page.
- Subscribe to and read the OnDemand bulletins from IBM. Darrell Bryant at IBM periodically e-mails an OnDemand for i5/OS bulletin with a lot of information that will help you make better use of OnDemand. These bulletins generally consist of news items and tips. The tips section is especially useful because it often highlights new features in the server or client and gives examples of how to use them. If you are not receiving these bulletins, send a note to Darrell Bryant (dbryant@us.ibm.com) and ask him to add your name to the bulletin distribution list.
- We recommend you change the default settings for the System Log application group as follows:
  - On the Storage Management tab, by default there is no Storage Set Name specified. We have found that it is best to define a migration policy just for the System Log so that you can change how long you keep information in the System Log without affecting any other application groups. Because the System Log contains many different kinds of information, there is no one set of values that is best for everyone.
  - On the Storage Management tab in the Cache Data group, the Cache Data for n Days is set to a default value of 10,000 days. This is normally much longer than what you need once you have assigned a storage group to the System Log. Remember that only the documents (items in the hit list that have Yes for the View column) are stored in cache. A value of 60 days or less is a good value for this field because typically only system administrators or report administrators work with the System Log.

- In the Storage Management tab in the Life of Data and Indexes group, carefully consider what to use for the Expire in n Days value. We initially would choose a very large value for this, then realized that smaller values may be appropriate. Originally, in order to use the RMVRPTOND command, you needed to know the Load ID for each set of documents that were loaded. This value was only shown in the System Log 87 message and there was no other way to find the value. A change was recently made to the client so that from the hit list or when viewing a document, you can right-click and display the properties of the document. One of the properties is the partial Load ID. You can change this partial Load ID into the Load ID that is needed by RMVRPTOND by appending onto it the value -0-0. For example, if the partial Load ID is 12072-2-0-12FAA, then the complete Load ID would be 12072-2-0-12FAA-0-0. Some of the things to consider when selecting a value for the life of data and indexes are:

    - The System Log contains documents that show what changes were made to applications, application groups, folders, users, groups, and storage sets and who made the changes. A report is also produced whenever one of these objects is deleted. The log provides a good audit trail of the changes and if you later encounter problems, you can see what changes were made and who made them. You can then contact the person who made the change to discover why they made the change and try to resolve the problem with their help.
    - The System Log may also be used to gather statistics about OnDemand usage. You need to keep data in the System Log long enough to provide meaningful information.
    - The System Log may also contain information needed for auditing purposes. Application groups may specify that the values for certain fields be logged in the System Log. For example, you may want to know who is looking at payroll information or checks and which ones they are looking at. If this is the case, you would need to keep the System Log for as long as you keep the documents themselves.

4

IBM Content Manager OnDemand for z/OS implementation guidelines

This chapter provides guidelines for implementing an IBM Content Manager OnDemand for z/OS solution.

We cover the following topics:

- Identify project resources.
- Content Manager OnDemand server sizing.
- Installation and configuration.
- Design.
- Application setup and verification.
- Functional testing.
- Performance testing and considerations.
- Training.
- Deployment into production.

4.1 Identify project resources

The implementation and complexity of Content Manager OnDemand for z/OS implementations varies considerably from customer to customer and even between multiple systems installed at a single customer. Thus, the project resources will vary considerably. In general, a team lead with sufficient organizational authority needs to be identified. Then a group of individuals, each of whom is responsible for one or more of the tasks and subtasks to be performed, need to be identified. The project team usually needs to include individuals with z/OS system skills, Content Manager OnDemand for z/OS management skills, and user process skills. Some individuals may have overlapping skills and thus play multiple roles in the implementation process.

The Content Manager OnDemand for z/OS administrative roles and responsibilities are further discussed in Chapter 2, “Preparing for OnDemand”, in *IBM Content Manager OnDemand for z/OS and OS/390: Introduction and Planning Guide*, GC27–1438. The section provides a good summary of the tasks required to administer Content Manager OnDemand for z/OS and helps you assign project responsibilities.

4.2 Content Manager OnDemand server sizing

Based on the scalability of the z/OS systems, the actual server sizings and their method of estimation will vary. The minimum z/OS server requirements for a Content Management OnDemand Server are discussed in Chapter 3, “Hardware and software”, in *IBM Content Manager OnDemand for z/OS and OS/390: Introduction and Planning Guide*, GC27–1438.

The basic factors that need to be examined are DASD and CPU.

4.2.1 DASD

Questions to be asked related to DASD are:

- How much DASD is required for the database. This includes:
  - Content Manager OnDemand system tables
  - Content Manager OnDemand Index tables
  - The DB2 index tables for both the Content Manager OnDemand System and Index tables.
- How much online archive storage is required?

- How much offline archive storage is required?
- How much offsite storage is required?
- How much online redundant storage is required (duplicate data/hot site backup)?

Storage requirements and their estimations are further discussed in Chapter 6, “Storage requirements”, in *IBM Content Manager OnDemand for z/OS and OS/390: Introduction and Planning Guide*, GC27–1438.

4.2.2 CPU

Depending on system performance and high availability requirements, a Content Manager OnDemand for z/OS system may run on one or more LPARS located on one or more systems. The number of CPUs assigned to each of the LPARs will be based on the processing and response time requirements for each LPAR.

CPU consumption is a function of:

- Number of concurrent users
- Number of exits and their functionality
- Type of indexers
- Report sizes, document size, and number of documents and indexes per report

Assistance with estimating the sizing requirements for a Content Manager OnDemand for z/OS system can be obtained from the IBM Americas TechWorks support center, found at:

`http://techworks.dfw.ibm.com/techworks/web.nsf/`

4.3 Installation and configuration

Before you begin your installation of Content Manager OnDemand for z/OS, you should understand the various types of configurations and setups that are available to you.

In this section, we discuss the design, installation, configuration, and implementation related tasks summarized below:

- Architecture
- Scalability
- Prerequisites and setup
- Installation and configuration summary
- Creation of a Content Manager OnDemand Instance
- Configure storage management
- Verify the installation
- Implement best practices

Refer to the following manuals, as they provide details about these different tasks: *IBM Content Manager OnDemand for z/OS: Configuration Guide*, GC27–1373 and *IBM Content Manager OnDemand for z/OS: Introduction and Planning Guide*, GC27–1438.

4.3.1 Architecture

Before beginning your installation and configuration, it is important to understand the general architecture of the Content Manager OnDemand server. This helps you determine the type of configuration that meets your business requirements.

As illustrated in Figure 4-1 on page 125, from an architectural perspective, the Content Manager OnDemand server consists of two components: a library server and one or more object servers. The library server contains the database system tables and the application group data tables. The object server contains the stored reports and documents.

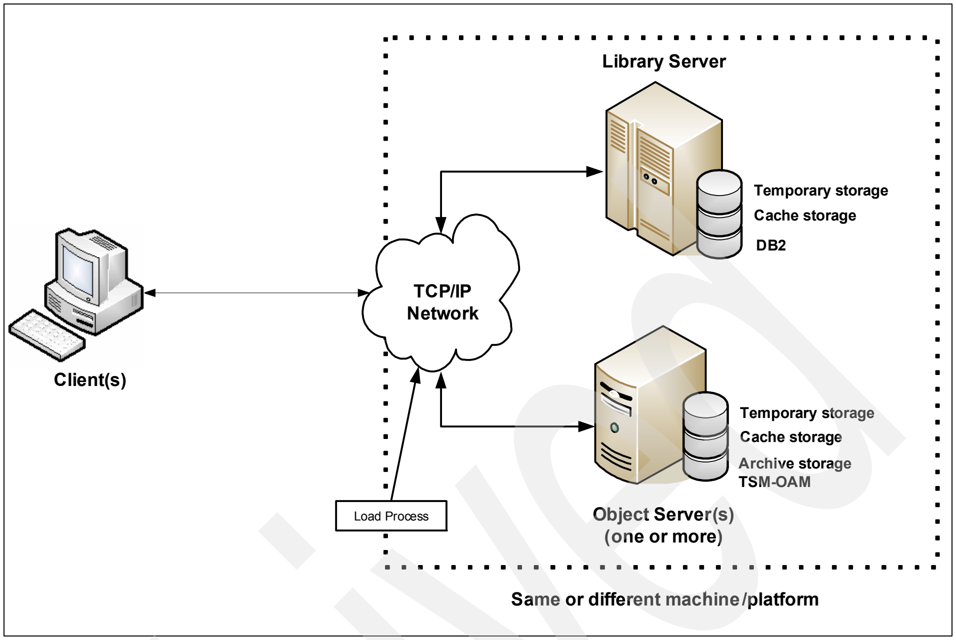


Figure 4-1 Content Manager OnDemand library server and object server

Data is loaded and retrieved from the Content Manager OnDemand server using TCP/IP. This allows for the library server and object server to be physically separate as long as a high speed TCP/IP network links them together. This linkage makes the library server and object server appear as a single server to the clients.

The most common types of configurations are:

- One library server with one object server. In this case, both the library server and object server are installed on a single z/OS LPAR.
- One library server with multiple object servers. In this case, several options are possible:
  - Library server in one LPAR and object server in a second LPAR.
  - Library server in one LPAR and multiple object servers (each in their own LPAR).
  - Library server in one LPAR and one or more object servers (in their own LPARs or on one or more object servers on multiplatforms servers).

Figure 4-2 illustrates the case of a library server in one LPAR and two object servers, each of which is installed in its own LPAR.

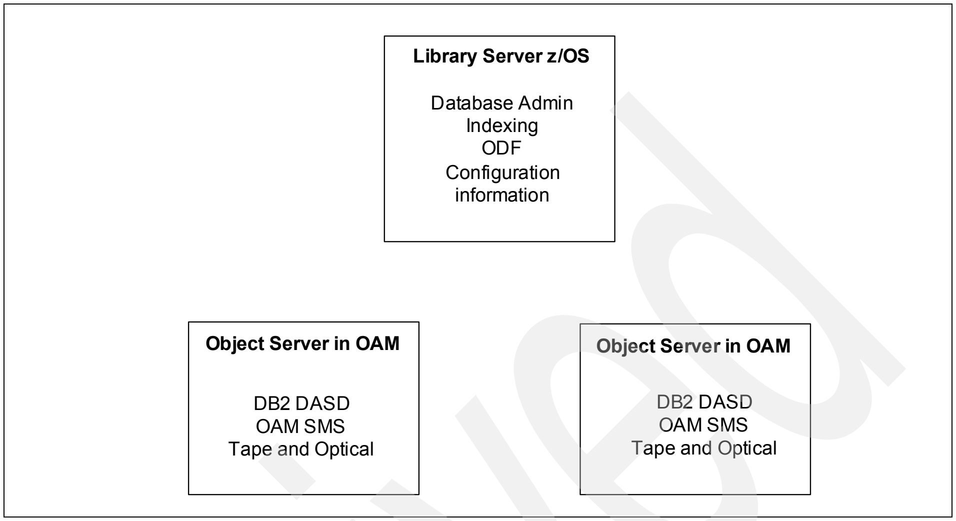


Figure 4-2 An OnDemand system with one library server and multiple object servers

Functionality is distributed between the library server and object server(s) as follows:

- Library server
  - Manages access to the administration definitions.
  - Provides data Integrity.
  - Maintains data archive index information, configuration, and user account information.
  - Controls access to data archives on object servers.
  - Directs query, retrieve, and print requests from the clients.
  - Routes store, retrieve, and delete requests from the clients.
  - Performs user authentication through internal security or external security (SAF) calls.
  - Performs logging.
- Object server
  - Provides the repository for Content Manager OnDemand data archives.
  - Stores archive storage policy information.
  - Manages retention of Content Manager OnDemand data archives.

- Controls transition of Content Manager OnDemand archives.
- Manages expiration of Content Manager OnDemand archives.

4.3.2 Scalability

A single Content Manager OnDemand system can be scaled from a single z/OS system image (Figure 4-3) that performs all of the required tasks (data loading, library storage, and object storage) to a Parallel Sysplex® complex (Figure 4-4 on page 128) on multiple systems (allowing for higher levels of performance and availability).

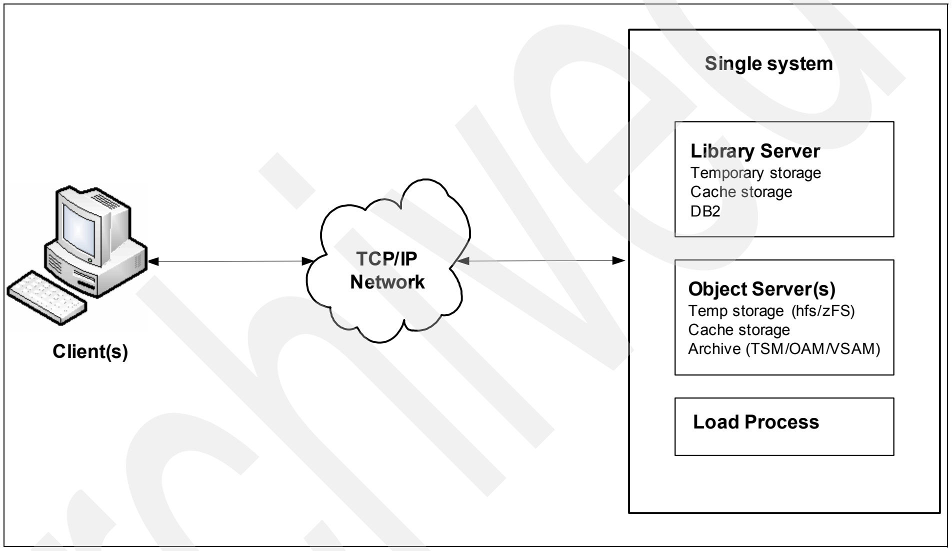


Figure 4-3 A single OnDemand system with a library server and object server architecture

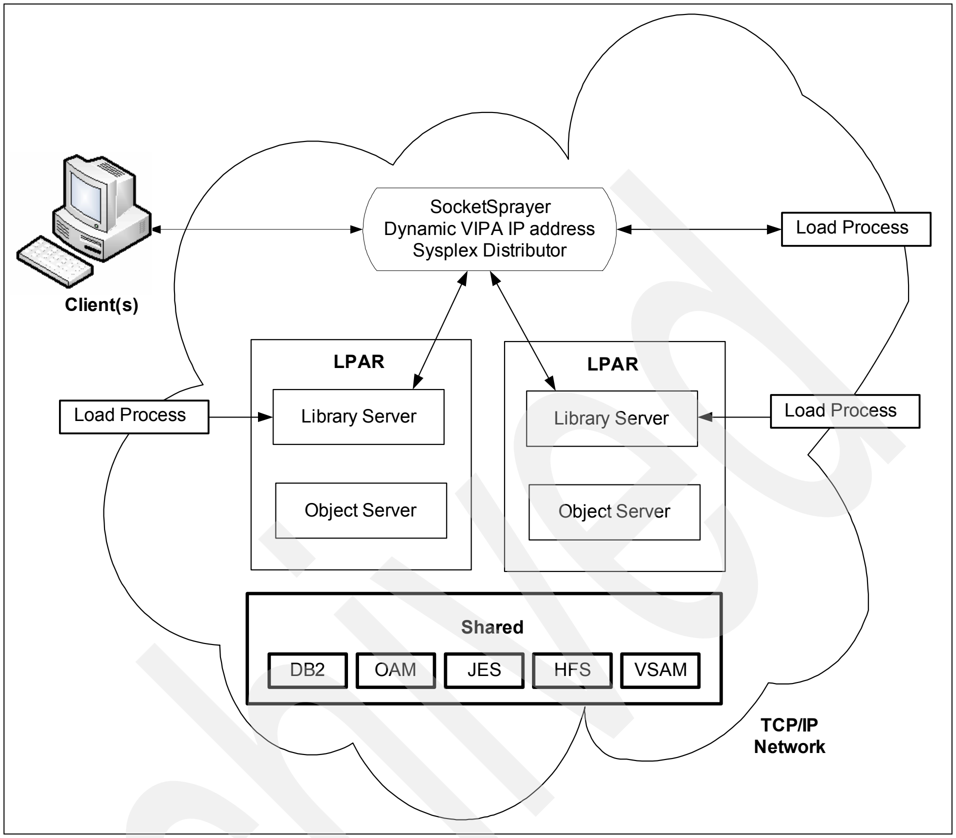


Figure 4-4 A Content Manager OnDemand system with parallel sysplex/multiple LPARs

A Parallel Sysplex system can be composed of multiple interconnected z/OS systems (globally distributed in the extreme case). Connectivity is maintained by the TCP/IP network.

In the example architecture illustrated in Figure 4-4, a Socket Sprayer provides a Virtual IP-address (VIPA), which will send an incoming connection request to a dispatcher or distribution manager. The distribution manager forwards the request to a target server based on load balancing rules.

There are three types of VIPA:

- Static VIPA
- Dynamic VIPA (DVIPA)
- Distributed DVIPA

The VIPA usage for the Content Manager OnDemand system is as follows:

- For Parallel Sysplex, the library servers must share the same port number.
- For sysplex (non-parallel means no socket sprayer), the library servers can have different port numbers.

In both cases, the library servers must access the same Content Manager OnDemand DB2 database.

Other factors that would affect the VIPA performance are:

- The kind of dispatcher or distribution manager (hardware, software, or both).
- The location at which the dispatcher or distribution manager is installed. This can be within a cluster, or on machines in the IP network, or in routers (in a primary and secondary) controlling the cluster.

4.3.3 Prerequisites and setup

When determining the hardware and software configuration requirements for your installation, consider the following items:

- The business requirements for ingesting and retrieving the data
- The types of indexers (based on input and stored data types)
- The volume of data that you plan to ingest
- The required exits (for pre- or post-processing the data)
- The retention polices for this data
- The number of concurrent users
- The types of viewers and transforms (based on stored and displayed data types)
- The performance requirements

At a minimum, you will need a single LPAR for a standard library server and object server:

- Server requirements:
  - z/OS.
  - DB2 with ODBC enabled.
  - UNIX System Services.
  - TCP/IP.

- Client options and requirements:
  - For Browser viewing, WebSphere® is needed on the middle tier.
  - For custom Java API based implementations, the Web Enablement Kit is needed on the middle tier.
  - For custom Structured API based implementations, the Web Enablement Kit is needed on the middle tier.
  - Windows Client.
  - For the 3270 client viewer, CICS is required on the server.
  - For the Content Manager OnDemand Distribution Facility, CICS is required on the server.

  Refer to the following manual, as it provides minimum requirements for these components: *IBM Content Manager OnDemand for z/OS: Configuration Guide,* GC27–1373.
- Custom code requirements:
  - Server exits:
    - Can be developed using COBOL, C, or Assembler.
    - Are dynamically invoked by the Content Manager OnDemand server (long running task/daemon).
  - Servlets and J2EE applications:
    - Require Java.
    - Run under WebSphere Application Server or as an independent Java application.
  - Structured APIs (SAPI):
    - Can be developed using COBOL or C.
    - Run under TSO Batch, or CICS, or IMS™.
  - Window Client extensions (two types):
    - The OLE code requires C++.
    - The Content Manager OnDemand Toolkit requires Visual Basic.

- Archive storage options:

  Both Object Storage Systems and File Storage Systems are available for archiving the data stored within Content Manager OnDemand. The type of storage system that is used is transparent to the Content Manager OnDemand user. Each type of storage system has advantages and disadvantages that differ by operating system and platform. In this section, we present the z/OS perspective:

  - Object Storage Systems

    Object Storage Systems are designed from the bottom up as storage systems for object data (where an object is defined as a Blob of data). Their main advantage is their ability to manage large quantities of data in a hierarchical manner. Built-in facilities exist for data migration, archiving, backup, and expiration. The two methods available on z/OS are:

    - Object Access Method (OAM).
    - Tivoli Storage Manager (TSM).

    OAM

    This has historically been the most commonly used archive storage methodology on the z/OS platform. OAM has been used since the inception of Content Manager OnDemand through the present time. OAM is an access method that supports a class of data known as *objects*. The content, format, and structure of the data objects is unknown to OAM. OAM's main characteristics include the functions necessary to manage the objects after they are stored. OAM provides the following functions:

    - Hierarchical data management (DASD, optical, and tape).
    - Transparent retrieval of offline data.
    - Built-in data migration and backup abilities.
    - Data expiration.

    Each OAM object is assigned to a *collection.* A collection is a group of objects typically having similar performance, availability, backup, retention, and class transition characteristics. This grouping enables efficient storage, retrieval, and deletion of objects.

    When using OAM as your Storage Manager, you will be required to set up the necessary SMS Constructs to support your storage and retention policies:

    - Storage Groups: Defines the storage location of the Content Manager OnDemand objects.
    - Management Classes: Defines the backup, retention, and class transition criteria for management of the objects.

- Storage Classes: Defines the type of object location, such as disk, tape, or optical.

Typically OAM data is initially stored on DASD (in DB2 tables), and later, when access to the data is minimal, the objects are migrated to secondary media, usually optical or tape. Regardless of the storage location, user access is identical, with the only difference being the document retrieval times, which would differ based on the storage device type. In general, OAM provides the fastest access method, as the data is stored in online DB2 tables. OAM is a strict z/OS implementation and does not function on any other platform. Historically, OAM has been regarded as the best and most simple archive alternative.

Figure 4-5 illustrates the general Content Manager OnDemand OAM architecture. The object storage and retrieval hierarchy is shown in the lower left hand corner. In the lower right hand side, the OAM data management DB2 tables are illustrated. These are the tables that OAM uses to manage the access to the OAM stored objects.

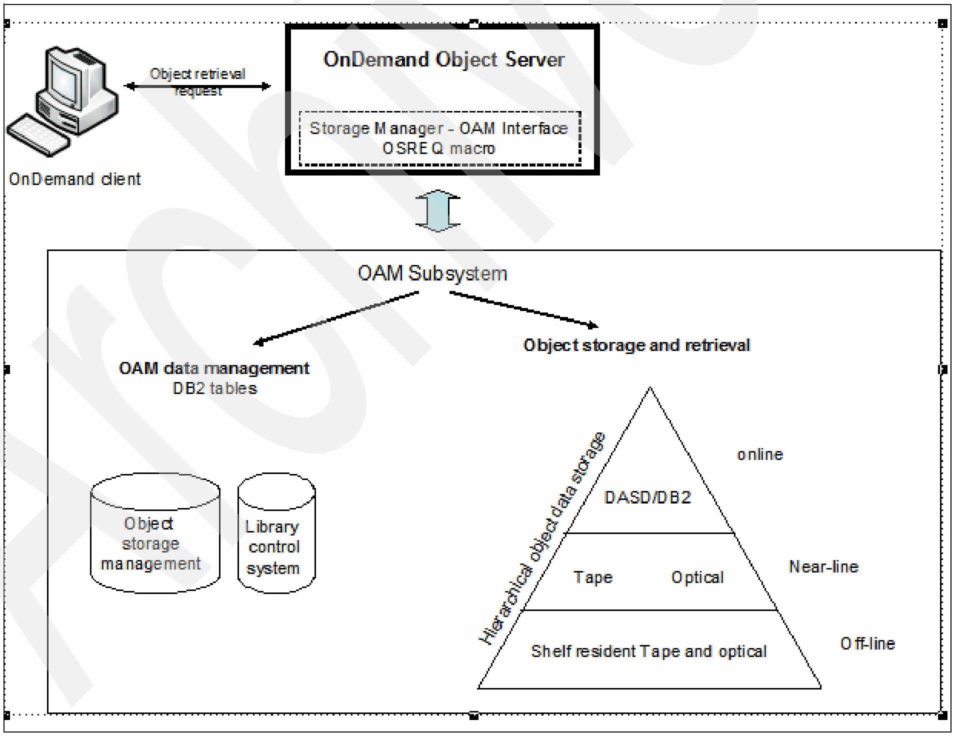


Figure 4-5 General Content Manager OnDemand OAM architecture

For more information about OAM Enablement for Object Support, see *z/OS DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support,* SC35-0426.

TSM

TSM is the IBM strategic storage manager direction. It was designed from the ground up as an enterprise-class data storage and recovery solution, protecting your business critical data from the mobile computer to the IBM eServer zSeries. TSM is a client-server system.

TSM supports HSM and moves inactive data from online storage to less expensive offline or near line storage. The TSM client software supports seventeen different operating systems and is thus available on all the Content Manager OnDemand server platforms. Tivoli Storage Manager's server software runs on eight operating systems, allowing for the server to be located in the most cost effective manner possible. Additionally, TSM supports more than 400 offline storage devices, including optical disk and tape.

TSM provides automated, policy-based, and distributed data and storage management for your environment. The TSM server provides the following functions:

- Data management.
- Storage device and media management.
- Instant archive and rapid recovery.

Figure 4-6 on page 134 illustrates the general Content Manager OnDemand TSM architecture. The Content Manager OnDemand Windows Client connects to the Content Manager OnDemand object server. The TSM client resides on the Content Manager OnDemand object server and is used to connect to the TSM server through TCP/IP. The TSM server can be located on any platform supported by TSM. The TSM administrative client is used to administer the TSM archive.

TSM storage and retrieval performance (compared to other local storage mechanisms) is mainly determined by the TCP/IP interface between the TSM client and server.

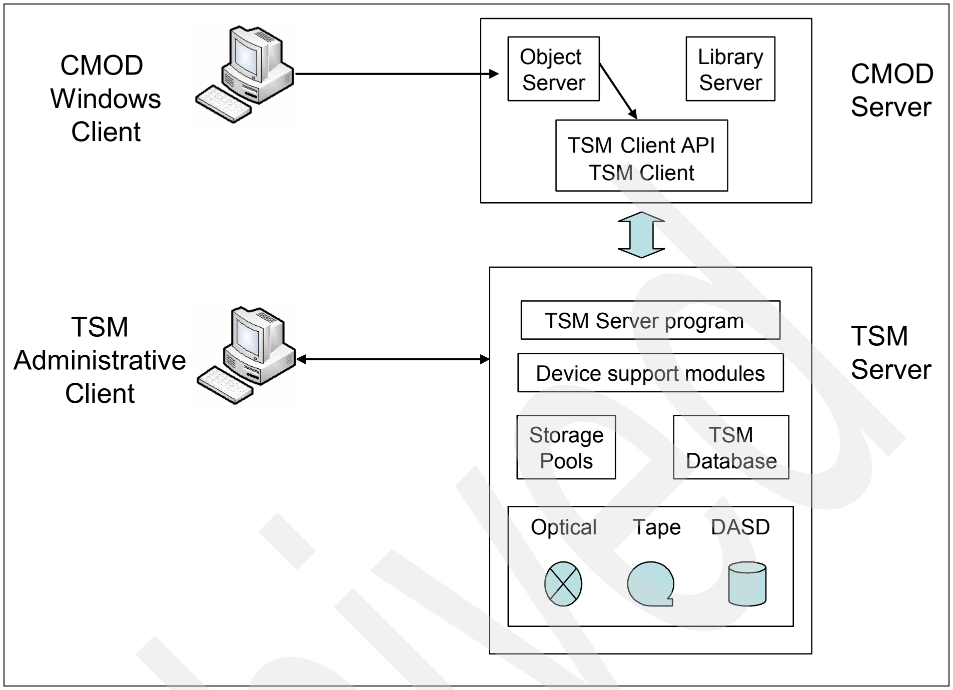


Figure 4-6 General Content Manager OnDemand TSM architecture

For TSM for z/OS implementation and setup requirements, see I*BM Tivoli Storage Manager for z/OS Quick Start,* GC32-0777.

– File Storage Systems

The three File Storage Systems available for use on z/OS are:

- Virtual Storage Access Method (VSAM).
- Hierarchical File System (HFS).
- z/OS File System (zFS).

VSAM

VSAM is an access method used to access MVS sequential datasets. Content Manager OnDemand archive data stored to VSAM can be backed up using DFSMS. The main advantage of using VSAM as an archival method is that it is part of z/OS (it comes with the system). The main disadvantage is that each storage object is stored in a single VSAM file and will thus consume a directory allocation. In a practical sense, VSAM is rarely used (due to the directory overhead).

HFS

The HFS was designed to provide UNIX file system type support on z/OS. A single VSAM file is defined as a mount point. This mount point is the top directory of an underlying directory structure that contains one or more files. Thus many (limited by DASD allocation) Content Manager OnDemand 10 MB storage objects can be stored in a single mount point. Although complete and partial backup solutions are available, the HFS does not support any HSM type of data movement. Typically, on z/OS systems, Content Manager OnDemand uses the HFS for:

- Temporary data storage.
- A cache file system (if TSM is used as an archive storage mechanism and the TCP/IP connection to the TSM server is such that a performance benefit is achieved by using cache).
- When OAM (tape) is used as the archive storage mechanism, data will usually be stored and retrieved from the HFS (cache) and copied at load time (transparently) to tape.

> **Note:** When OAM (DASD) is used as the archive storage mechanism, data will usually be stored and retrieved directly to OAM (No HFS).

zFS

zFS is conceptually identical to the HFS in that it emulates the UNIX file system. zFS's usage within Content Manager OnDemand is similar to that of the HFS and its use is equivalent to that of the HFS from a Content Manager OnDemand system perspective. The main difference between both systems is that the zFS was developed to take advantage of the z/OS operating system. It thus provides improved performance (especially for large files), reliability, and better monitoring and diagnostic facilities.

4.3.4 Installation and configuration summary

The following outlines the basic necessary steps for installing and configuring Content Manager OnDemand product software for z/OS:

1. Define and set up operating system groups and users.
2. Set up UNIX System Services profiles.
3. Define started tasks.
4. Create HFS datasets.
5. Install ACIF (optional).
6. Perform the SMPe product installation.
7. Configure the server control files.

Define and set up operating system groups and users

When installing the product software, you will need to set up the following IDs and groups:

- A security user ID that will be the used throughout the product install process, for all the various product components. This ID will require an OMVS Segment assigned to it. The recommended name is *ODADMIN*.
- A server owning ID that will be assigned as the user for each of the Content Manager OnDemand started tasks. The recommended name for this owning ID is *ARSSERVR*.
- A security owning group profile. You will need to ensure that these IDs have been added to the security profile group for Content Manager OnDemand. The recommended name for this group is *ARS*.

  You will also need to ensure, that for the HFS files, the ARS security group profile is provided ALTER access.
- The ARSSERVR ID will need to be set as the owner of the DB2 instance. This ID will need the appropriate DB2 access, such as DBADM or SYSADM, so you will be able to create the storage group, database, tablespaces, and others. Your systems database administrator (DBA) can adjust the access as required for the ARSSERVR ID, after the installation is completed.

Set up UNIX System Services profiles

You will need to establish UNIX System Services profiles if you plan to execute the Content Manager OnDemand server commands or programs, such as ARSADMIN and ARSLOAD.

Set and export the STEPLIB environment variable to the location of the latest Content Manager OnDemand code. This can be done in /etc/profile or in a.profile that gets executed when the user ID requiring this STEPLIB logs on.

An example of this is as follows:

```
STEPLIB=ARSPTF.ODMP710.SARSLOAD:$STEPLIB
export STEPLIB
```

Check with your system programmer to determine the best way to implement this requirement.

Define started tasks

The servers you will be installing require started tasks control (STC) names of:

- ARSSOCKD for the library server.
- ARSOBJD for the object server. This STC is required if the object server is being installed on a different system from the library server.

- ARSLOAD for the JES Spool load capture daemon.

> **Note:** These STC names are the supplied default names. You can use unique names based on your standards.

You will need to ensure that security profiles are created for each of these STCs with ARSSERVR assigned as the owning user ID and ARS as the owning group for each.

Create HFS datasets

There are a minimum of three HFS datasets that must be created for the Content Manager OnDemand installation. Example of these HFS datasets are as follows:

```
HLQ.PRODUCT.HFS - OnDemand product code
HLQ.SERVER.HFS - OnDemand CACHE
HLQ.TEMP.HFS - OnDemand TEMP
```

The following manual provides details about these HFS files and their suggested sizings: *IBM Content Manager OnDemand for z/OS: Configuration Guide*, GC27–1373.

Install ACIF (optional)

If you have report data that is mixed mode data or line data with form definitions and page definitions that are to be converted, you will require the ACIF(AFP Conversion Indexing Facility) Indexer to index these reports. To convert or index these reports with ACIF, you will need to install the ACIF software on your system.

ACIF is a separate program product that is delivered with the Content Manager OnDemand software.

See *Program Directory for Enhanced AFP Conversion and Indexing Facility for use with Content Manager OnDemand*, GI10–0211 for installation instructions.

Perform the SMPe product installation

You must perform the SMPe installation process described in *Program Directory for IBM Content Manager OnDemand for z/OS and OS/390, Version 7.1,* GI10–8441.

Configure the server control files

Following the SMPe install of FMID H272711 - Content Manager OnDemand, you will need to configure and tailor the Content Manager OnDemand control files for your installation and environment.

Create a working directory for a copy of the server files. Create a symbolic link from /usr/lpp/ars/config to this working directory.

Copy the ars.cache, ars.cfg, ars.ini, and arsprtjcl files from the Content Manager OnDemand product install directory ....../samples into the working directory. Table 4-1 lists these files and the purpose of the files.

Table 4-1 Sample directory files

File	Purpose
ars.cache	List of cache storage file systems.
ars.cfg	Content Manager OnDemand server configuration file.
ars.ini	Names of and configuration information for Content Manager OnDemand instances.
arsprtjcl	Infoprint server printing JCL. This file should be a copy of /usr/lpp/ars/samples/arsprtjcl1. The installation needs to rename the copy to /etc/ars/arsprtjcl.

Some important aspects of each of these configuration files are listed below. More details can be found in the *IBM Content Manager OnDemand for z/OS: Configuration Guide,* GC27–1373.

ars.ini

The ars.ini (Figure 4-7 on page 139) file contains information about the Content Manager OnDemand instances. An instance is composed of a DB2 database, one library server, and one or more object servers. The stanza name identifies the Content Manager OnDemand instance:

For example, [@SRV@_ARCHIVE] - ARCHIVE would be the name of this instance.

> **Note:** The ars.ini file must be in code page 1047. That is, the delimiters in the header line for the instances must be X'AD' (left bracket character) and X'BD' (right bracket character).

If you implement external security authentication, you will be required to include this parameter in the ars.ini file:

```
SRVR_FLAGS_SECURITY_EXIT=1
```

In you implement external security permissions checking, you will also need to include one or more of these parameters based on the level of external security permissions you have enabled:

```
SRVR_FLAGS_FOLDER_APPLGRP_EXIT=1
SRVR_FLAGS_DOCUMENT_EXIT=1
SRVR_FLAGS_SQL_QUERY_EXIT=1
```

Figure 4-7 shows a sample ars.ini file.

```
Y@SRV@_ARCHIVE
HOST=ondemand.ibm.com
PROTOCOL=2
PORT=0
DIRECTORY=/ars
SRVR_INSTANCE=ARSDBASE
SRVR_INSTANCE_OWNER=ARSSERVR
SRVR_OD_CFG=/odsars/usr/lpp/ars/config/ars.cfg
SRVR_SM_CFG=/odsars/usr/lpp/ars/config/ars.cache
SRVR_FLAGS_DOCUMENT_EXIT=1
SRVR_FLAGS_FOLDER_APPLGRP_EXIT=1
SRVR_FLAGS_SECURITY_EXIT=1
SRVR_FLAGS_SQL_QUERY_EXIT=1
```

Figure 4-7 Sample ars.ini file

ars.cfg

The ars.cfg file (Figure 4-8) contains information about the licenses, servers, temp storage, storage manager requirements, and DB2 and OAM threads. There are additional settings that can be included in the ars.cfg file as it relates to performance. Refer to the 4.3.8, “Implementation best practices” on page 144 for more information.

```
########################
# OnDemand Parameters  #
########################
ARS_NUM_LICENSE=10000
ARS_LANGUAGE=ENG
ARS_SRVR=
ARS_LOCAL_SRVR=
ARS_NUM_OBSRVR=10
ARS_TMP=/tmp
ARS_PRINT_PATH=/tmp
TZ=CST6CDT
ARSMVS_DEBUG_SM=1
ARS_DISABLE_ARSLOG=1
########################
# Database Parameters  #
########################
DB2_ENGINE=DB2ars.cache
#####################
# OAM Parameters    #
#####################
ARS_NUM_OAMSRVR=10
ARS_OAM_DB2SSID=DB1M
ARS_OAM_PLAN=CBRIDBS
ARSMVS_BPOOL_TSPACE=BP1
ARSMVS_BPOOL_INDEX=BP2
ARSMVS_NOMAXROWS_PRIQTY=72000
ARSMVS_NOMAXROWS_SECQTY=36000
ARSMVS_NOMAXROWS_INDEX_PRIQTY=36000
ARSMVS_NOMAXROWS_INDEX_SECQTY=18000
ARSMVS_MAXROWS_PRIQTY=72000
ARSMVS_MAXROWS_SECQTY=36000
ARSMVS_MAXROWS_INDEX_PRIQTY=38000
ARSMVS_MAXROWS_INDEX_SECQTY=18000
```

Figure 4-8 Sample ars.cfg file

ars.cache

The ars.cache file (Figure 4-9) contains a list of the cache file systems that you will use for cache storage. Typically on a z/OS implementation, you will not use the cache file system for storing the data. However, you are required to define a base cache file. This base cache file is used for storing Content Manager OnDemand control information. It is the first cache file system in the ars.cache file.

> **Important:** After you define the base cache storage file system to Content Manager OnDemand, you must not add or remove it from Content Manager OnDemand or change it in any way; otherwise, the system may fail.

Figure 4-9 illustrates an ars.cache example that includes the definitions for cache storage:

```
/ars/cache
/ars/more/cache
#
# NOTES:
# 1) The first filesystem defined in this file can NEVER change.
# 2) Always add new filesystems to the buttom of the list.
```

Figure 4-9 Sample ars.cache file

cli.ini

The cli.ini file (Figure 4-10) contains the configuration information for ODBC that ARSSOCKD uses to connect to the correct DB2 subsystem. This value could be already set within your installation by a DSNAOINI DD statement in JCL or as a HFS file, as in the example in this book.

```
YCOMMON
MVSDEFAULTSSID=DB1M
PLANNAME=DSNACL1
```

Figure 4-10 Sample cli.ini file

See *DB2 Universal Database for OS/390: ODBC Guide and Reference, SC26–9005* or *DB2 for OS/390: ODBC Guide, SC26–9941* for more information about configuring the cli.ini file.

4.3.5 Creation of a Content Manager OnDemand instance

Before you begin to define reports to Content Manager OnDemand, load data on the system, or use the system, you must create the Content Manager OnDemand database and initialize the system tables. The database resides on the Content Manager OnDemand library server.

Creating the Content Manager OnDemand instance involves the following tasks:

- Creating the storage group
- Creating the Content Manager OnDemand database
- Creating a tablespace for the Content Manager OnDemand system tables
- Creating the Content Manager OnDemand system tables
- Initializing the Content Manager OnDemand System Log component
- Initializing the Content Manager OnDemand system migration component

Refer to the following manual, as it provides Content Manager OnDemand Instance installation details: *IBM Content Manager OnDemand for z/OS: Configuration Guide,* GC27–1373.

4.3.6 Configure storage management

Refer to the appropriate Storage Management documentation when configuring your storage management policies:

- OAM: For OAM configuration for Object Support, see *z/OS DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support*, SC35-0426.
- TSM: For TSM for z/OS implementation and setup requirements, see *IBM Tivoli Storage Manager for z/OS Quick Start*, GC32-0777 and *IBM Tivoli Storage Manager for z/OS Administrator's Guide*, GC32-0775.

4.3.7 Verify the installation

To ensure everything is installed successfully and correctly, verify your system installation and perform report load and retrieval functions.

You can perform the following sequence of steps to verify that the system is installed successfully:

1. Start the Content Manager OnDemand server (ARSSOCKD).

2. If you have not already done so, install at least one of the Content Manager OnDemand Windows Client programs on a PC. See the *IBM Content Manager OnDemand: User's Guide,* SC27-0836 for details.
3. Start the Content Manager OnDemand Windows Client program. Content Manager OnDemand displays the Logon to Server dialog box.
4. Click **Update Servers**. Content Manager OnDemand displays the Update Servers dialog box.
5. Add the name of the Content Manager OnDemand library server, the IP address, and the port number that it is listening on. Click **Help** for information about the fields and options. Click **Close** to return to the Logon to Server dialog box.
6. Select the name of the server that you added in the Update Servers dialog box, if it is not already selected.
7. Type a Content Manager OnDemand user ID and password in the fields provided. (The first time that you log on to the system, you must specify the built-in Content Manager OnDemand user ID, *admin*. Initially, there is no password. However, you will be prompted to enter and verify a password. At some point in the future you need to change this password.)
8. Press Enter.
9. Open and search the System Log folder.

To verify that everything works as expected, we also recommend performing the report load procedure as follows:

1. Select a sample report to define to Content Manager OnDemand.
2. Define the application group:
   - Identify the data fields.
   - Determine storage management.
3. Define the application:
   - Define the logical and physical characteristics for the sample report.
4. Define the folder:
   - Map the application group to the folder.
5. Load the sample report using ARSLOAD.

After loading the sample report, perform the report retrieval procedure:

1. From the Windows viewer, open the folder and perform a search to obtain the hit list.
2. Retrieve and view the document.

4.3.8 Implementation best practices

This section describes the best practice recommendations for the IBM Content Manager OnDemand for z/OS Server. Before beginning, you should ensure that you are at the latest Content Manager OnDemand maintenance level.

You can follow these guidelines, but you should be aware that performance tuning has to be done on a case-by-case basis, because every system has different characteristics and usage requirements.

From a Content Manager OnDemand for z/OS perspective, these considerations include:

- Data type (such as AFP and line data), quantity of data, and any preprocessing requirements
- Number of indexes, report size, document size, and report to document ratio
- Data loading patterns
- Data retrieval patterns
- Environmental considerations (such as CPUs, disk, and network)

Content Manager OnDemand settings

The main, initial settings are located in, and controlled by, the server instance file (ars.ini), the server configuration file (ars.cfg), the cache file definitions (ars.cache), and the Content Manager OnDemand administrator parameters.

Server configuration files (ars.cfg)

The server configuration file (ars.cfg) contains several parameters that affect performance. These parameters are:

- ARS_NUM_DBSRVR: This is the maximum number of threads that are opened between the Content Manager OnDemand library server and DB2. Typically, this is set to a number between 4 and 40, depending on client access patterns.
- ARS_NUM_OAMSRVR: This is the maximum number of threads between the Content Manager OnDemand object server and OAM. Typically this is set to a number between 4 and 40, depending on client access patterns and object storage locations (DASD versus tape).

> **Note:** For both of the above parameters in large implementations, a value between 10 and 12 has been found to be a good tuning starting point value.

- ARS_EXPIRE_RECLIMIT: This parameter is used if storage manager expiration has been specified. It is the number of Load IDs that ARSADMIN will send to the server during expiration processing in a single request when using storage manager based expiration.
- ARS_TMP: This is the temporary work directory used by the load process. It is a separate non-shared mount point (configured as such through BPXPRMxx).

> **Important:** If the installation has multiple Content Manager OnDemand servers (ARSSOCKD), then each server must have its own ars.cfg and its own ARS_TMP directory.

Cache file definitions (ars.cache)

The cache files are defined in ars.cache. The cache file system can consist of one or more caches. Increasing the number of caches allows Content Manager OnDemand to distribute the data storage between them.

In a sysplex environment, each of the caches must be a shared zFS (or HFS) file system. Each cache has its own mount point (configured as such through BPXPRMxx). Performance can be further increased by placing each cache on a separate volume on a separate channel.

The first cache file system is the primary cache system in that it contains the Content Manager OnDemand control data and consequentially should never be placed offline.

> **Note:** Placing a cache file system offline prevents Content Manager OnDemand from accessing the stored documents.

GUI administrator parameters

There are a few GUI administrator parameters that you need to consider as well for system performance:

- Storage object size: The size of a storage object in kilobytes (KB). By default, Content Manager OnDemand segments and compresses report data into 10 MB storage objects. We recommend that you use the default value. Valid values are between 1 KB and approximately 150 MB. However, exercise caution when changing the value; specifying too large or too small a value can adversely affect the performance when loading data.

- Compressed object size: Determines the number of bytes in a fixed-size block of data stored on the system. By default, Content Manager OnDemand compresses input data into 100 KB blocks. You can specify a number from 1 to 99999. However, we recommend that you accept the default value; specifying too small of a value can result in less optimal document compression, while choosing too large of a value can result in less efficient document storage and retrieval. The value of the compressed object size must be less than or equal to the size of the storage object for the application group to which the application belongs.
- Compression: Content Manager OnDemand supports many different compression algorithms:
  - OD77 is the recommended compression algorithm.
  - OD77 Light is a variant on OD77 that utilizes less CPU but achieves less compression. However, the compression achieved is still typically greater than that achieved by other compression methods.
  - Selecting Compression Disable causes Content Manager OnDemand to not compress data during the load process and to not compress data for transmission over the network. This option is suitable for loading images (for example, TIFF) or PDF files, or any files that are already compressed and only need to be decompressed on the client after retrieval.
  - Selecting Compression None is generally discouraged. This setting will cause Content Manager OnDemand to not compress data for storage during the load process. During the retrieval process, Content Manager OnDemand will compress the data during transmission over the network, and then decompress the data on the client prior to viewing.
- Page-level index: This parameter is used only for large objects. It is used to display page-level index information within the document. This allows for easier navigation within the document. The page-level index information is not stored in the database and therefore cannot be used to search for and retrieve documents. Only group-level indexes can be stored in the database. Page-level indexes are stored with the document. After retrieving a document, the user can use the page-level indexes to move to a specific page in the document. If the parameter is not specified during loading, the page-level information is not displayed.
- Large object: The indexing program generates a large object by dividing very large documents into smaller parts and defining the indexing information that is used to retrieve the documents. When your users work with large objects, they should be able to retrieve documents more efficiently with less impact on the network. For example, suppose that a document contains 10,000 pages; using large object support, you divide the document into parts that contain 100 pages each. When users retrieve one of the documents, Content Manager OnDemand sends only the first part of the document (first 100

pages) to the client. Other parts of the document are automatically sent to the client when the user moves to different pages in the documents. Typically, a user will need to retrieve one or a few parts of the large objects, thus only a fraction of the large object document is downloaded to the client. This dramatically reduces network workload and client viewing responsiveness. Documents that require a large amount of storage (even when compressed) can also benefit from large object support.

- Segment field: We encourage you to define a segment field for the application group. Limiting a query to a specific table or set of tables significantly improves the performance of queries for applications that contain large amounts of data. If the segment field contains a date in the MMYY format, then Content Manager OnDemand deletes segments on the first day of the month (MM).
- Data migration from cache: This parameter determines how long the data is kept in cache (this should match the period of time in which the data will be frequently retrieved) before it is migrated to archive storage. After migration, the data can still be retrieved but with slower response times, depending on the archive storage device.
- Application ID Field: If you want to define more applications to an application group, you must define an Application ID Field to the application group at the time the application group is created. If you are not sure as to whether you will be adding one or more applications in the future, define the Application ID Field when the application group is created. The Application ID Field can be defined after the application group is created.
- Index definitions: It is important that you define the optimum number of application group indexes to meet your business requirements. Too few indexes will impact the users' ability to search reports, while too many indexes may slow down the systems load performance.
- Date field: Most of the time, a document includes at least one date. This is required from a user's point of view for organizing the document filing.

  For example, Invoice Number and Customer Number fields provide important information. Without them, we cannot associate the right invoice to the right customer. A date field, such as an Invoice Date, is also needed so we know when this invoice is generated. This information, as well as other date fields, such as Order Date and Delivery Date, are necessary for efficiently keeping documents organized.

  Similarly, with Content Manager OnDemand, to optimize its internal organization and ensure efficient document search and retrieval tasks, we recommend that you include a date field in the application group as a segment field.

You should identify the date field that Content Manager OnDemand can use to segment the application group index data. The segment field enables you to search specific tables of an application group's data rather than all of the tables.

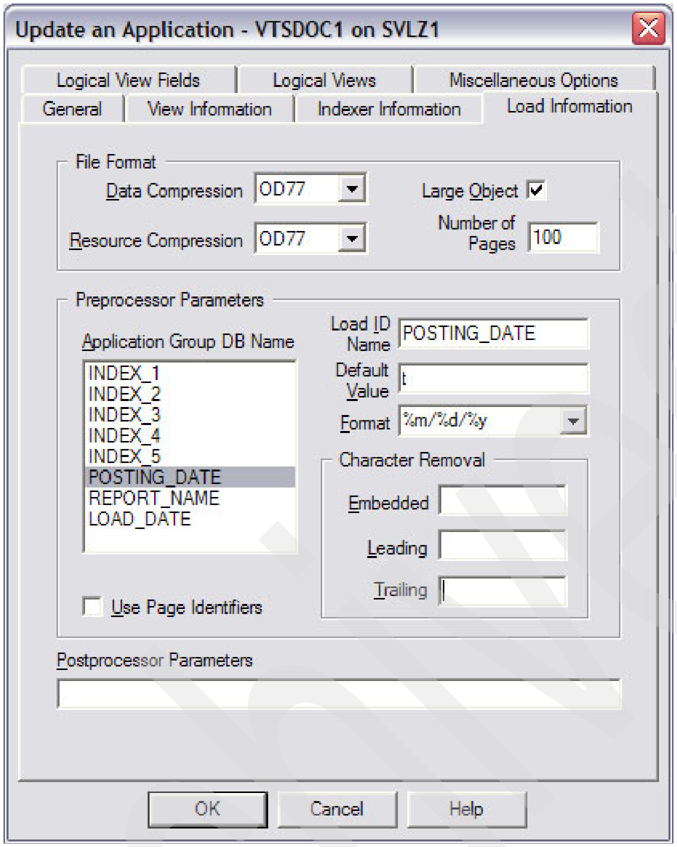
Important: After an application group is created, it is not possible to choose a date field as a segment.

Two date fields that require particular attention are:

- Load Date field: The Load Date field of the document might be different from any of the application dates. For example, the invoices can be loaded the day that they are printed or some days later. In this case, the Load Date is different from the Invoice Date. Accurate and easily accessible Load Date information helps avoid any misunderstandings.

  In addition to helping keep track of archiving activity, the availability of a Load Date index might be of great help in case of an audit or compliance request.

- Posting Date field: Most of the time, the report includes the date. If there is no date to be indexed in the report, you can define a Posting Date field as the date field and specify a lowercase “t” in the default value in the Load Information tab of the application definition. See Figure 4-11 on page 149.



File Format

Data Compression: OD77

Resource Compression: OD77

Large Object ☑

Number of Pages: 100

Preprocessor Parameters

Application Group DB Name:
INDEX_1
INDEX_2
INDEX_3
INDEX_4
INDEX_5
POSTING_DATE
REPORT_NAME
LOAD_DATE

Load ID Name: POSTING_DATE

Default Value: t

Format: %m/%d/%y

Character Removal

Embedded:

Leading:

Trailing:

□ Use Page Identifiers

Postprocessor Parameters:

Figure 4-11 POSTING_DATE field set up for an application

For step-by-step instructions about how to set up the configurations, refer to “Best practices”, in *Content Manager OnDemand Guide*, SG24-6915.

Data loading recommendations

Data loading recommendations are outlined as follows:

- Run parallel load jobs, to take advantage of multi-processors, multiple data paths, and multiple DASD.

- If the data source is on a remote system, it is possible to either:
  - Run ARSLOAD on the remote system and directly store the data to the specified IBM Content Manager OnDemand for z/OS (library server and object server)

    or
  - Upload the data to the specified Content Manager OnDemand for z/OS server through FTP, then run ARSLOAD on the selected Content Manager OnDemand for z/OS system.
- The -c indexer parameter should always be specified for ARSLOAD and should be unique for each ARSLOAD process.
- If ARSLOAD is running without TCP/IP, the -K parameter allows the DB2 connection to persist, potentially improving the load performance. The -K parameter causes ARSLOAD not to detach ARSADMIN. Normally, ARSLOAD attaches and detaches ARSADMIN several times for each file loaded. This might improve performance by not requiring the LE enclave associated with ARSADMIN to be constantly created or destroyed. The -K parameter is available only with APAR PQ91055 applied.

  > **Important:** Do not specify the -K parameter if you are using any Content Manager OnDemand for OS/390 Indexer exit routines until you verify that the exit routines function correctly in this environment. In particular, the LE enclave is no longer terminated between reports. This impacts exit routines that rely on enclave termination to perform cleanup, for example, closing files.

- All file systems should be dedicated file systems mounted on their own mount points.

Loading the data

You can load data using TCP/IP (in which case you can load the data from any system to any system across the network by default) or alternatively, you can store data directly to the Content Manager OnDemand library server and object server (directly using DB2 and OAM/TSM), in which case you must load the data on the same system where the data is being stored. This is illustrated in Figure 4-12 on page 151.

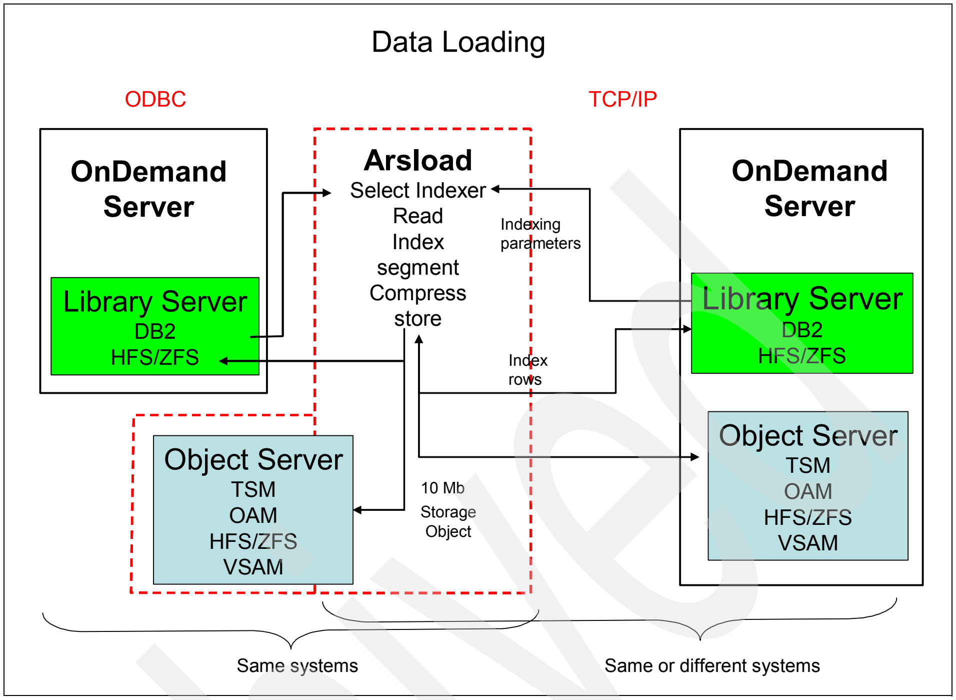


Figure 4-12 Content Manager OnDemand data loading using ODBC versus TCP/IP

By default, ARSLOAD will use TCP/IP to communicate with the server. This provides a great deal of flexibility with regard to where ARSLOAD is run and under which RACF® user and group IDs it is run under. This method of communication is advantageous when the reports to be loaded are on a separate system from the Content Manager OnDemand library server.

It is also possible for ARSLOAD to store data directly into DB2, OAM/TSM, or both. This method would be advantageous when:

- The reports to be loaded are on the same system as the Content Manager OnDemand database.
- The TCP/IP network is already overloaded.

To enable storage without the use of TCP/IP, the following conditions must be met:

- ARSLOAD must be running with the same RACF user and group ID as the server.
- ARSLOAD must be running on the same system as the server.

- The -h parameter must specify the instance name of the server in the ars.ini file and not a host name. The instance name must also be 16 characters or less.
- A DSNAOINI DD must be present.

> **Note:** If any condition above is not met, ARSLOAD will use TCP/IP regardless of the setting of ARSMVS_ARSADMIN_USETCPIP.

ARSLOAD can be forced to use TCP/IP even if all the above are true, by specifying the ARSMVS_ARSADMIN_USETCPIP=1 environment variable in the ars.cfg file.

Notes:

- Depending on your retrieval patterns and system hardware configuration, it might be advantageous not to store a duplicate set of documents in the Content Manager OnDemand cache when using OAM given that you already might be using OAM DASD.
- Generally speaking, the 390 indexer is faster than the ACIF indexer. This speed difference will vary and is a function of the data, the report and document size, and whether the data is line or AFP data.
- For the best overall performance, the indexer type needs to be matched to:
  - The type of data to be loaded
  - The input processing required prior to document storage
  - The preview processing during the document viewing process
- When using JCL to run the ARSLOAD program to capture reports:
  - Specify the -s ddname parameter to indicate the DD statement that points to the input report file that is being captured.
  - Specify the name of a temporary file as the last parameter. The ARSLOAD program uses the temporary file for work space during the load process. The directory to be used by ARSLOAD for temporary files is determined in the following order:

    i. The -c option in the ARSLOAD parameters.

    ii. The environment variable ARS_TMP.

    iii. The environment variable TEMP.

    iv. The environment variable TMP.

    v. The current working directory if none of the above are specified.

- ARSLOAD will store the document being processed in the directory specified by the -c option. This directory will need to be analyzed to make certain there is adequate additional space to hold the largest document that can be captured.
- The OS/390 indexer provides Content Manager OnDemand large object support for line print and AFP reports.
  - In general, large object can be used for all line print and AFP reports, but should be considered for most documents that exceed 100 pages in size.
  - Non-large object documents are limited in size by available processor storage. The entire document will be stored in memory during the load and retrieval processes. Excessively large documents might result in high storage requirements and significant paging activity and should be considered as candidates for large object.

Content Manager OnDemand database tablespace allocations

By default, the primary and secondary allocations for tablespaces created for the application group are based on the Maximum Rows setting for the application group. This setting is located in the Database Information window, which opens when you click **Advanced** in the General page for the Application Group.

If none of the parameters described below are specified, then the total amount of space required to contain 120 percent of the specified maximum number of rows is programatically determined. The PRIQTY of the tablespace is 50 percent of that total. The SECQTY is set to 12.5 percent of the total.

The following parameters (defined in ars.cfg) allow an installation to explicitly override and control the tablespace allocations:

- ARSMVS_MAXROWS_PRIQTY: For application groups with maximum rows specified, this parameter specifies the PRIQTY that will be used for the CREATE TABLESPACE. If not specified, the value calculated from the maximum rows will be used.
- ARSMVS_MAXROWS_SECQTY: For application groups with maximum rows specified, this specifies the SECQTY that will be used for the CREATE TABLESPACE. If not specified, the value calculated from the maximum rows will be used. This parameter is not honored unless the ARSMVS_MAXROWS_PRIQTY parameter is also specified.
- ARSMVS_MAXROWS_INDEX_PRIQTY: For application groups with maximum rows specified, this specifies the PRIQTY that will be used for the CREATE INDEX. If not specified, the value calculated from the maximum rows will be used.

- ARSMVS_MAXROWS_INDEX_SECQTY. For application groups with maximum rows specified, this specifies the SECQTY that will be used for the CREATE INDEX. If not specified, the value calculated from the maximum rows will be used. This parameter is not honored unless the ARSMVS_MAXROWS_INDEX_PRIQTY parameter is also specified.

For more information about specifying values for the above parameters, see the *DB2 Universal Database for z/OS: SQL Reference*, SC18-7426.

Content Manager OnDemand tablespace creation exit

The Content Manager OnDemand tablespace creation exit allows an installation to take action when Content Manager OnDemand creates a tablespace, table, or index tables that will be used to store application index data. The exit is not called for the Content Manager OnDemand system tables. For table and index creation, the installation can alter the SQL that will be used to create the table or index.

Content Manager OnDemand specifies the application group indexes as single columns. For example, a name and date field would be two indexes. One of those indexes could be a clustered index. Content Manager OnDemand for z/OS does not directly support the creation of composite indexes. Composite indexes may be created by means of the tablespace creation exit. This exit allows for the creation of other indexes during the creation of Application Group tablespaces.

Index creation is called once for each index that is being created on the table.

Refer to the *IBM Content Manager OnDemand for z/OS and OS/390: Configuration Guide*, GC27–1373 for further information.

Placing system tables in separate tablespaces

APAR PQ88578 changes the way the Content Manager OnDemand system tables are created. Previous to this APAR, all the Content Manager OnDemand tables were in a single tablespace. With this APAR, each Content Manager OnDemand system table is created in its own tablespace. This allows many tables to be placed in a 4K buffer pool, and allows row-level locking to be used for tables to allow for better concurrency when using those tables.

Installations that currently have all the Content Manager OnDemand system tables in a single tablespace do not need to migrate their tables to multiple tablespaces. For installations that desire to migrate, the following steps may be used:

1. Shut down the ARSSOCKD server.
2. Back up the ARSDBASE database (or the database associated with the server you are trying to migrate).

3. Use the `/usr/lpp/ars/bin/arsdb -x` command to export the Content Manager OnDemand system tables. Note that the PTF for PQ88578 must be applied before doing this step.
4. Drop TABLESPACE ARSTSPAC (or the tablespace associated with the server you are trying to migrate).
5. Run the ARSTSPAC job.
6. Run `/usr/lpp/ars/bin/arsdb -c` to create the tables in the new tablespaces.
7. Run `/usr/lpp/ars/bin/arsdb -i` to import the Content Manager OnDemand system tables.

Installations that migrate should examine any RUNSTATS jobs they use to ensure that the new tablespace names are used. The tablespaces associated with each table are listed in Table 4-2.

Table 4-2 Content Manager OnDemand system tables and the associated tablespace

Table	Tablespace
ARSAG	ARSAGT
ARSAGFLD	ARSAGFLT
ARSAGFLDALIAS	ARSAGFAT
ARSAG2FOL	ARSAG2FT
ARSAGPERMS	ARSAGPET
ARSANN	ARSANNT
ARSAPP	ARSAPPT
ARSAPPUSR	ARSAPPUT
ARSFOL	ARSFOLT
ARSFOLFLD	ARSFOLFT
ARSFOLFLDUSR	ARSFOLUT
ARSFOLPERMS	ARSFOLPT
ARSGROUP	ARSGROUT
ARSLOAD	ARSLOADT
ARSNAMEQ	ARSNAMET
ARSNODE	ARSNODET
ARSPRT	ARSPRTT

Table	Tablespace
ARSPRTOPTS	ARSPRTOT
ARSPRTUSR	ARSPRUST
ARSRES	ARSREST
ARSSEG	ARSSEGT
ARSSET	ARSSETT
ARSSYS	ARSSYST
ARSUSER	ARSUSERT
ARSUSRGRP	ARSUSRGT
ARSUSRGRPID	ARSUSGIT

Content Manager OnDemand data expiration

When expiring documents from OAM using the ARSEXOAM program, it is possible to improve the unload performance by using the ARS_EXPIRE_REQLIMIT parameter. This parameter controls the number of Load IDs sent at a time to the server in a single expiration request. The default value is 1, meaning a separate request for each Load ID being processed.

Load IDs for the same application group can be grouped together up to the ARS_EXPIRE_REQLIMIT value; groups must be for the same application group though. For example, adding ARS_EXPIRE_REQLIMIT=25 (in the ars.cfg file for the instance) will allow up to 25 Load IDs for an AG to be processed at a time.

Exits and logging

All exits and logs that are not being used should be disabled.

By default, most exits are disabled unless they are specifically enabled, that is, unless you issue the command **`setprog exit,add,exitname=ars.ptgn`**.

The system logging and the user exit logging should be turned off, unless they are needed. This is accomplished using the System Administration Client.

When you specify ARS_DISABLE_ARSLOG=1 in ars.cfg, the System Log Exit (ARSLOG) will not be invoked at all. Without specifying this parameter, even though all user exit logging is disabled in the System Parameters window of the System Administration Client, an attempt will still be made to call the ARSLOG exit for certain messages. If you are not planning on using the ARSLOG exit, you should specify ARS_DISABLE_ARSLOG=1 to minimize the overhead of attempting to call the ARSLOG exit.

If you are not using the Content Manager OnDemand Distribution Facility, you should disable it by renaming /usr/lpp/ars/bin/exits/arsuload to another name, such as /usr/lpp/ars/bin/exits/arsuload.sav.

There are special considerations for APKACIF exits written in COBOL. The ARSSPVIN sample APKACIF input exit is written as a COBOL main program. In order to prevent the Language Environment from creating and destroying the COBOL runtime environment, each time ARSSPVIN is called, a CEEUOPT CSECT must be assembled and link-edited with the COBOL object code. Information about how to construct a CEEUOPT CSECT is documented in *OS/390: Language Environment for OS/390 Customization*, SC28-1941.

A sample CEEUOPT CSECT is in data set CEE.SCEESAMP(CEEUOPT). You can use this sample as a model, but you must be sure that the following option is specified: RTEREUS= (ON).

We also recommend that you specify the ALL31(ON) option, but this option is not required. Stack and heap storage can potentially be allocated above the line while running with ALL31(OFF) specified.

In addition, you must be sure that the resulting module is link-edited as NOT RE-ENTRANT and NOT REUSABLE. These link-editing conditions are required to allow the local variables within the COBOL exit code to retain their values across multiple invocations

4.4 Design

During this phase of the implementation, you will need to determine the report types, security requirements, report retention requirements and how your users will access these reports. This will help you implement a Content Manager OnDemand archive solution in the most efficient way.

In this section, we discuss the following design tasks:

- Report selection
- Indexing requirements
- Retention
- Application group, application, and folder
- Security
- Document Audit Facility
- Indexers
- Exit points

4.4.1 Report selection

This section contains information that can help you plan for the reports that you will be storing into Content Manager OnDemand. This section lists questions that you might ask users of the reports, it provides information about the types of data that you can store in Content Manager OnDemand, and provides information about indexing reports.

As you are planning for the reports that you will be ingesting into Content Manager OnDemand, it requires an understanding of how the system will be deployed, who will use the system, how they will use it, and other user requirements. Answers to these questions will provide information that allows you to properly configure your Content Manager OnDemand system, including the storage and network components, to support your applications and users.

The questions to ask when planning for reports to be stored in Content Manager OnDemand are:

- What types of print data streams will you be ingesting?
- Will you require transforms to convert input data to other display data formats (such as AFP to PDF)?
- What is the logical organization of the print data stream?
  - Page organization: A consistent stream of pages of transaction or ledger data. Example: PAGE or PDOC report types.
  - Logical groups of information, such as statements or policies. Example: DOC report type.
  - Data that may not have a consistent format, such as reference materials or product literature. Example: NODX report type.
- What is the volume of input to process? How many reports and are their versions of the report?
- What index values do the users need to retrieve a specific version of a report (or a document)?
- How much time is available to load reports into Content Manager OnDemand? Daily? Weekly?
- How long do you plan to maintain report data on the system?
- How many concurrent, logged-on users do you anticipate on average? At peak times?
- How many active users do you anticipate?

- How will you handle securing access to these reports or documents? Will you use Content Manager OnDemand internal security or external security or both?

Attention: The maximum size of the input file is dependent on the indexing program that is being used. ACIF, the PDF Indexer, and the Generic Indexer process files in the HFS, so are limited in size to 2 GB. The OS/390 Indexer does not use the HFS to hold the input file, so the file size is limited only by the source of the input file.

Once you have an understanding of how the reports will be accessed, this will provide assistance with the indexing requirements.

Tip: We suggest that before you define any reports to Content Manager OnDemand, that you meet with the application group that owns these reports so that you have an understanding of the report data and how the users will be accessing the reports.

4.4.2 Indexing requirements

One of Content Manager OnDemand's main activities is indexing reports. When you index a report, Content Manager OnDemand extracts index values from the report and stores them in the database. The database fields that you define for your application groups hold the index values. When a user opens a folder, Content Manager OnDemand displays a list of search fields, which represent the database fields. To perform a query, the user enters values in the search fields. Content Manager OnDemand compares the search field values with the values in the database fields and retrieves the documents that match the query.

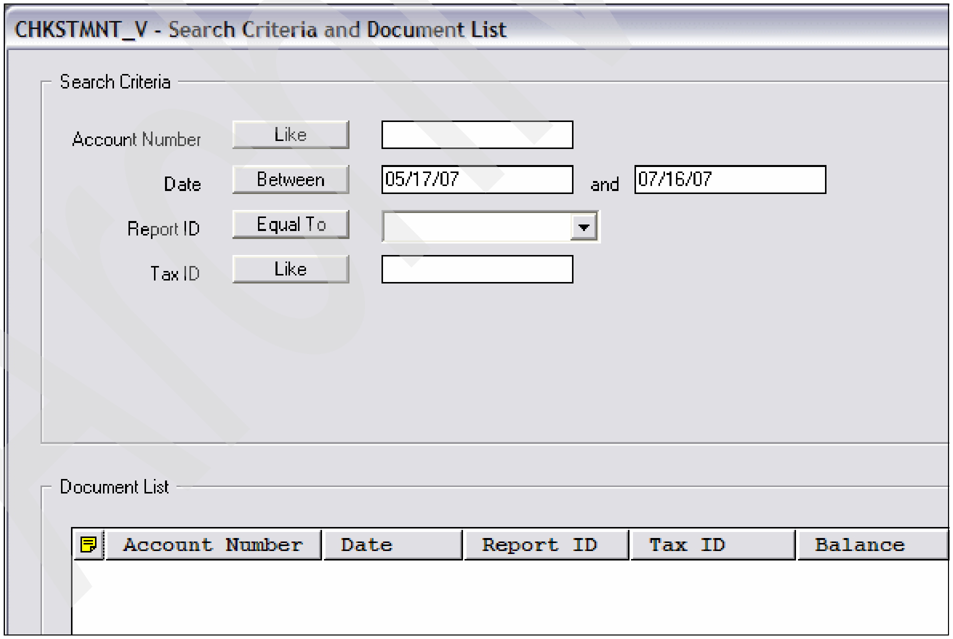
The questions to ask for each report are:

- How do you want to find a specific document within the report? In other words, what are the search fields?
- Should the search fields be filters or indexes?
- Which of these fields will be queried and which will only be displayed?
- When the value of a search field changes within the report, should this cause a break to a new document?

In the following check statement example (Figure 4-13), the users may:

- Prefer to search by account number, check date, and tax ID, and then see the balance amount displayed in the document list. In this case, you would make balance amount a filter and not an index, which is the default when you use the graphical indexer.
- If the users want to search by balance amount, but not as a stand-alone search; in other words, users may search for all balance values over a certain amount for a particular vendor number within a date range; in this case, balance amount would also be a filter.
- If the users want to search only for a balance amount, you would make that field an index so that an access path could be built for that field and the search would be faster.

> **Note:** Content Manager OnDemand has been enhanced so that you can change a field from being defined as an index to a filter or from a filter to an index after the application group has been created and reports have been archived. So if you want to alter your original choice, you may do so at a later time.



CHKSTMNT_V - Search Criteria and Document List

Search Criteria

Account Number	Like			
Date	Between	05/17/07	and	07/16/07
Report ID	Equal To			
Tax ID	Like			

Document List

Account Number	Date	Report ID	Tax ID	Balance

Figure 4-13 Check statement search window example

There are generally two types of indexing:

- Document indexing: Reports with logical information such as bills, statements, policies, and invoices.
- Report indexing: Reports with column data where the values in that column are in a sorted sequence like a general ledger.

Document indexing

Document indexing is used to index reports that contain unique values, such as an account number or a customer name. When searching and retrieving these types of reports, Content Manager OnDemand returns a list of the items that match the user's query and transfers the individual items to the Content Manager OnDemand Windows Client program for viewing and printing.

Content Manager OnDemand supports up to 32 fields as indexes or filters for document-type data. The fields do not have to be sorted and can contain numeric or text information. The fields are stored in the database as indexes or filters

Report indexing

Report indexing allows users to search sorted report data and retrieve the first occurrence of the value that is specified in the query. This type of reports contain line data with sorted values on each page, such as a transaction log or general ledger.

The OS/390 indexer can divide the report into groups of pages and generate index data for each group of pages. The first and last index values contained in each group of pages is stored in the database.

4.4.3 Retention

Content Manager OnDemand expects that the documents be kept by the storage manager for an amount of time equal to the value of the *Life of Data and Indexes* field. After that period of time expires, Content Manager OnDemand will delete the document index data and expects that the storage manager will delete the object data.

The Life of Data and Indexes field is specified on the Storage Management page of an application group using the Content Manager OnDemand administrative client. Content Manager OnDemand will keep index entries for the period of time specified in that field.

Historically, most customers used OAM as their storage manager for managing the life of the objects, so OAM needs to be configured to keep the objects around for that same period of time as defined in the Life of Data and Indexes field, if the storage manager (OAM or TSM) is not configured in that way,

it is possible that the storage manager is misconfigured such that it will delete objects before Content Manager OnDemand deletes the indexes that reference them. This causes a document list to be presented to the user that contains document entries for which no document data can be retrieved (the data no longer exists).

If an installation's owner does not want to configure their storage management policies to expire objects in accordance with the values that they have specified in the Life of Data and Indexes fields, it is possible to have the deletions of the object by the storage manager drive the deletion of the Content Manager OnDemand indexes.

4.4.4 Application group, application, and folder

In this section, we discuss application group, application, and folder creation.

Application groups

An *application group* is a collection of one or more applications that contain common indexing and storage management requirements. The application group contains the database information that is used to load, search for, and retrieve reports. The application group defines the data that is to be loaded into the database.

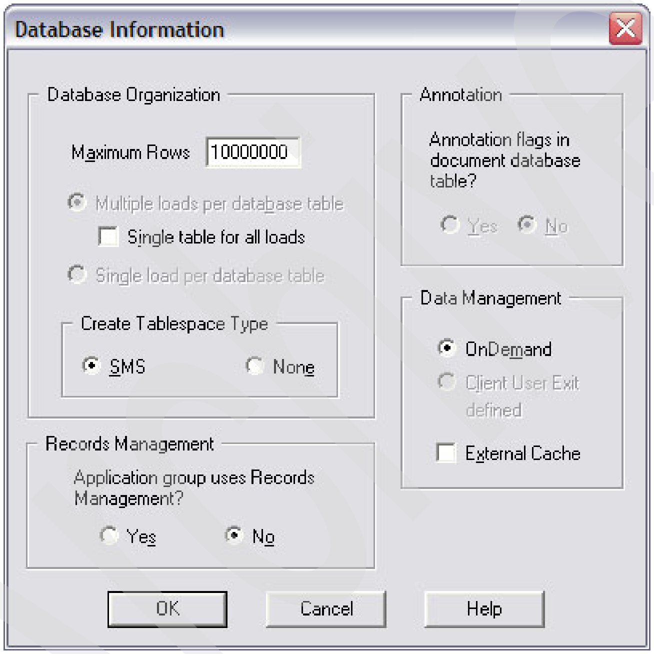
When you are designing your application group definition, there are some aspects that need to be considered as they can contribute to a successful Content Manager OnDemand system implementation:

- Database information
- Storage management
- Field definition
- Field information

Database information

The database information section of the application group definition process requires that decisions be made concerning the number of rows to be stored in each database table and the number of report loads to be included in each database table. These values are important to system performance and maintenance:

- Maximum Rows (Figure 4-14). Content Manager OnDemand uses this value to determine when to segment application group index data. When the table of the AG index data contains the number of rows specified in the Maximum Rows field, Content Manager OnDemand closes the table and initializes a new table. New index rows will be added to the newly initialized table. The closed table will still support queries.



Database Information

Database Organization

Maximum Rows 10000000

● Multiple loads per database table

□ Single table for all loads

○ Single load per database table

Create Tablespace Type

● SMS ○ None

Records Management

Application group uses Records Management?

○ Yes ● No

Annotation

Annotation flags in document database table?

○ Yes ● No

Data Management

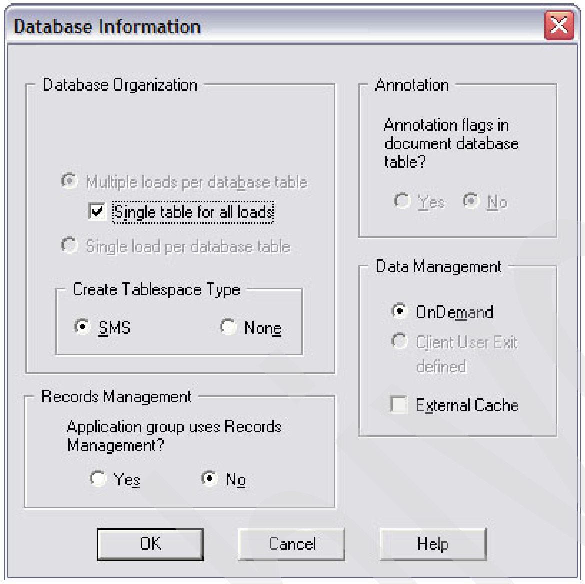
● OnDemand

○ Client User Exit defined

□ External Cache

Figure 4-14 Database Information - Maximum Rows

- Single table for all loads (Figure 4-15). If selected, Content Manager OnDemand will create one single index data table for each application group and all index loads for this AG will stored in one AG index data table. This option is most frequently used when the application group data tables are small.



Database Information

Database Organization

● Multiple loads per database table

☑ Single table for all loads

○ Single load per database table

Create Tablespace Type

● SMS ○ None

Records Management

Application group uses Records Management?

○ Yes ● No

Annotation

Annotation flags in document database table?

○ Yes ● No

Data Management

● OnDemand

○ Client User Exit defined

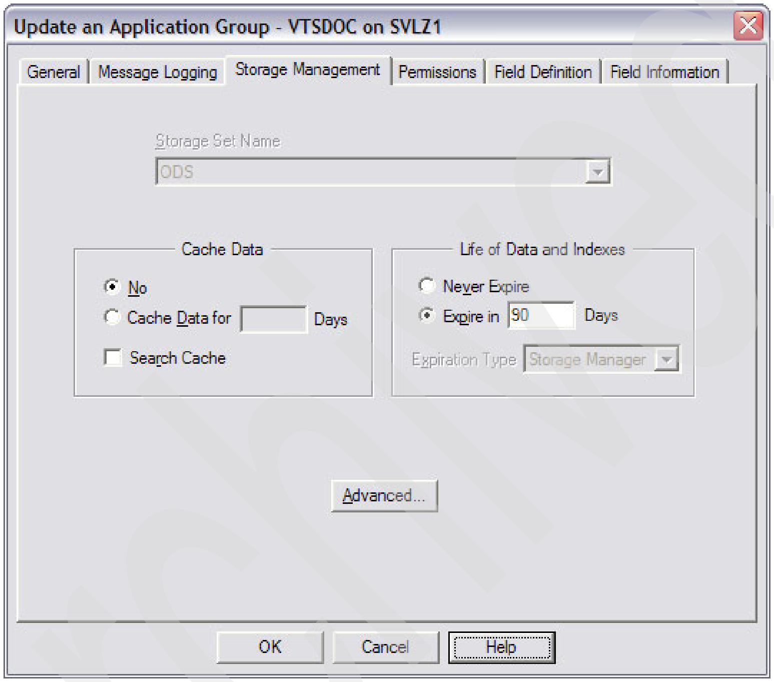
☐ External Cache

Figure 4-15 Database Information - Single table for all loads setup

Storage management

The storage management section (Figure 4-16) is where you specify the information Content Manager OnDemand uses to maintain application group data and provide the information that Content Manager OnDemand requires in order to work with the chosen storage manager, OAM, TSM, or CACHE.

The choices that you make will determine the length of time the report data, resources, and index data will be maintained.



Storage Set Name: ODS

Cache Data
● No
○ Cache Data for ____ Days
□ Search Cache

Life of Data and Indexes
○ Never Expire
● Expire in 90 Days
Expiration Type: Storage Manager

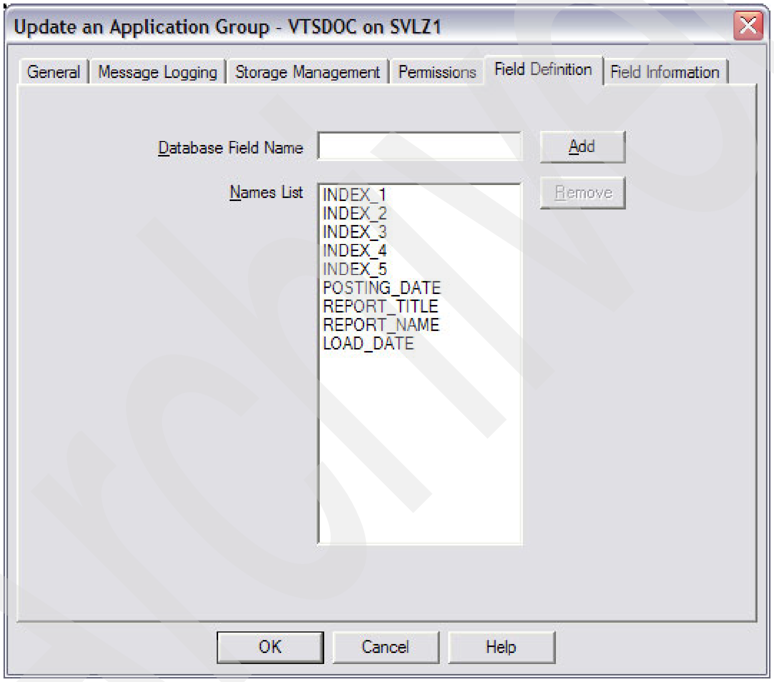
Figure 4-16 Application Group Storage Management tab

Field definition

The field definition page (Figure 4-17) is where you define database fields to the application group. The names you provide to these database fields can be arbitrary. They do not need to be the same names as the fields defined in the applications that are mapped to the application group.

We recommend you use field names like INDEX_1, INDEX_2, INDEX_3, and so on, as this allows you to include multiple applications with different index names to a single application group. There are two advantages with adopting this design:

- Reduction of application groups
- Reduction of the number of DB2 queries per folder search



Database Field Name

Names List

INDEX_1
INDEX_2
INDEX_3
INDEX_4
INDEX_5
POSTING_DATE
REPORT_TITLE
REPORT_NAME
LOAD_DATE

Figure 4-17 Application Group Field Definition tab

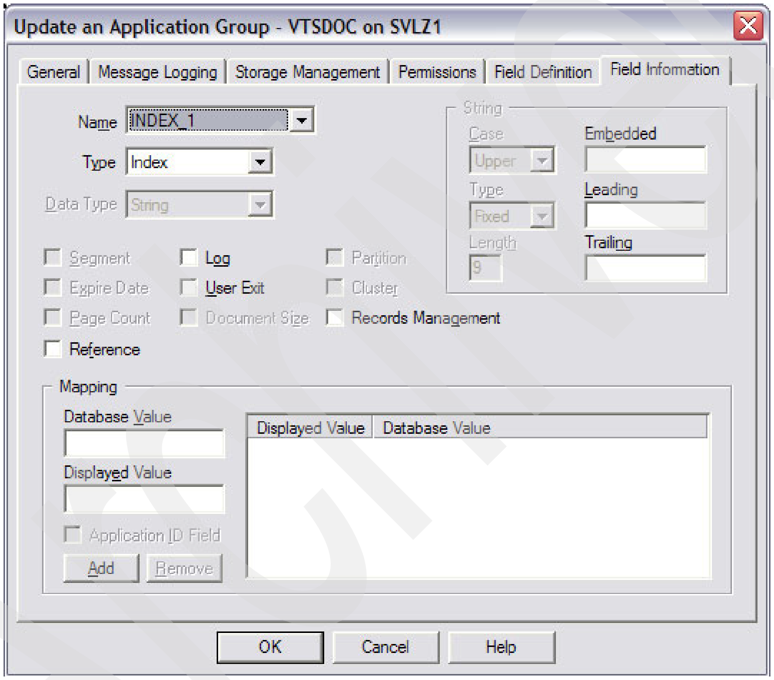
Field information

The Application Group Field Information tab (Figure 4-18 on page 167) is used to define the attributes of the database fields that make up the Content Manager OnDemand report index data. These attributes determine the characteristics of the index data and control many aspects of loading and processing data in the system. A database field must be added for each index value that is required by applications to be part of the application group.

Using the generic INDEX_1 name (Figure 4-18) allows you to define any application field to that index as long as it adheres to the common field information criteria. For example:

- Is of the same data type (string).
- Is equal to or less than the field length (9).
- At query time, we will only be querying a specific application within the application group.

More detailed information about setting up generic index usage is provided on a case by case basis during the IBM Content Management Lab Services implementation engagement.



Update an Application Group - VTSDOC on SVLZ1

Name: INDEX_1
Type: Index
Data Type: String

□ Segment □ Log □ Partition
□ Expire Date □ User Exit □ Cluster
□ Page Count □ Document Size □ Records Management
□ Reference

String
Case: Upper
Type: Fixed
Length: 9
Embedded:
Leading:
Trailing:

Mapping
Database Value:
Displayed Value:
□ Application ID Field

Displayed Value	Database Value

Figure 4-18   Application Group Field Information tab

Applications

Applications describe the characteristics of the report file and define how Content Manager OnDemand will index and load the report data.

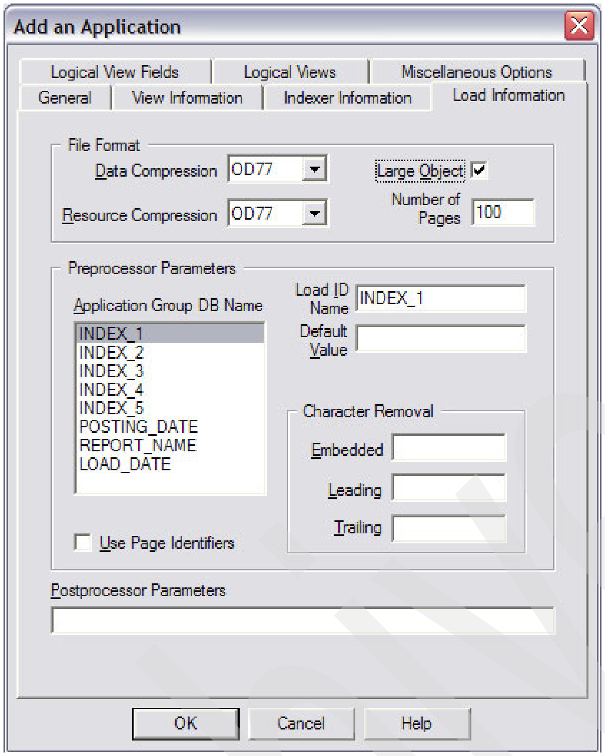
Applications associate the data with an application group, and specify the type of index processing to be performed on the data. Applications also define any logical views to be put in place for the users and determines any special print options to be used with the data.

Load Information attributes

The Application Load Information (Figure 4-19 on page 169) page specifies the processing and resource information that Content Manager OnDemand uses to load the input data onto storage volumes and to load the associated index data into the Content Manager OnDemand database. The File Format, Preprocessor Parameters, and Postprocessor Parameter are defined as part of the load information:

- File Format*:* Provides settings that control how the Content Manager OnDemand system compresses and stores documents and resources.
- Large Object Support: Large Object support is used to improve load and retrieve performance by dividing the document into smaller parts for loading and creating index information based on this document segmentation. Documents are retrieved faster due to the smaller segment sizes that are sent across the network.
- Preprocessor Parameters: Specify processing that is carried out on database fields prior to indexing data.
- Postprocessor Parameters: Specify a system command or exit program that will run against an index file before the index records are loaded into the database.

Note: An application group name can be updated after the application group is added, as long as the Application ID Field value has not been used as the identifier in an application; otherwise, you can no longer update the application group name.



File Format

Data Compression: OD77

Resource Compression: OD77

Large Object ☑

Number of Pages: 100

Preprocessor Parameters

Application Group DB Name:
- INDEX_1
- INDEX_2
- INDEX_3
- INDEX_4
- INDEX_5
- POSTING_DATE
- REPORT_NAME
- LOAD_DATE

Load ID Name: INDEX_1

Default Value:

Character Removal

Embedded:

Leading:

Trailing:

☐ Use Page Identifiers

Postprocessor Parameters:

Figure 4-19 Application Load Information tab

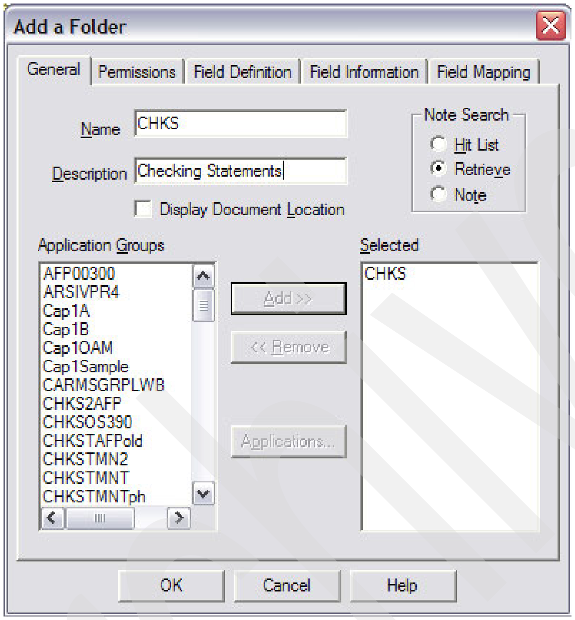
Folder

A folder provides a user with a simple way to query and retrieve data stored in Content Manager OnDemand. It provides users with a convenient way to find related information stored in Content Manager OnDemand, regardless of the source of the information and how the data was prepared.

A folder is set up by an administrator as a common query window for several application groups that might use different indexing schemes. This allows a user to retrieve the data with a single query. The user enters index search criteria for an application group into the folder search fields and a document hit list is constructed based on the results of the query. The folder can be customized to provide the look and feel that is desired for the users of the Content Manager OnDemand system. The folder definition process allows the Content Manager OnDemand administrator to grant specific permissions for users of the folders.

For example, you can set up a folder, called CHKS, that contains different banking statements (such as checking and savings) stored in different application groups, defined in different applications, and created by different programs.

Users can have access to different document types through one folder. They can limit their search to a specific document type, or they can see the document type that each hit-list entry represents.



Name: CHKS

Description: Checking Statements

□ Display Document Location

Note Search
○ Hit List
● Retrieve
○ Note

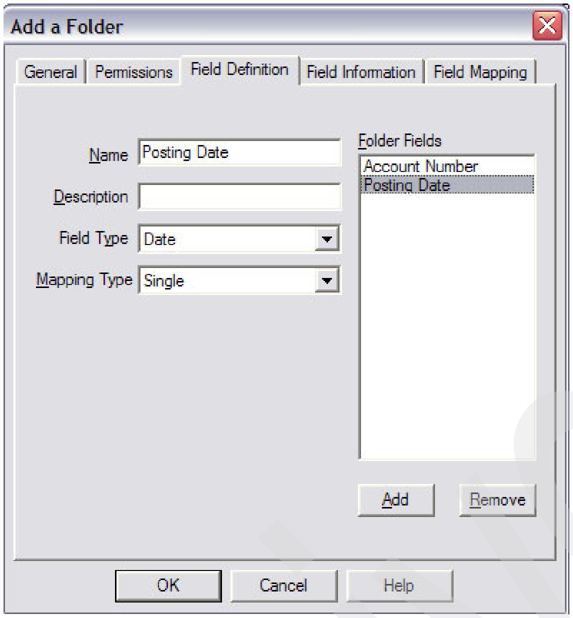
Application Groups:
AFP00300
ARSIVPR4
Cap1A
Cap1B
Cap1OAM
Cap1Sample
CARMSGRPLWB
CHKS2AFP
CHKSOS390
CHKSTAFPold
CHKSTMN2
CHKSTMNT
CHKSTMNTph

Selected:
CHKS

Figure 4-20 Folder General tab

Field information

When adding the folder fields that your users will be searching, it is important to note that the Field Type of the folder field *must* match the data type of the application group data field that it will be mapped to. You cannot define the folder field Posting Date as a Date/Time because the application group's data field POSTING_DATE is defined as a *Date* field. Figure 4-21 on page 171 and Figure 4-22 on page 172 provides these illustrations.



Name: Posting Date

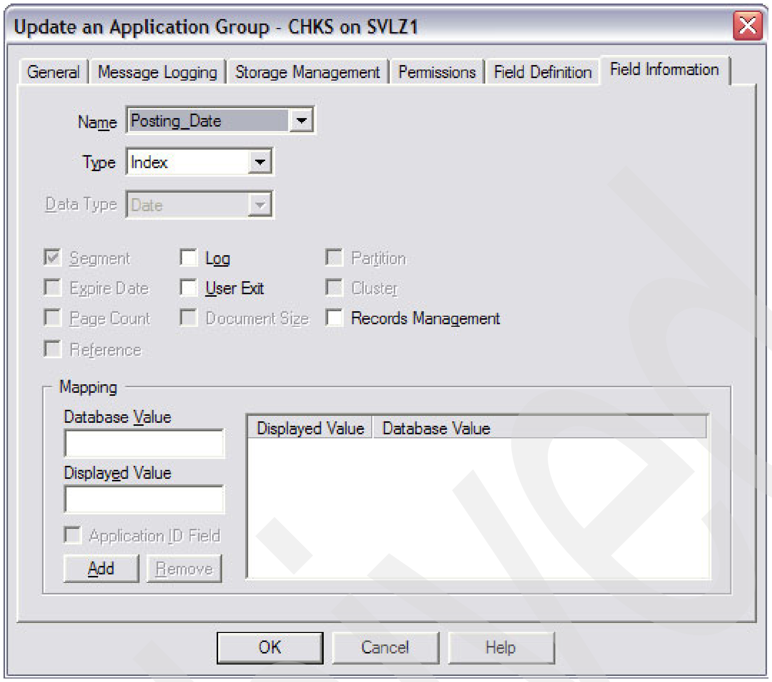
Description:

Field Type: Date

Mapping Type: Single

Folder Fields:
- Account Number
- Posting Date

Figure 4-21 Folder Field Definition tab


Update an Application Group - CHKS on SVLZ1

Name: Posting_Date

Type: Index

Data Type: Date

☑ Segment　□ Log　□ Partition

□ Expire Date　□ User Exit　□ Cluster

□ Page Count　□ Document Size　□ Records Management

□ Reference

Mapping

Database Value

Displayed Value

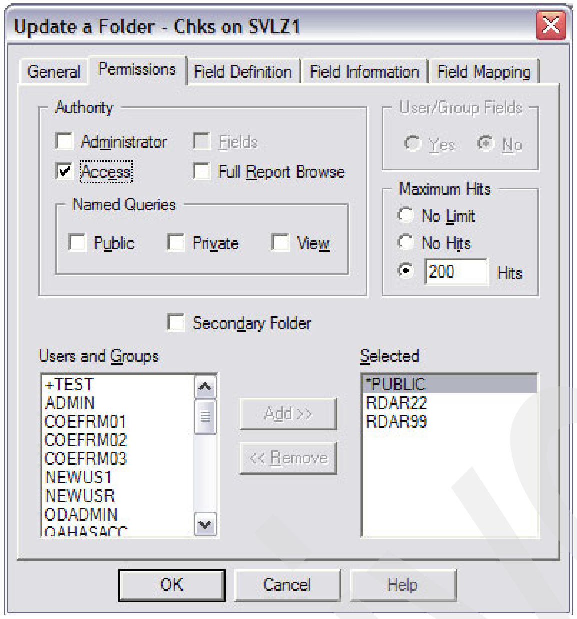
□ Application ID Field

Displayed Value	Database Value

Figure 4-22 Application Group Field Information tab

Maximum Hits setup

When defining your folder, the Maximum Hits (Figure 4-23 on page 173) sets the maximum number of document hit list entries to be returned by a folder query. Limiting the number of hits that can be returned from a query prevents performance degradation that might be experienced if you have an extremely large result that is to be returned from a query. If a query results in a large hit list that takes a long time to create, the cancel operation function on the Content Manager OnDemand Windows Client can be used to stop the creation of the hit list.


Update a Folder - Chks on SVLZ1

Authority

□ Administrator □ Fields
☑ Access □ Full Report Browse

Named Queries

□ Public □ Private □ View

User/Group Fields

○ Yes ● No

Maximum Hits

○ No Limit
○ No Hits
● 200 Hits

□ Secondary Folder

Users and Groups

+TEST
ADMIN
COEFRM01
COEFRM02
COEFRM03
NEWUS1
NEWUSR
ODADMIN
QAHASACC

Selected

*PUBLIC
RDAR22
RDAR99

Figure 4-23 Folder Permissions tab - Maximum Hits setup

4.4.5 Security

There are two types of security for Content Manager OnDemand:

- Authentication
- Permissions

Content Manager OnDemand provides internal and external security for authentication or permissions.

Authentication

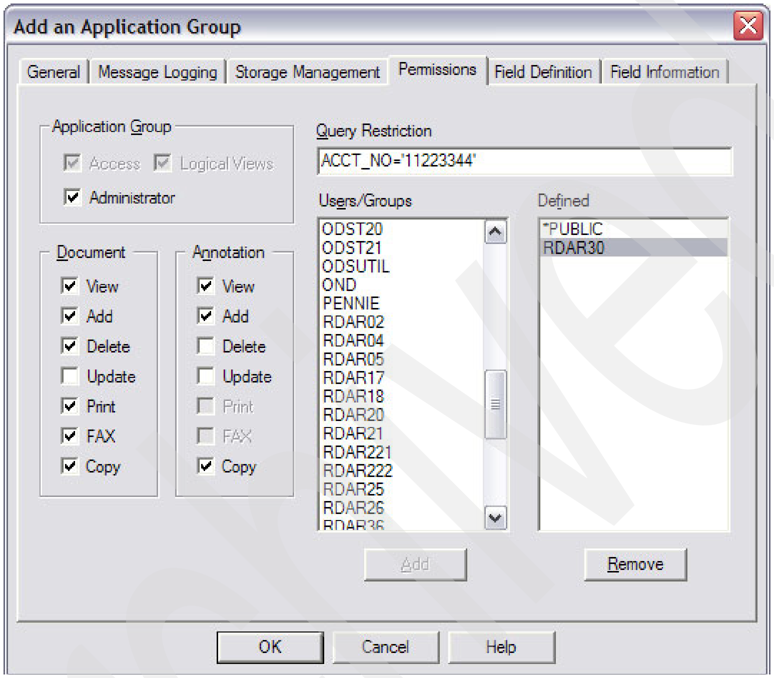
Content Manager OnDemand provides internal authentication through the use of user IDs and passwords that are stored in the Content Manager OnDemand system.

External authentication is provided by means of an installation modifiable security exit. Typically, this exit will be coded to issue a SAF call to the z/OS security manager. In this case, the User ID and Password are stored within the security manager, for example, RACF.

The authentication method is determined by means of a security flag setting in the ars.ini file: SRVR_FLAGS_SECURITY_EXIT=1.

Permissions

Content Manager OnDemand provides internal permissions checking by using the Query Restriction field to limit access to the application group data. In this case, it uses the user ID or group name to associate against the valid SQL statement in the query restriction field in the application group. This limits the results of any search that a user or group does. See Figure 4-24.



Application Group
☑ Access ☑ Logical Views
☑ Administrator

Document
☑ View
☑ Add
☑ Delete
□ Update
☑ Print
☑ FAX
☑ Copy

Annotation
☑ View
☑ Add
□ Delete
□ Update
□ Print
□ FAX
☑ Copy

Query Restriction
ACCT_NO='11223344'

Users/Groups
ODST20
ODST21
ODSUTIL
OND
PENNIE
RDAR02
RDAR04
RDAR05
RDAR17
RDAR18
RDAR20
RDAR21
RDAR221
RDAR222
RDAR25
RDAR26
RDAR36

Defined
*PUBLIC
RDAR30

Figure 4-24 Application Group Permissions tab - Query Restriction setup

External permissions is provided by means of an installation modifiable security exit. There are four external permission augments allowed:

- Access to a folder
- Access to an application group
- Restrict access to a specific document
- Control the SQL search criteria for searching folders

When a user attempts to access a folder, application group, or document or perform an SQL query, the arsuperm DLL is called. This executable must reside in the /usr/lpp/ars/bin/ directory and must have the APF extended attribute turned on.

The following statement must exist in the ars.ini file in order for the arsuperm DLL to be invoked for folder and application group permission checking:

```
SRVR_FLAGS_FOLDER_APPLGRP_EXIT=1
```

The following statement must exist in the ars.ini file in order for the arsuperm DLL to be invoked for document permission checking:

```
SRVR_FLAGS_DOCUMENT_EXIT=1
```

Note: Enabling document permission checking can greatly decrease Content Manager OnDemand performance when performing a document query.

The following statement must exist in the ars.ini file in order for the arsuperm DLL to be invoked for SQL query processing:

```
SRVR_FLAGS_SQL_QUERY_EXIT=1
```

Restriction: This exit runs in a threaded environment. The exit must be thread-safe.

Refer to the following manual for additional details for user security installation details: *IBM Content Manager OnDemand for z/OS: Configuration Guide,* GC27–1373.

4.4.6 Document Audit Facility

Document Audit Facility (DAF) is a feature of Content Manager OnDemand that lets users do basic approval routing through the Content Manager OnDemand Windows Client. An authorized user can change a status or audit field in the report, which actually changes the value of a database field in the application group. For example, you may want to allow certain users to click a button in the Client to approve vendor invoices, and allow other users to view only those approved invoices.

You may not need to use DAF with any of your reports, but if you do, it is better to plan ahead. You will need to create a field in the application group and in the folder to be used as the status or audit field that can be modified, so it is better to decide if you need to use this feature before setting up a definition for a report. To read more about Document Audit Facility, refer to *Content Manager OnDemand Guide*, SG24-6915.

4.4.7 Indexers

Content Manager OnDemand for z/OS has multiple indexers that you can choose from when defining an application. The right indexer depends on the type of data you are capturing and any data conversion that needs to be done during the indexing process.There are five different indexers that you can use with Content Manager OnDemand for z/OS:

- OS/390
- ACIF
- Generic
- PDF
- XENOS

OS/390 captures line print and AFP data. It has an exit point that can be used to capture most any other data type, much like the Generic Indexer.

ACIF captures line print and AFP data. ACIF can convert line print to AFP data.

The Generic Indexer can capture most any type of data. You provide it with an index control file that points to the offset and length of the data to capture from another data file.

The PDF indexer captures and indexes PDF documents.

The XENOS indexer lets you load AFP, Metacode/DJDE, or PCL print files into the system. You can use the Xenos transforms to extract index data from the input data and convert the input data into AFP or PDF documents.

In this section, we will focus on the OS/390 Indexer.

OS/390 Indexer

OS/390 Indexer allows the following input data types:

- Line print
- AFP (Does not convert line print to AFP.)
- Any other data type by using ANYSTORE exit

> **Note:** The OS/390 Indexer uses indexing parameters that look a lot like the parameters that are used by the ACIF Indexer. The OS/390 Indexer parameters are to a great extent interchangeable with the ACIF Indexer parameters. They may look alike, but there are differences.
>
> Be aware that you cannot take the indexing parameters from one indexer and expect it to work in the other.

The OS/390 Indexer is a one-pass indexer, where ACIF Indexer is a two-pass indexer. That is, the OS/390 indexer indexes and stores the data, including the indexes data, as it reads the data. ACIF Indexer first indexes the report, creates an output file, then goes back and stores the indexes and data.

The OS/390 Indexer handles line print and AFP, such as ACIF, but it does not convert line print to AFP. Other data types can be handled with the ANYSTORE exit.

When using the OS/390 Indexer, there is a unique parameter that reflects the report types you are loading in to Content Manager OnDemand. The parameter is:

```
INDEXSTYLE=
```

The valid values to use with this parameter are:

```
DOC
PAGE
PDOC
NODX
```

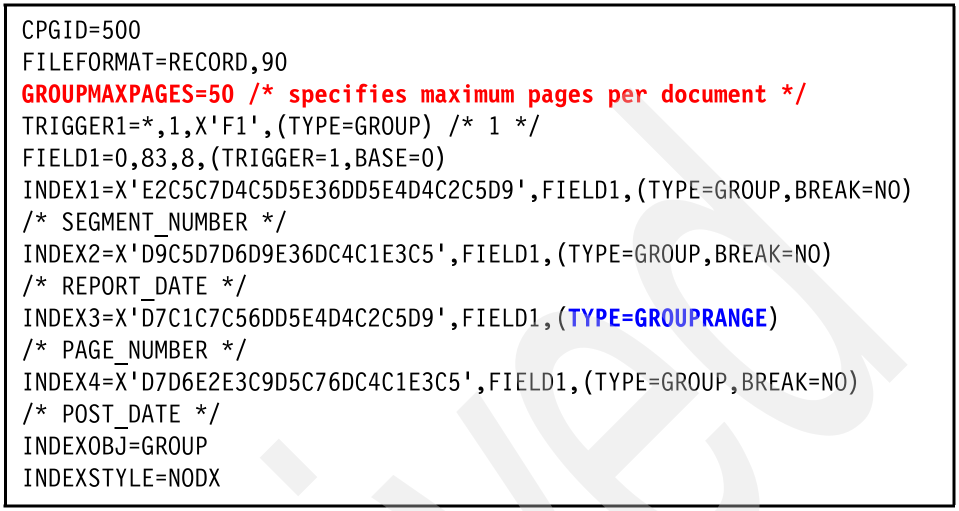
These are directly related to the Content Manager OnDemand V2 Report Types. If you are familiar with Content Manager OnDemand V2, these values for INDEXSTYLE will look very familiar. If you are new to Content Manager OnDemand, we will talk about what each of these mean below.

INDEXSTYLE=NODX

NODX (no index) reports are reports that either do not have obvious index values, or are very short and do not need to be broken up into documents.

The GROUPMAXPAGES parameter can be used to determine the number of pages included in each segment of the report. If no GROUPMAXPAGES value is specified, the default is 100 (pages).

Figure 4-25 illustrates sample indexing parameters for a typical NODX type report.



```
CPGID=500
FILEFORMAT=RECORD,90
GROUPMAXPAGES=50 /* specifies maximum pages per document */
TRIGGER1=*,1,X'F1',(TYPE=GROUP) /* 1 */
FIELD1=0,83,8,(TRIGGER=1,BASE=0)
INDEX1=X'E2C5C7D4C5D5E36DD5E4D4C2C5D9',FIELD1,(TYPE=GROUP,BREAK=NO)
/* SEGMENT_NUMBER */
INDEX2=X'D9C5D7D6D9E36DC4C1E3C5',FIELD1,(TYPE=GROUP,BREAK=NO)
/* REPORT_DATE */
INDEX3=X'D7C1C7C56DD5E4D4C2C5D9',FIELD1,(TYPE=GROUPRANGE)
/* PAGE_NUMBER */
INDEX4=X'D7D6E2E3C9D5C76DC4C1E3C5',FIELD1,(TYPE=GROUP,BREAK=NO)
/* POST_DATE */
INDEXOBJ=GROUP
INDEXSTYLE=NODX
```

Figure 4-25 sample indexing parameters for a typical NODX type report

INDEXSTYLE=PDOC

PDOC reports are transaction type reports, but have a high level index. For example, a bank may have a report that is organized by Branch Number. Within each Branch, the report is sorted on some column.

The GROUPMAXPAGES parameter can be used to determine the number of pages included in each segment. If no GROUPMAXPAGES value is specified, the default is 100 (pages). A new segment is started when either the high level index changes or the GROUPMAXPAGES value is reached.

Figure 4-26 on page 179 illustrates sample indexing parameters for a typical PDOC type report.

```
CPGID=500
FILEFORMAT=RECORD,90
GROUPMAXPAGES=50 /* specifies maximum pages per document */
TRIGGER1=*,1,X'F1',(TYPE=GROUP) /* 1 */
TRIGGER2=*,2,X'C2C1D5D2',(TYPE=GROUP) /*BANK */
TRIGGER3=*,46,X'4B',(TYPE=FLOAT) /* . */
FIELD1=1,11,3,(TRIGGER=1,BASE=0)
FIELD2=0,3,10,(TRIGGER=3,BASE=0)
FIELD3=0,83,8,(TRIGGER=1,BASE=0)
INDEX1=X'C2C1D5D26DC2D9C1D5C3C8',FIELD1,(TYPE=GROUP,BREAK=YES)
/* BANK_BRANCH */
INDEX2=X'D3D6C1D56DD5E4D4C2C5D9',FIELD2,(TYPE=GROUPRANGE2)
/* LOAN_NUMBER */
INDEX3=X'D7D6E2E3C9D5C76DC4C1E3C5',FIELD3,(TYPE=GROUP,BREAK=NO)
/* POSTING_DATE */
INDEXOBJ=GROUP
INDEXSTYLE=PDOC
```

Figure 4-26 sample indexing parameters for a typical PDOC type report

INDEXSTYLE=PAGE

PAGE reports are transaction type reports. The entire report is sorted by some column value. This sort key is used in the first and second indexes.

The GROUPMAXPAGES parameter can be used to determine the number of pages included in each segment. If no GROUPMAXPAGES value is specified, the default is 100 (pages).

Figure 4-27 illustrates sample indexing parameters for a typical PAGE type report.

```
CPGID=500
FILEFORMAT=RECORD,90
GROUPMAXPAGES=100 /* specifies maximum pages per document */
TRIGGER1=*,1,X'F1',(TYPE=GROUP) /* 1 */
TRIGGER2=0,2,X'D9C5D7D6D9E3',(TYPE=GROUP) /* REPORT */
TRIGGER3=*,46,X'4B',(TYPE=FLOAT) /* . */
FIELD1=8,3,10,(TRIGGER=1,BASE=0)
FIELD2=0,3,10,(TRIGGER=3,BASE=0)
FIELD3=0,83,8,(TRIGGER=1,BASE=0)
INDEX1=X'D3D6C1D56DD5E4D4C2C5D9',FIELD1,FIELD2,(TYPE=GROUPRANGE2)
/* LOAN_NUMBER */
INDEX2=X'D7C1C7C56DD5D66D',FIELD1,(TYPE=GROUPRANGE)
/* PAGE_NO_ */
INDEX3=X'D7D6E2E3C9D5C76DC4C2E3C5',FIELD3,(TYPE=GROUP,BREAK=NO)
/* POSTING_DATE */
INDEXOBJ=GROUP
INDEXSTYLE=PAGE
```

Figure 4-27 sample indexing parameters for a typical PAGE type report

INDEXSTYLE=DOC

DOC reports are traditional document reports, such as statements, invoices, and so forth. No indexes of type GROUPRANGE can be specified. If INDEXSTYLE parm not specified, it defaults to DOC.

Figure 4-28 on page 181 illustrates sample indexing parameters for a typical DOC type report.

```
CPGID=500
FILEFORMAT=RECORD,133
INDEXSTYLE=DOC
TRIGGER1=*,1,X'F1',(TYPE=GROUP) /* 1 */
TRIGGER2=*,3,’PAGE 1 ’,(TYPE=GROUP) /* PAGE 1 */
TRIGGER3=*,73,’ACCOUNT’,(TYPE=FLOAT) /* ACCOUNT */
TRIGGER4=*,3,’CONTENTS’,(TYPE=FLOAT) /* CONTENT */
TRIGGER5=*,19,’ENDING BAL’,(TYPE=FLOAT) /* ENDING BAL */
FIELD1=1,89,9,(TRIGGER=3,BASE=0)
FIELD2=1,87,11,(TRIGGER=4,BASE=0)
FIELD3=7,19,12,(TRIGGER=1,BASE=0)
FIELD4=0,87,16,(TRIGGER=5,BASE=0)
FIELD5=0,37,8,(TRIGGER=5,BASE=0)
INDEX1=‘ACCOUNT_NUMBER',FIELD1,(TYPE=GROUP,BREAK=YES) /* ACCT NUM*/
INDEX2=‘SSN_TAX_ID’,FIELD2,(TYPE=GROUP,BREAK=NO) /* SSN_TAX_ID */
INDEX3=‘CUST_NAME’,FIELD3,(TYPE=GROUP,BREAK=NO) /* CUST_NAME */
INDEX4=‘ENDING_BALANCE’,FIELD4,(TYPE=GROUP,BREAK=NO) /*ENDING BAL*/
INDEX5=‘POSTING_DATE’,FIELD5,(TYPE=GROUP,BREAK=NO) /*POSTING_DATE*/
```

Figure 4-28 sample indexing parameters for a typical DOC type report

Trigger with TYPE=GROUP and index with BREAK=YES determines when new documents start.

INDEXSTYLE=AFP

Advanced Function Printing (AFP) reports captured through the OS/390 indexer must already have been formatted into an AFP Data Stream (AFPDS). This can be done by using ACIF (AFP Conversion and Indexing Facility) or by any third-party program.

The OS/390 indexer looks for index values within the AFPDS, either in TLE or NOP records. ACIF, and other programs, can automatically generate the TLE records. The NOP records for use by the OS/390 indexer have a fixed format. For details on the TLE record formats, refer to the *MO:DCA Reference*, SC31-6802.

Figure 4-29 illustrates sample Indexer parameters for an AFP report type.

```
TRIGGER1=*,1,X’5A’,(TYPE=GROUP) /* AFP x’5A’ */
FIELD1=-0,1,14,(TRIGGER=1,BASE=0)
FIELD2=0,1,24,(TRIGGER=1,BASE=0)
FIELD3=0,1,18,(TRIGGER=1,BASE=0)
INDEX1=X’D796938983A8’,FIELD1,(TYPE=GROUP,BREAK=YES) /* Policy */
INDEX2=X’C39695A38595A3A2’,FIELD2,(TYPE=GROUP,BREAK=NO) /*Contents*/
INDEX3=X’C995A2A4998584’,FIELD3,(TYPE=GROUP,BREAK=NO) /* Insured */
INDEXSTYLE=AFP
```

Figure 4-29 Sample Indexer parameters for an AFP report type

ANYSTORE EXIT

The ANYEXIT parameter specifies a load module to call. This exit is responsible for reading the input file, breaking it apart into documents, and providing the index values for each document. This is usually used to capture data that is not line print or AFP, such as image data. For example: ANYEXIT=ANYSTI01

4.4.8 Exit points

In Content Manager OnDemand, it is possible to use exit points to customize and enhance the standard functionality of the product. There are a variety of exit points within the Content Manager OnDemand product. We provide some examples of the different types of operations and enhanced functions that are possible with the exits.

User exits allow you to execute a user-written program and then return processing control to Content Manager OnDemand after the completion of your user-written program. There are a few different kinds of exits, as you will see below. Content Manager OnDemand provides sample code for each exit that can serve as input to your user-written programs. When using these exits, it allows you the ability to update index values through a print request, cleaning up data as it is loaded into Content Manager OnDemand, and accessing external security managers. Infinite examples can be provided for what is possible using the Content Manager OnDemand exits. We provide some samples to serve as a guide for creating customized user exits programs:

- OS/390 Indexing exits (Figure 4-30 on page 183).
  - Input exit: Enables you to add, delete, or modify records in the input file before they are processed by the OS390 Indexer. The primary purpose of this exit allows modification of the input records before the indexer sees them. Common usages would be to add carriage control characters so as to truncate records.

- Index exit: Allows you to modify or ignore the records that the OS/390 indexer writes in the index object file. The program, specified in the OS/390 Indexer parameter, receives control just before a record is written to the index object file. The user-written program can tell OS/390 to use the record, not to use the record, or to perform some sort of editing on the record before inserting it into the index object file.
- Anystore exit: This exit is responsible for reading the input file, breaking it apart into documents, and providing the index values for each document. This is usually used to capture data that is not line print or AFP, such as image data.

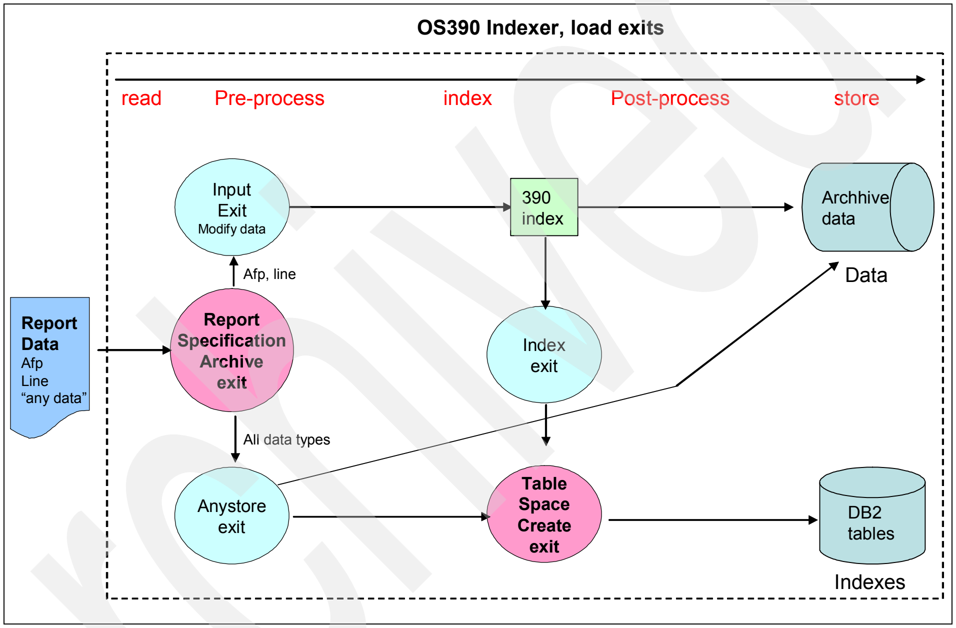


Figure 4-30 OS/390 indexer and load exits

- System administration: The system administration exits are illustrated in Figure 4-31 on page 185.
  - System Log exit: You can configure Content Manager OnDemand to record information, warning, and error messages in the system logging facility. Content Manager OnDemand can record messages about system activity, such as when users log on and log off the system, and application group activity, such as when clients query and retrieve data. In addition, you can configure Content Manager OnDemand to send the messages to the ARSLOG installation exit.

- Unified login exit: The Content Manager OnDemand unified login exit (ARS.PTGN) provides a means for an installation to allow a user to run the Content Manager OnDemand command-line utilities (such as **`arsload`**) without needing to specify a user ID and password.
- Report Specification Archive exit: The Content Manager OnDemand Report Specifications Archive Definition Exit allows an installation to modify some of the parameters used by Content Manager OnDemand when document data is being captured (loaded) by the ARSLOAD program. The following parameters can be modified:
  - The Application Group name.
  - The Application name.
  - The name of the object server to be used for data storage.
  - The name of the Storage Node to be used for data storage.
  - The indexer parameters set.
  - The input file control character type, logical record length, and record format.
- Security exit: The Content Manager OnDemand security system interface exit allows an installation to secure related processing of the following activities or events:
  - Logon or change password.
  - Add User ID or Delete User ID by using the Content Manager OnDemand administrative functions.
  - Access to a Content Manager OnDemand folder.
  - Access to a Content Manager OnDemand application group.
  - Restrict access to specific documents.
  - Control the SQL search criteria used for searching folders.
- Preview exit: The Content Manager OnDemand Windows Client preview exit allows an installation to process document data before the document is presented to the client. The client preview exit can be used to add, remove, or reformat data before the document is presented to the client.
- Tablespace create exit: The Content Manager OnDemand tablespace creation exit allows an installation to take action when Content Manager OnDemand creates a tablespace, table, or index tables that will be used to store application index data. The exit is not called for the Content Manager OnDemand *system tables.*

  For table and index creation, the installation can alter the SQL that will be used to create the table or index.

Note: Typically, the custom exits are called frequently and as a performance consideration, we recommend that your exits be placed in the Link Pack Area (LPA). You should also be aware that as you apply maintenance to your Content Manager OnDemand system, these custom exits may need to be recompiled and linked based on the applied modules. These custom exits should be thread safe and re-entrant.

Re-entrance means that a single copy of the executable code exists in memory and only program control structures and private memory are allocated for each concurrent instantiation of the program.

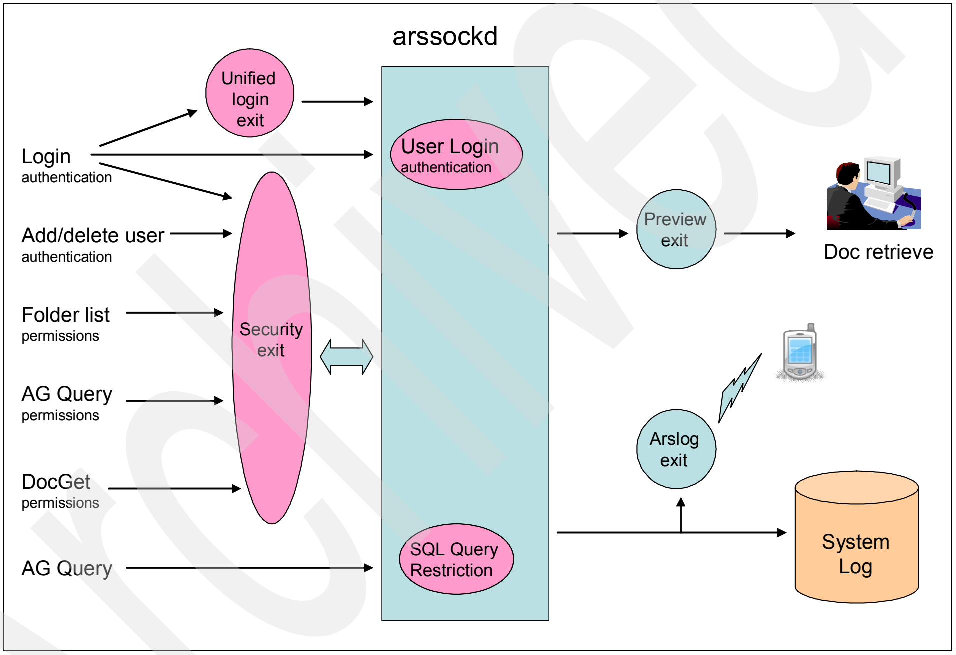


Figure 4-31 Content Manager OnDemand server security, preview and logging exits

- Application administration.
  - Structured APIs: Structured APIs allow customer applications in CICS, IMS, TSO, or batch environments the ability to invoke the server functions described in Table 4-3.

Table 4-3 Server functions and their description

Function	Description
LOGON	Establishes a connection to the Content Manager OnDemand library server. After a successful logon, the server returns a list of authorized folders that can be accessed by a specific user.
FOLDER OPEN	Identifies the folder name to be processed by subsequent search and retrieve requests.
HIT LIST	Requests the Content Manager OnDemand server return a list of items matching the user supplied search criteria.
RETRIEVE	Retrieves a document from a Content Manager OnDemand archive.
LOGOFF	Allows users to log off a Content Manager OnDemand server.
RELEASE	Frees storage areas used in the execution of the logon, folder open, hit list, and retrieve functions.
RELEASEC	Frees the FolderCriteriaStructure created by FOLDER OPEN.
RELEASED	Frees the DocumentStructure created by RETRIEVE.
RELEASEH	Frees the HitListStructure created by HIT LIST.
RELEASEL	Frees the FolderListStructure created by LOGON.

Refer to the following manual for additional details relating to Exit Points and their implementation: *IBM Content Manager OnDemand for z/OS: Configuration Guide,* GC27–1373.

4.5 Application setup and verification

In this section, we discuss the following application setup and verification tasks using Check Statement as a report example:

- Report setup
- Permissions setup
- Load, retrieve, and print verification
- User acceptance

4.5.1 Report setup

Report design and definition are key to a successful implementation of a Content Manager OnDemand system. Knowledge of the data that is to be indexed, loaded, retrieved, and printed, along with knowledge of Content Manager OnDemand best practices, results in the most efficient and easy-to-use system possible. In this section, we consider the processes that are followed when defining a Content Manager OnDemand report and present hints and tips to help in the design and implementation process.

The system components that are required for creating, retrieving, viewing, and printing a Content Manager OnDemand report are:

- A storage set
- An application group
- An application
- A folder
- A printer

These are the elements that are required by your Content Manager OnDemand administrator to define and create a report definition that can then be used to index and load data into Content Manager OnDemand.

Figure 4-32 depicts the relationship these elements have with a Content Manager OnDemand.

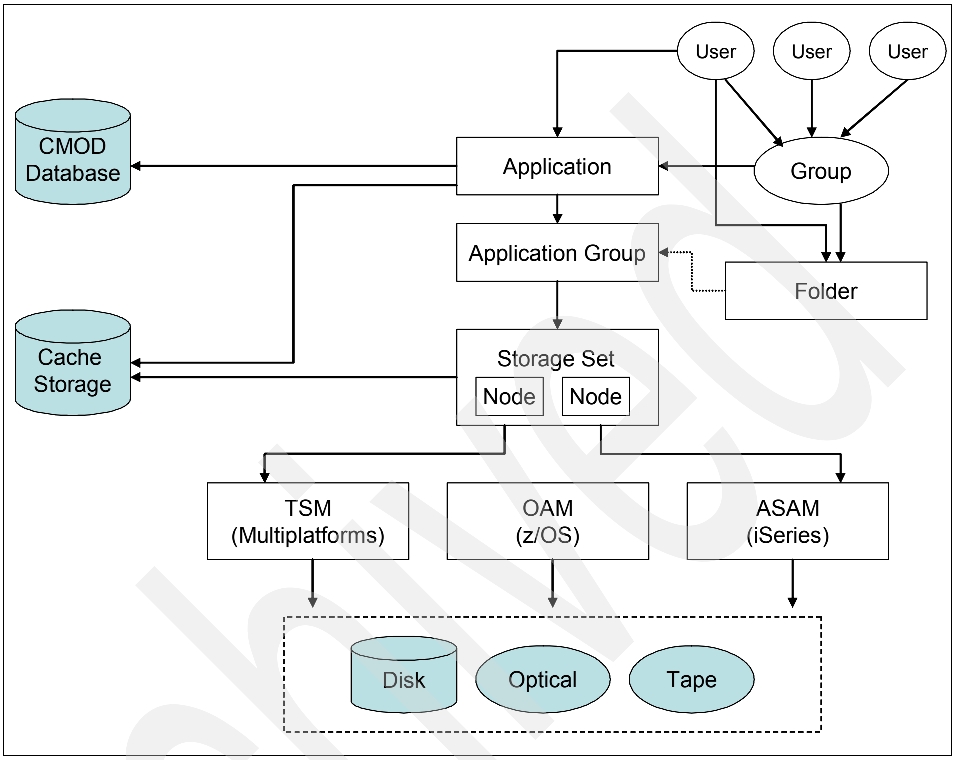


Figure 4-32 Illustrates the relationship of the elements with Content Manager OnDemand

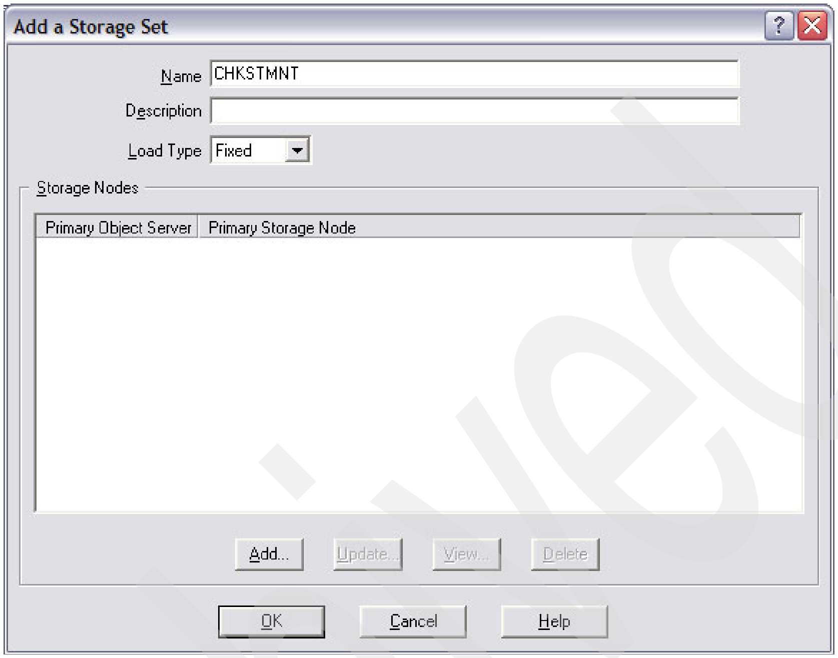
As we set up the Check Statement report, we use the following steps:

1. Define the storage set.

   A *storage set* contains one or more storage nodes that can be used by several application groups that have the same archive storage requirements. For example, a storage set can be used to maintain data from different application groups that must retain documents for the same length of time.

   When you are looking to begin your Report setup, one of the key questions that must be answered is what is the retention requirements for this report. Once that question has been answered, you can then add the storage node to the Storage Set.

We provide an example showing a storage node being added to the CHKSTMNT storage Set. Figure 4-33 shows the Add a Storage Set window.



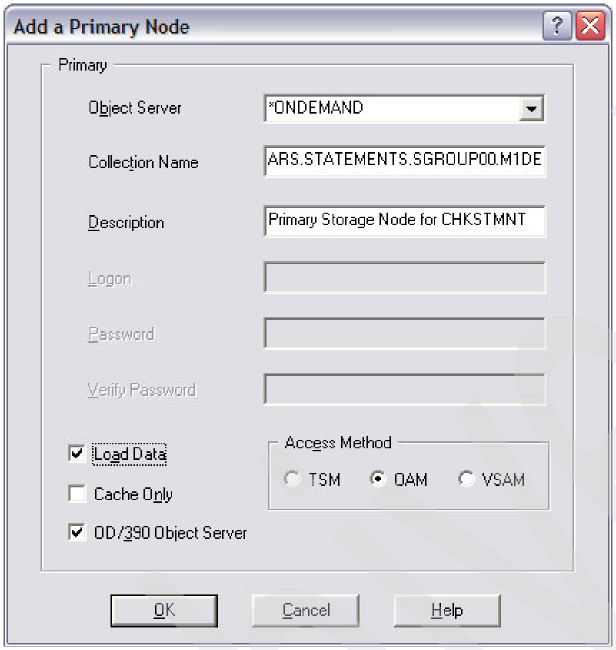
Name CHKSTMNT

Description

Load Type Fixed

Figure 4-33 Storage Set setup - Add CHKSTMNT

In Figure 4-33 on page 189, click **Add** to add a Primary Node, as shown in Figure 4-34.



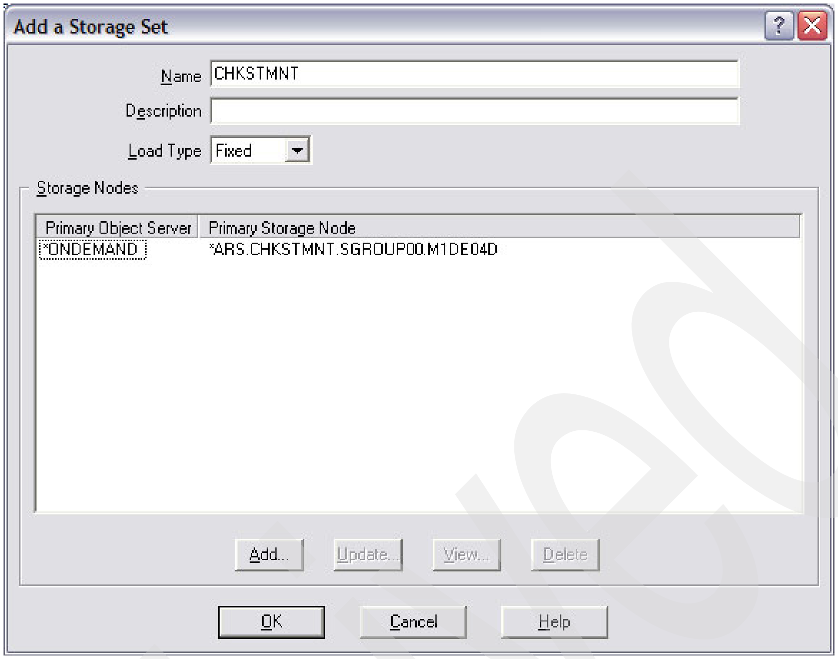
Object Server: *ONDEMAND
Collection Name: ARS.STATEMENTS.SGROUP00.M1DE
Description: Primary Storage Node for CHKSTMNT
Logon:
Password:
Verify Password:
☑ Load Data
□ Cache Only
☑ OD/390 Object Server
Access Method: ○ TSM ● OAM ○ VSAM

Figure 4-34 Primary Node setup for CHKSTMNT storage set

Fill out the information, and click **OK** and the Primary Storage Node is added to your Storage Set (see Figure 4-35 on page 191).

Note: If you are using the OAM Archive Manager, you need to ensure that the Primary Storage Node that you want to Load to in OAM has the Load Data box checked and an Access Method of OAM checked.

Verify that you have selected the correct Access Method for the Storage Manager that you will use.



Add a Storage Set

Name: CHKSTMNT

Description:

Load Type: Fixed

Storage Nodes

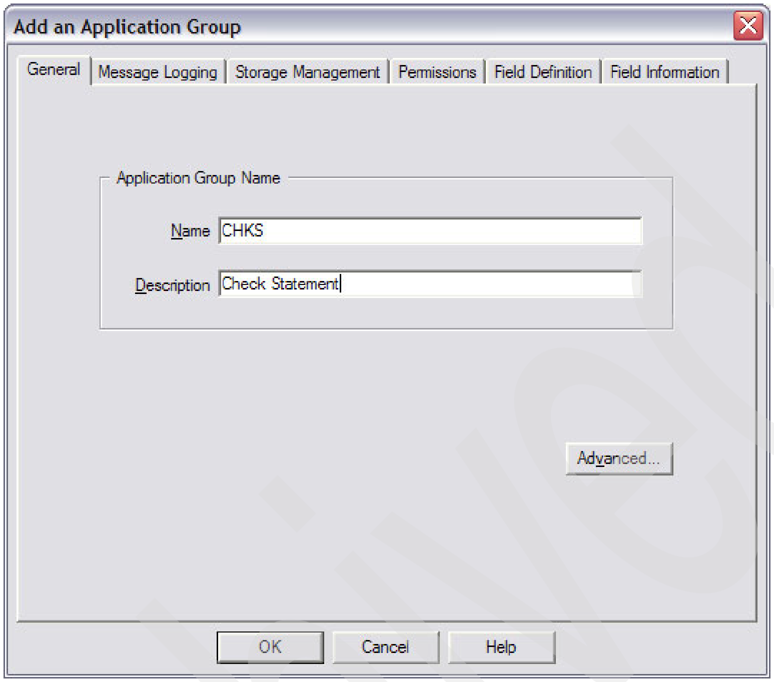
Primary Object Server	Primary Storage Node
*ONDEMAND	*ARS.CHKSTMNT.SGROUP00.M1DE04D

Figure 4-35 Storage Set setup - with added Primary Storage Node for CHKSTMNT

2. Create the application group.

   The *application group* is a collection of one or more applications that contain common indexing and storage management requirements. The application group contains the database information that is used to load, search for, and retrieve reports. The application group defines the data that is to be loaded into the database. Before you can begin the creation of the application group, you must understand, from the application, the fields that the user will want to search on, for example, Account Number and Report Date. In the following illustrations, we will demonstrate setting up an Application Group.

Figure 4-36 shows the Add an Application Group window.



Application Group Name

Name: CHKS

Description: Check Statement

Figure 4-36 Defining a new application group

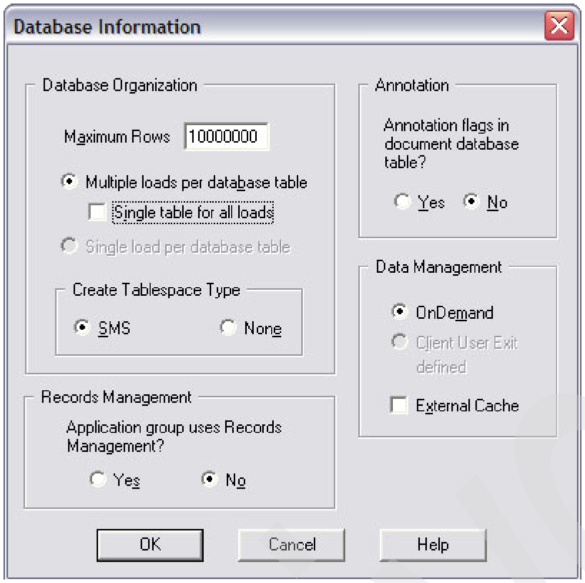
Enter the application group name and description. Click **Advanced** and the database information window displays (see Figure 4-37 on page 193). Define the following information:

– Database information

The database information section of the application group definition process (Figure 4-37 on page 193) requires that you decide how many rows will be stored in each database table or, alternatively, if you want to store all data loaded for the application group in a single database table. These values are important for system performance and maintenance.

When storing a fixed number of rows to a database table, the default value for Maximum Rows is 10 million (see Figure 4-37 on page 193).

Selecting **Single Table for All Data Loads** results in only one Application Group data table being created. This option is most frequently used when the total number of rows loaded to your application group data table will be relatively small.



Database Information

Database Organization

Maximum Rows 10000000

● Multiple loads per database table

□ Single table for all loads

○ Single load per database table

Create Tablespace Type

● SMS ○ None

Records Management

Application group uses Records Management?

○ Yes ● No

Annotation

Annotation flags in document database table?

○ Yes ● No

Data Management

● OnDemand

○ Client User Exit defined

□ External Cache

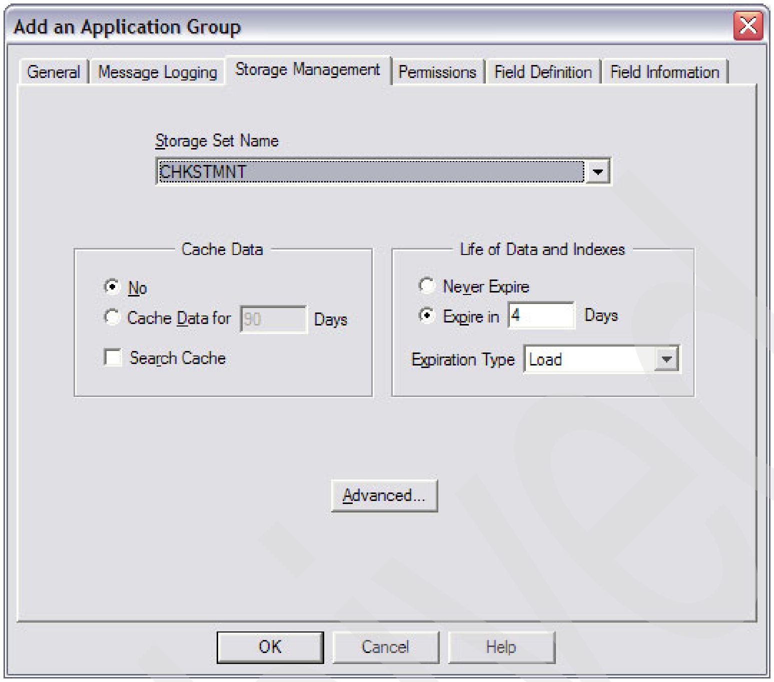
Figure 4-37 Application Group Database Information tab

- Storage Management tab

  The Application Group Storage Management tab shown in Figure 4-38 on page 194 allows you to define the life of the data and indexes and how they will be expired.

  Life of Data and Indexes settings determine the length of time that report data, indexes, and resources are maintained in the Content Manager OnDemand system before they are deleted from an application group. The report data, indexes, and resources can be maintained indefinitely if set to never expire, or they might be kept for up to four days, as shown in Figure 4-38 on page 194. Typically a value of years is selected.

  If you elect to use OAM storage manager to drive when you expire the Indexes in Content Manager OnDemand, the ARSEXOAM utility will be invoked. Refer to the following manual for additional details relating to ARSEXOAM expiration processing for details: *IBM Content Manager OnDemand for z/OS: Administration Guide*, GC27–1374.



Storage Set Name: CHKSTMNT

Cache Data

● No
○ Cache Data for 90 Days
□ Search Cache

Life of Data and Indexes

○ Never Expire
● Expire in 4 Days

Expiration Type: Load

Figure 4-38 Application Group Storage Management tab

The *Expiration Type* determines how report data, indexes, and resources are expired. If the expiration type is Load, an input file at a time can be deleted from the application group. The latest date in the input data, and the life of data and indexes, determine when Content Manager OnDemand will delete the data. Data that has been stored in archive storage is deleted by the storage manager based on the archive expiration date. We recommend that you set Expiration Type to Load.

– Field Definition tab

The Field Definition tab (Figure 4-39 on page 195) is where you define the database fields for the Application Group.

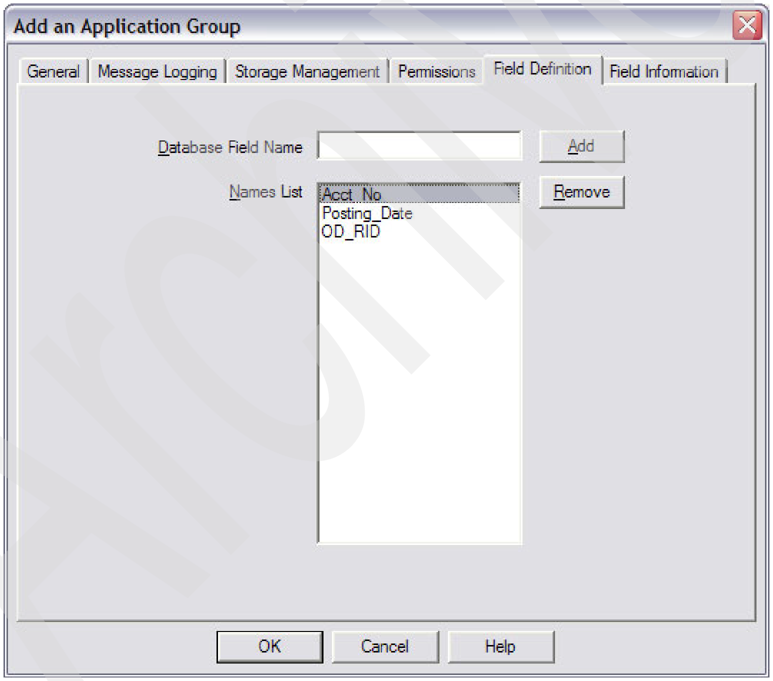
A *Database Field Name* should have the following characteristics:

- Can contain one to 18 characters (bytes).
- Must begin with A through Z; other characters can include A through Z, 0 through 9, @, $, _, and #.
- Cannot be any of the Content Manager OnDemand reserved keywords:

```
annot
comp_len
```

```
comp_off
comp_type
doc_len
doc_name
doc_off
doc_type
pri_nid
res_comp_type
resource
sec_nid
```

- Cannot be any of the keywords reserved by the database manager. See the documentation provided with your database manager product for a list of reserved words.
- Can be mixed case. However, the case does not create a unique name (repDate is the same as repdate).
- Must be unique to the application group.



Database Field Name

Names List
Acct_No
Posting_Date
OD_RID

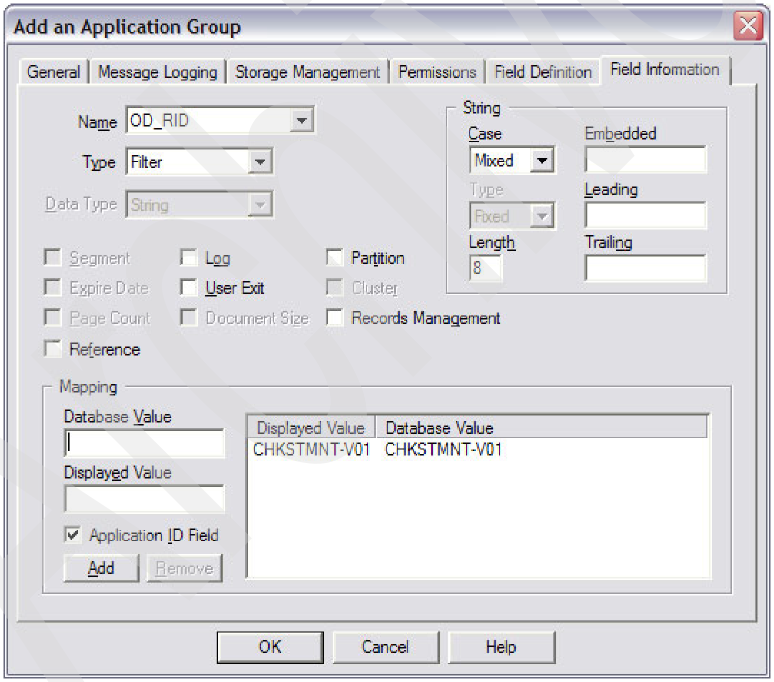
Figure 4-39 Application Group Field Definition tab

– Field Information tab

The Field Information tab (Figure 4-40) is used to define the attributes of the database fields that make up the Content Manager OnDemand report index data. These attributes determine the characteristics of the index data and control many aspects of loading and processing data in the system. A database field must be added for each index value that is required by applications to be part of the application group.

If multiple applications will be part of the application group, select the Application ID Field to uniquely identify each application in an application group. If it is possible that more than one application will be part of an application group, you should select the Application ID Field.

Note: If you need to add a database field after the application group is added to the Content Manager OnDemand system, you now have the ability to do this.



Name: OD_RID
Type: Filter
Data Type: String

□ Segment □ Log □ Partition
□ Expire Date □ User Exit □ Cluster
□ Page Count □ Document Size □ Records Management
□ Reference

String
Case: Mixed
Embedded:
Type: Fixed
Leading:
Length: 8
Trailing:

Mapping
Database Value:
Displayed Value:
☑ Application ID Field
Add Remove

Displayed Value	Database Value
CHKSTMNT-V01	CHKSTMNT-V01

Figure 4-40 Application Group Field Information tab

When defining the field attributes, you must take into consideration the following attributes:

- Type: The type of the database field determines how the field is used. There are two main types of fields: index and filter.

  An *Index* type should be used with a field definition if it is to uniquely identify a document or if it is frequently used when searching for documents in the application group. Designating a field as an index serves to enhance query performance.

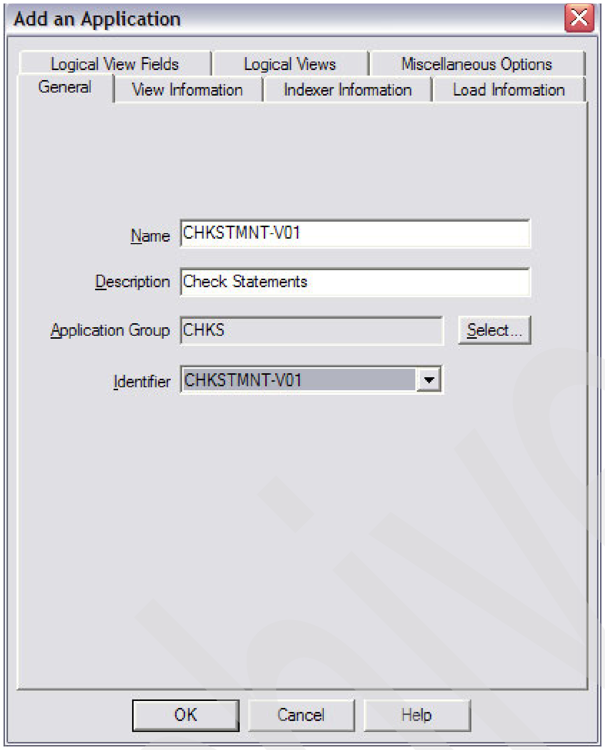
  A *Filter* type should be used if the field does not uniquely identify a document and it is usually used in conjunction with an index field during folder queries.

- Segment: Segment is the date or date and time field that is used to limit the number of tables that are searched during a folder query. If the application group is defined for multiple loads per database table, we highly recommend that you define a segment date for the application group.
- Application ID Field: The Application ID Field is used to identify an application within an application group when you create an application group that contains more than one application

3. Define the application.

   An application defines the physical and logical characteristics of the actual report data that is to be indexed and loaded, associates the data with an application group, and specifies the type of indexing process to be performed on the data. It also defines any logical views to be put in place for the users and determines any special print options to be used with the data.

Figure 4-41 shows the Add an Application window.



Name: CHKSTMNT-V01

Description: Check Statements

Application Group: CHKS

Identifier: CHKSTMNT-V01

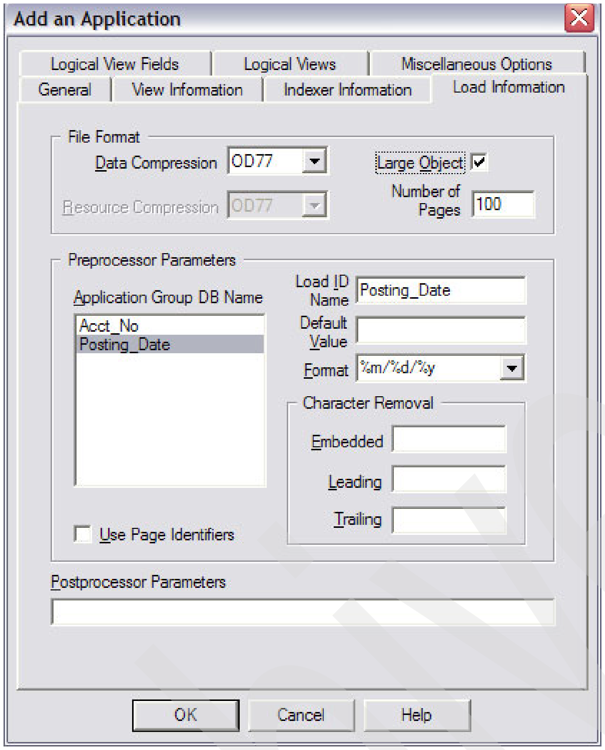
Figure 4-41 Define an application

In the Add an Application window, fill in the following information:

– Load Information tab

The Load Information tab (Figure 4-42 on page 199) allows you to specify the processing and resource information that the Content Manager OnDemand loader uses to load the input data onto storage volumes and to load the associated index data into the Content Manager OnDemand database. The File Format, Preprocessor Parameters, and Postprocessor Parameter are defined as part of load information:

- File Format: Provides settings that control how the Content Manager OnDemand system compresses and stores documents and resources.
- Preprocessor: Specifies processing that is carried out on database fields prior to indexing data.



File Format

Data Compression: OD77

Large Object: ☑

Resource Compression: OD77

Number of Pages: 100

Preprocessor Parameters

Application Group DB Name: Acct_No, Posting_Date

Load ID Name: Posting_Date

Default Value:

Format: %m/%d/%y

Character Removal

Embedded:

Leading:

Trailing:

□ Use Page Identifiers

Postprocessor Parameters:

Figure 4-42 Application Load Information tab

- Large Object support

  In the File Format section, you can select **Large Object**. Large Object support is used to improve load and retrieve performance by dividing the document into smaller parts for loading and creating index information based on this document segmentation. Documents are retrieved faster due to the smaller segment sizes that are sent across the network.

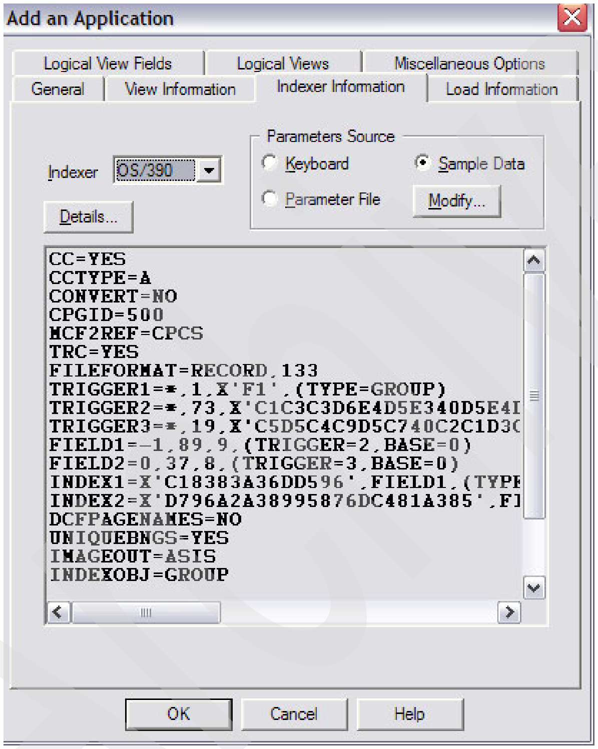
  When a document is retrieved for viewing, only the first part of the document is returned from the server to the client. Additional parts of the document are sent from the server to the client as the user moves to different pages in the document. Invoking Large Object support generates an INDEXOBJ=ALL entry in the indexing parameters that enables the generation of large object indexing information.

  When Large Object is selected, the number of pages parameter must also be specified. Number of pages determines how many pages will be included by Content Manager OnDemand in each large object segment.

– Indexer Information tab

When building the indexing parameters, you can use the Graphical Indexer to assist you with defining the Triggers, Indexes, and Fields. You will need to transfer a sample of the report data file down to your work station in *binary* format (FB with the same LRECL of the input report data file). For example, if the Check Statement data file is FB LRECL 133, then you would transfer the sample file as Binary FB LRECL 133.

Once you have it transferred, you can then open up the file by selecting **Sample Data** and clicking **Modify** in the Parameter Source panel on the Load Information tab (see Figure 4-43).

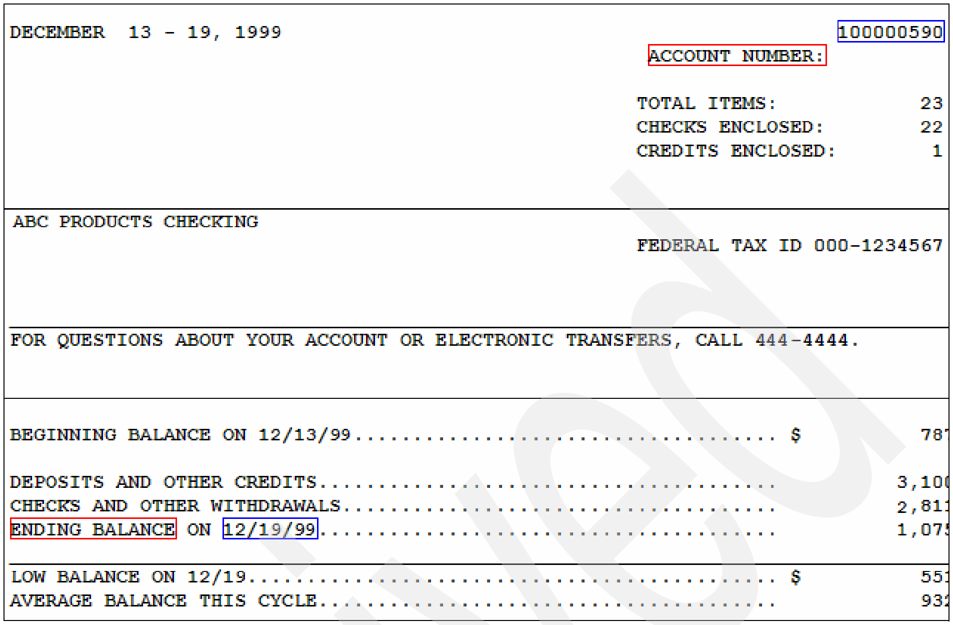


Indexer: OS/390

```
CC=YES
CCTYPE=A
CONVERT=NO
CPGID=500
MCF2REF=CPCS
TRC=YES
FILEFORMAT=RECORD,133
TRIGGER1=*,1,X'F1',(TYPE=GROUP)
TRIGGER2=*,73,X'C1C3C3D6E4D5E340D5E4I
TRIGGER3=*,19,X'C5D5C4C9D5C740C2C1D3C
FIELD1=-1,89,9,(TRIGGER=2,BASE=0)
FIELD2=0,37,8,(TRIGGER=3,BASE=0)
INDEX1=X'C18383A36DD596',FIELD1,(TYPE
INDEX2=X'D796A2A38995876DC481A385',FI
DCFPAGENAMES=NO
UNIQUEBNGS=YES
IMAGEOUT=ASIS
INDEXOBJ=GROUP
```

Figure 4-43 Application Indexer Information tab - Parameters Source modification

Figure 4-44 on page 201 depicts the Graphical Indexer highlighting the Triggers, Indexes, and Fields that were generated with the Graphical Indexer. This indexer information is what is used by the ARSLOAD program to perform the indexing and segmentation for each document of this report.



```
DECEMBER  13 - 19, 1999                                               100000590
                                                                ACCOUNT NUMBER:

                                                       TOTAL ITEMS:            23
                                                       CHECKS ENCLOSED:        22
                                                       CREDITS ENCLOSED:        1

 ABC PRODUCTS CHECKING
                                                       FEDERAL TAX ID 000-1234567

FOR QUESTIONS ABOUT YOUR ACCOUNT OR ELECTRONIC TRANSFERS, CALL 444-4444.

BEGINNING BALANCE ON 12/13/99....................................  $         78
DEPOSITS AND OTHER CREDITS.......................................          3,10
CHECKS AND OTHER WITHDRAWALS.....................................          2,81
ENDING BALANCE ON 12/19/99.......................................          1,07

LOW BALANCE ON 12/19.............................................  $         55
AVERAGE BALANCE THIS CYCLE.......................................            93
```

Figure 4-44 A sample check statement report indexing using the Graphical Indexer

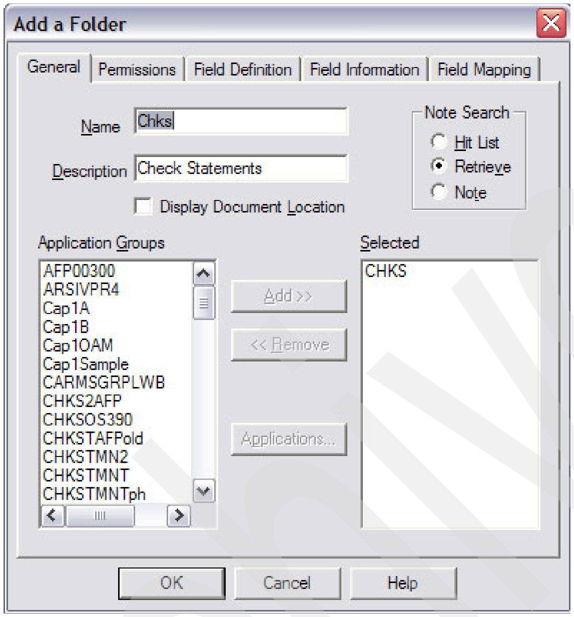
4. Define the folder.

   A folder is the interface that allows you to search for reports and documents that have been stored in the Content Manager OnDemand system. Content Manager OnDemand users will enter index search criteria for an application group into the folder search fields and a document hit list will be presented based on the results of the query. The folder definition process allows the Content Manager OnDemand administrator to grant specific permissions to various users as needed.

   When you are defining your folder (see Figure 4-45 on page 202), there are several key points that should be considered, as they can impact document retrieval performance:

   – Display Document Location: Displays an icon next to each document on the document hit list that tells you the location of this object (Disk, Tape, or Optical).

– Note Search: We recommend that you set the annotation parameter in the application group Advanced settings to handle annotation storage and display. When the application group annotation parameter is set to **Yes**, an annotation flag is set in the database when a user adds an annotation to a document. When an annotation exists for a document, a note icon is displayed in the document hit list.



Name: Chks

Description: Check Statements

□ Display Document Location

Note Search
○ Hit List
● Retrieve
○ Note

Application Groups:
AFP00300
ARSIVPR4
Cap1A
Cap1B
Cap1OAM
Cap1Sample
CARMSGRPLWB
CHKS2AFP
CHKSOS390
CHKSTAFPold
CHKSTMN2
CHKSTMNT
CHKSTMNTph

Selected:
CHKS

Figure 4-45 Define a folder CHKS

4.5.2 Permissions setup

For security measures, you can assign a user to a group. When you assign a user to a group, the user obtains the permissions of the group. For example, suppose you create a group and authorize the group to open the Chks Information folder. Any user that you assign to the group automatically obtains permission to open the Chks Information folder.

If you assign a user to more than one group, the user normally obtains the permissions of all of the groups. However, there are exceptions to this rule. See information about permissions in the *IBM Content Manager OnDemand for z/OS: Administration Guide*, SC27-1374 for details.

Folder permissions

You can set folder permissions at the folder, group, and user levels.

Setting permissions at the folder level provides all Content Manager OnDemand users and groups that are not otherwise given permissions with the permissions that you define.

Setting permissions at the group level provides all of the users that you assign to the group with the permissions that you define. Group level permissions override folder level permissions.

Setting permissions at the user level provides a specific user with the permissions that you define. User level permissions override group level permissions and folder level permissions.

Application group permissions

You can set application group permissions at the application group, group, and user levels.

Setting permissions at the application group level provides all Content Manager OnDemand users and groups that are not otherwise given permissions with the permissions that you define.

Setting permissions at the group level provides all of the users that you add to the group with the permissions that you define. Group level permissions override application group level permissions.

Setting permissions at the user level provides a specific user with the permissions that you define. User level permissions override group level permissions and application group level permissions.

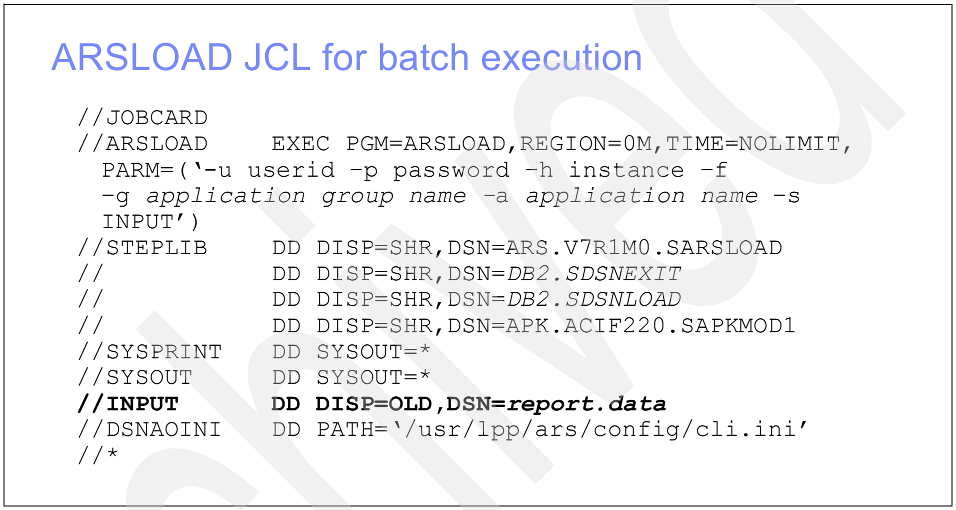
Query restriction

You can use the Query Restriction field within the application group to limit access to application group data. When you specify a user name to the application group and key a valid SQL statement in this field, you limit the results of any search for that user.

4.5.3 Load, retrieve, and print verification

After you set up a report definition (storage set, application group, application, and folder), the next step is to load and retrieve a sample report. There are two ways that you can load the report into Content Manager OnDemand:

- From the JES Spool with ARSLOAD executing as a started task. This task monitors the JES Spool based on a special class or class and destination.
- As a Batch Load using a special PARM field to provide the application group and application. Figure 4-46 provides an example.



ARSLOAD JCL for batch execution

```
//JOBCARD
//ARSLOAD     EXEC PGM=ARSLOAD,REGION=0M,TIME=NOLIMIT,
  PARM=('-u userid -p password -h instance -f
  -g application group name -a application name -s
  INPUT')
//STEPLIB     DD DISP=SHR,DSN=ARS.V7R1M0.SARSLOAD
//            DD DISP=SHR,DSN=DB2.SDSNEXIT
//            DD DISP=SHR,DSN=DB2.SDSNLOAD
//            DD DISP=SHR,DSN=APK.ACIF220.SAPKMOD1
//SYSPRINT    DD SYSOUT=*
//SYSOUT      DD SYSOUT=*
//INPUT       DD DISP=OLD,DSN=report.data
//DSNAOINI    DD PATH='/usr/lpp/ars/config/cli.ini'
//*
```

Figure 4-46 Example ARSLOAD JCL for a batch load

Note: The Content Manager OnDemand server (ARSOCKD) must be active when load jobs are run.

There are various types of viewers for retrieving your documents. Upon completion of the loading, you will need to access your installed viewers to validate the loaded data.

Once you have loaded the data successfully, you now have the ability to print the data to ensure your data prints successfully. You have the flexibility to print the data to your local server that may be defined to your desktop or to a predefined server printer.

4.5.4 User acceptance

The system implementation process is only successful if it is accepted by the users. Some ideas that help to increase user acceptance include:

- Identifying application team leads from the various users groups.
- Involving them early in the design process and decision making (folder naming and search criteria, document indexing, retention requirements, print and load requirements, and pre-processing and post-processing requirements.)
- Providing them with education that will allow them to train and provide consulting to their group members.
- Involving them in the functional testing process.

User acceptance is an ongoing process. As the system is enhanced, It is important to provide information updates to your team leads concerning these new enhancements. This will allow them to communicate the information to their groups. The team leads will be able to provide any additional training. They will be able to implement or recommend any procedural changes that are needed to maximize the benefits of using Content Manager OnDemand.

4.6 Functional testing

The goal of functional testing is to make sure that the system does what we expect it to do. The functional tests that you set up should reflect how the system is to be used in your organization.

In this section, we discuss functional testing for the following areas:

- Load processing
- Retrieval processing
- Printing
- Expiration processing
- Custom exits
- Backup and recovery

4.6.1 Load processing

Functional Load processing tests may include:

- Loading data through batch job:
  - With the various indexers that you will use
  - With your data types
- Loading data from the JES spool:
  - With the various indexers that you will use
  - With your data types
- Testing with your load exits.
- Analyzing the different compression methods to determine the best option. Data compression is a trade-off between CPU consumption, DASD consumption, and network traffic. You will have to decide, for each of your data types, the amount of compression that you require.

4.6.2 Retrieval processing

The retrieval processing functional tests ensure that what you loaded was loaded correctly and that it can be retrieved correctly.

- Retrieve the stored data using all the different clients that you will be using.
- Check that the document indexing is what you expect it to be.
- Check that the complete document is displayed.
- Make sure any (preview or security) exits you use are tested.
- Make sure that any transforms you use are tested.

4.6.3 Printing

The printing functional tests include:

- Printing all stored data types and verifying that the data has printed correctly.
- Printing to all printer types (both local and server printers) and verifying that the data has printed correctly.
- Validating the banner page and its contents are printed correctly.

4.6.4 Expiration processing

The expiration processing functional tests are designed to verify that the index data and the object data are both deleted based on the retention requirements. To establish a test expiration process, perform the following:

- Set the Life of Indexes and Data field so that the indexes are deleted by Content Manager OnDemand in one day.
- Set the SMS constructs for the Archive Manager such that the stored objects are expired and deleted in one day.
- Perform the load of your test data and verify its success.
- Execute the expiration process:
  - Performing ARSMAINT
  - Performing ARSEXOAM
- Verify that both the indexes and the stored objects are deleted.

4.6.5 Custom exits

Functional testing of the custom exits occurs in conjunction with the load and retrieval functional tests. These tests should also be run in parallel to make sure that there is no contention on resources and that the exits are re-entrant and thread safe (where needed).

4.6.6 Backup and recovery

For backup and recovery of Content Manager OnDemand, there are three areas that need to be included:

- Content Manager OnDemand Server Software, including:
  - Custom exits
  - Configuration files
  - User-defined files
- Content Manager OnDemand DB2 database, including:
  - Content Manager OnDemand system tables
  - Content Manager OnDemand application group (AG) data tables
- Storage Manager configuration testing, including:
  - Storage manager database(s)
  - Object data

For functional backup and recovery, the following testing and procedures should be taken in consideration:

- Create backup procedure.
- Create restore procedures.
- Document the process.
- When you have a tested sample of data (loaded and viewed), back up the data, delete the DB2 data and the storage manager data, and then perform a restore and verify that the data is still viewable correctly.
- Provision for storing your backup data at an off-site location.

4.7 Performance testing and considerations

Once functional testing is complete and you feel comfortable that the system is doing what you expect it to do, you need to do performance testing of the system. The goal of performance testing is to ensure that the system does what you expect it to do under load.

In this section, we discuss the following performance testing and considerations:

- Develop test cases based on requirements.
- Load testing.
- Retrieval testing.
- Tuning considerations based on test results.
- Periodic performance tuning.

4.7.1 Develop test cases based on requirements

The developed test cases should:

- Reflect the loads that we expect on the system.
- Reflect the document data types and sizes.
- Reflect realistic Content Manager OnDemand DB2 table sizes.
- Reflect the production system environment.

The goal is to run a set of tests that will either:

- Produce the same results that we expect to get in production. This can only happen if the test environment is identical to the production environment.
- Produce results that will allow us to extrapolate and predict what our results will be like in the production environment. This is the case when we are running the test environment under certain known constraints and we believe that we can predict that the removal of these constraints (in production) will

lead to a proportional increase in performance. The potential problem in this case is that when the constraints are removed, new bottlenecks will emerge that did not exit in the constrained environment.

4.7.2 Load testing

When loading reports into Content Manager OnDemand, there are two different methods:

- TCP/IP: In this case, the index data and the storage objects are transferred from the load process to the arssosckd server through the TCP/IP network. The main advantage of this process is that the load process can run anywhere (where the appropriate Content Manager OnDemand software components are installed) in the network, thus allowing the data to be stored to the library server and appropriate object server.
- DB2 direct: In this case, the load process stores the index data and storage objects directly to DB2 and the storage manager.

You will need to determine, based on your situation and your data, which is the best method to use.

Report data: The report data that is used in the testing should match the typical data that would be loaded in a production environment, in terms of the report size, number of indexes, document size, and data type.

Parallel loads: Content Manager OnDemand allows for multiple load jobs to run in parallel. Typical scenarios include:

- Running load jobs that are on multiple systems on which the report data to be loaded is located, rather than first transferring the report data to a specific system and then running the load job on that system.
- If the quantity of data to be loaded exceeds the capacity of a single load task to load the data within a specified time frame (window), then running parallel load jobs will allow for a reduction of the load elapsed time.

4.7.3 Retrieval testing

Retrieval tests should also be run in an environment as reflective as possible of the production environment. Automated testing tools and scripts can be used to mirror the expected user work load. Care should be taken to simulate both the numbers of users and the behaviors of these users (what types of documents would they retrieve, how often, and how many indexes).

4.7.4 Tuning considerations based on test results

The first set of performance tests will produce some results. If these results are acceptable, then all is well; otherwise, you will need to investigate (with the help of your systems' personnel) the reasons for the lack of performance. There are potentially many areas that need monitoring and tuning. A full discussion is beyond the scope of this book. However, we include these areas for your references:

- DB2
- OAM
- TCP/IP
- UNIX System Services
- WebSphere Application Server
- Java
- Exits

4.8 Training

Typically, there are three groups of individuals that are critical for training. In this section, we will discuss these types of training:

- System personnel
- Content Manager OnDemand report administrators
- User training

4.8.1 System personnel

This training includes the knowledge to perform everyday monitoring and ongoing product support. If new systems or sub-system skills are needed, then those are usually obtained through specialized IBM education courses.

Training in Content Manager OnDemand system internals is typically delivered during the IBM Content Management Lab Services engagement. This training is tailored more for your specific implementation environment.

4.8.2 Content Manager OnDemand report administrators

Report administrators require an in-depth understanding of the Content Manager OnDemand z/OS report analysis and index processes. They typically would attend training provided by IBM Learning Services. These courses are periodically offered at IBM training centers world-wide; alternatively, they can be offered onsite for groups of administrators or if timing constraints exist. The two main courses are:

- IM110 - Introduction to IBM Content Manager OnDemand: This course is designed for individuals responsible for creating and loading applications into the Content Manager OnDemand for z/OS system. It provides the basic understanding of all areas of Content Manager OnDemand for z/OS.
- OD105 - IBM Content Manager OnDemand System Administration: This course is specifically designed for individuals who are experienced at indexing and loading documents and have a need for a greater in-depth knowledge of the Content Manager OnDemand system, either for system administration, maintenance, or troubleshooting purposes.

4.8.3 User training

User training is typically developed in-house. The Content Manager OnDemand administrators provide the Train the Trainer courses. This training can either be on the job or instructor based, and in some cases video or Web-based technology is used.

> **Note:** Training schedules need to be in sync with the implementation plan. To maximize training costs payback, training should be delivered no more than a week before the skills learned will be applied.

4.9 Deployment into production

In this section, we discuss the following training:

- Automating the system process
- System documentation
- System monitoring

4.9.1 Automating the system process

The more automation you can build into the system, the easier it will be to maintain it in the future. Processes to automate include:

- Loading data. For example:
  - Automate your batch load by capturing report data from the JES Spool.
  - Schedule batch jobs to run at specific times.

- Expiring data. For example, automate the expiration processing of your Content Manager OnDemand object data and Content Manager OnDemand index data.
- Backups.

  Automate the backup of all of your Content Manager OnDemand system components.
- Server recovery processes.

  Automate the fail-over process or re-start processes for the Content Manager OnDemand servers.
- Event notification.

  Automate the event notification process of failed Content Manager OnDemand components allowing the appropriate personnel resource to be directly notified.

4.9.2 System monitoring

After completing your initial performance testing, you will put the system into production (and maybe perform some additional tuning as real users start using the system). At this point, you may believe that the system's performance will meet your expectations and the performance tuning will have been successfully concluded, but that is incorrect. The problem is that over time the system environment will change. For example, more and different types of data will be added to the system, the number of users may increase and the usage by each user may increase, new hardware may be installed, new software may be installed, and disk drives may fill up.

One way of dealing with this is to establish a performance baseline after the system has gone into production (and is fully utilized), then to periodically (based on you environment) run the performance tests to see if any performance degradation has occurred and to address the issues as necessary. The time period for running these tests could be anything between three and 18 months, depending on the volatility of your system's environment.

Between your periodic performance tests, system's personnel will notify you if your applications are negatively impacting the system and users will notify you if the performance degrades dramatically.

4.9.3 System documentation

There are different types of system documentation:

- The design documentation needs to be started at the very initial stage of the implementation process.
- The procedural documentation needs to be created and verified during the functional testing period.
- The system monitoring documentation needs to be created and tested during the load testing period.

Note: All documentation needs to be kept current. As your Content Manager OnDemand system changes, you will need to update the documents accordingly.

4.9.4 System references

There are multiple sources for system references:

- Content Manager OnDemand manuals for system administrators:
  - *Introduction and Planning Guide*, GC27-1438: This reference is of primary interest to administrators who plan to install, administer, and use Content Manager OnDemand. It is also intended for people in an organization who plan hardware, software, network, recovery, and applications for business systems.
  - *Configuration Guide*, GC27-1373: This reference is of primary interest to people responsible for installing and configuring software products. This book provides the information an installer needs to install and configure a Content Manager OnDemand system. This book specifically addresses the Content Manager OnDemand server software and related software programs, configuring the services required to operate the server, and setting up batch jobs and maintenance tasks to run automatically on a regular schedule.
  - *Administration Guide*, SC27-1374: This reference is of primary interest to administrators that are responsible for working with and maintaining a Content Manager OnDemand system. Some administrators can use this book and the tools described in it to define reports to the system. Other administrators can use this book and the tools described in it to maintain users, groups, printers, storage sets, and so forth. Still other administrators can use the administrative commands described in this book to maintain the database and cache storage, extract documents from the system, and so forth.

  - *Indexing Reference*, SC27-1375: This reference guide is of primary interest to administrators and other people in an organization who are responsible for preparing data to be stored in Content Manager OnDemand.
  - *Messages and Codes*, SC27-1379: This reference guide is of primary interest to Content Manager OnDemand administrators who require more detailed information about returned messages.
- Content Manager OnDemand manuals for optional products:
  - *Migration Guide*, LY37-3746: This reference is intended for Content Manager OnDemand system administrators, planners, and programmers and for database administrators who support V2 Content Manager OnDemand and are migrating to V7 of Content Manager OnDemand.
  - *Content Manager OnDemand Distribution Facility Installation and Reference Guide*, SC27-1377: This guide is for people that plan for, install, configure, and upgrade ODF for an organization.
  - *Content Manager OnDemand Web Enablement Kit Implementation Guide*, SC27-1376: This guide is for people responsible for planning, installing, and configuring IBM Web Interface for Content Management.
- Content Manager OnDemand user manuals:
  - *User's Guide*, SC27-0836: This publication introduces you to the basic features of the Content Manager OnDemand Windows Client. It is of primary interest to people who use Content Manager OnDemand to search for, retrieve, and view documents.
- Online help windows: The online help includes details of things that can be done using the client. The Windows Client online help complements the *User's Guide*, while the Administration Client online help complements the *Administration Guide*.
- Online Content Manager OnDemand manuals at IBM Content Manager OnDemand Information Center can be found at:

  http://publib.boulder.ibm.com/infocenter/cmod/v8r4m0/index.jsp
- Overall online Content Manager OnDemand information can be found at:

  http://www.ibm.com/software/data/ondemand

  Go to the specific product page by selecting the product (either Content Manager OnDemand for Multiplatforms, for i5/OS, or for z/OS and OS/390) and click **Go**. From the specific product page, you can:
  - Click the **Information Center** link to get the online Information Center.
  - Click the **Product manual** link to obtain all manuals (in different languages) for the specific product.

  - Click the **Product support** link to get to the IBM Redbooks publications, technotes, and white papers.
  - Click other links such as **Demos**, **Developer resources**, and **Web casts** to get other information.
- White papers:
  - *IBM Content Manager OnDemand for OS/390 and z/OS Best Practices for Performance*, found at:

    `http://www.ibm.com/support/search.wss?rs=2207&tc=SSQHWE+SSEPC6&rank=8&dc=DA480+DB100&dtm`
  - *IBM Content Manager OnDemand for z/OS System Monitoring*, found at:

    `http://www.ibm.com/software/sw-library/en_US/products/C191233C99643W62/`

    Search for “System Monitoring”
- IBM Redbooks publications:
  - *Content Manager OnDemand Backup, Recovery, and High Availability*, SG24-6444
  - *Content Manager OnDemand Guide*, SG24-6915
  - *Integrating IBM Tivoli Workload Scheduler and Content Manager OnDemand to Provide Centralized Job Log Processing*, SG24-6629

Part 2

Case studies

For the chapters in this part, we provide implementation case studies for each platform.

We cover the following case studies based on the platforms:

- Content Manager OnDemand for Multiplatforms:
  - Large commercial bank case study
  - Application service provider case study
- Content Manager OnDemand for i5/OS:
  - International bank case study
  - Telephone company case study
- Content Manager OnDemand for z/OS
  - Financial institution case study
  - Federal agency case study
  - Educational institution case study

5

Case studies for multiplatforms

In this chapter, we describe two case studies for Content Manager OnDemand for Multiplatforms implementations. For each case study, we describe the company background and the business requirements. Using the information, techniques, and recommendations we addressed earlier in the book, we discuss how to plan for the implementation and the implementation process for each case study.

The cases studies we cover in this chapter are for:

- A large commercial bank
- An application service provider

5.1 Large commercial bank case study

This case study details the implementation of Content Manager OnDemand to provide report management for internal users and banking statements to both internal users and external customers for a fictitious US commercial bank holding company. In addition, it detaisl the steps taken to consolidate multiple archive systems into a single Content Manager OnDemand system.

5.1.1 Company background

We refer to this fictitious large commercial bank as the bank.

This bank is a large US-based commercial bank holding company with branches in more than a dozen states and other offices in over 40 states. The bank has over 150 billion in assets under its control.

5.1.2 Business requirements

The following requirements were identified as deliverable for this project:

- A single archive large enough to support over 8000 existing different reports and statements while allowing for growth of over 20% per year.

  This bank had in place 10 existing archive and reporting systems due to mergers over the years. Most of these systems were no longer supported or were costing the bank a large amount of money to maintain. In addition, there was no central query capability to find documents or reports across the many archives. This bank desired this ability in the new system.

  The systems ranged from mainframe based to stand-alone PCs. Support for these archive systems was across multiple departments within the bank. This needed to be consolidated into a single support group and a single architecture.

- Reduction of costs.

  The existing mainframe system was expensive to manage and maintain. The system used older optical based hardware to manage data and also used extensive amounts of mainframe processing power to perform the nightly load. In addition, it had reached capacity and was plagued by numerous outages.

  Other systems had reached their end of life. Some companies who produced these systems were no longer in existence. In addition, it was getting harder to locate or acquire spare parts for the existing systems, some of which were PC based.

- Customer access to statements.

  The bank wanted to provide a new service to allow customers to view their banking statements online. This new system needed to be able to support this capability.

- Internal Web-based access to all data.

  The bank wanted to eliminate the need to support PC based software internally to provide searching and viewing capability. The bank's goal was to only support basic internet browsers for searching and retrieval of the report and statement data.

5.1.3 Planning

A team from IBM and the bank was formed to implement the Content Manager OnDemand solution. IBM was contracted to convert all existing reports from the existing archive systems to Content Manager OnDemand and to install and configure the solution.

The following information was gathered during the planning phase of the project:

- The existing system, which was mainframe based, had over 5000 different report definitions. The system had no versioning to help isolate changes to the reports over the years of its operation. After analysis, it was determined that over 8000 different definitions would be needed to support this migration and implementation.
- 95% of the data was EBCDIC line data. Large Object support will be used to segment the larger reports for online viewing.
- 5% of the data was IBM Advanced Function Printing (AFP) data streams. This data is comprised of customer statements that would be viewed by external customers in addition to bank employees working in a customer service role.
- It was critical to get day-forward data loaded to Content Manager OnDemand as soon as possible to alleviate the load on the existing archive systems. This was necessary due to capacity and performance concerns.
- Simplified indexing consisting of a report name and date would be used for all but the top 200 accessed reports. This will allow for a faster implementation.
- Some internal users had a requirement to extract data from various reports to be used in analysis. This data needed to be extracted and re-purposed for use in spreadsheets. This customer was a licensed user of Monarch, a report mining and data analysis tool by Datawatch, and it will use the integration that exists between Content Manager OnDemand and Monarch for this purpose.

- New IBM eServer pSeries hardware would be acquired for this project. The production systems would be pSeries 570s and contain eight CPUs and 32 GB of memory. DASD storage would be EMC DMX disk arrays. Archive storage would be mirrored EMC Centera boxes with 20 TB of available space. Web servers running IBM WebSphere Application Server would be Windows 2000 based servers.
- Internal users would use the Content Manager OnDemand Custom Client for report viewing. This client runs as a WebSphere application.
- External users would access statements through a custom banking application developed internally by the bank. This application would use the Content Manager OnDemand Web Enablement Kit JAVA APIs and IBM AFP2PDF transforms to deliver these statements as PDF documents to the bank's customers.
- Migration of data from the existing archive systems would be performed. Data would be extracted and reloaded into Content Manager OnDemand. Retention would be maintained by the using the date of the original file.

5.1.4 Implementation

The implementation of Content Manager OnDemand involved many steps. Software was installed in four separate server environments: a development server, quality assurance (QA) server, production server, and disaster recovery server. Workstations to be used by the application developers were set up with the Content Manager OnDemand administrative client along with the Windows client to test retrieval. We also set up test Web servers to begin development of the custom banking interface to Content Manager OnDemand and to test the Custom Client.

Installation and configuration

The Content Manager OnDemand software was installed on pSeries 570 AIX boxes using the following products:

- AIX V5.2
- DB2 UDB V8.2
- Tivoli Storage Manager V5.2
- Content Manager OnDemand for Multiplatforms V8.3
- MVS download (Used for downloading reports from the JES Spool)
- WebSphere Application Server V5.1
- OnDemand Custom Client V1 Services Offering

Note: The software versions listed were current at the time of the implementation. Always verify currently released and supported levels before implementing a system.

Each UNIX based product was installed onto local disk attached directly to the pSeries. In this way we did not need to be concerned about being able to activate each node individually. All configuration files were duplicated and installed on each individual node except for the TSM history and volume files, which were placed on mirrored disk.

Figure 5-1 illustrates the implemented architecture. The production setup used mirroring technology from EMC to create the disaster recovery (DR) site. Symmetrix Remote Data Facility (SRDF) was used to mirror all activity at the production site to the DR site and a set of mirrored disks. These disks contained the Content Manager OnDemand database, Tivoli Storage Manager database, and Content Manager OnDemand cache file systems. EMC Centera replication was used to create the DR copy of all data objects stored by TSM. Additionally, development and servers were installed, although these configurations were smaller and with four CPUs and 16 GB of memory and without the mirroring configuration.

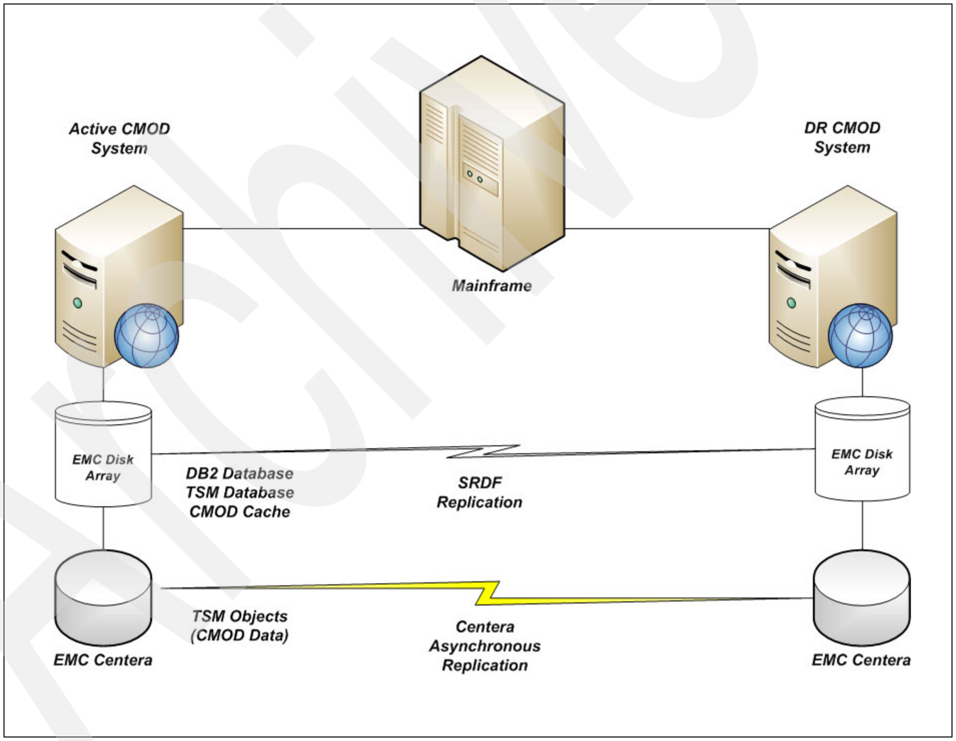


Figure 5-1 Content Manager OnDemand for Multiplatforms case study 1 - System architecture

"Content Manager OnDemand for Multiplatforms case study 1 - Client architecture" on page 224 illustrates the client architecture. Internal power users had the Content Manager OnDemand Windows Client installed. These users included the internal group that needed to access the report and re-purposed the data through Monarch. Internal casual users were given access to Content Manager OnDemand using the Content Manager OnDemand Custom Client developed by IBM Content Management Lab Services. This WebSphere application was installed on a number of WebSphere servers running in a Network Development (ND) environment. The final access method configured was to allow external banking customers to access their monthly statements from a Web browser using a custom application developed by the bank. This application was tied into the bank's Internet Banking Portal.

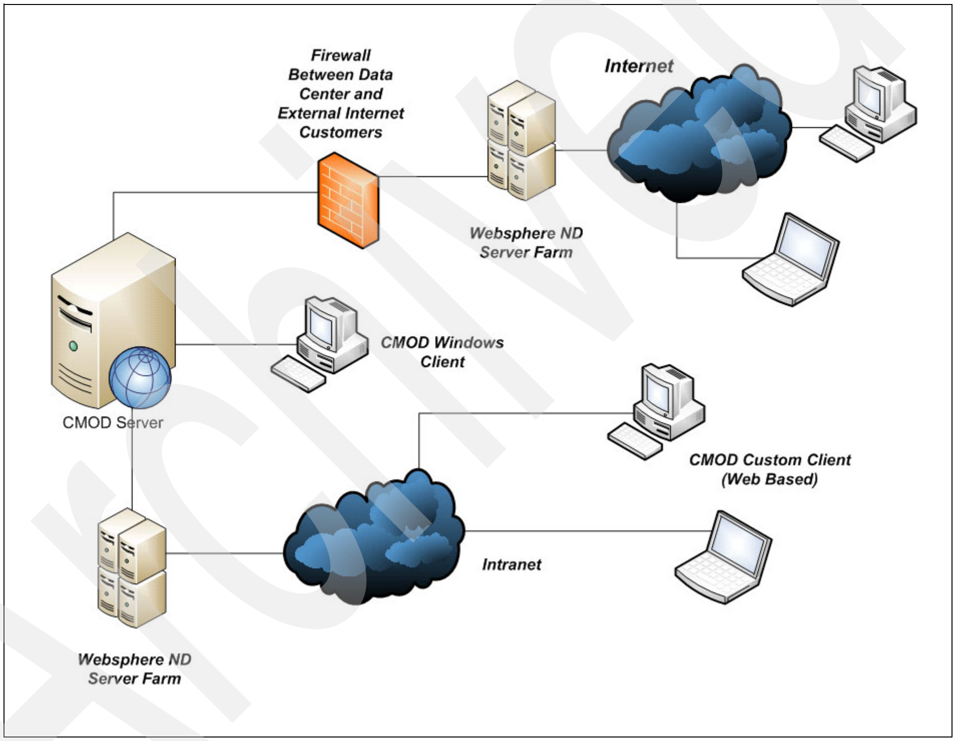


Figure 5-2 Content Manager OnDemand for Multiplatforms case study 1 - Client architecture

Application design

The bank needs to support thousands of users and will be storing terabytes of data in its Content Manager OnDemand archive. The need for a high performance design to support both the ingestion of this data and the retrieval of it made this stage critical to the success of this project.

The application design phase of this project began with a detailed analysis of the customer's existing systems. Extensive investigations of the number of existing report types and volumes was necessary to properly configure and size Content Manager OnDemand. To develop the application design, the following information was gathered:

- The number of reports
- Retention needs and legal requirements
- Total number of report types
- Report ingestion requirements, time window, and volumes
- Total number of users and client environment
- Network considerations for data download and system mirroring
- Network considerations for Web and Windows Client access

The analysis of the above information allowed the team to build an architecture with Content Manager OnDemand to support the bank's requirements:

- System requirements

  Critical to this design was the ability to handle the existing volumes and to support a growth of 20% per year. The pSeries 570 was chosen due to its ability to expand the existing hardware without the need to replace it. The DASD storage requirements and archive storage requirements were given to the bank's storage team to allocate and provide. The high level architecture was shown previously in Figure 5-1 on page 223.

- Application requirements

  It was during this phase that decisions were made to streamline the application development environment. During the investigation, over 8000 separate report types were found to exist. But no usage information was available to determine who was using these reports. It was decided to simplify the indexing requirements to allow the greatest number of reports to be ingested in the shortest length of time. This was done by limiting the indexing to just the Report Name and Date. Other decisions were made to provide additional query capabilities for external customers and internal customer support staff on a specific set of reports and all statements. We will detail two representative samples, one report, and one statement application in the following section.

- Storage requirements

  The reports and statements were spread across multiple TSM storage pools. These pools were designed using the following criteria. The first criteria used was data age. We kept the older migrated data in its own storage pool, separate from the day forward data[1]. In addition, we needed to separate data based on different retention. The bank had needs for two, five, seven, and 10 years. We also separated the reports from the statements based on retrieval patterns. Since the statements were to be retrieved by specific users and some of these were external customers, we wanted to ensure that there was no internal contention for the TSM volumes.

Application group, application, and folder design

The reports-based application groups were defined with two database fields. The first field we defined was for rptid. This field had a length of 8 bytes and was defined as an index. It would be the primary key people used to locate the particular report for that day. The second field used was report date. This field was defined as a filter and also used as the segment date. This simplified approach allowed us to both create common applications we were able to duplicate, but also to both support the bank's need for individual folders and combined folders. In the combined folders, we added an application group field that was used to generate a drop-down box for multiple report searching when using a combined folder.

The statement application groups contained a total of five fields. The first field and primary index used was account number. This number based on the type of account was either 12 or 16 bytes. The different lengths were supported in separate application groups, although not due to any Content Manager OnDemand limitation. We also created an Index for Customer Name at 30 bytes. This was used on only 20% of the searches when a customer forgot their account number or a customer service representative wanted to locate multiple accounts by name. We decided to make this large a field an index to provide quicker searches. In addition, three filter fields were used. All were defined as filters, since they were only used to narrow down searches or used as display fields in the hitlist. These fields were Account type for 3 bytes, Ending Balance as a Decimal field, and Statement Date. The date field was also used as the segment date.

Reports were indexed using ACIF and compressed using OD77 compression. The statements were sent to Content Manager OnDemand as AFP and contained embedded Tagged Logical Elements (TLEs), which were used to index this data.

[1] Day forward data: This is new data that would be loaded and archived after Content Manager OnDemand was in production. The data is not considered historical or part of the historical migration.

Folders were created at multiple layers. Each and every application group had a dedicated folder created that made it easy to secure individual application groups and also mimicked the customers' previous archive systems. With this setup, the training requirements were reduced. We also created combined folders to utilize the power of Content Manager OnDemand to do a single search across multiple application groups to provide a hit list of combined document or report types.

Performance testing

The bank had identified Content Manager OnDemand as a critical application and determined that the system needed to be fully redundant and available 24x7. Due to this need, they made the investment in hardware and software needed to support this environment. This system was implemented on a pair of IBM pSeries Model 570s with eight CPUs and 32 GB of memory. These systems were set up in separate data centers using Disk Mirroring technology to create the DR site. A daily ingestion of 300 GB of data needed to be supported by this platform with larger loads during month end and year end processing. This data needed to be available for users to view by 8 AM in the morning and for the bank's external customers the next day. The system was configured as follows:

- Four arsloads running on a 24/7 basis. The majority of loading occurred during the hours of 12 AM - 6 AM.
- The size of the reports vary from three pages to 400,000 pages.
- The size of the statement documents ranged from 2 - 2000 pages.

This data load environment was tested on the bank's development system. This system was also a p570 with half the number of CPUs and memory, a 4-core, 16 GB box. The tests were developed to simulate the production environment as it related to files and throughput. It was understood that this environment was smaller, but we used the amount of data ingested per CPU as a benchmark. We also developed tests that would simulate peak loads on the system from a retrieval perspective.

Initial tests revealed that the system would be able to meet the required amount of data to be ingested into the production environment. An issue was uncovered though concerning the ability of the retrieval side to process the statement files through the bank's Web environment. This processing entailed the need to convert the files from the AFP format to a PDF format. Configuration changes were needed on the Web servers to support greater throughput of this process. Additional disks and temporary space was added to support a more distributed capability. In addition, larger Windows based Web servers with more processors and additional memory were added. Once this was done, the system was determined to be production ready.

Training

The bank had previous experience with a number of archiving systems. They needed specific Content Manager OnDemand training to support the product but were familiar with the theory and requirements. They were trained in the following areas:

- System skills: The bank had a large UNIX support staff that was able to support the server environment. They also had in house support for WebSphere and the home grown application and Content Manager OnDemand Custom Client. Specific training was provided by the IBM Content Management Lab Services group during the installation and testing phases of the project.
- Report administration skills: These skills were acquired by first attending training provided by IBM Learning Services that was delivered on site. This class was designed specifically for the bank and its implementation. In addition, the bank sent two people to the following classes to acquire more general skills.
  - IM110 - Introduction to IBM Content Manager OnDemand: This course is designed for the individuals responsible for creating and loading applications into the Content Manager OnDemand for Multiplatforms system. It provides the basic understanding of all areas of the Content Manager OnDemand for Multiplatforms.
  - OD105 - IBM Content Manager OnDemand System Administration: This course is specifically designed for individuals who are experienced at indexing and loading documents and have a need for a greater in-depth knowledge of the Content Manager OnDemand system, either for system administration, maintenance, or troubleshooting purposes.
- User skills: Specific training classes were developed for users at the bank. Since the bank was using the Content Manager OnDemand Custom Client developed by IBM Content Management Lab Services, no IBM Learning Services Class existed. A custom class was developed and delivered to the users at multiple locations around the country. In addition, Train the Trainer courses were given on the use of the Content Manager OnDemand Windows Client and administrative client. The banks support staff then held additional training sessions using the skills they were taught in these cases.

Deployment

The bank’s new Content Manager OnDemand archive system was deployed in stages. The initial stage was designed to load all day forward reports to Content Manager OnDemand. This allowed the bank to reduce the load on its existing system and to eliminate the need for any additional storage on that system. Users were told to log onto a specific system based on the date of the report they were looking for. Once this phase was complete, the loading of the converted data from the bank's previous archive systems started. This was done using a newest first method to assure the greatest usability of the Content Manager OnDemand system and to allow the bank to reduce the need for the previous systems. After all the bank's data loading and performance objectives were accomplished using the new Content Manager OnDemand system, the bank was able to turn off and decommission the previous archive system. This approach from the user perspective allowed for the smallest disruption to there daily jobs since all reports were available in either the old repository or in the new Content Manager OnDemand archive.

Since the completion of the initial implementation of Content Manager OnDemand, the bank has converted additional older archive systems to Content Manager OnDemand and continues to add new day forward reports and customer facing documents. The system has performed tremendously and has the room to support the growth the bank needs.

5.2 Application service provider case study

This is a case study for a fictitious multinational company interested in offering Content Manager OnDemand as a service. While this is not the typical Content Manager OnDemand implementation scenario, it does offer valuable insight into situations where multiple Content Manager OnDemand instances are required.

5.2.1 Company background

We refer to this fictitious multinational company as the company.

The company is an IBM Business Partner that has customers throughout North America, Europe, Latin America, and the Pacific Rim. The company has extensive knowledge and expertise in records management, data protection, and information destruction solutions and helps clients address the problems most companies face today:

- Rising storage costs
- Litigation
- Regulatory compliance
- Disaster recovery

5.2.2 Business requirements

Many of this company's customers believe they cannot justify an in-house document archive system. Content Manager OnDemand is advertised as a highly reliable, yet flexible, system to meet data archive and retrieval requirements. This, combined with National Language Support (Arabic countries, Brazil, Canada, China, Czech Republic, Denmark, Finland, France, Germany, Italy, Japan, Korea, Netherlands, Norway, Portugal, Spain, Sweden, Taiwan, and all English speaking countries), made Content Manager OnDemand an obvious candidate for a software application as a service. The business requirement from the company is to be able to offer Content Manager OnDemand as a service.

5.2.3 Planning

The Business Partner had approached IBM for a hardware recommendation for their planned service provider offering. The ensuing discussion revealed that the Business Partner was looking at using a separate Content Manager OnDemand instance for each customer. This was motivated by the following factors:

- Keep customer data separate.
- Separate instances allow the customers to completely administer their own applications, users, and groups.
- Each instance can be tuned for the customer (Content Manager OnDemand configuration files and database parameters) and use the appropriate ARS_CODEPAGE parameter.
- Backups and maintenance should be easier to schedule and perform for multiple smaller instances as opposed to a single large instance.

Primary performance testing: What is the effect of multiple instances compared to a single instance? For example, compared to a single instance with 500 users, how does library performance behave for five instances each with 100 users, 10 instances with 50 users each, and 20 instances with 25 users? In order to focus on the primary testing criteria, the following assumptions were made:

- Storage: All documents would be saved on disk in a Storage Area Network (SAN).
- Client: All users would access documents through the WEBi client, which would utilize a separate WebSphere Application Server.
- Ingestion: All loading would take place through object servers on other CPUs.

5.2.4 Implementation

This particular implementation of Content Manager OnDemand was performed on two different systems. The first was on an AIX system in the Content Manager OnDemand Development Lab and was intended only for development of the test setup scripts prior to the actual performance testing at the IBM Innovation Center in Waltham, MA with the Business Partner. The IBM Innovation Center was chosen for the actual testing for two reasons:

- Hardware availability
- Knowledgeable personnel (AIX, DB2, and storage)

A complete description of the IBM Innovation Center services can be found at:

http://www.ibm.com/jct09002c/isv/spc/

The first implementation in the Content Manager OnDemand Development Lab used a p630 AIX box, a 4-core (two 1 GHz PPC4 CPUs) deskside server with 4 GB of memory, two internal disks (36.4 GB each) and 100 Mbps Ethernet running AIX 5L™ V5.2. Limited testing was performed on this box and the results are reported for comparison purposes. This testing was extremely valuable because several obstacles were encountered and solved before going to the IBM Innovation Center for the formal testing.

The second implementation at the IBM Innovation Center used a p570 AIX mid-range system, a 32-core (eight 1.9 GHz PPC5 CPUs) server with 32 GB of memory, six internal disks (36.4 GB each) and 1 GB Ethernet running AIX 5L V5.3. This system was configured with a single LPAR using one CPU and 5 GB of memory for the performance tests (see Figure 5-3).

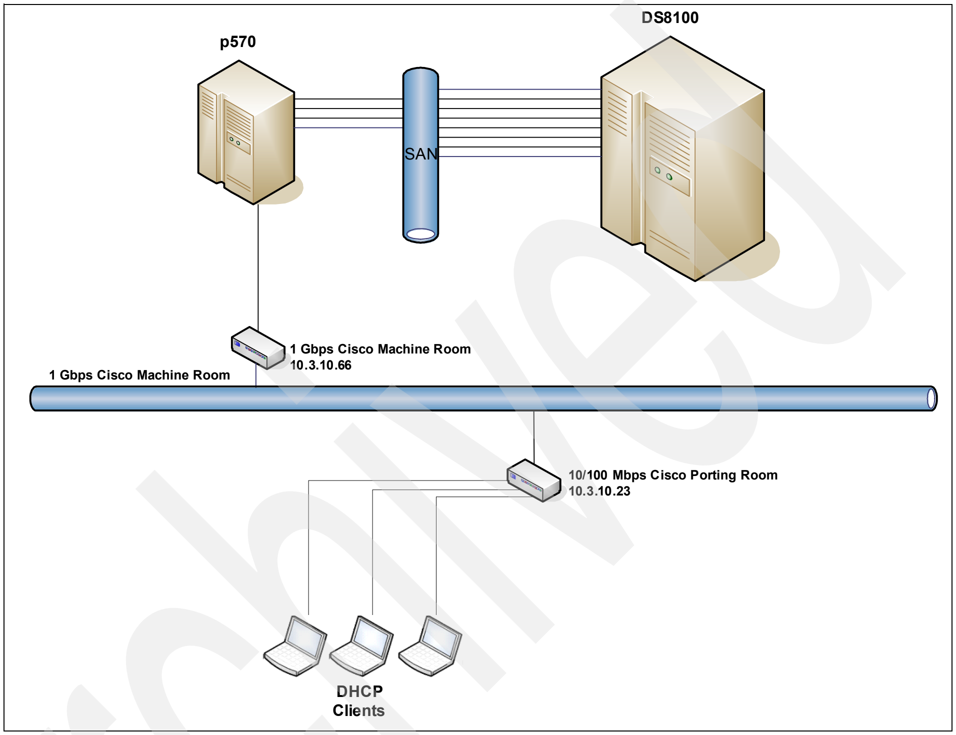

DS8100
p570
SAN
1 Gbps Cisco Machine Room
10.3.10.66
1 Gbps Cisco Machine Room
10/100 Mbps Cisco Porting Room
10.3.10.23
DHCP
Clients

Figure 5-3 IBM Innovation Center system architecture

> **Tip:** The most efficient method of determining the system configuration information for an AIX box is to use `prtconf` on the command line.

The test scripts were initially run on a T60p mobile computer with 2 GB of memory, a 100 GB disk, and 802.11a/b/g Ethernet running Windows XP. In order to run tests over night, the scripts were later moved to a p610 AIX deskside box with 2 GB of memory running AIX 5L V5.3. The test results were the same from both machines, although the p610 showed about half of the CPU activity of the mobile computer. This told us that we were not having a problem generating a maximum load on the Content Manager OnDemand server.

Note: On a Windows platform, be sure to increase the MaxUserPort and TcpTimedWaitDelay values in the registry. The default value for MaxUserPort is 5000 (port number) and should be changed to 65534 (range is 5000 to 65534). The default values for TcpTimedWaitDelay is 240 (seconds) and should be changed to 30 (range is 30 to 300). Both registry entries can be found in HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters.

The software stack included Content Manager OnDemand V8.3 (V7.1.2), Fix Pack 9, DB2 V8.1 Fix Pack 14, and Java V1.4.2. DB2 and Content Manager OnDemand were installed using the standard configuration. Initially, separate DB2 instances were created for each Content Manager OnDemand instance, but the final configuration used a single DB2 instance for all Content Manager OnDemand instances (no performance penalty and easier to manage with one instance owner).

Installation and configuration

Following the software installation, a single Content Manager OnDemand instance was created. Because document search and retrieval was the only objective of the performance testing, the /arsload and /arstmp file systems were shared between all Content Manager OnDemand instances. The /arscache, /arsdb, /arsdb_primarylog, and /arsdb_archivelog file systems were created for each Content Manager OnDemand instance using a naming convention to uniquely identify each instance, for example, /arscacheinst1, /arsdbinst1, and so on. To expedite instance creation, a shell script was written, add_instance.sh, (see Example 5-1) that performs the following steps:

- Create and mount the file systems.
- Change ownership and permissions.
- Invoke the **arsdb** command to create the system tables for each instance.
- Create the System Log.
- Start the Content Manager OnDemand instance.

Example 5-1 Sample add_instance.sh script

```
#!/bin/ksh
#
# add instance
#
if [ -z "$1" ]
then
  echo YOU FORGOT THE INSTANCE NUMBER !
  exit
fi
```

```
#
VG=arsvg
DB2VG=
ODVG=cmod${1}vg
INST=inst$1
INSTOWNR=arsinst1
#
# DB2 filesystem (256 MB)
#
if [ ! -d /arsdb${INST} ]
then
  echo "crfs -v jfs -g ${VG} -m /arsdb${INST} -A yes -a size=524288"
  crfs -v jfs -g ${VG} -m /arsdb${INST} -A yes -a size=524288
#  echo "crfs -v jfs2 -g ${DB2VG} -a size=512M -m /arsdb${INST} -A yes
-p rw -a agblksize=512"

  DEV=`lsfs | grep "arsdb${INST} " | awk '{ print $1}'`
  echo "mount $DEV /arsdb${INST}"
  mount $DEV /arsdb${INST}
fi
#
# DB2 primary_log filesystem (192 MB)
#
if [ ! -d /arsdb${INST}_primarylog ]
then
  echo "crfs -v jfs -g ${VG} -m /arsdb${INST}_primarylog -A yes -a
size=393216"
  crfs -v jfs -g ${VG} -m /arsdb${INST}_primarylog -A yes -a
size=393216
#  echo "crfs -v jfs2 -g ${DB2VG} -a size=192M -m /arsdb${INST} -A yes
-p rw -a agblksize=512"

  DEV=`lsfs | grep "arsdb${INST}_primarylog" | awk '{ print $1}'`
  echo "mount $DEV /arsdb${INST}_primarylog"
  mount $DEV /arsdb${INST}_primarylog
fi
#
# DB2 archive_log filesystem (192 MB)
#
if [ ! -d /arsdb${INST}_archivelog ]
then
  echo "crfs -v jfs -g ${VG} -m /arsdb${INST}_archivelog -A yes -a
size=393216"
  crfs -v jfs -g ${VG} -m /arsdb${INST}_archivelog -A yes -a
size=393216
```

```
#  echo "crfs -v jfs2 -g ${DB2VG} -a size=192M -m /arsdb${INST} -A yes
-p rw -a agblksize=512"

  DEV=`lsfs | grep "arsdb${INST}_archivelog" | awk '{ print $1}'`
  echo "mount $DEV /arsdb${INST}_archivelog"
  mount $DEV /arsdb${INST}_archivelog
fi
#
# change ownership and permissions on database filesystems
#
chown -R ${INSTOWNR}:sysadm1 /arsdb${INST}*
chmod 2770 /arsdb${INST}*
#
# cache filesystem (2048 MB)
#
if [ ! -d /arscache${INST} ]
then
  echo "crfs -v jfs -g ${VG} -m /arscache${INST} -A yes -a
size=4194304"
  crfs -v jfs -g ${VG} -m /arscache${INST} -A yes -a size=4194304
#  echo "crfs -v jfs2 -g ${ODVG} -a size=100G -m /arscache${INST} -A
yes -p rw -a agblksize=4096"

  DEV=`lsfs | grep "arscache${INST}" | awk '{ print $1}'`
  echo "mount $DEV /arscache${INST}"
  mount $DEV /arscache${INST}
fi
#
chown -R ${INSTOWNR}:sysadm1 /arscache${INST}*
chmod 700 /arscache${INST}
#
# only create the DB2 instance if it doesn't exist
#
cd /usr/opt/db2_08_01/instance
EXISTS=`db2ilist | grep ${INSTOWNR} | wc -l`
if [ "$EXISTS" -eq 0 ]
then
  echo "db2icrt -u db2fenc1 ${INSTOWNR}"
  db2icrt -u db2fenc1 ${INSTOWNR}

  cd /home/${INSTOWNR}/sqllib
  echo "EXTSHM=ON" >> userprofile
  echo "export EXTSHM" >> userprofile
  echo "db2set DB2ENVLIST=EXTSHM" >> userprofile
fi
```

```
DB2INSTANCE=${INSTOWNR}
#
# configure ars.ini, ars,cfg, ars.dbfs and ars.cache
#
PORT=`expr 1450 + $1`
#
cd /usr/lpp/ars/config
#
EXISTS=`grep ARSINST${1} ars.ini | wc -l`
if [ "${EXISTS}" -eq 0 ]
then
  echo "[@SRV@_ARSINST${1}]" >> ars.ini
  echo "HOST=" >> ars.ini
  echo "PROTOCOL=2" >> ars.ini
  echo "PORT=${PORT}" >> ars.ini
  echo "SRVR_INSTANCE=ARSINST${1}" >> ars.ini
  echo "SRVR_INSTANCE_OWNER=arsinst1" >> ars.ini
  echo "SRVR_OD_CFG=/usr/lpp/ars/config/ars.${INST}.cfg" >> ars.ini
  echo "SRVR_DB_CFG=/usr/lpp/ars/config/ars.${INST}.dbfs" >> ars.ini
  echo "SRVR_SM_CFG=/usr/lpp/ars/config/ars.${INST}.cache" >> ars.ini
  echo "" >> ars.ini
fi
#
cp ars.inst0.cfg ars.${INST}.cfg
ed - ars.${INST}.cfg << !
1,$ s/inst0/${INST}/
1,$ w
!
#
cp ars.inst0.dbfs ars.${INST}.dbfs
ed - ars.${INST}.dbfs << !
1,$ s/inst0/${INST}/
1,$ w
!
#
cp ars.inst0.cache ars.${INST}.cache
ed - ars.${INST}.cache << !
1,$ s/inst0/${INST}/
1,$ w
!
#
# login as the instance owner and create the Content Manager OnDemand
instance
#
cd /usr/lpp/ars/bin
```

```
#
arsdb -I ars${INST} -gcv
#
# create System Log
#
arssyscr -I ars${INST} -l
#
# start the Content Manager OnDemand instance
#
arssockd start ars${INST}
#
echo Done !
#
```

At this point, it was necessary to perform the following manual steps:

- Update the servers with System Administration Client.
- Log on as admin, change the password, and update the System Log to use the cache-only storage set.
- Create ten dummy user accounts (user0 through user9).
- Create sample reports (application groups, applications, and folders).

The sample data was then loaded using a shell script, load_instance.sh (see Example 5-2).

Example 5-2 Sample load_instance.sh script

```
#!/bin/ksh
if [ -z "$1" ]
then
  echo YOU FORGOT THE INSTANCE NUMBER !
  exit
fi
#
INST=arsinst$1
USER=admin
PASS=ondemand
PROG=/usr/lpp/ars/bin/arsload
#
${PROG} -g APG-line2afp-xxxK-HFS-IL -I ${INST} -fnv -u ${USER} -p
${PASS} perf-line2afp-25k
#
${PROG} -g APG-line-xxxK-HFS-IL -I ${INST} -fnv -u ${USER} -p ${PASS}
perf-line-25k
#
```

```
${PROG} -g APG-line-lo-HFS-IL -I ${INST} -fnv -u ${USER} -p ${PASS}
perf-line-lo
#
${PROG} -g APG-pdf-HFS-IL -I ${INST} -fnv -u ${USER} -p ${PASS}
perf-pdf
#
${PROG} -g APG-tif-checks-15k-HFS-IL -I ${INST} -fnv -u ${USER} -p
${PASS} perf-tif-checks-15k
#
${PROG} -g APG-tif-checks-33k-HFS-IL -I ${INST} -fnv -u ${USER} -p
${PASS} perf-tif-checks-33k
#
${PROG} -g APG-afp-fc-HFS-IL -I ${INST} -fnv -u ${USER} -p ${PASS}
perf-afp-fc-150k
#
${PROG} -g APG-afp-lo-HFS-IL -I ${INST} -fnv -u ${USER} -p ${PASS}
perf-afp-fc-150k
#
${PROG} -g APG-line-trans-HFS-IL -I ${INST} -fnv -u ${USER} -p ${PASS}
perf-line-trans
#
```

For all subsequent Content Manager OnDemand instances, the following procedure was used:

- Run **add_instance.sh <new_instance_number>**.
- Update Servers, log on as admin, change the password, and update the System Log to use the cache-only storage set.
- Export the dummy users from the first instance to the new instance.
- Export the application groups (and applications) from the first instance to the new instance (leave No Storage Set unchecked).
- Export the folders from the first instance to the new instance.
- Run **load_instance.sh <new_instance_number>**.
- Log in as each dummy user to change the password (the password was not changed).

Application design

Application design for this implementation was atypical due to the fact that a service provider would have minimal influence over the reports to be archived. In order to test searching and retrieval in a multiple instance configuration, representative sample data was chosen based on the following considerations:

- Data type
- Document size
- At least one unique index per application
- Large numbers of documents

A total of nine applications were used with the characteristics shown in Table 5-1.

Table 5-1 Sample data applications

Application name	**Characteristics**
APG-afp-fc	Fully composed AFP, 150,356 rows, 4 fields (1 index, 3 filter), OD77 compression
APG-afp-lo	AFP large object, 584 rows, 4 fields (1 index, 3 filter), OD77 compression
APG-line2afp	Line data converted to AFP, 200,000 rows, 4 fields (2 index, 2 filter), OD77 compression
APG-line-lo	Line data, large object, 100 rows, 3 fields (1 index, 2 filter), OD77 compression
APG-line-trans	line data, 5000 rows, transaction field, 5 fields (1 index, 4 filter), OD77 compression
APG-line-xxxK	Line data, 400,000 rows, 4 fields (2 index, 2 filter), OD77 compression
APG-PDF	PDF data, 2905 rows, 4 fields (1 index, 3 filter), compression disabled
APG-tif-checks-15k	TIFF image data, 15,000 rows, 4 fields (1 index, 3 filter), compression disabled
APG-tif-checks-33k	TIFF image data, 33,000 rows, 7 fields (1 index, 6 filter), compression disabled

Performance testing

Performance testing is broken down into the following sub-sections:

- Preparation
- Procedure
- Results
- Analysis
- Findings
- Recommendations

Preparation

The Java test script (actually two scripts, one for line data and one for all other data) is designed to log on *U* number of users and then initiate a search and retrieve thread for *N* iterations. Only ten user accounts were created for each instance, so multiple logons were made as necessary. The search is based on a where clause read from an external file and is chosen randomly. If more than one hit is returned, the document to be retrieved is also chosen randomly. The performance measures (document search and document retrieval time) are kept track of from the time the last thread is started until a thread completes (that is, only while all *U* threads are running simultaneously). A batch file was written to invoke the Java test script for the desired number of instances (see Example 5-3).

Example 5-3 Batch script UNIX version

```
#!/bin/ksh
if [ "$#" -eq 0 ]
then
  echo $0 instances users iterations
  exit
fi

export CLASSPATH=/usr/lpp/ars/www/api/ODApi.jar:.:$CLASSPATH

SERVER=p57003
USER=admin
PSWD=ondemand
CONFDIR=/femt

JSCRIPT=JavaApiDriver_Line
#JSCRIPT=JavaApiDriver

FOLDER="FLD-line-xxxK-HFS-IL"
DATAFIL=line-400k.query

NUMINST=$1
```

```
NUMUSER=$2
NUMITER=$3

RAND=ON

if [ "$NUMINST" -gt 20 ]
then
  NUMINST=20
fi

echo $NUMINST instances
echo $NUMUSER users
echo $NUMITER iterations

OUTFIL=of${NUMINST}insts_${NUMUSER}thrds.out

if [ "$NUMINST" -gt 19 ]
then
  PORT=1470
  java -Xms2m -Xmx768m $JSCRIPT $SERVER $PORT $USER $PSWD $CONFDIR
$FOLDER $DATAFIL $NUMUSER $NUMITER $RAND > 20$OUTFIL &
fi

if [ "$NUMINST" -gt 18 ]
then
  PORT=1469
  java -Xms2m -Xmx768m $JSCRIPT $SERVER $PORT $USER $PSWD $CONFDIR
$FOLDER $DATAFIL $NUMUSER $NUMITER $RAND > 19$OUTFIL &
fi

...

if [ "$NUMINST" -gt 2 ]
then
  PORT=1453
  java -Xms2m -Xmx768m $JSCRIPT $SERVER $PORT $USER $PSWD $CONFDIR
$FOLDER $DATAFIL $NUMUSER $NUMITER $RAND > 3$OUTFIL &
fi

if [ "$NUMINST" -gt 1 ]
then
  PORT=1452
  java -Xms2m -Xmx768m $JSCRIPT $SERVER $PORT $USER $PSWD $CONFDIR
$FOLDER $DATAFIL $NUMUSER $NUMITER $RAND > 2$OUTFIL &
fi
```

```
PORT=1451
java -Xms2m -Xmx768m $JSCRIPT $SERVER $PORT $USER $PSWD $CONFDIR
$FOLDER $DATAFIL $NUMUSER $NUMITER $RAND > 1$OUTFIL &

wait

echo All Done !
```

The external query file was created by using DB2 select statements to create a list of all values for a particular column, and then editing the list to create the where clauses. A portion of the simple query file is shown in Example 5-4.

Example 5-4 Simple query file

```
where account = '000-000-000'
where account = '000-000-001'

...
where account = '250-000-997'
where account = '250-000-998'
where account = '250-000-999'
```

The complex query file was based on two database fields and an OR clause (see Example 5-5 on page 242).

Example 5-5 Complex query file

```
where account = '000-000-000' or name = 'Compan A A'
where account = '000-000-001' or name = 'Company A B'

...
where account = '250-003-997' or name = 'Company X'
where account = '250-003-998' or name = 'Company Y'
where account = '250-003-999' or name = 'Company Z'
```

The query file could have been much more complex and more realistic including for example, different columns and wildcard searches, but the time required to generate such a file was excessive given that an actual production application was not used for testing and typical query characteristics were not known.

Procedure

On the Content Manager OnDemand library server:

1. Log in as root (member of DB2 administration group).

2. Start the DB2 instance.
3. Start all Content Manager OnDemand instances using **`arssockd start arsinst1`**:
   - Export TERM=vt100.
   - Start the performance monitor by running, for example, **`topas -i 5`**.

On the PC:

1. Change directory (**`cd`**) to the test scripts directory.
2. Run setup.bat (sets PATH and CLASSPATH):

   Run batch file and multiple invocations of the Java test script (# instances, # users/instance, and # iterations).

The initial testing was performed on a p630, much of which was done remotely over a slow broadband connection (3 Mb down, 256 Kb up). This testing allowed us to find most of the problems in our test environment before we arrived at the IBM Innovation Center, where our time was limited.

Table 5-2 Initial testing result on p630,100 total users, slow dial-in connection

# Instances	Users/instance	Milliseconds/ document	CPU% (Kernel/User)	CPU% (Idle)
1	100	147	10-20	80-90
2	50	282	10-20	80-90
3	33	384	10-20	80-90
4	25	512	10-20	80-90

Results

Results from each simulation were written to an output file (see Example 5-6) whose name contains the number of instances and users (threads) per instance.

Example 5-6 Sample output file, 7of10insts_100thrds.out, complex query

```
*****************************************************************************
* ENVIRONMENT CONFIGURATION AND TEST SETTINGS
* OD Server to load            : p57003
* OD Folder to retrieve        : FLD-line-xxxK-HFS-IL
* Number of Concurrent Users   : 100
* Iterations per each user     : 25
sServer: p57003
iPort: 1457
sUserID: admin
```

```
sPassword: ondemand
sConfDir: /femt
sFolder: FLD-line-xxxK-HFS-IL
vFileLines.size: 100000
iNumUsers: 100
sRandFlag: ON
iNumIterations: 25
* Test Started                 : Fri Aug 24 12:21:50 EDT 2007
everyone logged on
100 threads started
* Test Finished                : Fri Aug 24 12:31:29 EDT 2007
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* LOAD STATISTICS AND PERFORMANCE SUMMARY RESULTS
* Time (doc search+retrieve)   : 374091 milliseconds elapsed
* Duration (doc search only)   : 324412 milliseconds elapsed
* Duration (doc retrieve only) : 49679 milliseconds elapsed
* Duration (unable to connect) : 0 milliseconds elapsed
* Volume                       : 1739 documents retrieved
* Unable to connect count      : 0
* Throughput                   : 4.648614 documents retrieved per second
* Service Rate                 : 215 milliseconds per document
* Run Time                     : 578 seconds elapsed
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

A summary shell script (see Example 5-7) was created to scan each output file and average the results from all instances. For example, an eight instance run would have eight result files and the script would find the average response time and execution time and write one line of summary information (# of instances, # of users/instance, response time in ms, and execution time in seconds) in comma delimited format to a report file (see Example 5-8 on page 246). The execution time was used to estimate total time required for testing when multiple tests were included in a batch file. Batch testing was performed overnight and sometimes required 12 hours for all runs to complete.

Example 5-7 Summary report script

```
#!/bin/ksh
# j is total instances
j=1
while [ $j -le 20 ]
do
# k is users per instance
  k=10
  while [ $k -le 2000 ]
#  for k in 10 20 40 70 100 150 200 300 700 1000 2000
```

```
  do
#   i is instance number
    i=1
    while [ $i -le $j ]
    do
      CT=0
#     SUM1, TIME1, AVE1 are response times in ms
      SUM1=0
#     SUM2, TIME2, AVE2 are run times in sec
      SUM2=0
      FILE=${i}of${j}insts_${k}thrds.out
      if [ -f "$FILE" ]
      then
#       echo ${i}of${j}insts_${k}thrds.out
        TIME1=`awk '/Service/ { print $5}' < $FILE`
        TIME2=`awk '/Run Time/ { print $5}' < $FILE`
        if [ -z "$TIME2" ]
        then
           TIME2=0
        fi
#       echo TIME1 $TIME1
#       echo TIME2 $TIME2
        CT=`expr $CT + 1`
        SUM1=`expr $SUM1 + $TIME1`
        SUM2=`expr $SUM2 + $TIME2`
      fi
#     increment i = 1,2...j
      i=`expr $i + 1`
    done
    if [ "$CT" -gt 0 ]
    then
      AVE1=`expr $SUM1 / $CT`
      AVE2=`expr $SUM2 / $CT`
      echo ${j},${k},${AVE1},${AVE2}
    fi
  k=`expr $k + 10`
  done
# increment j = 1,2,4,6,8,10,12,14,16,18,20
  if [ "$j" -eq 1 ]
  then
    j=`expr $j + 1`
  else
    j=`expr $j + 2`
  fi
done
```

Example 5-8 Sample report file

```
1,10,19,7
1,20,12,10
1,40,12,18
1,70,11,32
1,100,11,45
1,200,11,90
1,400,11,181
1,700,12,318
1,1000,12,455
1,2000,,
2,10,22,8
2,20,23,14
2,40,24,30
2,70,25,58
2,100,29,90
2,200,41,240
2,400,80,861
2,700,128,2511
2,1000,223,5781
4,10,48,14
4,20,51,30
4,40,56,65
4,70,65,129
4,100,73,203
4,200,116,610
4,400,250,2359
...
18,10,276,84
18,20,301,171
18,40,342,384
18,70,411,785
18,100,474,1324
18,200,,
20,10,315,174
20,20,334,199
20,40,387,427
20,70,475,893
20,100,,
```

Analysis

Once the summary report file was generated, a third-party software application was used to analyze the results. This software will solve non-linear equations with two independent variables, and allowed us to create mathematical models (functions) and then compute coefficients (regression analysis) that resulted in a best fit based on the data provided. A visual examination of the results led us to believe the response time was linear with respect to the number of instances and total users (number of instances times users per instance), so our first model was as follows:

y = a + b*x1 +c*x2 +d*(x1*x2)

Where:

- y is the response time in milliseconds.
- x1 is the number of instances.
- x2 is the number of users per instance.
- x1*x2 is the total number of users.

The calculated values for the coefficients were as follows:

- a = -22.73
- b = 13.87
- c = -0.1032
- d = 0.1465

According to the application, this model and these coefficients account for 98.0% of the variance in the test results. Plotting the model and data points for 10 and 20 instances revealed a less than ideal fit (see Figure 5-4 on page 248).

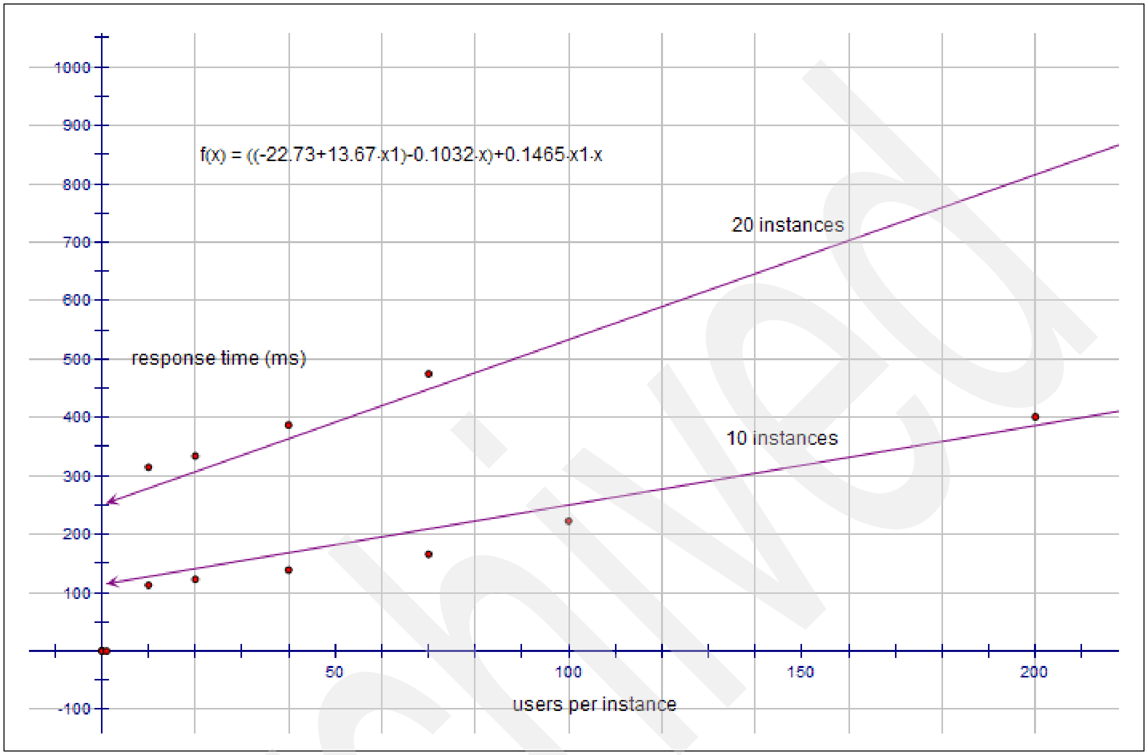


Figure 5-4 Initial simple query model plot

We decided to change the model and make each term an exponential:

y= a + b*x1^e + c*x2^e + d*(x1*x2)^e

Where:

- a = 12.18
- b = 4.256
- c = -0.0068
- d = 0.0069
- e= 1.404

Performing regression analysis on this model showed an even better fit (99.2% of the variance explained), and the plot looked much better (see Figure 5-5).

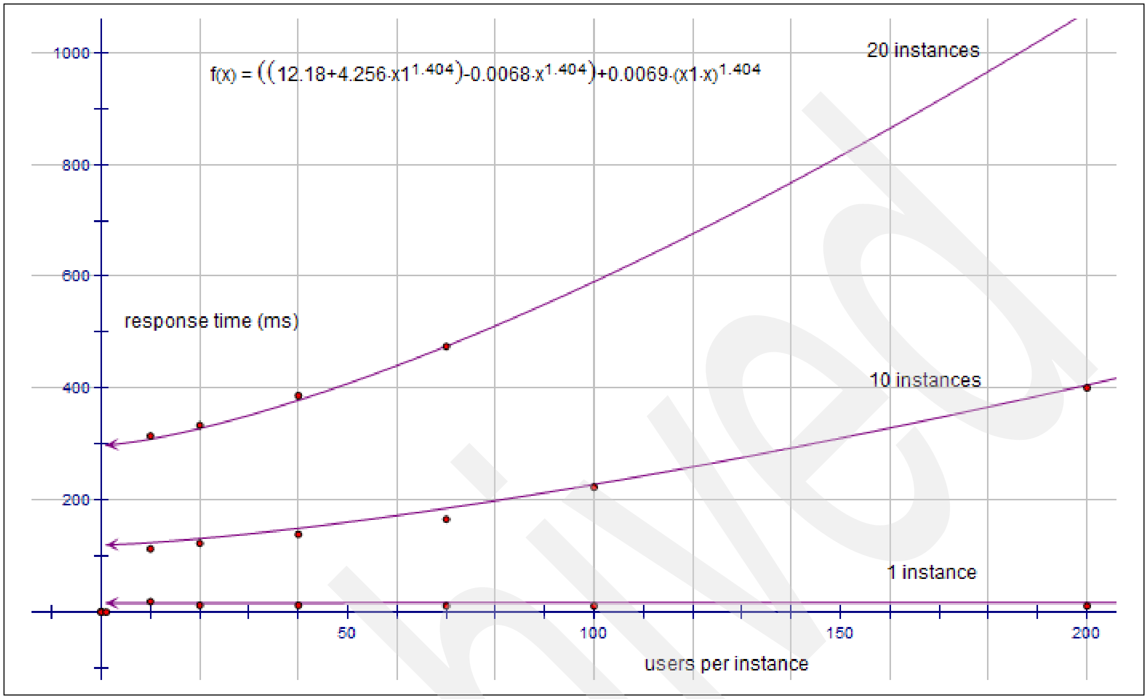


Figure 5-5 Better simple query model plot

The simulation runs were then repeated using a more complex query (exact matches on two database columns with indexes, account = <account> OR name = <name>). The exponential model was used and the calculated coefficient values were as follows:

y= a + b*x1^e + c*x2^e + d*(x1*x2)^e

Where:

- a = 18.78 (12.18 for simple query)
- b = 4.770 (4.256)
- c = -0.0063 (-0.0068)
- d = 0.0059 (0.0069)
- e= 1.399 (1.404)

Regression analysis showed 98.7% of the variance in test results explained and the plot looked very similar (see Figure 5-6) to the simple query model. The primary difference was that the search component of the response time was larger for the complex query than the simple query.

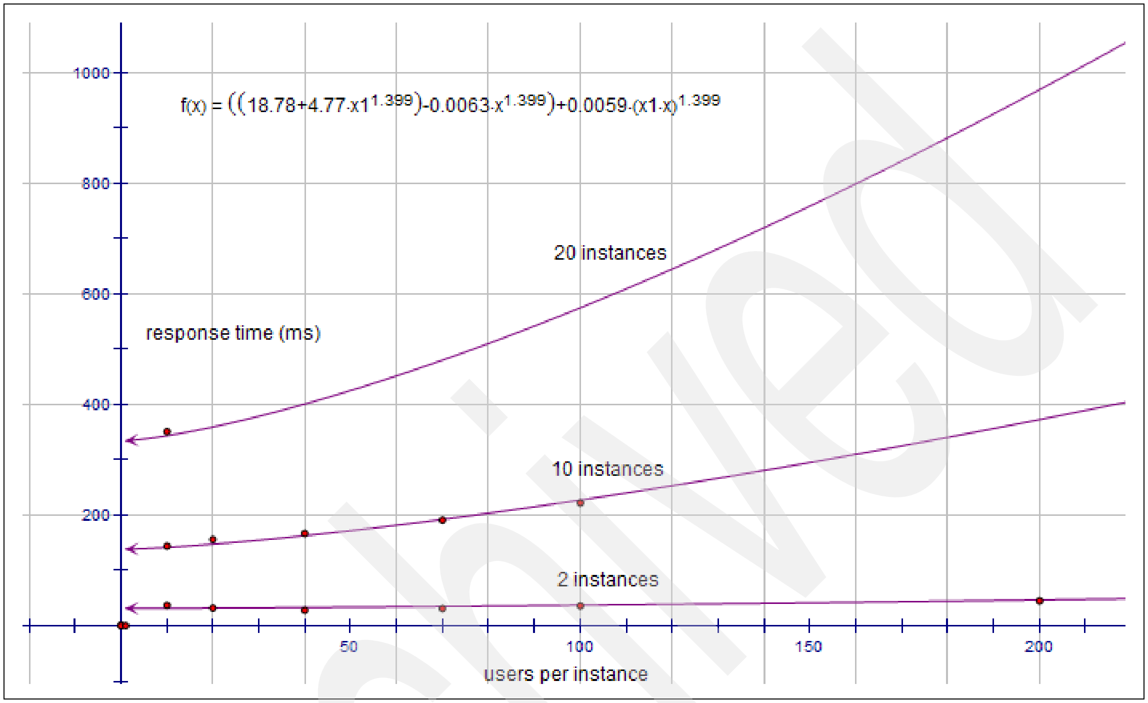


Figure 5-6 Complex query model plot

The final plot presents the findings in a slightly different way: response time versus number of instances, where each curve represents the total number of users over all instances (simple query). From this plot, we can conclude that response time (search and retrieval only) is linear with respect to the number of instances (once the number of instances is five or greater), assuming a constant number of total users (see Figure 5-7).

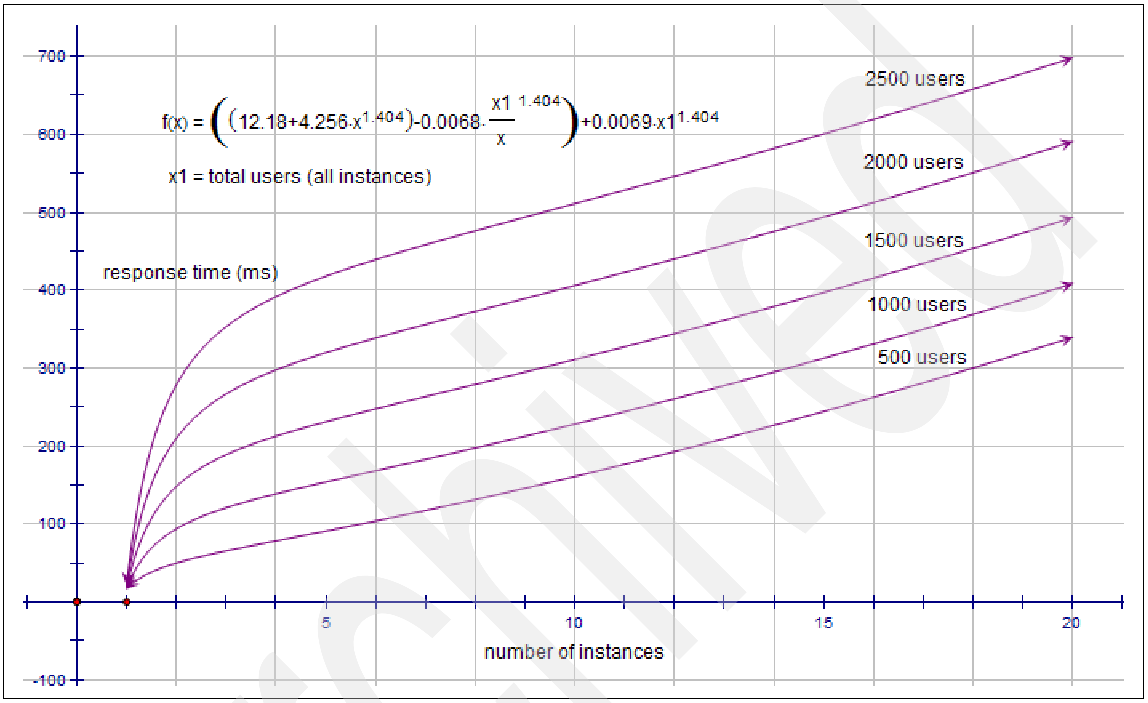


Figure 5-7 Response time versus number of instances, constant total users, simple query

Findings

Response time: Even with 2000 total users and 20 instances (100 users per instance), the total response time for well constructed queries and document retrieval was less than one second, well below what was viewed as an acceptable response time. It should be noted that these simulations were designed to stress the Content Manager OnDemand server to its maximum, and that the total number of users in these tests probably equate to fifty or even a hundred times that number in an actual production environment (100,000 to 200,000 users).

CPU limited process: CPU activity, as measured with **topas** and **nmon**, was consistent once the steady-state segment of the simulations was reached (all users logged on and all threads started). Typical values for CPU activity were:

- Kernel: 30% to 35%. The high kernel percentage was attributed to system calls.
- User: 60% to 65%. These were due to arssockd and DB2 processes.
- Wait - < 1%. Note that the wait percentage was very low, indicating that disk I/O was not a problem. This was attributed to a fast disk subsystem with a substantial hardware cache/buffer.
- Idle - 1% to 5%. This also includes network I/O wait time, in addition to no CPU activity. Since the simulated queries and retrievals were generated on another box, this value would make sense.

At one point in the testing, we added a second CPU to the LPAR to confirm our belief that we were CPU limited. As expected, the addition of a second CPU cut the response times in half.

Memory requirements: The base AIX memory footprint was measured immediately after booting the LPAR (no DB2 or arssockd processes started). The DB2 memory requirements were based on the DB2 monitor and we used **svmon** for the arssockd processes:

- AIX: 1.85 GB
- DB2: 1.65 GB (63 MB/db, 10 MB/db2 instance, 1 MB/db2agent; 20 *63 + 10 + 20 * 21 * 1 = 1690 MB)
- Content Manager OnDemand: 2.25 GB (5.5 MB/arssockd; 20 instances * [20 numdbsrvr+1] * 5.5 MB = 2310 MB)

Note that the total is 5.75 GB. Memory measurements were made when testing 10 instances, so the actual total memory for 10 instances would be 1.85 + 0.82 + 1.12 = 3.79 GB. During these tests (10 instances), **topas** showed approximately 75% computational memory and 15% non-computational memory. Assuming 75% computational memory, 0.75 * 5 GB = 3.75 GB, which agrees with the computed memory estimate.

Paging: PgspIn (pages read from paging space) and PgspOut (pages written to paging space) were both 0 during most of the testing. Paging did occur when testing more instances (14 and above) with ARS_NUM_DBSRVR=20, although this was easily corrected by lowering the value to 10. The response time did not appear to increase appreciably with the lower value.

Disk subsystem: A virtual disk subsystem (DS8000®) with 32 GB of hardware cache was divided into 20 database hdisks (1 GB each) and 20 document hdisks (100 GB each). The database hdisks were only busy a maximum of 5% during the simulation runs, and the document hdisks were only slightly more busy.

Java test script: The Java test script became a limiting factor for the upper range of instances and users per instance, due to a 1 GB limit on the Java maximum heap size. The only work around for this was to use more than one computer to run the test script, but this was not done due to the difficulty in synchronizing individual runs and results.

Recommendations

Physical memory must be sufficient to support the number of Content Manager OnDemand instances running. As soon as physical memory is completely used and paging space is utilized, performance can and will deteriorate rapidly. Because AIX has a large memory footprint, you might consider decreasing the number of LPARs and assigning more memory (and CPUs) to each LPAR.

DB2 buffer pools were sized at 10,000 pages per database (Content Manager OnDemand instance). This equates to approximately 39 MB (4096 bytes/page) per Content Manager OnDemand instance, and was sufficient for the sample database when well constructed queries were used for search criteria. When substantially larger buffer pool sizes (50,000 pages) were specified, we found the search time to improve only slightly (less than 10%). We believe that the buffer pool size only needs to accommodate the indexes for the Content Manager OnDemand system tables and applications, unless the search criteria requires table scans (wildcard searches and searches on filter fields).

Content Manager OnDemand was configured for 10 database connections per instance during performance testing (ARS_NUM_DBSRVRS=10). This number was chosen to support the maximum number of users per instance (2000 users for one instance), and could certainly be decreased for fewer users. At 5.5 MB per arssockd process, the memory freed up by fewer database connections could be dedicated to DB2 buffer pool memory.

AIX file caching was restricted to a small portion of memory (5%) with the **vmo** command:

```
vmo -r -o maxperm=5
```

This was done in order to provide the greatest amount of computational memory possible. Retrieval times did not increase, since documents were not read repetitively.

During performance testing, only one DB2 instance was used. This made it easier to make database parameter changes, but it required stopping all Content Manager OnDemand instances at the same time. In a production environment, it may make administration easier if separate DB2 instances for each Content Manager OnDemand instance are employed. The memory overhead for the DB2 instance was only 10 MB, so the option should be considered.

DB2 tuning with the production application(s) and typical search criteria is advisable. While complex queries will lengthen search times, this testing has shown that multiple Content Manager OnDemand instances provide acceptable response times even with the overhead incurred as a result of multiple instances.

Training

Training was not performed in this Content Manager OnDemand implementation, but several administrative recommendations were made.

Deployment

This Content Manager OnDemand implementation was a test environment and not an actual deployment. The system configuration, test results, and recommendations were documented for a future implementation.

6

Case studies for System i

In this chapter, we describe two case studies for Content Manager OnDemand for i5/OS implementations. Because these implementations were done before the name of the system and software were changed, the iSeries and OS/400 names are used here. The technique and suggestions here are still applicable even though the names of the hardware and software have changed to System i and i5/OS.

The cases studies we cover in this chapter are for:

- An international bank
- A telephone company

6.1 International bank case study

In this case study, we describe a fictitious international bank implementation. We describe how the bank with many locations made use of Content Manager OnDemand to provide data for users not only in its headquarters location but in all its branch offices.

6.1.1 Company background

We refer to the fictitious international bank as the bank.

The bank is a large international bank that operates in 10 different countries with more than 80 branch offices and over 1700 users who need to access the data that is archived in Content Manager OnDemand.

6.1.2 Business requirements

We identified the following business requirements:

- A single tool is needed to archive and view the bank's reports.

  Most reports are printed at the branches and some of those reports are also placed on CDs that are then delivered to the branches for historical retention. Statements are mailed directly to customers and copies placed on CDs for use in the branches.

- Improve the quality of the current process.
  - Make the information from nightly processing available to each of the branches before they open in the morning.

    The nightly processing runs until early in the morning. Because there is only one printer in each branch office, the reports do not always finish printing until after the branch is already open for the day.

  - Secure the data so only the authorized users see the data.

    Reports printed in the branch offices are many times left on desks where anyone can see it. Also, only branch managers are allowed to see data for all branches within their country. Only some groups at the headquarters location are allowed to see data for all countries.

  - Provide a responsive system even though the communication links to the branches are not all high speed.
- Enable viewing the reports over the Web.

6.1.3 Planning for implementation

We assembled a team to analyze and implement the Content Manager OnDemand solution. This team consisted of a Content Manager OnDemand Business Partner, a project manager, an IT representative, a Content Manager OnDemand administrator, and a user representative.

We gathered the following information about the environment:

- Over 370 reports need to be archived in Content Manager OnDemand.
- There are over 1700 employees that need to access the data to be archived.
- The reports consist of System Network Architecture Character Stream (SCS) line data.
- There are two general types of reports:
  - Bank reports either do not contain a branch number in them or if they do contain a branch number, the branch number changes in the middle of a page.
  - Branch reports must have a branch number and if there are multiple branches in the report, it must change at a page break.
- The same report is produced for each country, but it is placed in an output queue specific to that country.
- Reports are uniquely identified by the combination of the spooled file name and user data.
- There are two reports that currently use pre-printed forms: statements and notices.
- Current network traffic going to the branches can be heavy at times and slow response times are seen.
- Some users currently use the report data to do calculations on the data using a calculator.
- Because the bank operates in 10 different countries, the legal requirements for data retention vary depending on the country.
- The current iSeries system has excess capacity.

Based on the information gathered, we made the following decisions:

- Select only five reports for the initial work.
- Use multiple types of administrators for Content Manager OnDemand.

  The three types of administrators are: system, report, and user. The primary reason for doing this is the number of reports and users. There is a group defined for each branch, and the user administrator for that branch is responsible for adding and removing users in that branch.

- Convert reports that use pre-printed forms to AFP reports with overlays to show the pre-printed forms.
- Use large object support for reports that have more than 20 to 50 pages in each document to reduce the impact on network traffic.
- Have two migration policies that retain data for one year and seven years.
- Use a monitor exit program to handle the determination of the application group and application name used when loading data.

  This is necessary because the application group and application names do not match any of the spooled file attributes. The exit program has to look at the combination of the user data and spooled file name to determine the application group name and application name to use.
- No additional iSeries hardware is needed.
- Use the ArsWWWServlet interface of the Content Manager OnDemand Web Enablement Kit (ODWEK) running on a Windows WebSphere Application Server.

  We chose the servlet interface rather than CGI because the servlet is managed by an application server. This is more efficient because it runs as a single instance and the memory is managed by the application server. If the CGI interface is used, it runs as multiple instances and it has to manage the memory for each of these threads.
- Users who now do calculations on the reports need to use the thick Content Manager OnDemand Windows Client so they can bring the data into Monarch or Excel for analysis.

6.1.4 Implementation

Implementing Content Manager OnDemand is a major project involving several steps. We first installed and configured the product on a test machine, then decided how to define the reports, archive and retrieve the data, do performance testing to ensure adequate service levels, train the users, and finally move the definitions from the test machine to the production machine. The overall design of the Content Manager OnDemand system at the bank is shown in Figure 6-1 on page 259.

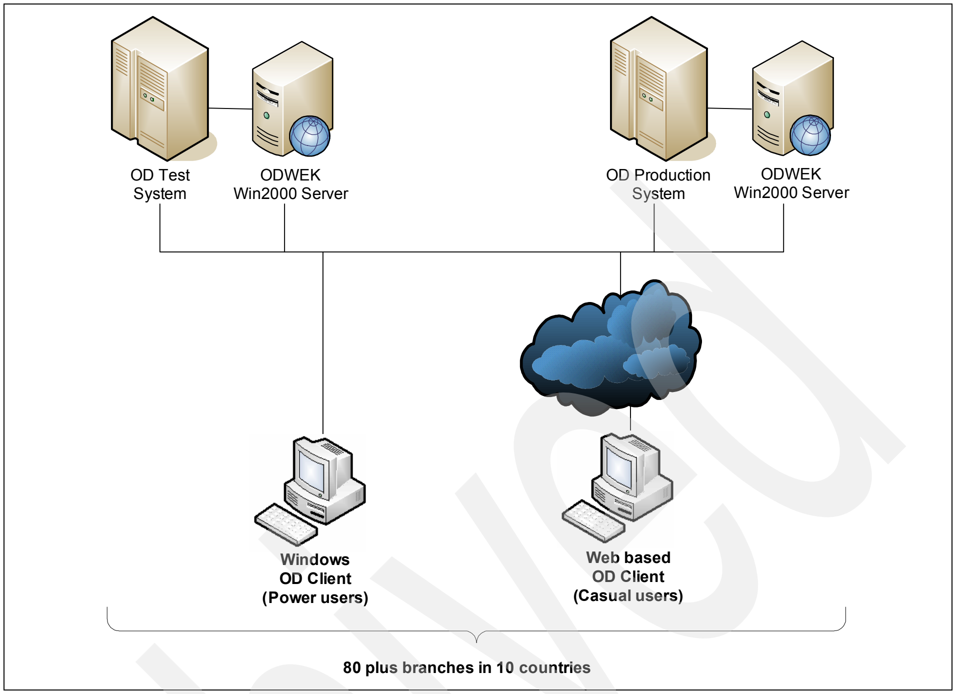

OD Test System
ODWEK Win2000 Server
OD Production System
ODWEK Win2000 Server
Windows OD Client (Power users)
Web based OD Client (Casual users)
80 plus branches in 10 countries

Figure 6-1 Content Manager OnDemand for i5/OS case study 1 - System design

Installation and configuration setup

We installed the prerequisite OS/400 software and the Content Manager OnDemand Licensed Program Product (5722RD1). After installing the latest OS/400 and Content Manager OnDemand PTFs, we created the default instance QUSROND. We then used System i Access Navigator to create a migration policy called Y01 to retain the data for one year and another one called Y07 to retain the data for seven years to handle the data retention requirements.

We then installed the IBM HTTP Server, WebSphere Application Server, and ODWEK on a Windows 2000 Server.

Application design

During application design, we decided how to define the users, groups, folders, application groups, applications, and storage sets, in order to fulfill the business requirements that we had identified.

Users

Because all users already have OS/400 user profiles, OS/400 authentication was used for Content Manager OnDemand. This also simplifies password management for the users because there would be only one common password. Because of the number of users in many different locations, we decided to have a user administrator in each branch office to administer the users in that branch by assigning them to the group for that branch. We also decided to have report administrators who would be responsible for knowing the report and document data and for defining the indexes that are extracted from the data. Report administrators are also responsible for designing the user interface to the reports through the folder definition process and for controlling access permissions to the reports that they administer.

Groups

There is no relationship between user groups in Content Manager OnDemand and OS/400 group profiles. Even if a Content Manager OnDemand group has the same name as an OS/400 group user profile, the members of the group are independent from the group user profile.

We decided that all Content Manager OnDemand user administrators and users should be placed into Content Manager OnDemand groups and permissions should be defined at the group level rather than the user level. This allows user administrators to easily add, remove, or transfer a user without having to individually update the permissions for that user.

We used the following naming convention to easily identify the groups:

country-branch

For example, users in branch 006 in country B would be in group BBB-006 while users in branch 626 in the country E would be in group EEE-626.

If a user is placed in multiple groups and each group is given *specific* permission to an application group or folder, then the permissions are determined from the group with the lowest Group ID number (GID). Because of the way permissions are determined when a user is a member of multiple groups, we tried to avoid placing anyone in multiple groups. However, in order to provide future flexibility, we reserved 10 low number GIDs by creating a small number of groups by the name ZReserver*nn* (where nn is a two digit number from 01 to 10) before we defined any other groups. This reserved the first 10 group ID numbers for future use. By starting the name with a Z, they will also show up at the end of the list of groups and do not require that you always have to scroll past them.

Note: If you export the groups to another instance, the groups are exported in alphabetical order and assigned GIDs by that order. Therefore, to preserve the low numbers for these groups, you should first export the reserved groups separately before the other groups that you have defined.

Security

Security is handled by Content Manager OnDemand permissions to control access to reports and define the capabilities that a user has. The security model is hierarchical with the permission checking done in the following order:

1. User
2. Group
3. *PUBLIC

If no permission is specified for a user, the group that the user is a member of is checked next. If the user is a member of more that one group, the groups are checked in ascending group ID number. If none of the groups have any permissions defined for the user, the permissions defined for *PUBLIC is used.

Note: Not having any permissions defined for a group or user means that when looking at the folder or group permissions in the Administration Client, the user or group is not listed in the right hand column titled *defined*.

Permissions control the user’s access to folders and application groups. Table 6-1 shows how we set the folder and application group permissions for the four types of users.

Table 6-1 Permission settings by user type

	System Administrator	Report Administrator	User Administrator	User
Folder Permissions				
- Access	*	*	O	O
- Administrator	*	Y		
- Field	*	Y		
Application Group Permissions				
- Access	*	*	O	O
- Logical Views	*		O	O
- Administrator	*	Y		
Document				
- View	*		O	O
- Add	*		O	O
- Delete	*		O	O
- Update	*		O	O
-Print	*		O	O
- FAX	*		O	O
-Copy	*		O	O
Annotations				
- View	*		O	O
- Add	*		O	O
- Delete	*		O	O
- Update	*		O	O
- Copy	*		O	O

*	Has authority/permissions because of user type
Y	Give authority/permission to this
O	Optional; can be specified as needed to achieve desired security

The general security requirements for users are:

- Only headquarters’ users can see reports across countries.
- Headquarter’s users and branch managers can see data for all branches within a country.
- Branch users can only see information for their branch.

Because we chose a design where reports for all branches and countries are in the same application group, query restrictions must also be specified to meet the security requirements. To facilitate the implementation of these query restrictions, we used the standard application group field names shown in Table 6-2 on page 266.

For example, the following query restriction allows a user to see documents for only branches 345 and 437 in country A:

```
branch IN (345,437) and country = ‘AAA’
```

> **Note:** Consider using a short field name to minimize the amount of typing you have to do to enter the query restriction. In this case, you might consider using a field name of b for the branch and c for the country.

Reports

A report definition is specified within Content Manager OnDemand by the definitions in the folder, application group, application, and storage sets. You need to carefully consider the names you use for each of these.

Here are some considerations for these definitions:

- Folders:

  When a folder contains multiple application groups, the folder should represent the type of information contained in the reports. If a report does not have indexes that are common to the other reports, then that report should be placed in a separate folder.

  Some folder names we used were:

  - Customer Information Reports
  - Checking Statements (only one application group in the folder because the index fields are unique)
  - General Ledger Reports
  - Process Server Reports
  - Loan Reports
  - Transaction Reports

Note: You normally do not want to have many application groups in a single folder because it generates a separate query for each application group. In this case, the bank wanted the users to see only a few folders in their list. Because one of the folder fields had a type of application group, there was a drop-down menu displayed for that field when the folder was opened that allowed the user to select one report (application group). Users were trained to always select one report (application group) from the list so only one query would be sent to the server. If users did not select anything from the list, then a query would be sent for each application group in the folder and they would soon learn that it paid to select a report from the list. Unfortunately, you cannot make an application group type folder field a required field, so you cannot force them to choose a report.

The default search operator must be carefully chosen for each folder field because the search operator cannot be changed by the user when using a thin client (internet browser).

For *PUBLIC, we always had the country folder field defined with a Query order of 0 and Hit List order of 0 so it was never seen by the user. The branch folder field was defined with a Query order of 0 and a Hit List order of 2 (if there is a branch application group field defined). Even though the user cannot search for the branch number, having it in the hit list provided feedback that they are looking at their data.

For the headquarters groups, folder fields were defined differently because they need to search across multiple countries and branches. The country folder field was defined as Query field 1 and Hit List field 2. The branch folder field was defined as Query field 2 and Hit List field 3.

- Application groups:

  Because we have multiple application groups in a folder and one of the folder fields has a field type of application group, the users will see the application group name when it is used as a search field or a hit list field. This name cannot be mapped to another value so we had to choose a name that is descriptive of the report and that the user will recognize. For example, rather than using the name RP0135 (which is the value in the user data spooled file attribute), we used the name of the report, which is the Paid Exception Journal. This requirement means that an output monitor exit must be used to specify the application group name.

- Applications:

  The name of the application is controlled by the fact that all reports must have a query restriction for the country application group field. The country cannot be extracted by the indexer because it is not in the body of the report, but is only identified by the output queue where the report is produced. We decided to use the application ID field (called country) to provide the country code.

  Based on this requirement, we decided to use the following format for naming the applications:

  name-country

  where

  – name is the user data spooled file attribute for the report.

  – country is the first three characters used as the DB value, as shown for the country application group field in Table 6-2 on page 266.

  The country application group field (which is the application ID field) is four characters long with the first three characters identifying the country and the last character representing the version. The country codes we used are shown in Table 6-2 on page 266. We also added these codes as mapping values for the application ID field in each application group so the user would see the country without any version number.

  For example, the application name RP0135-BBB was the name for the Paid Exception Journal for country B. Because this was the first version of the application definition, the mapping value for the country application group field was:

Display value	**DB value**
Country B	BBB1

  To make it easier, we defined and tested an application definition for only one of the countries. After we were satisfied that the definition was correct, we then copied that definition nine more times and named it to indicate the country that was selected as the application ID. This was safe to do because the report format for each country was exactly the same, so once one definition was correct, all the others could be copied from it.

Note: When you need to create a new version, you should rename the old application to include the version number, then copy it to a new application with the name without the version number and make the modifications to that new application definition. This keeps the application name of the current version always the same so you do not have to make changes to the report archiving process.

- Storage sets:

  The name of the storage set is always the same as the name of the migration policy on the iSeries. We decided to have the migration policy name describe the attributes of the migration policy. For example, we named the migration policy that retains the data on optical for seven years, Y07. When you create an application group, you specify the storage set to use for that application group.

> **Note:** You cannot create or modify a storage set using the Administration Client when connected to an iSeries server. Creating, updating, and deleting of storage sets can only be done indirectly by working with the migration policy using the Content Manager OnDemand Common Server plug-in in the iSeries Navigator, which is part of the Access for Windows product.

Before we started defining individual reports, we first identified some common fields that were found in many of the reports and selected names, attributes, and mapping values to use across all definitions. Table 6-2 shows the standard definitions we chose for the application group field names.

Table 6-2 Standard application group fields and mapping values

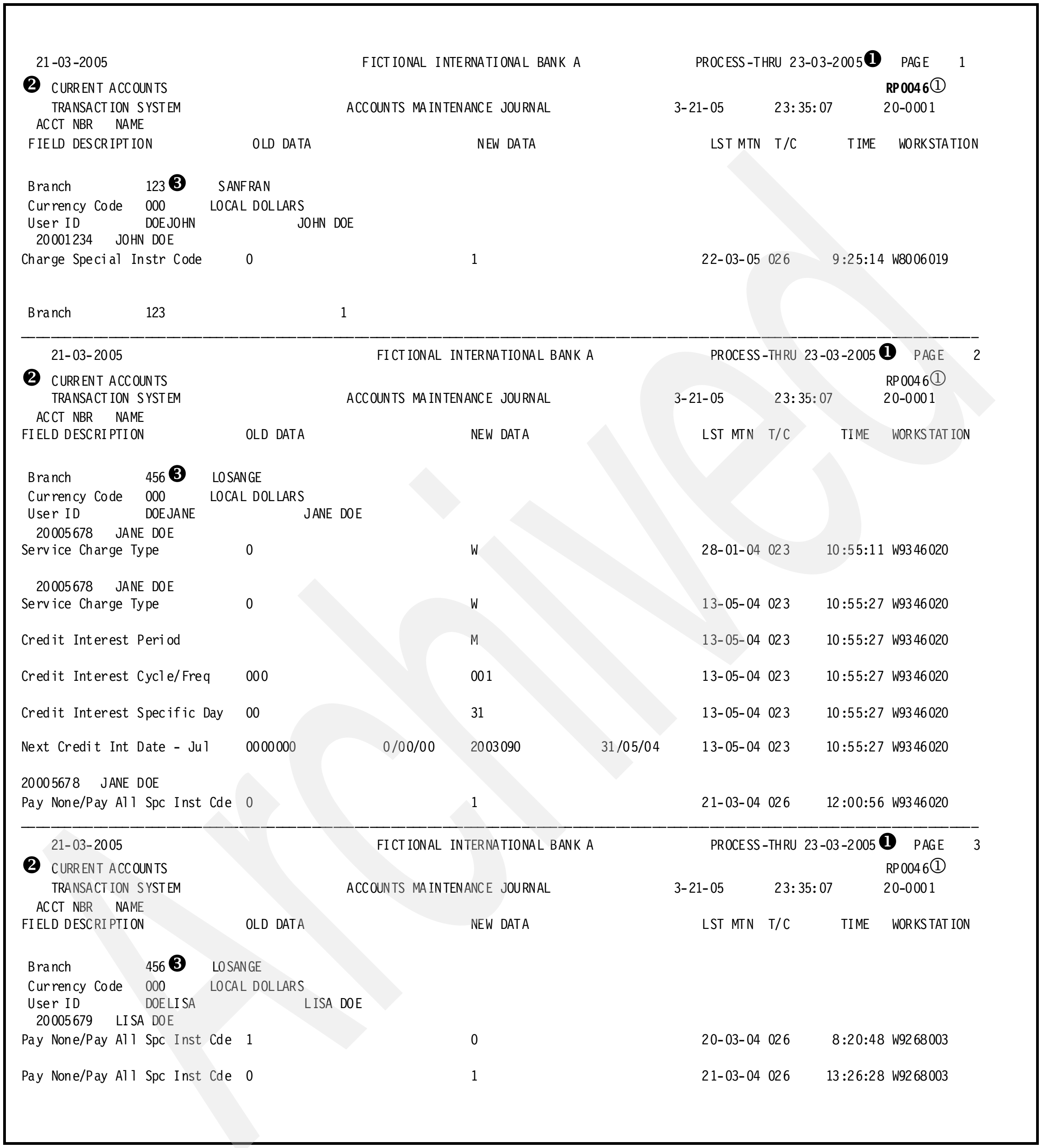
		Mapping values	
Name	**Attribute**	**Displayed value**	**DB value**
actnbr	big int		
acttype	string 1	Current Accounts Savings	C S
branch	small int		
country	string 4	Country A Country B Country C Country D Country E Country F Country G Country H Country I Country J	AAAn BBBn CCCn DDDn EEEn FFFn GGGn HHHn IIIn JJJn
currency	small int	Local Currency Canadian Dollars US Dollars British Pounds	0 1 2 3
custname	string 40		

Name	Attribute	Mapping values	
		Displayed value	DB value
participant	string 12	Participant 1 Participant 2 Participant 3 Participant 4	001BNK0001 001BNK0002 002BNK0002 001BNK0003
rptdate	date		

These eight names provided all the application group field names we needed to use for more than 90% of the application groups we had to define for the project.

Because the names we needed to use for the application group and applications were not in any of the spooled file attributes, an output queue monitor exit program had to be written. The first challenge was to limit the number of exit programs we needed to write. Normally a monitor looks for a program by the name of the application that is determined by the APPSRC parameter for the STRMONOND command. Because our design was to use the User Data value to determine the application name, we would need hundreds of exit programs. For limit this we decided to use *FORMTYPE for the APPSRC value because it always contained *STD. This would call a program by the name of STD as the exit program (Content Manager OnDemand strips off characters that are not valid as a program name and changes any hyphens in the name to underscores). This allowed us to write a single exit program to handle all our spooled files. The function of this program was to take the values passed to it (spooled file name, user data, output queue name) and based on these values, determine what application group and application should process the report. We decided that we would use a database table to define the relationship between the spooled file's user data and spooled file name to come up with the application group name. In most cases, the user data provided a unique mapping, but when it did not, the spooled file name was used to get a unique mapping for the application group name. The application name was the User Data with part of the output queue name appended to it.

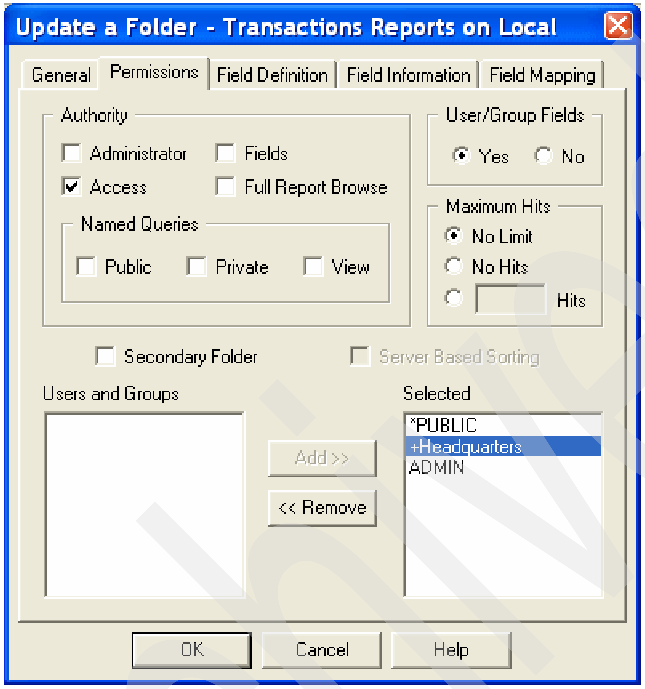
We decided we would start with five reports to verify that the design was appropriate for our reports. We started with the account maintenance journal. The first three pages of the report are shown in Figure 6-2 on page 268. This report is a branch type report, so we needed to separate it into documents by branch number. We first selected trigger 1① to be the name of the program that appears in the header at the right side of each page. Based on that trigger, we located the three index fields: rptdate❶, accttype❷, and branch❸. We set BREAK to Yes for the branch field so each branch would be in a new document. The accttype has values of Current Accounts and Savings Account in the reports. Because the first character of the values is unique, the accttype was defined as a string with a length of one to minimize the size of the index.



```
  21-03-2005                                FICTIONAL INTERNATIONAL BANK A            PROCESS-THRU 23-03-2005❶  PAGE    1
❷  CURRENT ACCOUNTS                                                                                            RP0046①
    TRANSACTION SYSTEM                    ACCOUNTS MAINTENANCE JOURNAL              3-21-05       23:35:07       20-0001
  ACCT NBR   NAME
 FIELD DESCRIPTION              OLD DATA                      NEW DATA                    LST MTN  T/C       TIME   WORKSTATION

 Branch          123❸      SANFRAN
 Currency Code   000      LOCAL DOLLARS
 User ID         DOEJOHN              JOHN DOE
  20001234   JOHN DOE
Charge Special Instr Code      0                             1                            22-03-05 026      9:25:14 W8006019


 Branch          123                     1
-------------------------------------------------------------------------------------------------------------------------------
    21-03-2005                                FICTIONAL INTERNATIONAL BANK A            PROCESS-THRU 23-03-2005❶  PAGE    2
❷  CURRENT ACCOUNTS                                                                                            RP0046①
    TRANSACTION SYSTEM                    ACCOUNTS MAINTENANCE JOURNAL              3-21-05       23:35:07       20-0001
  ACCT NBR   NAME
FIELD DESCRIPTION              OLD DATA                      NEW DATA                    LST MTN  T/C       TIME   WORKSTATION

 Branch          456❸     LOSANGE
 Currency Code   000      LOCAL DOLLARS
 User ID         DOEJANE               JANE DOE
  20005678   JANE DOE
Service Charge Type            0                             W                           28-01-04 023     10:55:11 W9346020

  20005678   JANE DOE
Service Charge Type            0                             W                           13-05-04 023     10:55:27 W9346020

Credit Interest Period                                       M                           13-05-04 023     10:55:27 W9346020

Credit Interest Cycle/Freq     000                           001                         13-05-04 023     10:55:27 W9346020

Credit Interest Specific Day   00                            31                          13-05-04 023     10:55:27 W9346020

Next Credit Int Date - Jul     0000000           0/00/00     2003090          31/05/04   13-05-04 023     10:55:27 W9346020

20005678   JANE DOE
Pay None/Pay All Spc Inst Cde  0                             1                           21-03-04 026     12:00:56 W9346020
-------------------------------------------------------------------------------------------------------------------------------
    21-03-2005                                FICTIONAL INTERNATIONAL BANK A            PROCESS-THRU 23-03-2005❶  PAGE    3
❷  CURRENT ACCOUNTS                                                                                            RP0046①
    TRANSACTION SYSTEM                    ACCOUNTS MAINTENANCE JOURNAL              3-21-05       23:35:07       20-0001
  ACCT NBR   NAME
FIELD DESCRIPTION              OLD DATA                      NEW DATA                    LST MTN  T/C       TIME   WORKSTATION

 Branch          456❸     LOSANGE
 Currency Code   000      LOCAL DOLLARS
 User ID         DOELISA               LISA DOE
  20005679   LISA DOE
Pay None/Pay All Spc Inst Cde  1                             0                           20-03-04 026      8:20:48 W9268003

Pay None/Pay All Spc Inst Cde  0                             1                           21-03-04 026     13:26:28 W9268003
```

Figure 6-2 Accounts Maintenance Journal sample data

A summary of its definition is displayed in Table 6-3 on page 272. Because this is a branch report, the folder fields have different attributes for headquarters' users and all other users (*PUBLIC). This is done by enabling User/Group Fields for the Headquarters group, as shown in Figure 6-3.



Authority

☐ Administrator ☐ Fields

☑ Access ☐ Full Report Browse

Named Queries

☐ Public ☐ Private ☐ View

User/Group Fields

● Yes ○ No

Maximum Hits

● No Limit

○ No Hits

○ Hits

☐ Secondary Folder ☐ Server Based Sorting

Users and Groups

Selected

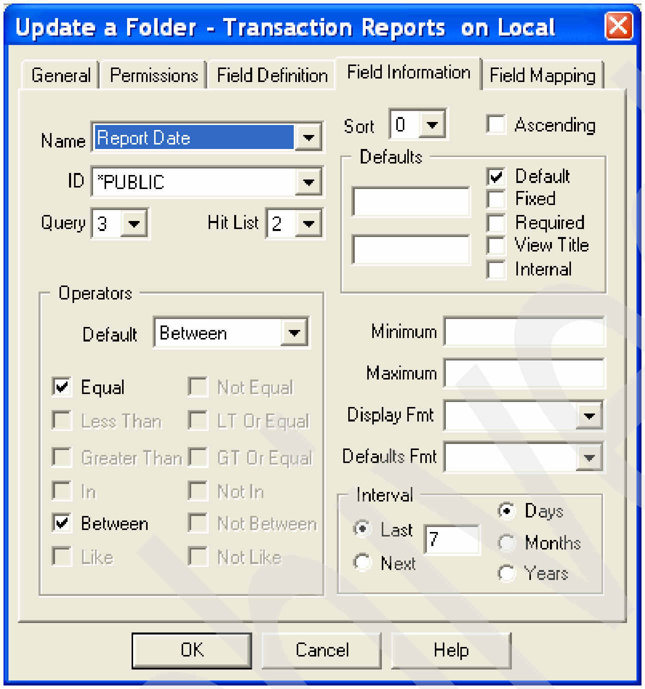
*PUBLIC

+Headquarters

ADMIN

Figure 6-3 Selecting User/Group Fields

Then, in the Field Information tab, you are able to specify different attributes for each folder field for *PUBLIC, as shown in Figure 6-4 and for the Headquarters group, as shown in Figure 6-5 on page 271.



Name: Report Date
ID: *PUBLIC
Query: 3
Hit List: 2
Sort: 0
☐ Ascending

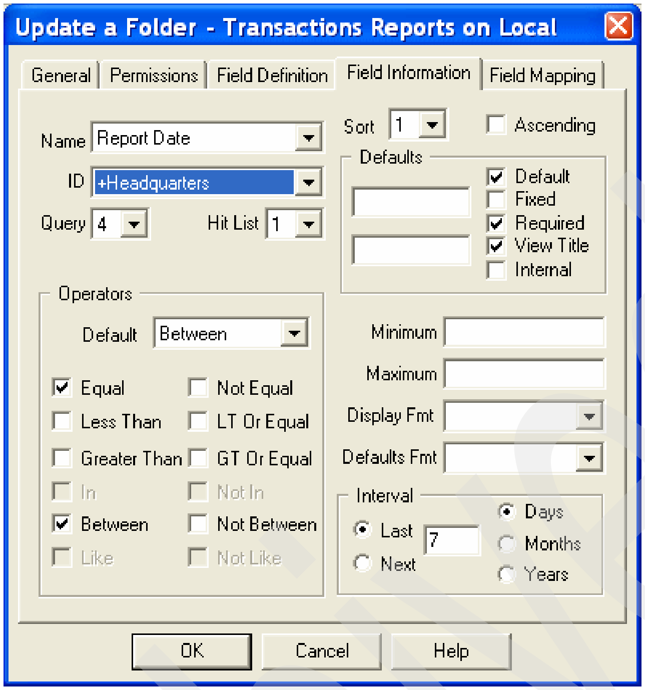
Defaults
☑ Default
☐ Fixed
☐ Required
☐ View Title
☐ Internal

Operators
Default: Between
☑ Equal ☐ Not Equal
☐ Less Than ☐ LT Or Equal
☐ Greater Than ☐ GT Or Equal
☐ In ☐ Not In
☑ Between ☐ Not Between
☐ Like ☐ Not Like

Minimum:
Maximum:
Display Fmt:
Defaults Fmt:

Interval
● Last 7
○ Next
● Days
○ Months
○ Years

Figure 6-4 Specifying Field information for *PUBLIC



Name: Report Date

ID: +Headquarters

Query: 4

Hit List: 1

Sort: 1

□ Ascending

Defaults

☑ Default
□ Fixed
☑ Required
☑ View Title
□ Internal

Operators

Default: Between

☑ Equal
□ Not Equal
□ Less Than
□ LT Or Equal
□ Greater Than
□ GT Or Equal
□ In
□ Not In
☑ Between
□ Not Between
□ Like
□ Not Like

Minimum:

Maximum:

Display Fmt:

Defaults Fmt:

Interval

● Last 7
○ Next

● Days
○ Months
○ Years

Figure 6-5 Specifying Field Information for Headquarters Group

The name of the application must be RP0046-*country* because the user data for this report’s spooled file is RP0046. Note that the query restriction in the application group for the headquarters group does not contain any restriction for the branch application group field, so they can see information for all branches.

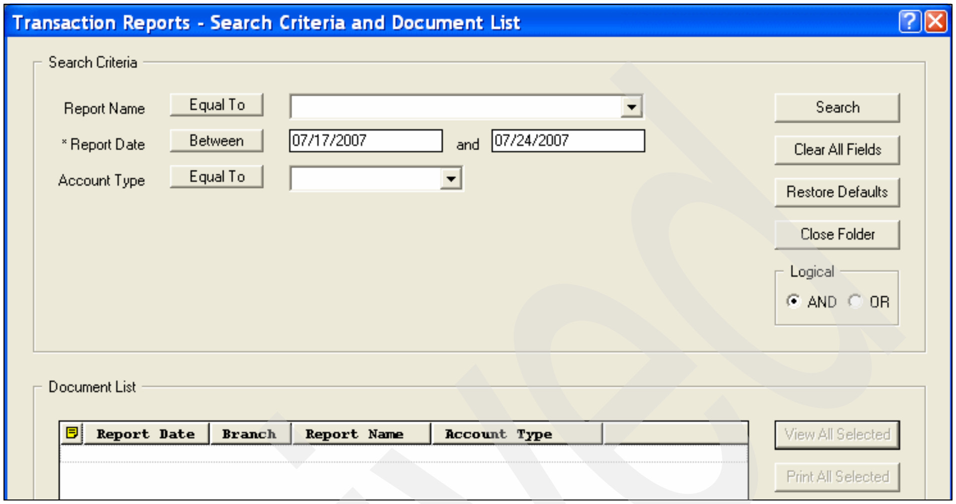
Table 6-3 summarizes the Account Maintenance Journal report definition.

Table 6-3 Account Maintenance Journal report definition

Folder:	**Transaction Reports**				
*PUBLIC	**Folder Field**	**Query**	**Hit List**	**Sort**	**Miscellaneous**
	Report Name	1	3	0	Default operator - Equal
	Branch	0	2	0	
	Account Type	3	4	0	Default operator - Equal
	Report Date	2	1	1 descending	Default operator - Between, Default, Required, View Title, Date interval - last 7 days
	Country	0	0	0	
Headquarters	**Folder Field**	**Query**	**Hit List**	**Sort**	**Miscellaneous**
	Report Name	3	4	0	Default operator - Equal
	Branch	2	3	0	Default operator - Equal
	Account Type	5	5	0	Default operator - Equal
	Report Date	4	1	1 descending	Default operator - Between, Default, Required, View Title, Date interval - last 7 days
	Country	1	2	0	Default operator - Equal
ApplGrp:	**Accounts Maintenance Journal**				
	Storage Set	**Cache**	**Life**	**Indexes**	**Migrate data from cache**
	Y07	90 days	2555 days	No Migr	When Data is Loaded
	DB Name	**Type**	**Data Type**	**Seg field**	**Miscellaneous**
	branch	Index	Integer(2)		
	acttype	Index	String(1)		Upper case
	rptdate	Index	Date	Yes	
	country	Filter	String(4)		Application ID field
Note:	This is a branch report so query restrictions for branch number and country must be specified.				
Application:	**RP0046-*country***				
	Load ID	**Embedded**	**Leading**	**Trailing**	**Miscellaneous**
	branch	,.	■[a]	■	
	acttype				
	rptdate	■	■	■	Format - %e-%m-%Y

a. This symbol (■)represents the blank character in the tables in this chapter.

Notice how the Search Criteria and Document List window looks different for *PUBLIC in Figure 6-6 and the Headquarters Group in Figure 6-7 because we are using User/Group Fields in the folder definition.



Transaction Reports - Search Criteria and Document List

Search Criteria

Report Name Equal To

* Report Date Between 07/17/2007 and 07/24/2007

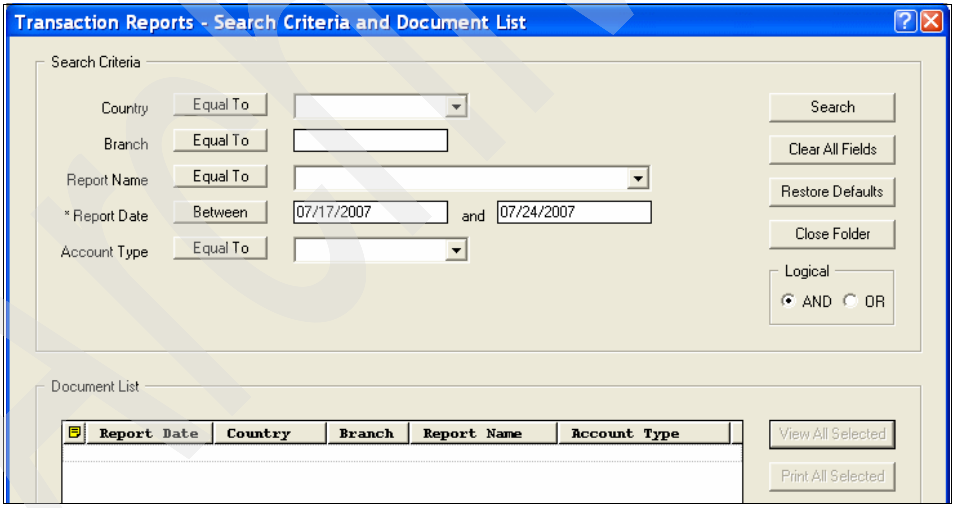
Account Type Equal To

Logical ● AND ○ OR

Document List

Report Date	Branch	Report Name	Account Type

Figure 6-6 Transaction Reports Folder for *PUBLIC



Transaction Reports - Search Criteria and Document List

Search Criteria

Country Equal To

Branch Equal To

Report Name Equal To

* Report Date Between 07/17/2007 and 07/24/2007

Account Type Equal To

Logical ● AND ○ OR

Document List

Report Date	Country	Branch	Report Name	Account Type

Figure 6-7 Transaction Reports Folder for Headquarters group

The second report is the Paid Exception Journal report and the definitions for it are shown in Table 6-4. It is a branch report and is categorized as a transaction report, so it is part of the Transaction Reports folder. The user data for this report's spooled file contains RP0135, so the name of the application must be RP0135-*country*.

Table 6-4 Paid Exception Journal report definition

Folder:	**Transaction Reports (See details for this in Table 6-3 on page 272.)**				
ApplGrp:	**Paid Exception Journal**				
	Storage Set	**Cache**	**Life**	**Indexes**	**Migrate data from cache**
	Y07	90 days	2555 days	No Migr	When data is loaded.
	DB Name	**Type**	**Data Type**	**Seg field**	**Miscellaneous**
	branch	Index	Integer(2)		
	acttype	Index	String(1)		Upper case.
	currency	Filter	Integer(2)		
	rptdate	Index	Date	Yes	
	country	Filter	String(4)		Application ID field.
Note:	This is a branch report so query restrictions for branch number and country must be specified.				
Application:	**RP0135-*country***				
	Load ID	**Embedded**	**Leading**	**Trailing**	**Miscellaneous**
	branch	,.	■[a]	■	
	acttype				
	currency		■	■	
	rptdate	■	■	■	Format - %e-%m-%Y.

[a] This symbol (■)represents the blank character in the tables in this chapter.

The third report is the Posting Rejects Journal and you can see a summary of its definition in Table 6-5. The name of the application is RP0130-*country*, because the user data for this report's spooled file is RP0130. Note that the query restriction for the headquarters' groups did not contain any restriction for the branch application group field so they could see information for all the branches.

Table 6-5 Posting Rejects Journal report definition

Folder:	**Transaction Reports (See details for this in Table 6-3 on page 272.)**				
ApplGrp:	**Posting Rejects Journal**				
	Storage Set	**Cache**	**Life**	**Indexes**	**Migrate data from cache**
	Y07	90 days	2555 days	No Migr	When data is loaded.
	DB Name	**Type**	**Data Type**	**Seg field**	**Miscellaneous**
	branch	Index	Integer(2)		
	acttype	Index	String(1)		Upper case.
	currency	Filter	Integer(2)		
	rptdate	Index	Date	Yes	Segment.
	country	Filter	String(4)		Application ID field.
Note:	This is a branch report so query restrictions for branch number and country must be specified.				
Application:	**RP0130-*country***				
	Load ID	**Embedded**	**Leading**	**Trailing**	**Miscellaneous**
	branch	,.	■[a]	■	
	acttype				
	currency		■	■	
	rptdate	■	■	■	Format - %e-%m-%Y.

[a] This symbol (■)represents the blank character in the tables in this chapter.

The fourth report is the Stop/Hold/Earmarks Journal and the definitions for it are shown in Table 6-6. It is a bank report (there is no branch index information) and is categorized as a transaction report so it is part of the Transaction Reports folder. The user data for this report's spooled file contains RP0060, so the name of the application must be RP0060-*country*.

Table 6-6 Stop/Hold/Earmark Journal report definition

Folder:	**Transaction Reports (See detail s for this in Table 6-3 on page 272.)**				
ApplGrp:	**Stop/Hold/Earmark Journal**				
	Storage Set	**Cache**	**Life**	**Indexes**	**Migrate data from cache**
	Y07	90 days	2555 days	No Migr	When data is loaded.
	DB Name	**Type**	**Data Type**	**Seg field**	**Miscellaneous**
	acttype	Index	String(1)		Upper case.
	rptdate	Index	Date	Yes	
	country	Filter	String(4)		Application ID field.
Note:	This is a bank report so query restrictions for country must be specified.				
Application:	**RP0060-*country***				
	Load ID	**Embedded**	**Leading**	**Trailing**	**Miscellaneous**
	acttype				
	rptdate	■[a]	■	■	Format - %e-%m-%Y.

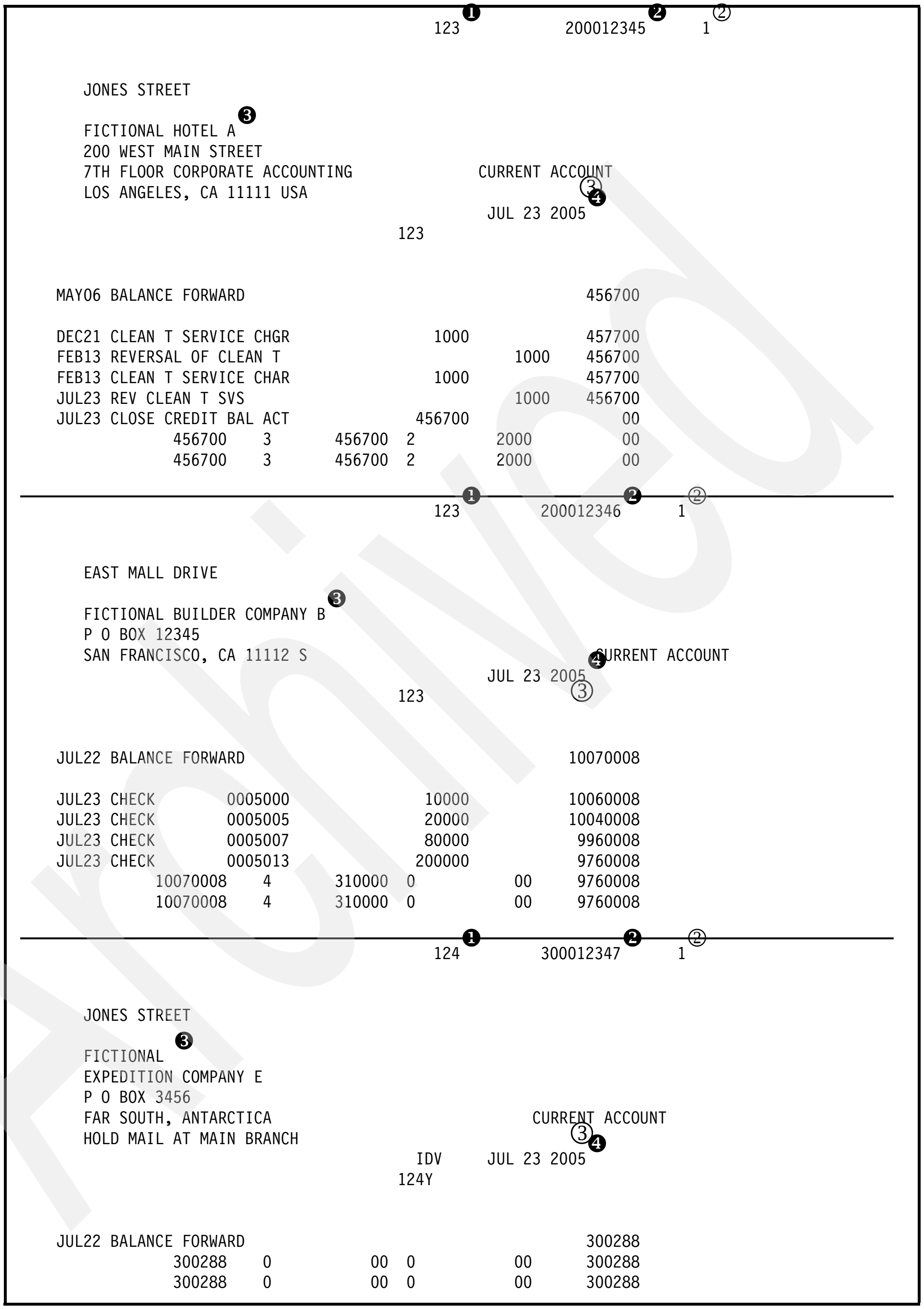
[a] This symbol (■)represents the blank character in the tables in this chapter.

The final report is the Checking Statements and the first three pages of a sample report are shown in Figure 6-8 on page 278. This report is a branch type report, so we needed to separate it into documents by branch number. Getting the index data is more difficult for this report because it is printed on a pre-printed form. It is not unusual for reports like this to have a variable number of records in the spooled file between index fields and this is no exception. To overcome this problem, we have to define multiple triggers so that we can find the various index fields. We first selected trigger 1 to be the new page carriage control character. We then defined trigger 2 ② to be the page number 1 so we will not try to locate the fields on the second and following pages of a multi-page statement. Trigger 3 ③ is a record range trigger and is the century portion of the statement date. It is a record range trigger because the statement date prints below the customer name and address, which will vary in the number of lines. The first three index fields, branch number❶, account number❷, and customer name❸, are all located by trigger 2. The statement date❹ is located by trigger 3. We set BREAK to Yes for the account number field so each statement would be in a new document.

Note: Many times when you are working with AFP reports or reports that are printed on pre-printed forms, it is very difficult to find constants in the actual report data that can be used as triggers. For these checking statements, trigger 3 is using the first two digits of the four digit year. This will work fine until the century changes and then this trigger will need to be changed. If this date only contained a two digit year, then you could only match on the decade and this would then have to be changed at the start of a new decade. Although it is not desirable to do this, sometimes there is no other alternative unless you are able to change the report by adding something else to the report that can be used as a trigger. For example, you might add a period to the statement date line that would print at the far right of the statement and could be used as a trigger.

This is in a folder by itself called Checking Statements because the index fields are different from all other reports. User/Group fields were specified for the auditing group because the auditing group in headquarters needs the capability to see checking statements in any country and branch. For the auditing group, both the branch and country need to be searchable folder fields. The user data for this spooled file contains RP9142 so the name of the application must be RP9142-*country*.

The sample data output is shown in Figure 6-8 on page 278. A summary of the definitions for Checking Statements is displayed in Table 6-7 on page 279.



```
                                              ❶                    ❷      ②
                                          123            200012345      1


   JONES STREET
                     ❸
   FICTIONAL HOTEL A
   200 WEST MAIN STREET
   7TH FLOOR CORPORATE ACCOUNTING             CURRENT ACCOUNT
   LOS ANGELES, CA 11111 USA
                                                           ③❹
                                                JUL 23 2005
                                      123


MAY06 BALANCE FORWARD                                      456700

DEC21 CLEAN T SERVICE CHGR                1000             457700
FEB13 REVERSAL OF CLEAN T                          1000    456700
FEB13 CLEAN T SERVICE CHAR                1000             457700
JUL23 REV CLEAN T SVS                              1000    456700
JUL23 CLOSE CREDIT BAL ACT              456700                 00
             456700    3       456700  2         2000          00
             456700    3       456700  2         2000          00
----------------------------------------------❶-----------------❷------②---------------------
                                          123         200012346        1


   EAST MALL DRIVE
                               ❸
   FICTIONAL BUILDER COMPANY B
   P O BOX 12345
   SAN FRANCISCO, CA 11112 S                                 CURRENT ACCOUNT
                                                            ❹
                                                JUL 23 2005
                                                          ③
                                      123


JUL22 BALANCE FORWARD                                    10070008

JUL23 CHECK        0005000              10000            10060008
JUL23 CHECK        0005005              20000            10040008
JUL23 CHECK        0005007              80000             9960008
JUL23 CHECK        0005013             200000             9760008
           10070008    4       310000  0           00     9760008
           10070008    4       310000  0           00     9760008
----------------------------------------------❶-----------------❷------②---------------------
                                          124         300012347        1


   JONES STREET
              ❸
   FICTIONAL
   EXPEDITION COMPANY E
   P O BOX 3456
   FAR SOUTH, ANTARCTICA                              CURRENT ACCOUNT
   HOLD MAIL AT MAIN BRANCH                                  ③ ❹
                                        IDV     JUL 23 2005
                                      124Y


JUL22 BALANCE FORWARD                                      300288
             300288    0           00  0           00      300288
             300288    0           00  0           00      300288
```

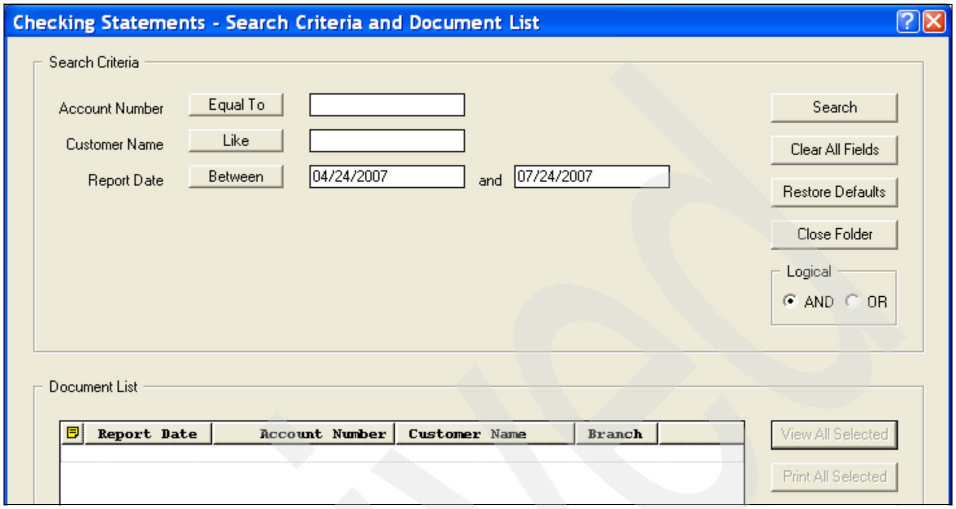
Figure 6-8 Check Statements sample data

Table 6-7 Checking Statements

Folder:	**Checking Statements**				
*PUBLIC	**Folder Field**	**Query**	**Hit List**	**Sort**	**Miscellaneous**
	Branch	0	4	0	
	Account Number	1	2	0	Default operator - Equal.
	Customer Name	2	3	0	Default operator - Like, Wild Card Prepend, Append.
	Report Date	3	1	1 descending	Default operator - Between, Default value, Date interval - last 3 months.
	Country	0	0	0	
Auditing	**Folder Field**	**Query**	**Hit List**	**Sort**	**Miscellaneous**
	Branch	2	4	0	Default operator - Equal.
	Account Number	3	2	0	Default operator - Equal.
	Customer Name	4	3	0	Default operator - Like, Wild Card Prepend, Append.
	Report Date	5	1	1 descending	Default operator - Between, Default, Date interval - last 60 days.
	Country	1	5	0	Default operator - Equal.
ApplGrp:	**Checking Statements**				
	Storage Set	**Cache**	**Life**	**Indexes**	**Migrate data from cache**
	Y07	90 days	2555 days	No Migr	When data is loaded.
	DB Name	**Type**	**Data Type**	**Seg field**	**Miscellaneous**
	branch	Index	Integer(2)		
	actnbr	Index	Big Int		
	custname	Index	String 40		Upper case.
	rptdate	Index	Date	Yes	Segment.
	country	Filter	String(4)		Application ID field.
Note:	This is a branch report, so query restrictions for branch number and country must be specified.				
Application:	**RP9142-*country***				
	Load ID	**Embedded**	**Leading**	**Trailing**	**Miscellaneous**
	branch	,.	■[a]	■	
	actnbr	,.	■	■	
	custname				
	rptdate	■	■	■	Format - %b%e%Y.

[a] This symbol (■)represents the blank character in the tables in this chapter.

Notice how the Search Criteria and Document List window looks different for *PUBLIC in Figure 6-9 and the Auditing Group in Figure 6-10 because we are using User/Group fields in the folder definition.



Checking Statements - Search Criteria and Document List

Search Criteria

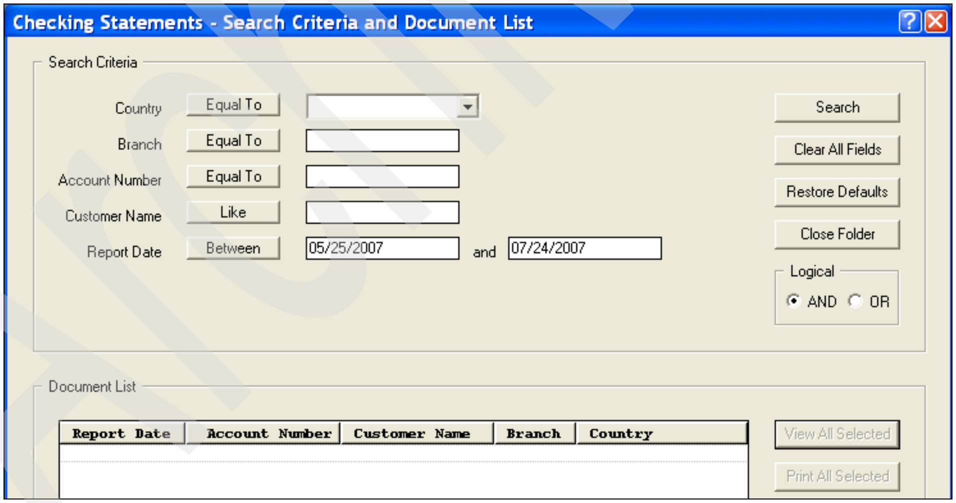
Account Number: Equal To
Customer Name: Like
Report Date: Between 04/24/2007 and 07/24/2007

Logical: ● AND ○ OR

Document List

Report Date	Account Number	Customer Name	Branch

*Figure 6-9 Checking Statements Folder for *PUBLIC*



Checking Statements - Search Criteria and Document List

Search Criteria

Country: Equal To
Branch: Equal To
Account Number: Equal To
Customer Name: Like
Report Date: Between 05/25/2007 and 07/24/2007

Logical: ● AND ○ OR

Document List

Report Date	Account Number	Customer Name	Branch	Country

Figure 6-10 Checking Statements Folder for Auditing Group

Performance testing

The primary performance concern was the time that it took users in the branches to retrieve a report. We set up a controlled environment in a couple of branch offices to take measurements. Tests were done using the Content Manager OnDemand Windows Client and Internet Explorer using ODWEK. We were pleasantly surprised to find that the use of ODWEK actually improved the performance for the remote users. This is because the ODWEK client is running on a Windows server right next to the iSeries and is on the same Ethernet network. When either client (ODWEK or the thick windows) opens a folder, all the application group definitions and application definitions in the folder are downloaded to the client. This can be a significant amount of data and the fact that ODWEK is on the same Ethernet loop provided a noticeable difference in response time.

In addition to report retrieval, testing was done archiving data to verify that all reports could be archived after the nightly batch processing and before the branches open in the morning.

Training

We first had a class to train the administrators so that they understood how to define reports, users, groups, and assign permissions. This training class was held for four days where each administrator received hands-on training.

User training was provided to a group of 15 people for one-half day who then trained the general users in the various countries before each country started production use of Content Manager OnDemand. This user training focused on using the Web client to search for documents and view them. It also reviewed the use of the icons that are on the viewer toolbar.

Deployment into production

Deployment was done by taking the definitions created on the test machine and using the Administration Client to copy (drag and drop) them to the server on the production machine. The following steps detail how this was done:

1. Migration policies were created on the production machine that matched those on the test machine. This allows them to be used for the application groups when the application groups are copied to the production machine. There is no way to copy migration policies between machines (servers).
2. Users were copied from the test machine to the production machine. We checked the **Ignore Warnings** box so the copy would not fail if the user already existed on the production machine.
3. Groups were copied next from the test machine to the production machine. We checked the **Ignore Warnings** box so the copy would not fail if the group already existed on the production machine

4. Application groups were then copied from the test machine to the production machine. We checked the **Ignore Warnings** box so the copy would not fail if there were errors and left the **No Storage Set** check box cleared so the existing storage sets in the production machine would be associated with the copied application groups. This also copied all the applications that were defined for each application group.
5. Finally, we copied the folders from the test machine to the production machine. We checked the **Ignore Warnings** box so the copy would not fail if the folder already existed on the production machine.

After the definitions were deployed to the production machine, we added the following jobs to the job scheduler:

1. A job to run a CL program to start Content Manager OnDemand after the backups were completed. This job did the following:
   a. Start the Content Manager OnDemand server.
   b. Start the Content Manager OnDemand output queue monitors.
2. A job to run a CL program to end Content Manager OnDemand before the backups were started:
   a. End the Content Manager OnDemand output queue monitors.
   b. End the Content Manager OnDemand server.
3. A job to run DSM and ASM. This ended the output queue monitors, ran DSM and ASM, and then re-started the output queue monitors.

We also set up the Windows 2000 Server to start ODWEK at 7:00 AM and to end it at 7:00 PM every day.

Once this was done, we began to use the production machine to archive and retrieve documents.

6.2 Telephone company case study

In this case study, we describe a fictitious telephone company implementation. We describe how this company was able to stop printing thousands of pages of paper and provide information quickly to customers and to internal users.

6.2.1 Company background

We refer to the fictitious telephone company as the company.

The implementation of this company reflects the implementation of several small to medium-sized telephone and communications companies.

The company is a telephone cooperative owned by the members it serves. It consists of three separate companies and has been in business for over 50 years. As technology has changed, the company has expanded its services and offers voice, video, high-speed data, and wireless communications.

6.2.2 Business requirements

The company identified the following requirements for the Content Manager OnDemand project:

- Stop printing thousands of pages of paper each week and storing them in a warehouse. This requirement is aimed at achieving the following goals:
  - Improve the speed of finding information.
  - Reduce the cost of paper.
  - Reduce the cost of warehouse space.
- Replace microfiche with Content Manager OnDemand for archiving the telephone bills. This requirement is aimed at achieving the following goals:
  - Improve customer service.
  - Reduce the cost of microfiche.
- Access the signed membership cards electronically instead of looking for them in the store room.
- Access the vendor invoices and check copies electronically instead of keeping them in cabinets in the Accounts Payable department.

6.2.3 Planning for implementation

We provided Content Manager OnDemand administrator training to three people from the Information Technology department. At the end of the training, they understood the capabilities and concepts of Content Manager OnDemand, had practice creating sample report definitions, and were able to make good decisions on how they could use Content Manager OnDemand with their reports. We decided to set up reports in ways that seemed reasonable to us, then show the results to the users for feedback and possible modifications. We gathered the following information about the environment:

- Over 200 green bar reports of data type *SCS.
- Almost all green bar reports have a header page with the report name and date, and are less than 100 pages.

- Vendor checks and phone bills of data type *AFPDS (Advanced Function Presentation Data Stream).
- Boxes of signed membership cards to be scanned and archived.
- Invoices from vendors to be scanned and archived.
- Security by output queue so users see reports only for their department or job function.
- Three companies within the cooperative, but no special security by company.
- Reports and documents need to be kept for either three or seven years.
- The current iSeries system has plenty of disk space and the CPU is not overutilized.

Based on the information gathered, we made these decisions:

- Start archiving phone bills and stop using microfiche to see immediate cost savings and increased productivity and satisfaction in customer service.
- Divide the green bar reports into categories, then define and start archiving reports by category.
- Install two scanners and set up Kofax Ascent Capture to start scanning and archiving invoices and membership cards.
- Send boxes of membership cards to a company who can scan the cards and return CDs containing TIFF images and index records that we can load into Content Manager OnDemand.
- Use large object support for green bar reports that will be archived as single documents and that are typically larger than 300 pages.
- Have two migration policies that retain data for three years and seven years, respectively.
- Create a separate migration policy for the System Log, so those documents will be aggregated together.
- No additional iSeries hardware is needed.

6.2.4 Implementation

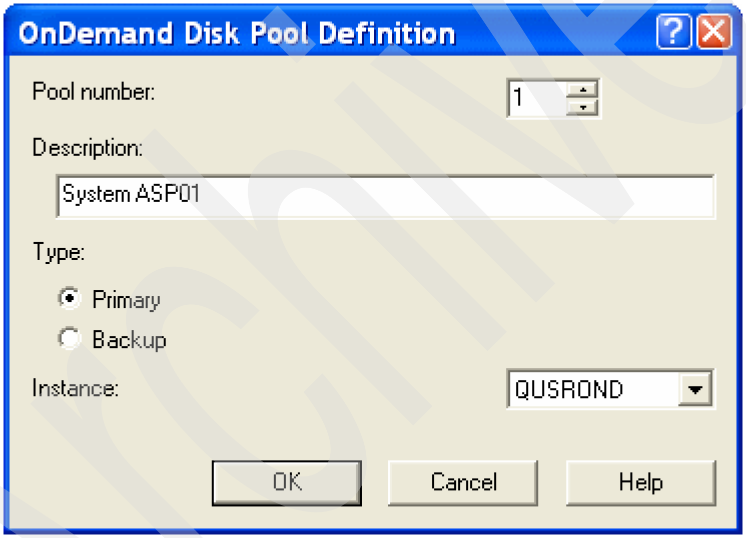
Implementing Content Manager OnDemand consists of software installation and configuration, application design, functional testing, performance testing, training, and final deployment into production.

Installation and configuration setup

We installed the prerequisite OS/400 software and the Content Manager OnDemand Licensed Program Product. After installing the latest OS/400 and Content Manager OnDemand PTFs, we created the default instance QUSROND. We changed three user profiles so they could be Content Manager OnDemand system administrators and added them as Content Manager OnDemand users (refer to the steps in Chapter 3, “IBM Content Manager OnDemand for i5/OS implementation guidelines” on page 75 for details).

Application design

During application design, we decided how the folders, application groups, applications, storage sets, users, and groups should be defined. We had already determined that we would keep some reports and documents on disk for three years and some for seven years. So we used iSeries Navigator to add the System ASP as disk pool 1. See Figure 6-11.



OnDemand Disk Pool Definition

Pool number: 1

Description: System ASP01

Type:

● Primary

○ Backup

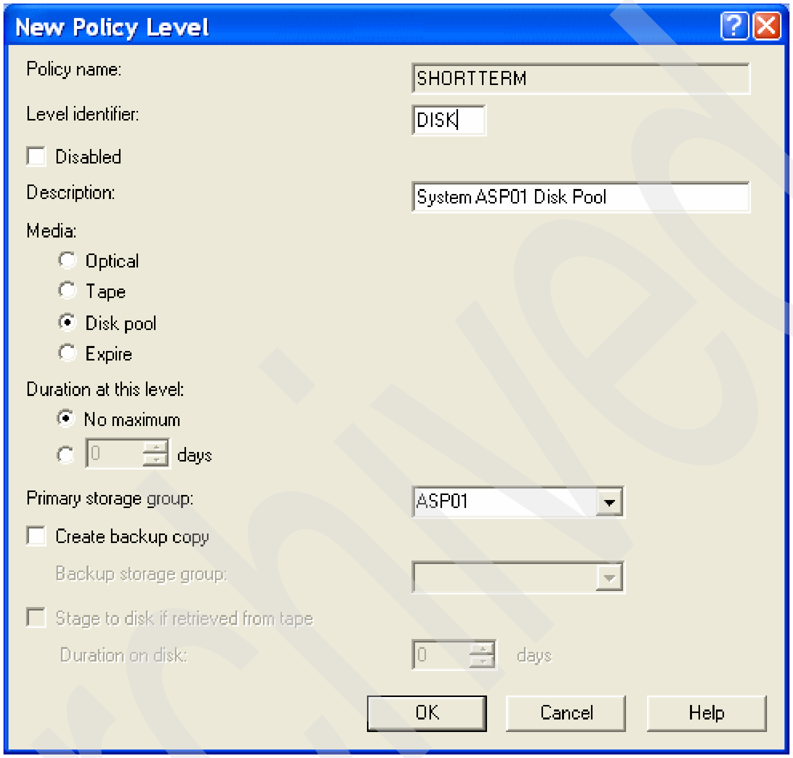
Instance: QUSROND

Figure 6-11 Disk Pool Definition

We created three migration policies that used the disk pool:

SHORTERM	Documents with a retention period of three years or less
LONGTERM	Documents with a retention period of seven years
SYSTEMLOG	Documents from the System Log

In each policy, we checked **No maximum for Duration at this level** and we enabled aggregation, as you can see in Figure 6-12 and in Figure 6-13 on page 287. Each report will expire based on the number of days in the Life of Data and Indexes parameter in the application group, as you see in Figure 6-14 on page 287.



New Policy Level

Policy name: SHORTTERM

Level identifier: DISK

□ Disabled

Description: System ASP01 Disk Pool

Media:

○ Optical

○ Tape

● Disk pool

○ Expire

Duration at this level:

● No maximum

○ 0 days

Primary storage group: ASP01

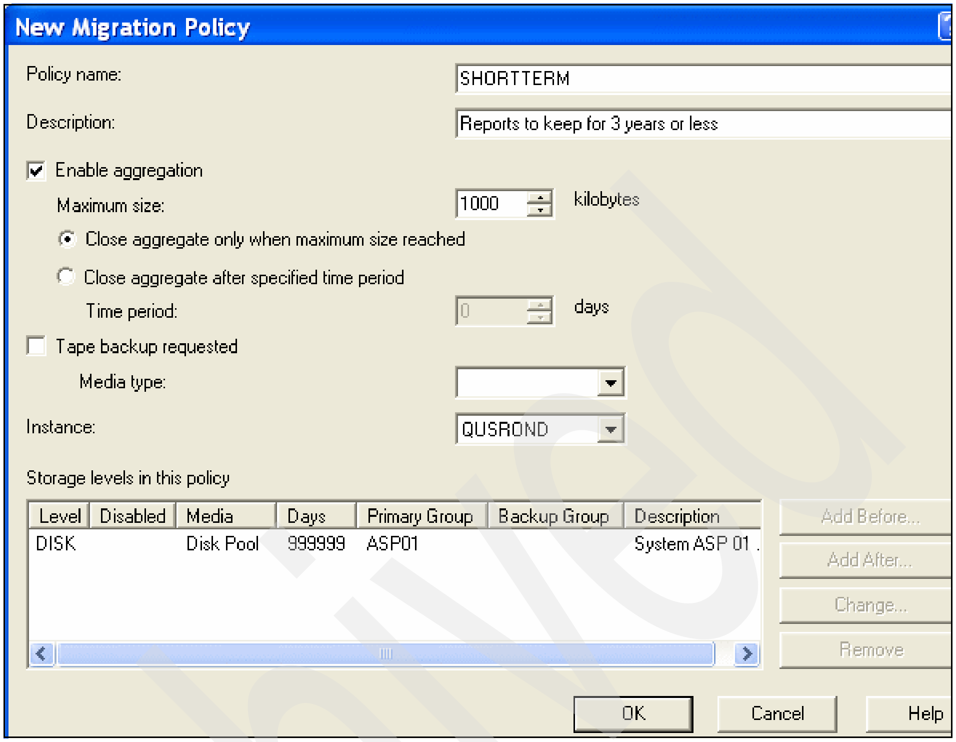
□ Create backup copy

Backup storage group:

□ Stage to disk if retrieved from tape

Duration on disk: 0 days

Figure 6-12 Policy Level



Policy name: SHORTTERM

Description: Reports to keep for 3 years or less

☑ Enable aggregation

Maximum size: 1000 kilobytes

● Close aggregate only when maximum size reached

○ Close aggregate after specified time period

Time period: 0 days

□ Tape backup requested

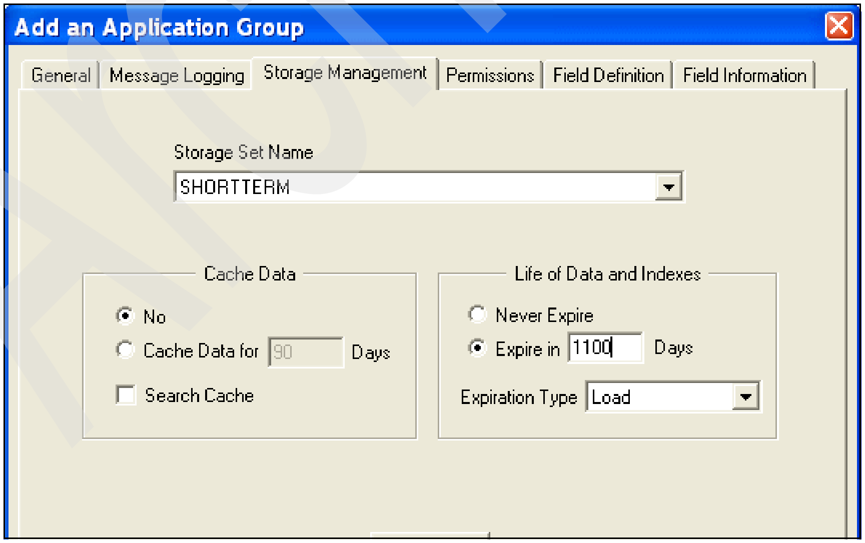
Media type:

Instance: QUSROND

Storage levels in this policy

Level	Disabled	Media	Days	Primary Group	Backup Group	Description
DISK		Disk Pool	999999	ASP01		System ASP 01 .

Figure 6-13 SHORTTERM Policy



Storage Set Name: SHORTTERM

Cache Data

● No

○ Cache Data for 90 Days

□ Search Cache

Life of Data and Indexes

○ Never Expire

● Expire in 1100 Days

Expiration Type: Load

Figure 6-14 Application Group Storage Management

We could have used a single migration policy for all application groups since they all use the same disk storage level. But we decided not to do that because different reports would be aggregated (joined) together, and reports that should expire after three years would physically still exist if they happened to be aggregated with reports that are kept for seven years. The indexes would be deleted, but the compressed data would still be on disk, although not accessible. This is usually not a concern; the data is compressed, and there is enough disk space, but it is still a better idea to keep the data separate.

If we had a lot of large reports, we would have used an aggregation size larger than the default of 1000 kilobytes in order to have fewer objects in the disk pool. The size can be changed at any time if necessary.

You may prefer to name the migration policies by the type of business application. For example, create a PAYROLL policy for all application groups used by the payroll department, which all need to be retained for 10 years. Or if you plan to use optical storage, you may choose to put all reports that use optical in a single policy. If you have different reports that you need to keep for a very short time, for example, 30 days, create a policy called TEMPREPORTS and let all those reports use this policy. You can still specify **No maximum** for the storage level duration because the storage management in the application groups will determine when the reports will expire.

We discussed the indexing requirements for each of the reports, obtained sample spooled files, set up the definitions, and reviewed the results with the users. For the scanned documents, we set up definitions in Content Manager OnDemand and in Kofax Ascent Capture. Table 6-8 shows the design for the first set of reports.

Table 6-8 Folders for the company

Folder	App group /Application	Data type	App group fields	Folder fields	Field type
Phone Bills	BILLS	AFP	bdate phone # name version (appl ID)	Date Phone # Name	date string string (var) string
Billing and Inventory folders	Several application groups in each folder	SCS	rdate company rptname rptname2 version	Report Date Company Report Name Report Name2	date string string string

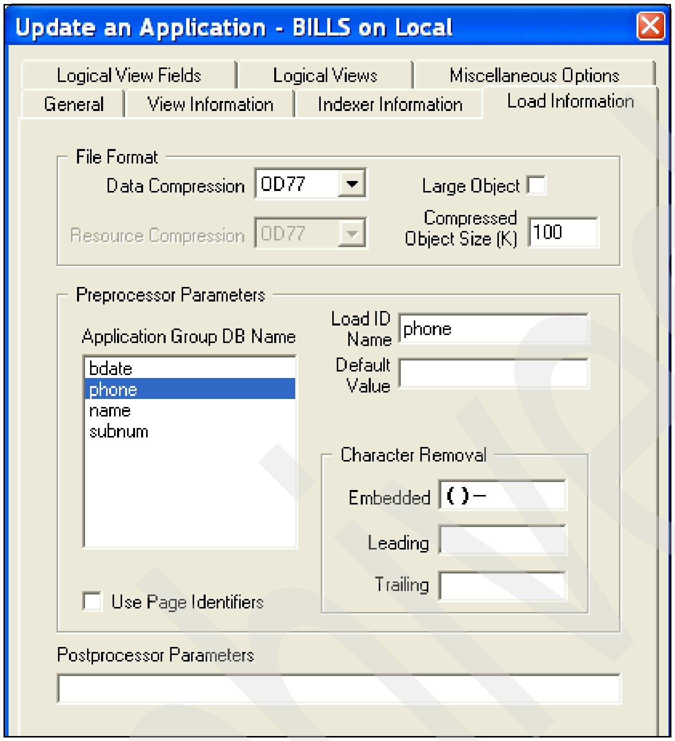
Folder	App group /Application	Data type	App group fields	Folder fields	Field type
Vendor Invoices and Checks	INVOICES VENCHECKS	AFP TIFF	invdate/ckdate invnum vendnum vendname cknum ckamt doctype (appl ID)	Date Invoice # Vendor # Vendor Name Check # Amount Document Type	date string integer string (var) integer decimal string
Membership	MEMBSCAN	TIFF	date memnum memname applid	Date Member # Member Name	date integer string (var) string

Each application group has one application, and the names are the same for both the application group and the application. The name matches the user data field in the spooled file, so we can use the output queue monitor program to automatically archive the files. MEMBSCAN and VENCHECKS (the scanned TIFF images) match the batch and document classes we defined in Kofax Ascent Capture.

We included an application ID field (version) in each application group so that if the layout of the spooled file changes, we can easily add another application in the application group. In that case, we would rename the existing application and add the application ID as a suffix, for example, ABC would become ABC-01. Next, we would add the value “02” as an application ID value in the application group. Then we would copy the ABC-01 application to a new application named ABC with application ID 02, and make the necessary changes to the field locations in the application. With this method, the currently used application has the same name as the application group, so the STRMONOND command will continue to work.

In each application that has a variable length string field, we removed trailing spaces for that field. We also noted that in the user training we would need to make sure to tell the users to select the option **Document Column List Autosize** so that the width of the fields in the document list would adjust to the actual size of the fields.

In the BILLS application, we removed embedded characters from the phone number so that only the numbers would be loaded into the database field. For example, (800)555-1212 would load as 8005551212. We also removed the embedded characters in the application group field so that if a user entered those special characters, Content Manager OnDemand would remove them before searching for the document index. You can see this technique in the application definition in Figure 6-15 and in the application group definition in Figure 6-16.



Update an Application - BILLS on Local

File Format

Data Compression: OD77

Large Object □

Resource Compression: OD77

Compressed Object Size (K): 100

Preprocessor Parameters

Application Group DB Name:
- bdate
- phone
- name
- subnum

Load ID Name: phone

Default Value:

Character Removal

Embedded: ()–

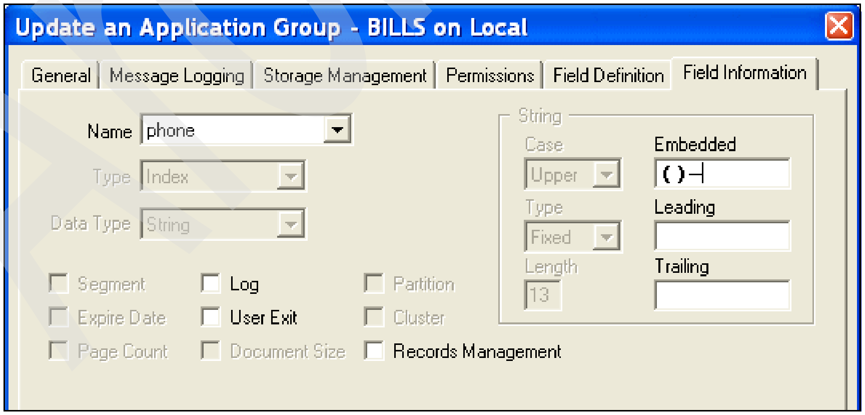
Leading:

Trailing:

□ Use Page Identifiers

Postprocessor Parameters:

Figure 6-15 Application remove characters



Update an Application Group - BILLS on Local

Name: phone

Type: Index

Data Type: String

□ Segment □ Log □ Partition

□ Expire Date □ User Exit □ Cluster

□ Page Count □ Document Size □ Records Management

String

Case: Upper

Type: Fixed

Length: 13

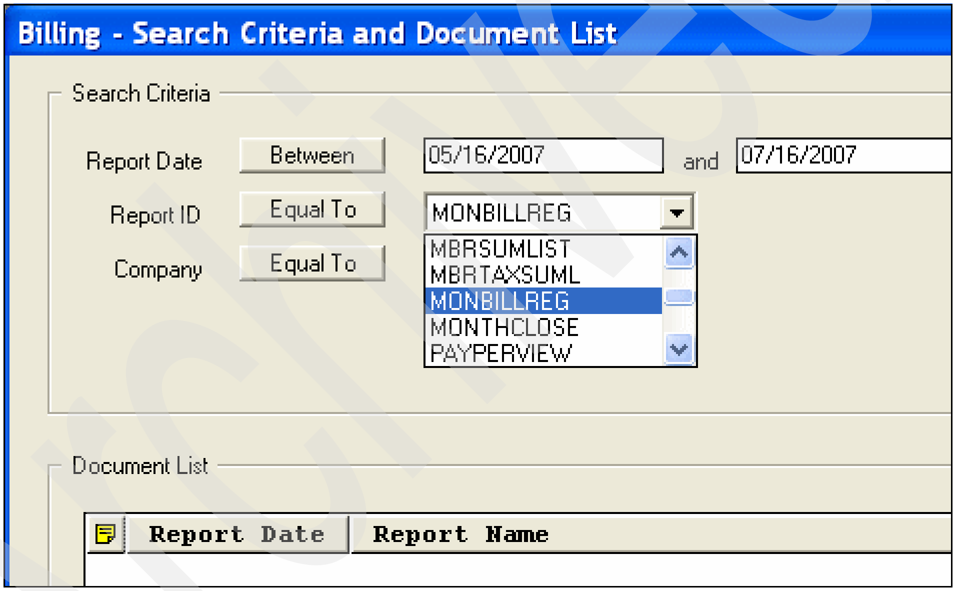
Embedded: ()-

Leading:

Trailing:

Figure 6-16 Application group remove characters

We used a very simple design for the billing and inventory reports. Each report has a header page that contains a box in the center of the page. Within the box is a one- or two-line description of the report, the date of the report, and the company number. The users know each of their reports by the long report name and also by the short name (which is the user data in the spooled file, and also the name we used for the application and application group). For example, they know that the short name for the Monthly Billing Register is MONBILLREG. The users wanted to have a single place to find all the billing reports, so we created one folder that contained all the billing reports, with a separate application group for each report. Normally, it is not a good idea to have multiple application groups in a single folder. But in this case, the application group name is one of the search fields in the folder, referred to as the Report ID. A user selects only one report ID, so only one application group database file is opened for searching. See Figure 6-17.



Billing - Search Criteria and Document List

Search Criteria

Report Date Between 05/16/2007 and 07/16/2007

Report ID Equal To MONBILLREG

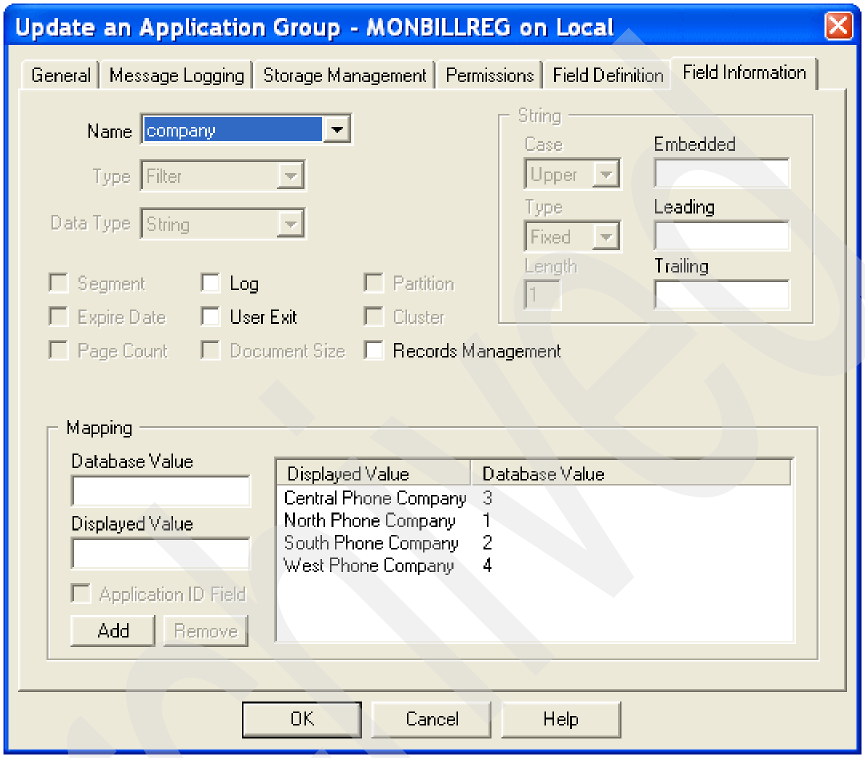
Company Equal To

Document List

Report Date	Report Name

Figure 6-17 Billing folder search by report ID

The company number is a single character string field. In each application group we map that field to a displayed value that is more descriptive. You can see the field in the application group in Figure 6-18, and you can see how the users search for a Monthly Billing Register by company in Figure 6-19 on page 293.



Name: company

Type: Filter

Data Type: String

□ Segment □ Log □ Partition

□ Expire Date □ User Exit □ Cluster

□ Page Count □ Document Size □ Records Management

String

Case: Upper

Type: Fixed

Length: 1

Embedded:

Leading:

Trailing:

Mapping

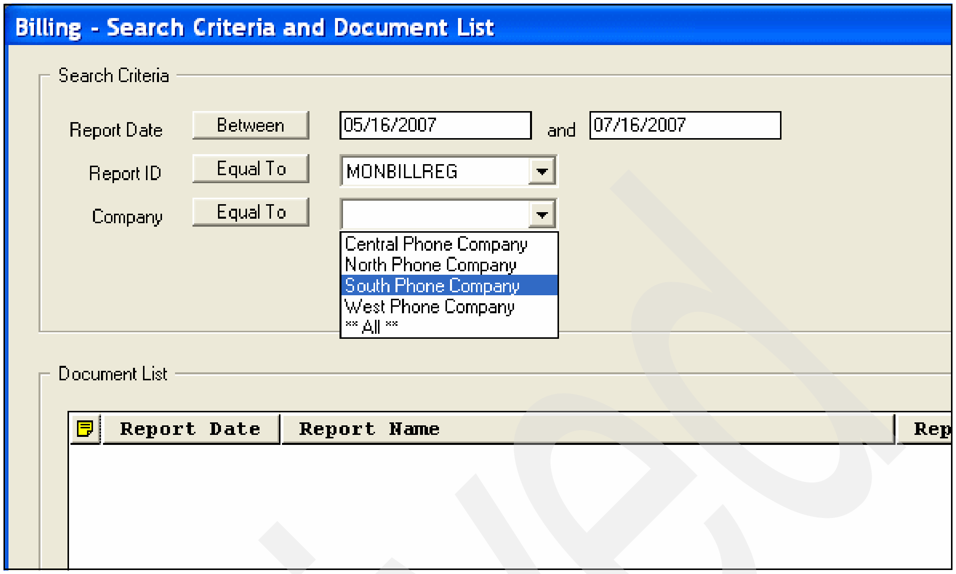
Database Value:

Displayed Value:

□ Application ID Field

Displayed Value	Database Value
Central Phone Company	3
North Phone Company	1
South Phone Company	2
West Phone Company	4

Figure 6-18 Application group company field



Billing - Search Criteria and Document List

Search Criteria

Report Date	Between	05/16/2007	and	07/16/2007
Report ID	Equal To	MONBILLREG		
Company	Equal To			

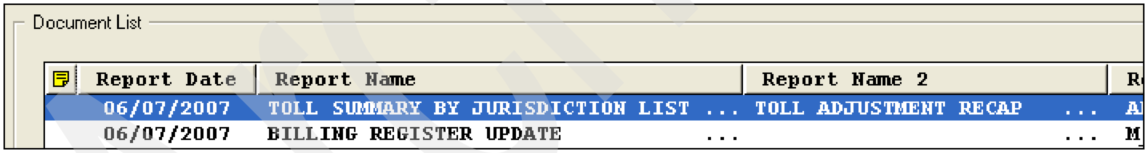
Central Phone Company
North Phone Company
South Phone Company
West Phone Company
** All **

Document List

	Report Date	Report Name	Rep

Figure 6-19 Billing folder search by company

The user does not search on Report Name or Report Name 2. These are filter fields that are only displayed in the document list. There is always a value for Report Name, but there is a value for Report Name 2 only when there is a second line description on the report's header page. See Figure 6-20 for an example.



Document List

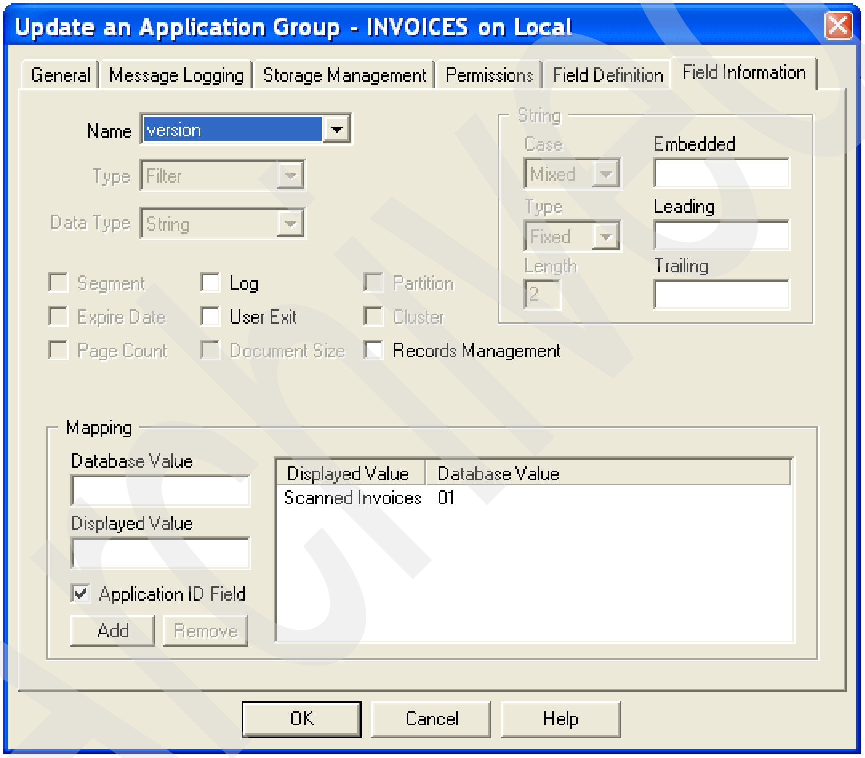
	Report Date	Report Name	Report Name 2	R
	06/07/2007	TOLL SUMMARY BY JURISDICTION LIST ...	TOLL ADJUSTMENT RECAP ...	A
	06/07/2007	BILLING REGISTER UPDATE ...	...	M

Figure 6-20 Billing document list

We created a folder for Billing and one for Inventory. Then we used the Report Wizard to create each of the billing and inventory definitions and deleted each of the individual folders the Wizard created. We mapped the fields from each application group to the folders.

The INVOICES application group and application are for the invoices that the company receives from vendors. We installed Kofax Ascent Capture to scan and index the documents, then release them to an IFS directory on the iSeries. We issued the command `arsload` to monitor that directory and archive the files into Content Manager OnDemand. We used database validation in the document class in Kofax Ascent Capture. A user scans an invoice and types in only the invoice number and the check number. Ascent Capture sends a request to a database file on the iSeries that contains all the information needed for each check. The user reviews the values that are found for vendor number, vendor name, check date, and check amount, then presses the Enter key to process the next check.

We mapped the application ID fields to descriptive names for the VENCHECKS and INVOICES application groups. We used the name Document Type for the application ID field in the folder, although the field is called version in each application group. See how we mapped the application group field in Figure 6-21. See how the user searches the folder in Figure 6-22 on page 295.


Update an Application Group - INVOICES on Local

Name: version
Type: Filter
Data Type: String

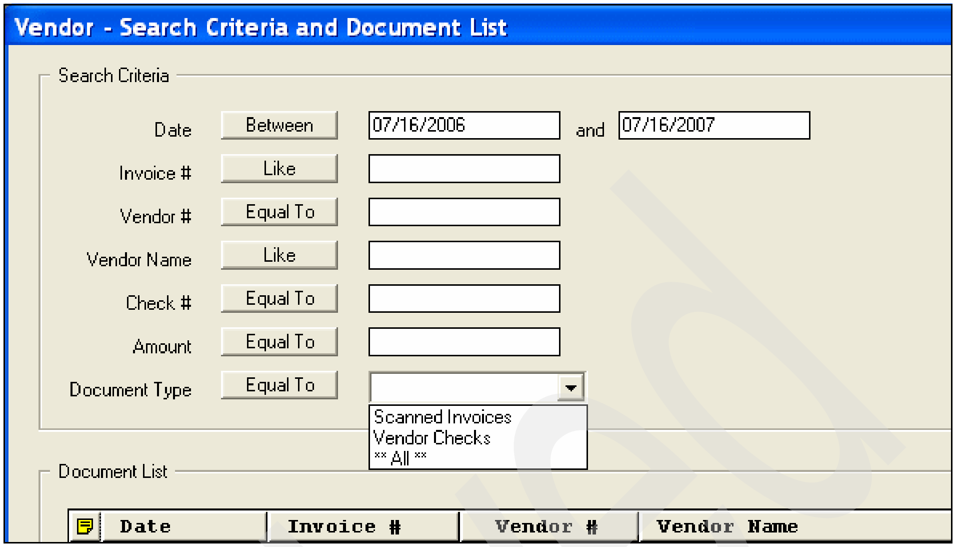
String
Case: Mixed
Type: Fixed
Length: 2
Embedded:
Leading:
Trailing:

□ Segment
□ Expire Date
□ Page Count
□ Log
□ User Exit
□ Document Size
□ Partition
□ Cluster
□ Records Management

Mapping
Database Value:
Displayed Value:
☑ Application ID Field

Displayed Value	Database Value
Scanned Invoices	01

Figure 6-21 INVOICES application ID field



Date Between 07/16/2006 and 07/16/2007

Figure 6-22 Vendor folder search

Security

Security at the company is easy to manage. We have only two user types: user and system administrator. We had already added the three administrators when we set up Content Manager OnDemand and did the administrator training. We created four user groups: Accounting, Financial, Customer Service, and Payroll. Then we added users to one or more of those groups. As we created report definitions, we gave Logical Views permission to *PUBLIC for each application group. For each folder, we assigned permissions to user groups with no access to *PUBLIC.

Functional testing

We created an output queue called ONDTEST and started a monitor for it. For each report, we used the Report Wizard with a sample spooled file to create the application group, application, and folder. Then we updated the storage management and permissions in the application group and the date format, if necessary, in the application. We also removed embedded and trailing characters for variable length string fields. Then we moved the spooled file to the ONDTEST queue. If it loaded successfully, we viewed the report in the Content Manager OnDemand Windows Client to make sure the results were what we expected. Then we showed the report to the users for approval.

Performance testing

We had no performance problems archiving or retrieving reports or scanned documents. We avoided the problem of slow logons by inserting this line in the file /qibm/UserData/OnDemand/QUSROND/ARS.CFG:

```
ARSSOCK_RESOLVE_CLIENT_NAME=0
```

When we trained the users, we showed them how to limit their searches by date range and how to press the hand icon to stop the search.

Training

We gave informal hands-on training classes to the users so they could learn how to find their reports and documents. We showed them many features of the Content Manager OnDemand Windows Client, such as those described in Chapter 3, “IBM Content Manager OnDemand for i5/OS implementation guidelines” on page 75.

When we installed the Client, we told the users to check the Auto Note Retrieval option, as shown in Figure 6-23. Now when they open a document that contains a note, they see the yellow sticky note displayed and can easily click it to open it. We told them to check the **Set Document List Column Autosize** option so that the columns in the hit list would be sized correctly when variable length fields are used.

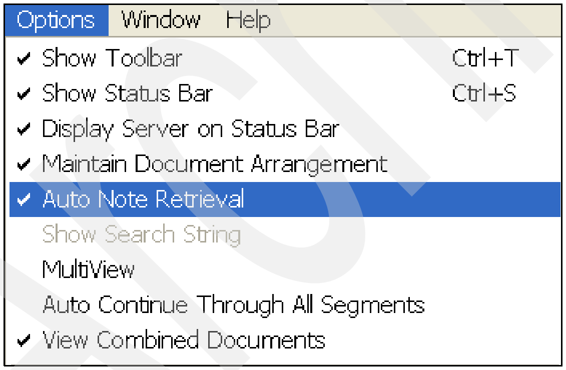

Options Window Help
✓ Show Toolbar Ctrl+T
✓ Show Status Bar Ctrl+S
✓ Display Server on Status Bar
✓ Maintain Document Arrangement
✓ Auto Note Retrieval
Show Search String
MultiView
Auto Continue Through All Segments
✓ View Combined Documents

Figure 6-23 Auto note retrieval

We also trained the users who would be scanning the membership cards and invoices. We taught the system administrators how to use Kofax Ascent Capture to create batch and document classes. One of them attended a Kofax class to get more education on the product.

Deployment into production

We created two output queues,ONDPRT01 and ONDDLT, and started a monitor on each of them. The spooled files in ONDPRT01 are archived into Content Manager OnDemand and then sent to the PRT01 output queue to be printed. The files in ONDDLT are archived, then deleted. We also created an error output queue for each of the monitors instead of using the default QUSRRDARS/ONDERR. That way we would know which output queues to use after we fixed the application problems for the spooled files in the error queues. And we changed the business applications so that the spooled files were created in the output queues monitored by Content Manager OnDemand.

We added several jobs to the OS/400 job scheduler so that we could automate many of the Content Manager OnDemand processes. Backups are run after midnight. After that, the server is started and the output queue monitors are started. In the evening, the monitors are ended (the end time is in the STRMONOND command), DSM and ASM are run with the STRDSMOND command, the server is ended, and the ASP01 file system is unmounted. We submit an **arsload** command from the job scheduler once a week, after weekly backups when the system is shut down and re-started. The arsload program is used to monitor the IFS directory where the files are sent from the PC scanning station. The command we used is:

```
QSH CMD('arsload -d /ASCENT/SCAN01 -f -t 120')
```

Every two minutes (120 seconds), the /ASCENT/SCAN01 directory is checked to see if there is a new .ard (header) file, which indicates that an .ind file and an .out file have been sent from the PC. We used the default naming conventions, so the batch and document classes match the names of the applications and application groups. The files are deleted after being processed. If there are errors, the .ard file is renamed to have a .failed extension.

Each day an operator looks to see if there are any spooled files in one of the error output queues, or if there are any .failed files in the /ASCENT/SCAN01 directory. If so, an administrator gets involved to solve the problem.

The company sent their membership cards to a company who scans documents and creates TIFF images. They also sent them a text file that contained a list of member numbers and names. The scanning company created a CD with the TIFF images and a text file with a record for each document. Each record included the application group database field names and values for the MEMBSCAN application group, along with the locations on the CD for the associated images. We used this information to import and archive the files into Content Manager OnDemand.

The administrators have just started using the Content Manager OnDemand Toolbox to index and archive PC files. They plan to work with different users to see if they have spreadsheets or documents they would like to store in Content Manager OnDemand.

7

Case studies for z/OS

In this chapter, we describe three case studies for Content Manager OnDemand for z/OS implementations. For each case study, we describe the company background and the business requirements. Using the information, techniques, and recommendations we addressed earlier in the book, we discuss how to plan for the implementation and the actual implementation steps for each case study.

The cases studies we cover in this chapter are for:

- A financial institution
- A public sector
- An educational institution

7.1 Financial institution case study

In this case study, we describe a fictitious customer which has issued a request for proposal (RFP) for a project involving electronic delivery of customized reports through a state of the art report delivery mechanism. To start, we will profile the customer so that you understand the customer's business environment and the corporation's position in the world financial market. Then, we will describe how Content Manager OnDemand for z/OS was able to:

- Meet and exceed the customers business needs by utilizing innovative implementation practices.
- Help the customer gain a competitive edge in today's on demand world.
- Minimize costs by delivery of information to the right people at the right time easily and cost effectively.

7.1.1 Company background

We refer to the fictitious financial institution as the company. The company is one of the largest custodian of financial transactions in the world. It is one of the leaders in managing customer assets, providing securities finance services and foreign exchange services. It is also one of the top investment managers of institutions and institution tax-exempt assets worldwide. Within the U.S., it leads in mutual funds and accounting services, and U.S. pension funds management.

The timely delivery of information and reports is fundamental to maintaining this leadership status. Value added products and services that provide real time, online access to client's account and fund information are key to competitive differentiation and are key to client retention. The company's clients want personalized fund information, in a variety of standard formats, delivered through both Web and desktop interfaces.

7.1.2 Business requirements

To become an *On Demand* business, the company recognized the need to make information available across processes throughout its worldwide enterprise, bringing new levels of integration to processes and applications. The company also recognized it could not realistically achieve those goals with their current infrastructure for content managed reports and documents. They needed a better solution. This led to the creation of the RFP. After extensive study of the RFP responses, the company recognized the business benefits of implementing the Content Manager OnDemand solution. This solution addressed both content delivery and infrastructure improvements within their enterprise.

By selecting Content Manager OnDemand for z/OS as their enterprise solution, the company was in position to address the following critical business requirements:

- Immediate access to fund accounting reports 24/7.

  Eliminate the potential for lost revenue. The company was losing revenue due to their inability to provide Web access to fund accountants and their daily transactions, which resulted in missed deadlines to meet the daily Stock Exchange and NASDAQ quotes.

- Implement a state of the art report distribution system.

  Replace an antiquated in-house system with Content Manager OnDemand for z/OS using the Content Manager OnDemand's OnDemand Distribution Facility (ODF) to meet the needs of its global user base and package and deliver client data in a time critical, distributed, and highly demanding environment. Sharply reduce report delivery and print.

- Presentation layer migration

  Provide a new user experience to their user community by changing from the *Green Screen* viewer to a Windows based and Web-enabled interface that is rich in features and functions.

- Publishing of the application reports (and associated data) to external clients.

  Retrieve content that has been stored in its native format and dynamically convert it into e-content formats such as PDF, HTML, or XML for viewing, distribution, or application ingestion.

- User productivity gains with the introduction of a defined workflow.

  Establish a business process that initiates a workflow when fund account transactions are captured. This positions reviewers to approve or disapprove the transaction based on validity and then make them accessible to a final reviewer for approval. External fund accountants would then see the final report based on fund accounts with all the daily activity.

- Utilize current infrastructure.

  Using the company's existing system components, such as the current security package, database system, storage manager, and communication protocol, is critical in the decision to implement the Content Manager OnDemand for z/OS solution. Reduce application transition time.

- Eliminate current vendor product constraints.

  The company was struggling with development and supporting its current content delivery infrastructure. Its inability to provide APIs and scalability restraints would be removed with a Content Manager OnDemand for z/OS solution.

7.1.3 Planning for implementation

Understanding the business benefits that could be achieved with Content Manager OnDemand for z/OS, the company positioned itself for success by the realization that in every project there would be entities that would help move the project forward and hindering forces that may oppose the project team or have a negative impact on the vision of the project. Replacing a solution means change, which can at times be a culture shock to many individuals.

In addition to the knowledge background that the company had available to them, they decided to also include the IBM Content Management Lab Services team in the implementation process. The reasoning behind this was quite simple: They wanted to minimize any implementation risk and take advantage of other IBM customers' implementation experience, thus ensuring a successful implementation.

A team was assembled that included individuals from the application areas that would see immediate benefits with the new solution: a Project Manager, IT representatives (including z/OS system administrators, z/OS system programmers, network administrators, storage administrators, and DB2 specialists), Content Manager OnDemand report administrators, and IBM Content Management Lab Services. An end-to-end detailed project plan was created addressing the following:

- Verify that the prerequisite software products are installed and operational.

  Enable the subsystems and interfaces required by Content Manager OnDemand for z/OS, including UNIX System Services and TCP/IP. Verify availability of proper levels of prerequisite z/OS server software. Customize host-based software as required by Content Manager OnDemand.

- Implement a dual store strategy of reports into current and new CMOS z/OS applications.

  The company required zero impact to its current content management system, which forced a dual store of all current reports approach. This approach continued throughout the conversion and implementation cycle.

- Implement a client interface strategy to support the fund accounting report approval process.

  Facilitate the workflow approval process development using the Document Audit Facility (DAF).

- Incorporate Content Manager OnDemand for z/OS authentication and entitlements into the current security infrastructure.

  One of the design guidelines for the proposed Content Manager OnDemand for z/OS security mechanisms was the integration with the existing security infrastructure. Another design point was to leverage, where possible, existing authentication and authorization information stores. This would avoid the need to create, populate, and maintain a separate authorization/entitlement store just for Content Manager OnDemand for z/OS.

- Replace existing report bundling and distribution infrastructure.

  Design a new report delivery system to eliminate costly and unnecessary printing.

- Task identification.

  IBM and the company worked to identify every task and assigned a task owner to each task. This was essential to the success of the project. The assignments were announced during a project kick-off meeting that included the project team and all key stakeholders.

7.1.4 Implementation

Implementation of the Content Manager OnDemand for z/OS solution consists of system design, installation and configuration, application design, performance testing, training, and final deployment.

System design

Figure 7-1 on page 304 illustrates the Content Manager OnDemand system overview as implemented for the company. The system has the following setup and configuration:

- Content Manager OnDemand is configured across two CPUs.
- A DB2 shared environment and an OAMPLEX are implemented for data sharing.
- VIPA is used for user request routing.
- A coupler is used for system synchronization.
- Multiple instances of ARSLOAD are used for ingesting data into the system that is acquired through JES.
- ARSODF is used for package and deliver client data in a time critical, distributed, and highly demanding environment.
- A sysplex distributor and workload manager performs workload distribution and provides maximum throughput on a continuous basis.

Not shown in the diagram are additional ARSLOAD jobs that monitor several UNIX System Services HFS directories. These directories contain reports coming into the system that have been indexed with the Generic Indexer from the Open Systems side of the house and are to be ingested into Content Manager OnDemand for z/OS.

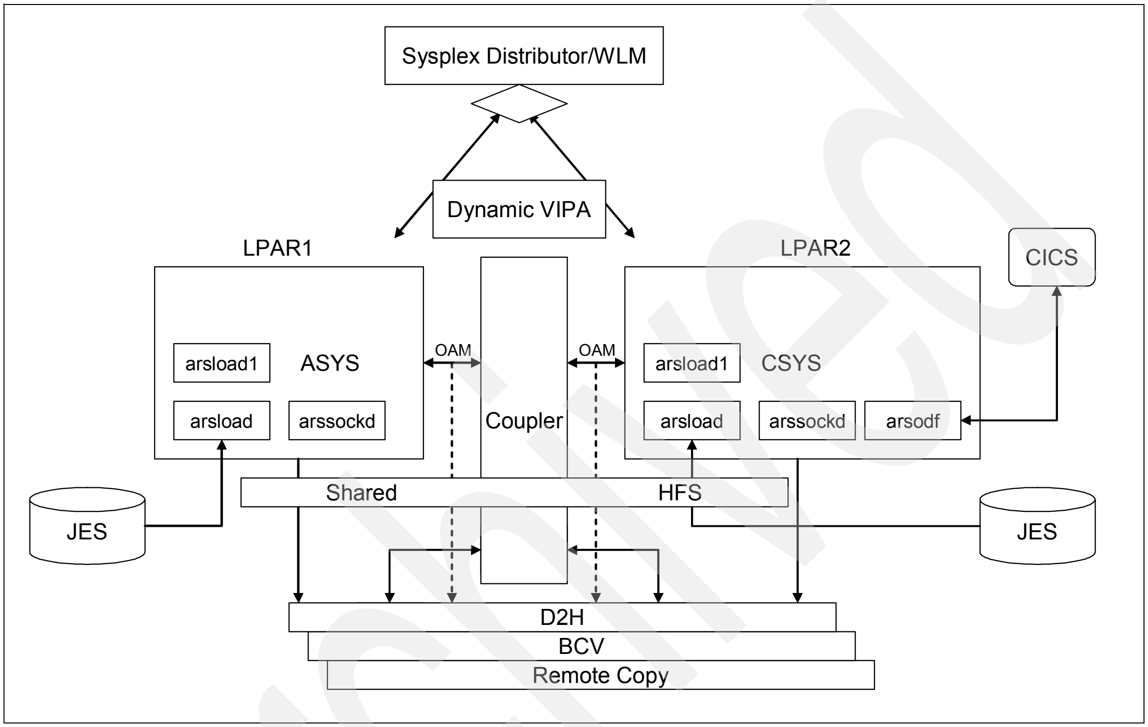


Figure 7-1 Content Manager OnDemand for z/OS case study 1 - System overview

Document retrieval architecture

Figure 7-2 on page 305 provides a more detailed view of the company's document retrieval architecture.

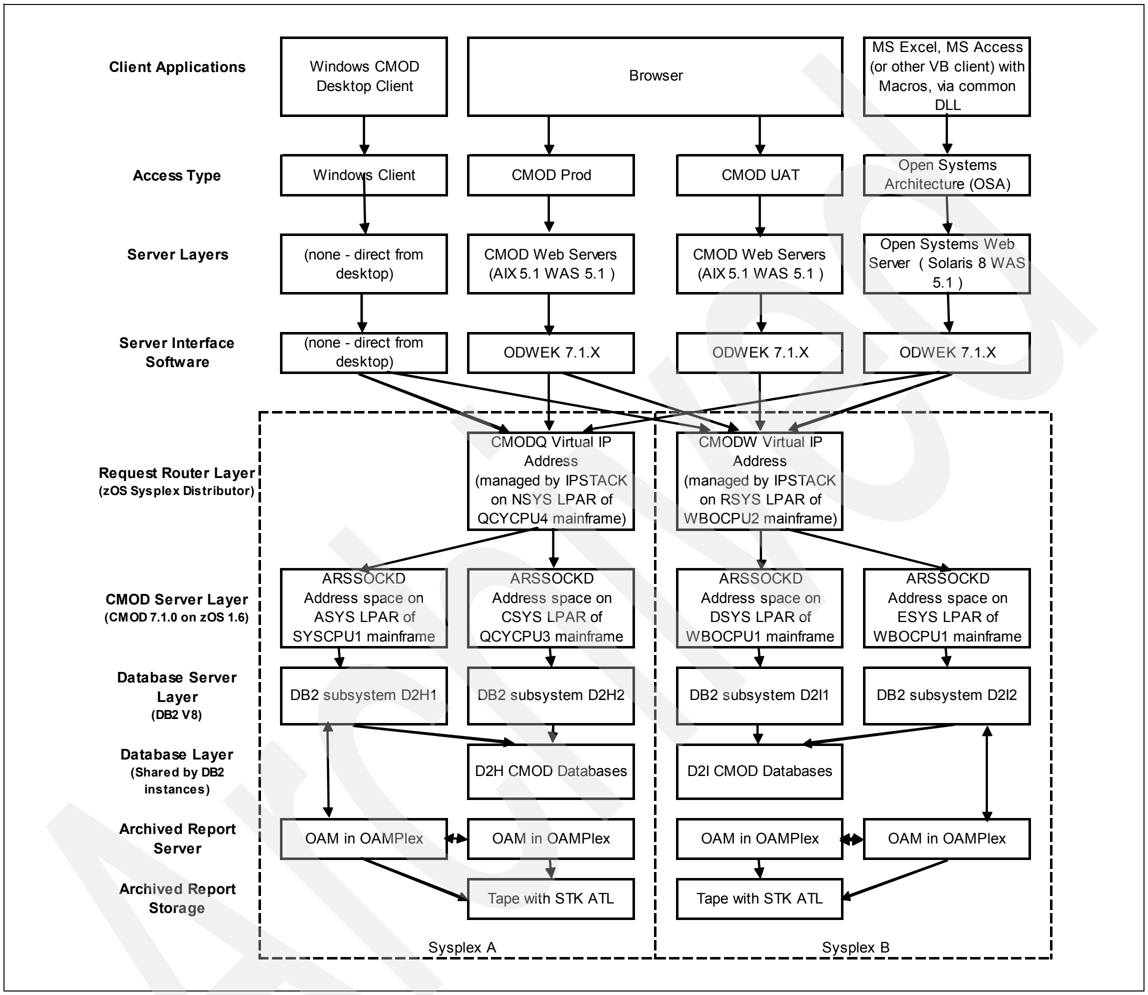

Client Applications
Windows CMOD Desktop Client
Browser
MS Excel, MS Access (or other VB client) with Macros, via common DLL
Access Type
Windows Client
CMOD Prod
CMOD UAT
Open Systems Architecture (OSA)
Server Layers
(none - direct from desktop)
CMOD Web Servers (AIX 5.1 WAS 5.1)
CMOD Web Servers (AIX 5.1 WAS 5.1)
Open Systems Web Server (Solaris 8 WAS 5.1)
Server Interface Software
(none - direct from desktop)
ODWEK 7.1.X
ODWEK 7.1.X
ODWEK 7.1.X
Request Router Layer
(zOS Sysplex Distributor)
CMODQ Virtual IP Address (managed by IPSTACK on NSYS LPAR of QCYCPU4 mainframe)
CMODW Virtual IP Address (managed by IPSTACK on RSYS LPAR of WBOCPU2 mainframe)
CMOD Server Layer
(CMOD 7.1.0 on zOS 1.6)
ARSSOCKD Address space on ASYS LPAR of SYSCPU1 mainframe
ARSSOCKD Address space on CSYS LPAR of QCYCPU3 mainframe
ARSSOCKD Address space on DSYS LPAR of WBOCPU1 mainframe
ARSSOCKD Address space on ESYS LPAR of WBOCPU1 mainframe
Database Server Layer
(DB2 V8)
DB2 subsystem D2H1
DB2 subsystem D2H2
DB2 subsystem D2I1
DB2 subsystem D2I2
Database Layer
(Shared by DB2 instances)
D2H CMOD Databases
D2I CMOD Databases
Archived Report Server
OAM in OAMPlex
OAM in OAMPlex
OAM in OAMPlex
OAM in OAMPlex
Archived Report Storage
Tape with STK ATL
Tape with STK ATL
Sysplex A
Sysplex B

Figure 7-2 Content Manager OnDemand for z/OS case study 1 - Top to bottom view at document retrieval architecture

The illustrated layers starting at the top are:

- Client applications: They provide the user interfaces. There are three client types. The Windows Client and the browser client allow users to retrieve and view documents stored within Content Manager OnDemand. The client based applications (for example Microsoft Excel and Microsoft Access) retrieve data from Content Manager OnDemand for further processing.

- The next three layers, labeled as access type, server layers, and server interface software: They are responsible for retrieving the documents from the Content Manager OnDemand server through TCP/IP and presenting them to the client applications layer.
- Request router layer: It encompasses both the WLM and the VIPA. It controls the distribution of the incoming TCP/IP requests and optimizes the distribution of the system workload. This functionality provides both automated workload balancing from the user perspective and automated failover.
- Content Manager OnDemand server layer: It shows four ARSSOCKD tasks (two per LPAR) that are the Content Manager OnDemand server components responsible for locating and retrieving the requested documents. The Content Manager OnDemand server will first authenticate the user, then check to see if the user is allowed access to the requested data. It will then query its DB2 tables in order to locate the data. Finally, it will retrieve the data from OAM and return it to the requester through TCP/IP.
- Database server and database layers: Together, they comprise the components of the shared DB2 environment. The DB2 tables provide the indexes that identify the location of the data.
- Archive report server and archive report storage layers: Together, they compose the OAMPLEX, which is responsible for the storage of the document data.

Installation and configuration

For the Content Manager OnDemand for z/OS implementation, the following components were installed and configured:

1. Content Manager OnDemand for z/OS base server code software
2. Content Manager OnDemand Windows Client
3. Content Manager OnDemand for z/OS Web Enablement Kit (ODWEK)
4. OnDemand Distribution Facility (ODF)

Content Manager OnDemand for z/OS base server software

The installation and configuration of Content Manager OnDemand for z/OS base server software includes completing the following steps:

1. Install server function using SMP/E.
2. Establish and modify appropriately a copy of the configuration files.
3. Configure the file structure.
4. Define the cache storage file system.
5. Configure system initialization.
6. Create and initialize the Content Manager OnDemand database.

7. Load sample files.
8. Perform installation verification test (IVP).

Content Manager OnDemand Windows Client

The Content Manager OnDemand Windows Client runs on the users' workstations and interfaces to the host Content Manager OnDemand for z/OS system. The Windows Client installation and configuration includes completing the following steps:

1. Verify availability of the proper levels of the prerequisite workstation software.
2. Install Content Manager OnDemand Windows Client, including one copy of the administrative interface (administrative interface is loaded through the custom option during the wizard installation).
3. Perform installation verification test (IVP).
4. Define report definitions in the Content Manager OnDemand for z/OS environment, which includes:
   - Define the entries for the report group, reports, and folders.
   - Define the indexing requirements.

Content Manager OnDemand for z/OS ODWEK

ODWEK enables search and retrieval access to the Content Manager OnDemand for z/OS archived documents through a Web interface. ODWEK installation and configuration includes completing the following steps:

1. Determine the security level to be used with ODWEK:
   - ODWEK programs and Web page access.
   - Data access to and from the OnDemand server.
2. Install and configure the ODWEK software to WebSphere Application Server:
   - Install the ODWEK software.
   - Customize the WebSphere Application Server.
   - Install the required plug-ins for viewing the data.
   - Customize the sample applications and perform functional testing.
   - Customize the default HTML template file.
3. Transformation implementation:
   - Install and configure the software.
   - Implement transformation of text to CSV.
   - Implement transformation of text to PDF.

Content Manager OnDemand for z/OS ODF

The ODF installation and configuration includes completing the following steps:

1. Install ODF function using SMP/E.
2. Define DB2 data structures.
3. Define CICS resources.
4. Bind DB2 packages and plans.
5. Load sample files.
6. Define ODF started task.
7. Create a bundle, distribution, and recipient.
8. Perform installation verification procedure (IVP).

Application design

The company supports tens of thousands of users and stores tens of terabytes of data in its Content Manager OnDemand for z/OS archives. High performance is achieved by fully understanding the users' business requirements and carefully designing the system to meet their business needs. The application design architecture was based on the illustration shown in Figure 7-3 on page 309.

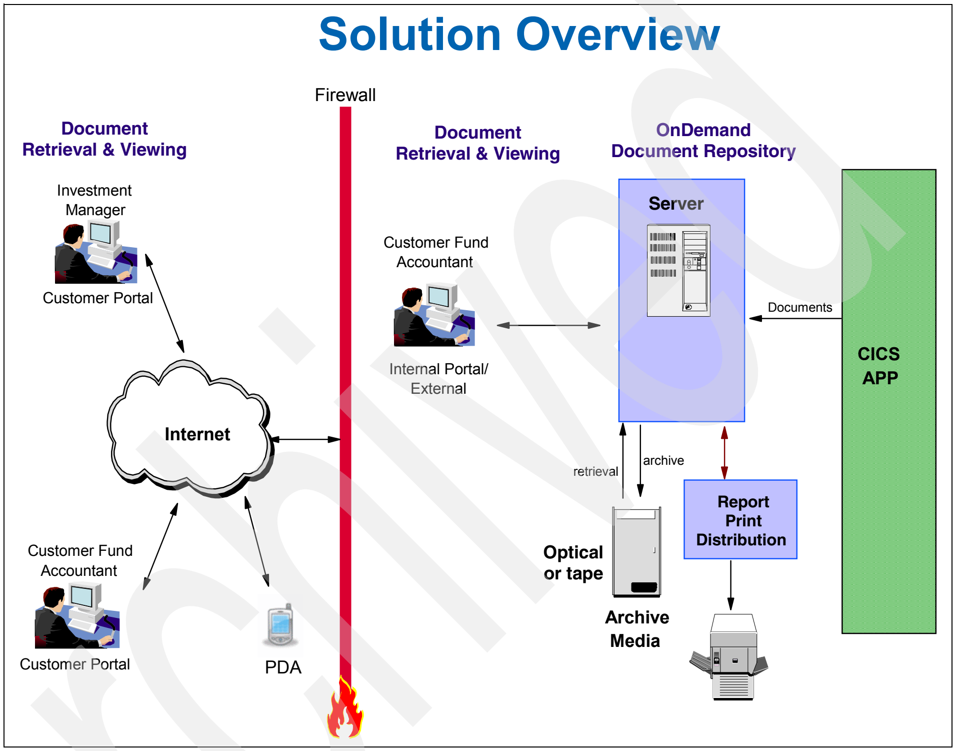


Figure 7-3 Content Manager OnDemand for z/OS case study 1 - Solution overview

To design the Content Manager OnDemand for z/OS application, the following information had to be gathered and reviewed:

- Number of reports
- Retention policies (data storage capacity)
- Report types (data conversion mechanisms)
- Capture mechanism (data source)
- Report ingestion requirements (data capture rate)
- Total number of users and client environment

- Concurrent number of users and requested response times (data retrieval rate)
- Network considerations (available and required bandwidth)

Analysis of the gathered information provided both the system requirements and the application design requirements.

The system requirements, which was the basis for the architecture previously described, includes an initial sizing for MIPS and DASD storage requirements. The MIPS sizing was considered initially as the system to be implemented was new, so although it is possible to predict the number of reports that will be stored in the system with a reasonable degree of accuracy, it is far more difficult to predict users' usage patterns of the new system. The existing retrieval patterns were based on a system that did not meet their needs, so it was expected that the new system would generate much higher volumes in terms of data retrieval requests, and it did. The system requirements also included DASD storage requirements.

The application design requirements resulted in the metadata definition, data storage definitions, and report delivery decision.

Metadata definition

The Content Manager OnDemand report structure was identified and implemented across two separate ingestion systems: Each of the ingestion systems serves a different customer base, so there is no sharing of data between them. It includes two sites:

- SITEA

Application	7691
Application Groups	502
Folders	664
Storage Groups	95

- SITEB

Application	3056
Application Groups	354
Folders	685
Storage Groups	59

Data storage

One of the unique features of the company's implementation was that not only were very large volumes of data being ingested 24/7 but also the retention period for the data was short (months). In fact, it was typical for multiple copies of a report to be stored on a daily basis when only the last stored copy was to be kept over the true retention period (months). So these extra copies that were stored during the day had to be programatically identified and deleted during the daily OSMC cycle. Thus, there was a need to support very large volumes of deletions as well as the very large volumes of ingestion.

Reports and documents were spread across multiple OAM storage groups. Typically, a single storage group is allocated for all reports that have the same retention requirements. The company decided that it would create several storage groups that were defined with the same retention. This was done strictly for availability and capacity planning purposes. A large segment of the stored reports have the same retention values. By creating multiple storage groups, it allowed the company to:

- Balance the report storage distribution so that no one storage group gets too big.
- Allow for multiple OSMC tasks to be executing in parallel.
- Avoid lock contention during report ingestion.

Report delivery

Report delivery was looked at as a new way of doing business going forward, rather then allowing users to continually receive the printed reports that they may or may not require to be printed any longer. The following points were used as justification presented to management in the Return On Investment (ROI) document:

- Elimination of the endless paper trail. Save a tree. Go paperless.
- I do not need the report printout anymore but never told anyone.
- The mailroom cannot find you.
- Do you know what you are getting? Nobody does.

Their approach to distribution was drastic. They turned all paper distributions off and created an automated request process that required management approval. Thus, users could request the reports to be delivered to them, and when management approval was granted, these reports would be delivered to them electronically as they were generated. This eliminated the report printing and distribution costs. It also eliminated all distribution delays and lost reports. The end result was a reduction of *85%* in distribution overhead. Within the first year, the company had a complete ROI.

Performance testing

The company identified Content Manager OnDemand for z/OS as a mission-critical application and made the needed investment in hardware to support their storage and retrieval needs. The system was implemented on a stand-alone z900 with 4,000 MIPS. Ingestion requirements had to meet or exceed storing 500 reports/minute at peak load time. The reports are viewed as three logical reports per physical report to Content Manager OnDemand for z/OS. So in essence, 1,500 Content Manager OnDemand reports were being ingested per minute. The report loading environment consists of:

- 16 ARSLOADs running on two LPARs (eight on each) on a 24/7 basis.
- The size of the reports vary from three pages to 100,000 pages.

The load (concurrent users and data throughput) on the system plus the different system and data architectures meant that it would be necessary to create a set of scripts to run against the two-tiered and three-tiered access environment. These tests would simulate peak work loads on the system from a load and retrieval perspective.

These tests were run continuously through the company’s development cycle. The company’s new Java code was being developed at the same time that the OnDemand code was being installed. This necessitated both tuning the Java application as well as tuning the Content Manager OnDemand for z/OS environment to maximize throughput for the business case scenarios. The performance tests were run daily over several months allowing code or environmental adjustments to occur between tests.

Initial tests revealed that the system would not meet the required amount of data to be ingested in the predicted production environment. Through code improvements and tuning efforts that occurred during the performance test period, response times were improved beyond the original goals and the system was now ready for production.

Training

The company identified three groups of individuals that were critical for training. These were the system personnel, report administrators, and general users.

- System personnel: Training involved the knowledge to perform everyday monitoring and on going product support. This training was delivered during the IBM Content Management Lab Services engagement.

- Report administrators: Require a more in-depth understanding of the Content Manager OnDemand for z/OS report analysis and indexing. They attended training provided by IBM Learning Services. The two courses attended were:
  - IM110 - introduction to IBM Content Manager OnDemand is designed for individuals responsible for creating and loading applications into the Content Manager OnDemand for z/OS system. It provides the basic understanding of all areas of Content Manager OnDemand for z/OS.
  - OD105 - IBM Content Manager OnDemand System Administration is specifically designed for individuals who are experienced at indexing and loading documents and have a need for a greater in-depth knowledge of the Content Manager OnDemand system, either for system administration, maintenance, or troubleshooting purposes.
- General user training: User training was developed in-house. The Content Manager OnDemand administrators provided Train the Trainer courses. The approach was to develop a reusable script for all application areas and external users that could be used to provide training at the enterprise level and can be used for future users of the system. External users were provide training using Webcast technology.

Deployment

After all the performance objectives were accomplished, the new Content Manager OnDemand for z/OS system was activated by switching from the test system to the production system by changing the TCP/IP port number in the VIPA. This allowed all new incoming requests to be routed to the Content Manager OnDemand for z/OS server. The change was transparent to the existing users. The phased in approach from the user perspective was used for the cut over to production. Since all reports were being stored in both the old archival repository and the new Content Manager OnDemand for z/OS repository, the company guaranteed no interruption of report access to its users community.

As users were trained in using the new client interfaces to Content Manager OnDemand, access to the old system was eliminated.

Today, the system is fully operational and exceeds all users expectations. The company is constantly adding new reports and documents to its electronic archive. All aspects of the production system are highly automated. The activity that requires the most work is the help that needs to be provided to new power users when they first start developing their own code to interface with the Java APIs.

7.2 Federal agency case study

In this case study, we describe a fictitious federal agency implementation. This agency is an existing OnDemand V2 customer and is migrating to OnDemand V7 to benefit from the new functional enhancements.

We will begin our discussion by profiling the customer and describing their business requirements. This will provide the reader with an understanding of the scope and complexity of the problem that is to be solved. Then we will describe how Content Manager OnDemand for z/OS was able to meet and exceed the customers business needs by utilizing innovative implementation practices. These practices enable the customer to minimize operational costs and complexity by delivering the right information to the right people in a timely manner.

7.2.1 Company background

We refer to the fictitious federal agency as the agency.

The agency is one of the largest Federal Agencies in the US government. It has offices in all of the 50 US states, as well as tens of thousands of employees and provides services to the US population as a whole.

As a service provider, the agency needs timely access to both current and historic information about all US citizens. The timeliness of the access is critical to server the public on an *On Demand* basis. The agency's clients want to be assured that their data is safely stored and readily available whenever they request it. At the same time, the agency's employees do not know who the next customer will be or what their requirements are. So, the accurate indexing and fast retrieval of the stored data are critical.

7.2.2 Business requirements

In order to improve its On Demand business operations and to position itself for future improvements, zCentirc Sam realized the need to migrate to Content Manager OnDemand for z/OS V7.

An analysis of the existing system and the current business requirements led the agency to consider the migration to Content Manager OnDemand for z/OS V7. A follow-up study on the costs and benefits of migrating to Content Manager OnDemand for z/OS V7 added clarity to the agency's vision of the need for and potential benefits of such a migration.

By selecting Content Manager OnDemand for z/OS V7 as their enterprise solution, the agency was able to address the following critical business requirements:

- Immediate and secure access to customer information 24x7.

  Eliminate customer complaints due to the unavailability of timely data. Implement a home grown security environment.

- Support variable workloads.

  There are 80,000 Content Manager OnDemand user IDs defined. The number of users accessing data varies throughout the day. With different time zones, they start and end work at different times.

  Most of the batch load jobs would run nightly, with a few jobs run during the day as needed.

- Re-position for future growth.

  Allow for easy integration of new hardware and storage devices. Allow for easy software upgrades and federation with other Content Manager products. As system usage grows, allow for future increases in the number of concurrent servers.

- User productivity gains.

  Provide 24x7 data availability. Provide users with a user friendly, Web-based customer interface. Provide users with the right data On Demand.

- Utilize the current infrastructure and skill base.

  Utilize the existing hardware and network infrastructure. Utilize the existing systems and programming skills.

- Eliminate Content Manager OnDemand V2 constraints.

  Eliminate dependence on TSQ space availability and single server dependency.

- Simplify system operations

  Create a redundant server environment. Eliminate client software distributions. Automate system operations.

7.2.3 Planning for implementation

The migration to Content Manager OnDemand for z/OS V7 was to be handled by the same personnel that handled the Content Manager OnDemand V2 system. This enabled the agency to take advantage of their current knowledge and experience covering Content Manager OnDemand V2, archiving requirements, user requirements, and system knowledge (including hardware and network). In addition, the IBM Content Management Lab Services team is also involved in the implementation process to leverage their existing migration expertise and ensure a successful implementation.

A team was assembled that included representatives from each of the stakeholders, These included Content Manager OnDemand administrators, IT representatives (including z/OS system administrators, z/OS system programmers, network administrators, storage administrators, and DB2 specialists), a project manager, and IBM Content Management Lab Services.

An end-to-end detailed project plan was created that served to:

- Identify the proposed architecture, its components, and its ability to meet the business requirements.

  A Web-based system was selected as the most suitable architectural implementation for the agency's needs. This would minimize software distributions and any dependencies on client PC's hardware or software.

  WebSphere Application Server was selected. WebSphere Application Server was already functional and producing good results within the existing operational environment. All the required operational and diagnostic skills were available in-house, further ensuring a successful implementation.

- Verify that the prerequisite software products are installed and operational.

  Verify the availability of the proper levels of the prerequisite z/OS server software.

  Enable all the subsystems and interfaces required by Content Manager OnDemand for z/OS. The WebSphere Application Server, UNIX System Services, DB2, OAM, and TCP/IP subsystems were already operational and in use by Content Manager OnDemand V2. TCP/IP would require further customizing to support the increased network throughput abilities of Content Manager OnDemand V7.

  Customize host-based software as required by Content Manager OnDemand for z/OS.

- Use a dual store strategy of reports for the current and new Content Manager OnDemand for z/OS application.

  The agency required zero impact to its current content management system. So during the Content Manager OnDemand V7 installation and testing period, a strategy was implemented that would allow access from both Content Manager OnDemand V2 and Content Manager OnDemand V7. This entailed new reports to be stored to both systems, and the existing report indexes to be migrated to the new system.

- Client interface development strategy.

  The agency developed a new Web-based client that included an enhanced user interface with links to the agency's other systems. Sample Java API code provided by IBM was used as a base, because the code already included all the techniques and methods required to connect to the back-end server. The agency developed the enhanced user interface to allow for customized access to the archived data.

- Security infrastructure.

  As in most On Demand computing environments, access security and data security are extremely important. The agency used the internal security mechanism provided by Content Manager OnDemand for z/OS and overlaid it with a proprietary in-house developed security system that they were creating. Exits provided by Content Manager OnDemand for z/OS and good programming techniques implemented by the agency thus ensured secure and timely access to both the system and the data.

- Task identification

  IBM and the agency worked to identify every task and assign a task owner. This was essential to the success of the project. The assignments were announced during a project kick-off meeting with the project team and the key stakeholders.

7.2.4 Implementation

Implementation of the Content Manager OnDemand for z/OS solution consists of system design, installation and configuration, application design, performance testing, training, and final deployment.

System design

Figure 7-4 illustrates the implemented system architecture.

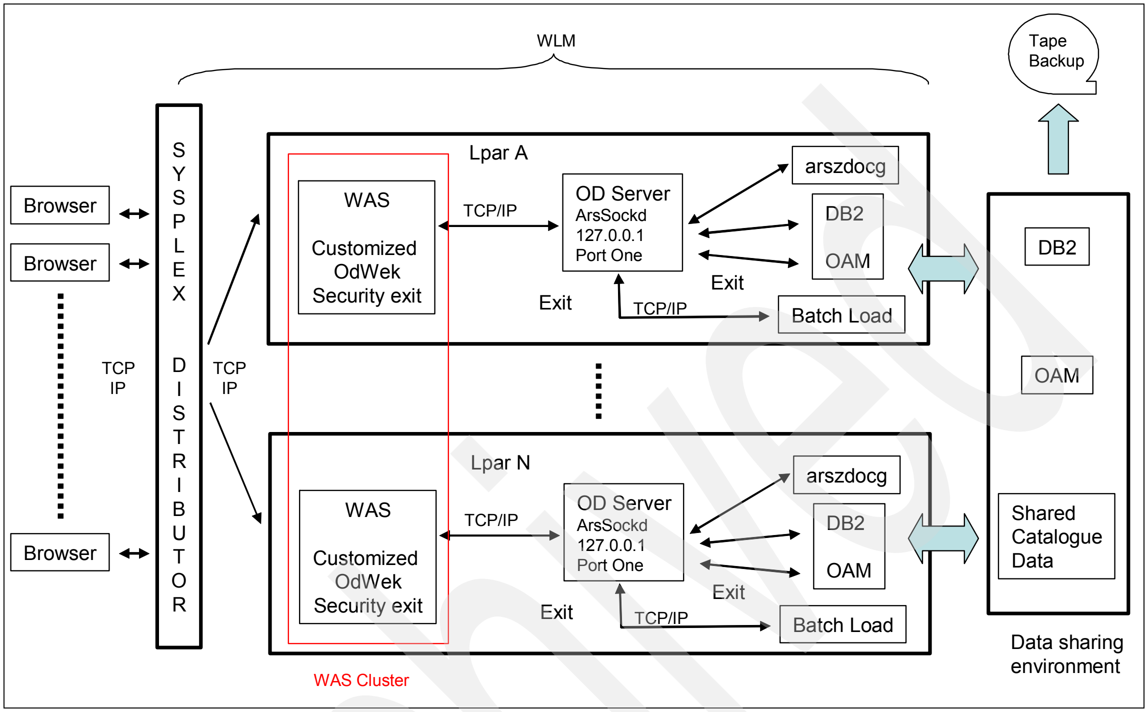


Figure 7-4 Content Manager OnDemand for z/OS case study 2 - System architecture

The system components include:

- A Sysplex distributor for distributing incoming TCP/IP requests to one or multiple LPARs.
- Each LPAR contains a complete Content Manager OnDemand for z/OS WebSphere implementation. This includes WebSphere Application Server, the agency's customized Java code, Java APIs, customized security code, the Content Manager OnDemand server (ARSSOCKD), and V2 compatibility code (ARSZDOCG).
- Each of the LPARs connects to DB2 and OAM. Both DB2 and OAM are installed in a data sharing environment and act as the data archive backbone of the system. The data sharing environment allows Content Manager OnDemand index and object data stored from all of the LPARs to be placed in a single DB2 and OAM datastore. Thus, any single LPAR is independently capable of accessing all the OnDemand data and functioning independently of any other LPAR.

- Multiple LPARs are used to provide good response times to incoming user requests and system redundancy. If one LPAR fails, any of the other LPARs can pick up the workload transparent to the users. Each of the LPARs are on separate z/OS systems.
- From the browser perspective, the entire complex appears as a single IP address and a single port.

Report data retrieval

From an operational perspective, the communication flow is as follows:

- Multiple browsers access the system from either the internet or the intranet. Browsers coming in from the internet would pass through a firewall first (not shown for simplicity purposes).
- The browsers connect to the sysplex distributor that, based on system workload and session affinity, will route the request to the appropriate WebSphere Application Server.
- At this point, the agency's custom Java code performs the appropriate security checking and then calls the OnDemand Java APIs that will communicate with the back-end OnDemand server to retrieve the stored the document.
- When the document request reaches the OnDemand server, it will determine whether the data to be retrieved is newly archived data (stored in the V7 Content Manager OnDemand archive) or whether the request is for older data stored in the V2 Content Manager OnDemand archive. Based on the determination, different code paths will be followed and the requested document data will be retrieved from the appropriate archive.
- The document data is returned through WebSphere Application Server to the browser.

From the average user perspective the complete process takes a couple of seconds.

Report data loading

The implemented architecture allows for the batch load jobs to be run on any system. Since the data to be loaded comes from an SMS shared catalogue environment, it is accessible from any system. There are no performance penalties regardless of the system from which the data comes from and on which system/LPAR the load runs.

There are multiple identical copies of ars.cfg, one defined to each system. So regardless of the system on which the batch load runs, it will always point to the local host 127.0.0.1 and the same instance name, same port one (reserved in TCP profile). Thus, the load will always be local (to the ARSSOCKD on which the load job is run) through the local TCPIP stack, ensuring optimum TCP/IP throughput.

Operationally, approximately 50 reports each containing thousands of documents are loaded per night by means of up to 22 parallel load jobs. The load jobs are automatically distributed by the workload manager between multiple LPARs.

Installation and configuration

The agency's system required that the following Content Manager OnDemand components to be installed and configured:

- Content Manager OnDemand for z/OS base server code.
- Content Manager OnDemand Windows Client (Windows Client), distributed on a limited scale for preliminary testing.
- Content Manager OnDemand Administration Client (Admin Client), distributed on a limited scale to administrators of the Content Manager OnDemand for z/OS the system.
- Content Manager OnDemand for z/OS Web Enablement Kit (ODWEK). This includes the Java APIs that are needed to build customized Web client interfaces.

The following general steps were followed in installing and implementing the Content Manager OnDemand for z/OS System. For details, refer to both the *IBM Content Manager OnDemand for z/OS: Introduction and Planning Guide*, GC27–1438 and the *IBM Content Manager OnDemand for z/OS: Configuration Guide,* GC27–1373.

Content Manager OnDemand for z/OS base server software

The installation and configuration of Content Manager OnDemand for z/OS base server software includes completing the following steps:

1. Install the server function using SMP/E.
2. Establish and modify appropriately a copy of the configuration files.
3. Create any necessary temp directories.
4. Data is being archived directly to OAM, so there was no cache storage definition needed for the storage of cache data. Only a single cache storage file system needs to be defined for the storage of Content Manager OnDemand for z/OS internal system data.
5. Initialize the Content Manager OnDemand system.
6. Create and initialize the database.
7. Load sample files.

8. Perform the installation verification test (IVP).

Content Manager OnDemand Windows Client and Admin Client

The Content Manager OnDemand Windows Client software runs on the users' workstations and interfaces to the host Content Manager OnDemand for z/OS system through a TCP/IP connection. The agency installed only a handful of clients within the Content Manager OnDemand operations group. These included both the Windows Client and the Administration Clients. These clients are to be used for system administration, initial validation of report definitions, and preliminary testing. Installation includes completing the following steps:

1. Verify availability of the proper levels of prerequisite Workstation Software.
2. Install Content Manager OnDemand Windows Client, including a copy of the administrative client interface. The administrative client interface is installed through the custom option during the wizard installation process.
3. Perform the installation verification test (IVP).

Content Manager OnDemand for z/OS ODWEK

ODWEK enables search and retrieval access to the Content Manager OnDemand for z/OS archived documents through a Web interface. ODWEK includes a simplified, fully functional out-of-the box Web implementation that could be used as a Web interface or as a test client to verify the correct installation of the Web environment. ODWEK also includes the Java APIs upon which the agency built their customized Web interface. The tasks required to build the customized interface included:

1. Determine security level to be used with ODWEK:
   - ODWEK programs and Web page access.
   - Data access to and from the OnDemand server.
2. Install and configure the ODWEK software to WebSphere Application Server:
   - Install the ODWEK software.
   - Customize the WebSphere Application Server.
   - Install the required plug-ins for viewing the data.
   - Customize the sample applications and perform functional testing.
   - Customize the default HTML template file.
3. Install and configure AFP to PDF transform software.

Application design

The agency supports tens of thousands of users and stores tens of terabytes of data in its Content Manager OnDemand archives. High performance was achieved by fully understanding the users' data needs and carefully designing the system so as to meet these needs. The main issues that were addressed during the application design were:

- Indexes: An analysis of the report data revealed that there was a single index that could be assigned as the main index. This was defined as a required index field. Two other index fields were also identified as being of use but not required, so they were also identified as index fields. The time spent in studying the reports and determining this smaller number of indexes enabled faster loading of the report data and the minimization of the DASD space required to store the index data. The indexes themselves are in the terabyte range. So, the agency was successful in defining the correct number of index fields that met their business requirements.
- Date field: Analysis showed that users typically search the last two months of stored reports. So, the load date field was defined as a segment field to enable Content Manager OnDemand to search only the appropriate application group data tables when a search request was submitted. To make the process easier on the users, default date values were provided on the folder window with the date search range being set to **Current Date** and **Current date - 60 days**.
- Application group data table sizes: The table sizes were selected such that the number of rows equalled (approximately) the amount of data indexes stored in a single month. This allowed for index searches to be limited to two or three tables (in most cases) and also allowed for easier maintenance and migration of the data tables. On a monthly basis when a table is closed (that is, no more indexes will be stored to that table), then a final **`runstat`** is run against the table and DB2 access to that table is optimized without the need for any further **`runstats`** (on that table).

Application, application group, and folder definitions

In general, the application, application group, and folder definitions followed the pattern illustrated in Figure 7-5 on page 323. There are on average 50 applications per application group and three defined folders for each application group.

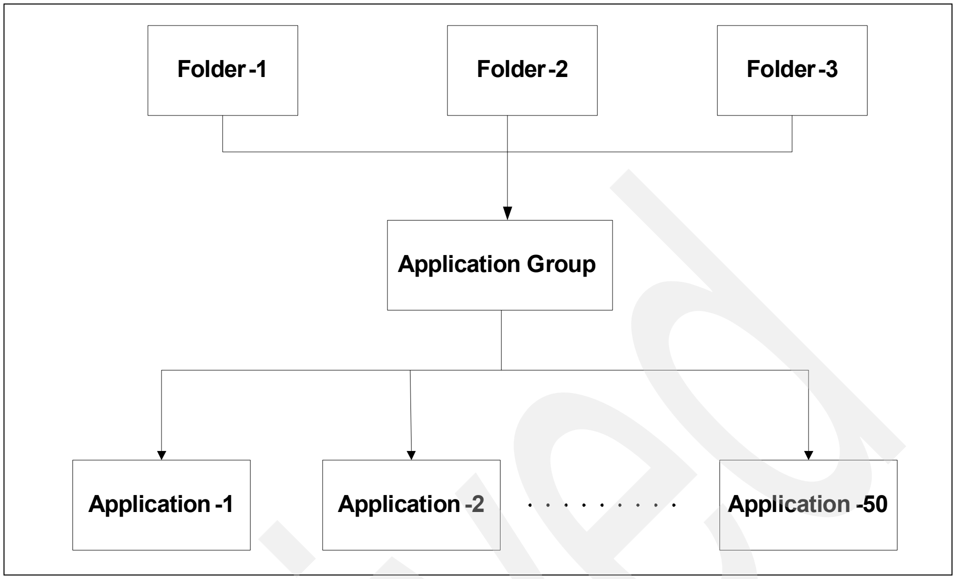


Figure 7-5 General report assignment pattern

The design points were as follows:

- The application groups were defined based on their functionality and usage within the organization. For example, all reports (documents) that were to be accessed by the accounting department would be placed in an accounting application group. Analysis also revealed that all the accounting department documents share the same keys, regardless of the data type or source.
- The index keys were defined generically, as described in Chapter 4, “IBM Content Manager OnDemand for z/OS implementation guidelines” on page 121. This allowed for multiple report types with different number of keys or different key types to be stored in the same application group. This technique made it easier (for example) to store all accounting related reports in the accounting application group.
- Collecting the 50 applications in a single application group allowed for:
  - A single logical search for all the documents from those 50 applications stored in the application group.
  - The application ID field was used to differentiate between the different applications in the application group.
  - If new applications were added in the future, they could simply be added to an existing application group that met the business requirements.

- The three folders were defined for different purposes:
  - The first folder was for the internet users. This was defined so that the users could only enter a specific key and the date range was fixed. This narrowed internet users searches to a specific subset of data, essentially the data that they were allowed access to. This also reduced the possibility of internet users submitting complex queries that would consume more CPU time than needed.
  - The second folder was for internal users. This folder was defined such that aside from the required key field, two other key fields could be selected. Also, the date range could be changed. This allowed internal users to perform more extensive tailored searches of the data.
  - The third folder is to be used by auditors and administrators. This folder was defined to include all the features of the second folder as well as the ability to do table scans (search for non index fields).
- The Max rows value for the application group data tables was set to 100,000,000. This corresponded to the number of documents loaded per month. This size helped in limiting the number of DB2 queries during a search request and with the DB2 table maintenance. At the end of each month when the application group data table was closed, a final **`runstat`** is performed after which no more **`runstats`** are needed for that table. Also, at migration or expiration time, it was more practical from a business perspective to migrate or expire data by month.

The application group data table expiration type was set to LOAD. This meant that at expiration time, when all the documents within the table had expired, expiration could be carried out efficiently by simply dropping the table.

Backup

The DB2 and OAM backups are run daily. Backups are performed to tape and the data is stored both locally and offsite. All databases (DB2 and OAM) are backed up during the nightly cycle. Only the active tables need to be backed up on a regular basis. Closed tables are backed up on a monthly basis.

In OAM, the following parameter is set:

```
set auto backup = yes
```

This ensured that backups of the objects were also written to a backup storage media.

Performance testing

The agency identified Content Manager OnDemand for z/OS as a mission critical application, since it is being used for service delivery to its customers on a daily basis. Originally, with V2 of Content Manager OnDemand, the agency had personnel, hardware, and network dedicated to the Content Manager OnDemand application. With the migration to Content Manager OnDemand V7, additional networking and hardware capacity was added. The same number of personnel and work schedule was maintained. The additional hardware and network capacity were deemed necessary and beneficial due to the following facts:

- V7 now supported a larger concurrent user base, with better user response times.
- V7 had additional reports defined to it.
- More data is being stored in V7 on a daily basis, so more load jobs are being run in parallel.
- V7's multiple LPAR configuration provides scalability and redundancy that was not available in V2.

The additional load (users and data) on the system plus the different system and data architectures meant that it would not be possible to extrapolate performance numbers from V2 previous performance data, so the agency devised two sets of tests that would to a large extent emulate their production environment:

- A set of script-driven, browser-based tests that it could run to simulate the new expected retrieval load on the system. These tests were run nightly through the agency's development cycle. As the agency's new Java code was being developed at the same time that the Content Manager OnDemand code was being installed, this necessitated both tuning the Java application as well as tuning the Content Manager OnDemand environment to maximize throughput for the agency's scenario.
- A set of load tests that simulated loading a variety of report size with different number of indexes and document sizes. These tests were run nightly through the agency's development cycle. Initially, the tests were run on alternate nights from the retrieval tests and towards the end of the tests period were run in conjunction with the retrieval tests so as to load the system beyond what was expected.

The performance tests were run nightly over several months, with code or environmental adjustments being made between tests.

Initial tests revealed that the system would be unusable in production. The code improvements and tuning efforts (based on the best practices document) that occurred during the performance test period improved response times to the desired goal and the system was now ready for production.

Training

The agency's training requirements were minimal. There are generally three types of skills that are needed:

- System skills: The many years of experience that the agency had with V2 OnDemand provided them with most of the system skills needed. The additional system skills needed for V7 versus their V2 implementation, mainly WebSphere Application Server, UNIX System Services, and TCP/IP, had already been acquired from other WebSphere Application Server z/OS implementations. The main knowledge that they lacked was an understanding of the internals of V7 and its finer tuning points for their implementation. They were able to acquire this knowledge from the IBM Content Management Lab Services group during the joint installation and testing phases of the project.
- Report administration skills: Even though the agency's report administrators were highly skilled V2 Content Manager OnDemand administrators, because V2 of Content Manager OnDemand is different from V7 of Content Manager OnDemand, they decided that there would be significant payoff to acquiring a full understanding of V7 administration features and capabilities. These skills were acquired by first attending training provided by IBM Learning Services. Further implementation specific skills were then transferred from IBM Content Management Lab Services both in a classroom environment and hands on throughout the project life cycle. The classes attended were:
  - IM110 - Introduction to IBM Content Manager OnDemand is designed for individuals responsible for creating and loading applications into the Content Manager OnDemand for z/OS system. It provides the basic understanding of all areas of the Content Manager OnDemand for z/OS.
  - OD105 - IBM Content Manager OnDemand System Administration is specifically designed for individuals who are experienced at indexing and loading documents and have a need for a greater in-depth knowledge of the OnDemand system, either for system administration, maintenance, or troubleshooting purposes.
- User skills: User training was developed in-house. The Content Manager OnDemand administrators provided Train the Trainer courses. The trained trainers then trained the internal user community. A help desk was maintained to answer any usage questions from users or trainers. External users were provide training using Web help windows. The help desk was also the focal point for any Content Manager OnDemand problems encountered by the users. It thus became the focal point for keeping track of the users "comfort level" with the new system.

Deployment

The agency had done everything right, so the deployment should have gone off without any problems. That did not happen.

Users started logging in to the system on the first day of deployment, and they were so happy with the new system that their usage exceeded that which had been predicted and tuned for during the performance testing stage. Their excess usage caused the LPARs to intermittently (five times per day) run out of memory and crash, which was not a good thing. But because of the built-in redundancy (the multiple LPAR configuration), user requests would automatically be routed to another LPAR and would continue with their work. The users were thus unaware that one of the servers had crashed. Also, the automation tools picked up the fact that the server crashed and automatically re-started the server. This process took less than a minute.

So the good news was that the redundant server worked. The bad news was that some more tuning needed to be done. Over the next two weeks, the problem was tracked down and resolved. It was simply a matter of increasing the WebSphere Application Server capacity by increasing the number of servant regions within the WebSphere Application Server.

7.3 Educational institution case study

In this case study, we describe a fictitious educational institution implementation. This institution is an existing Content Manager OnDemand V2 customer and is migrating to Content Manager OnDemand V7 in order to benefit from the new functional enhancements.

We will begin our discussion by profiling the customer and describing their business requirements. This will provide the reader with an understanding of the scope and complexity of the problem that is to be solved. Then we will describe how Content Manager OnDemand for z/OS was able to meet and exceed the customers business needs by utilizing innovative implementation practices, thus allowing them to minimize operational costs and complexity by delivering the right information to the right people in a timely manner.

7.3.1 Company background

We refer to our fictitious educational institution as the institution.

The institution is a medium sized public educational institution. It is composed of a main campus with multiple satellite campuses statewide. There are over 20,000 students that attend the institution’s daytime program.

The institution had been using Content Manager OnDemand V2 to store administrative type reports for several years. The institution decided to expand its use of OnDemand to store both administrative reports and faculty data. The administrative reports consist of financial records and student records. The faculty data is varied and consists of published articles, lab experiment datasets, and results, as well as student assignments. Both the timeliness of the access and the security of the data are considered critical in order to serve the institution's goals. The institution's faculty and staff want to be assured that their data is safely stored and readily available whenever they request it. So, the confidentiality, accurate indexing and fast retrieval of the stored data are critical.

7.3.2 Business requirements

In order to improve its OnDemand business operations, to position itself for future improvements, and to implement additional functionality only available in Content Manager OnDemand for z/OS V7, zCentirc Edu decided that it would migrate to Content Manager OnDemand for z/OS V7.

By selecting Content Manager OnDemand for z/OS as their enterprise solution, the institution was able to address the following critical business requirements:

- Immediate access to administrative and faculty data 24x7.

  While administrative documents are normally accessed Monday through Friday during regular working hours with few exceptions, access to faculty data is on a more sporadic, unregulated basis, with data being accessed outside of normal working hours on a regular basis.

- Support variable workloads.

  While the number of users is relatively small (thousands of users) and part of the workload presented to the system is variable, the administrative workload is reasonably predictable on a daily basis with end of month and end of year peaks. The faculty workload is total unpredictable and is largely dependant on the faculty needs of the moment.

  Most of the administrative batch load jobs are run at night, with a few jobs run during the day as needed. On the other hand, the faculty and academic workloads run on an unpredictable schedule unrelated to any daily routine.

- Re-position for future growth.

  Allow for easy integration of new hardware and storage devices. Allow for easy software upgrades and federation with other Content Manager product. Allow for the storage and retrieval of a wide array of data types.

- User productivity gains.

  Provide 24x7 data availability. Provide administrative users with a user friendly Windows Client interface. Provide faculty users with a user friendly Web-based interface. Provide power users with Java APIs that allow them to construct their own Web-based interfaces.

- Utilize current infrastructure and skill base.

  Utilize existing hardware and network facilities. Utilize existing system and programming skills.

- Eliminate Content Manager OnDemand V2 constraints.

  Eliminate dependence on TSQ space availability

- Simplify system operations.

  Create an auto restart on failure capability.

7.3.3 Planning for implementation

The migration to Content Manager OnDemand for z/OS was to be handled by the same personnel that handled the Content Manager OnDemand V2 system. This enabled the institution to take advantage of their current knowledge and experience covering Content Manager OnDemand V2 archiving requirements, user requirements, and system knowledge (hardware and network).

In addition to the extensive Content Manager OnDemand V2 and system knowledge that the institution had available to them, they decided to also include the IBM Content Management Lab Services team in the implementation process.

A team was assembled that included representatives from each of the stakeholders. These included Content Manager OnDemand administrators, IT representatives (including z/OS system administrators, z/OS system programmers, network administrators, storage administrators, and DB2 specialists), a project manager, and IBM Content Management Lab Services.

An end-to-end detailed project plan was created addressing the following items:

- Identify the proposed architecture, its components, and ability to meet the business requirements.

  A Windows Client was selected as the most suitable architectural implementation for the institution’s administrative staff needs. A Web-based interface was selected as the most suitable architectural implementation for the institution’s faculty needs. Additionally, Java APIs were provided for faculty (power users) that wanted to build their own interfaces.

WebSphere Application Server was selected since it was already installed and operational for other applications. It was producing good results within the existing operational environment. The availability of the required WebSphere Application Server operational and diagnostic skills would further ensure a successful implementation.

- Verify prerequisite software products are installed and operational.

  Enable all the subsystems and interfaces required by Content Manager OnDemand for z/OS. WebSphere Application Server, UNIX System Services. DB2, OAM, and TCP/IP were already in use. Verify the availability of the proper levels of the prerequisite z/OS server software.

  Customize host-based software as required by Content Manager OnDemand.

  Some tuning of the host-based components would be required to support the increased workload to be implemented in CMDO z/OS.

- Use a dual store strategy of reports for the current and new Content Manager OnDemand for z/OS application.

  The institution had a relatively small data archive (hundreds of gigabytes), so the decision was made to transfer all the data from the V2 archive to the V7 archive.

  During the migration period, all newly loaded data would go into both the V2 and V7 archive. When the migration process completes, all report data would be stored in the V7 archive. The migration period was short. Jobs were run on a 24x7 basis and all the reports were migrated over a two week period. Migration was scheduled during the summer vacation so as to minimize any potential impact on the administrative staff. There was only administrative data to migrate, as the old system did not provide support for faculty data.

- Client interface development strategy.

  The institution used the Windows Client as the interface for administrative users. This is a fully functional out of the box client. A customized Web interface was developed that would allow faculty to store and retrieve their own data.

  Sample Java API code provided by IBM was used as a base, since this already included all the techniques and methods required to connect to the back-end server. These Java APIs are also available to be used for interface development by an interested faculty.

- Security infrastructure.

  As in most On Demand computing environments, access security and data security are extremely important. The institution used the internal security mechanism provided by Content Manager OnDemand z/OS.

Access to the system is firewall protected and only allowed through the institution's Virtual Private Network (VPN).

- Task identification

  IBM and the institution worked to identify every task and assign a task owner. This was essential to the success of the project.

  Assignments were announced during a project kick-off meeting with the project team and the key stakeholders.

7.3.4 Implementation

Implementation of the Content Manager OnDemand for z/OS solution consists of system design, installation and configuration, application design, performance testing, training, and final deployment.

System design

Figure 7-6 on page 332 illustrates the retrieval architecture of the implemented system retrieval architecture. The architecture consists of the following design:

- A two tier-system access is implemented for the administrative staff using the Windows Client and for the system administrators using the Windows Administration Client.
- A three-tier system access is implemented for faculty (using the institution's enhanced browser) and is also provided through the Java APIs for power users supporting their own implementations.
- All communications between the clients and the Content Manager OnDemand server is through a TCP/IP connection.

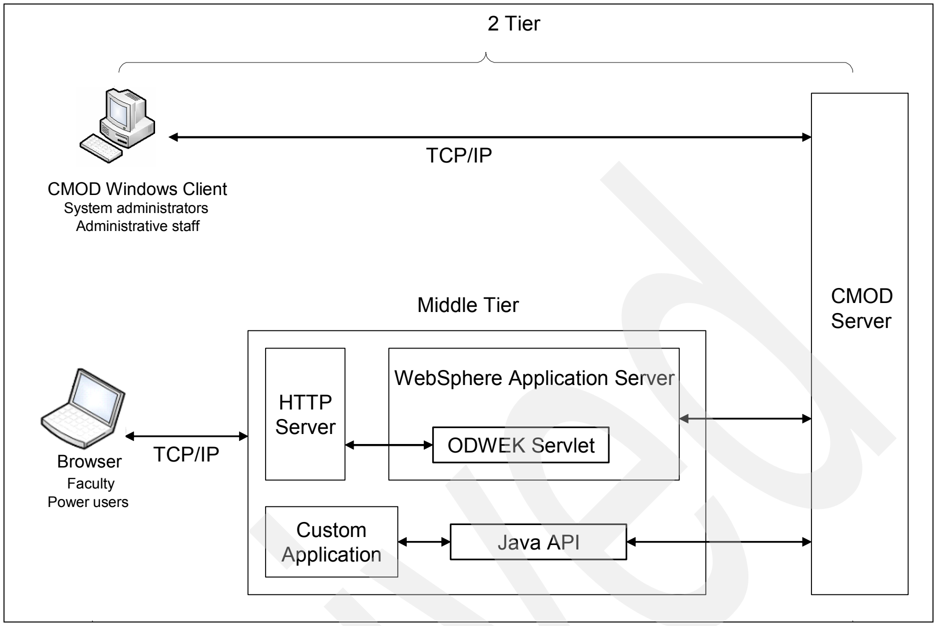


Figure 7-6 Retrieval architecture

Installation and configuration

The institution requirements dictated that the following Content Manager OnDemand components be installed and configured:

- Content Manager OnDemand for z/OS base server code.
- Content Manager OnDemand Windows Client (Windows Client), for university staff and faculty.
- Content Manager OnDemand Administration Client (Admin Client), for administrating the system.
- Content Manager OnDemand for z/OS Web Enablement Kit (ODWEK). This includes the Java APIs that are needed to build customized Web client interfaces.

The following general steps were followed in installing and implementing the Content Manager OnDemand for z/OS system. For details, refer to both the *IBM Content Manager OnDemand for z/OS: Introduction and Planning Guide*, GC27–1438 and the *IBM Content Manager OnDemand for z/OS: Configuration Guide,* GC27–1373.

Content Manager OnDemand for z/OS base server software

The installation and configuration of Content Manager OnDemand for z/OS base server software includes completing the following steps:

1. Install server function using SMP/E.
2. Establish and modify appropriately a copy of the configuration files.
3. Create any necessary temp directories.
4. Data is being archived directly to OAM, so there was no cache storage definition needed for the storage of cache data. Only a single cache storage file system needs to be defined for the storage of Content Manager OnDemand for z/OS internal system data.
5. Initialize the Content Manager OnDemand system.
6. Create and initialize the database.
7. Load sample files.
8. Perform the installation verification test (IVP).

Content Manager OnDemand Windows Client and Admin Client

The Content Manager OnDemand Windows Client software runs on the users' workstations and interfaces to the host Content Manager OnDemand for z/OS system. The institution was already using the Windows Client with V2 of Content Manager OnDemand, so procedures were already in place for distribution and installation of a new client release. Additionally, this installation could be done incrementally, since the old version of the client software continued to function with the new Content Manager OnDemand server. Regardless, the institution decided to install the new client right away so that they could benefit from the additional functionality. Installation includes completing the following steps:

1. Verify availability of the proper levels of prerequisite Workstation Software.
2. Install Content Manager OnDemand Windows Client, including a copy of the administrative client interface. The administrative client interface is installed through the Custom option during the wizard installation process.
3. Perform the installation verification test (IVP).

Content Manager OnDemand for z/OS ODWEK

ODWEK enables search and retrieval access to the Content Manager OnDemand for z/OS archived documents through a Web interface. ODWEK includes a simplified fully functional out-of-the box Web implementation that could be used as a Web interface or as a test client to verify the correct installation of the Web environment. ODWEK also includes the Java APIs upon which the institution built their customized Web interface. The tasks required to build the customized interface included:

1. Determine security level to be used with ODWEK:
   - ODWEK programs and Web page access.
   - Data access to and from the OnDemand server.
2. Install and configure the ODWEK software to WebSphere Application Server:
   - Install the ODWEK software.
   - Customize the WebSphere Application Server.
   - Install the required plug-ins for viewing the data.
   - Customize the sample applications and perform functional testing.
   - Customize the default HTML template file.
3. Install and configure AFP to PDF transform software.

Application design

The institution implemented three different applications:

- The first application was a follow on to the previous work from Content Manager OnDemand V2.

  The application supported the institution’s administrative staff. It provided archive storage and online access to financial, personnel, and student records. It was implemented using the Windows Client.

- The second application was new and was in support of the faculty.

  The application allowed for the storage, retrieval, and viewing of predefined data types. These data types include line data, AFP data, Excel spreadsheet, and Word documents.

  In addition, it allowed for the storage and retrieval of data files. These files were of other types that were not defined to the system. For example, readings obtained from a spectrometer would be stored as a data blob in the archive. The indexing and data type information is entered by the faculty member as part of the store process.

- The third application was to support power users who wanted to store their own data into Content Manager OnDemand for z/OS. These users would write their own applications that interfaced with the Java APIs and were thus able to store and retrieve their data.

Applications, application groups, and folders

Figure 7-7 illustrates the relationship between the application, application group, and folder definitions for the administrative staff users:

- At the application level, each application represents a single document type or version of the document type.
- Each application group is composed of a set of applications that represent a specific function. For example, all documents that are specific to student enrollment would be in an application group, while other documents related to student payments would be in different application group.
- At the folder level, the general rule is to define the folder based on the staff members access requirements (for a specific task). For example, an admissions employee may need access to the application groups that contain the student's history, student's application, and student's payment history.

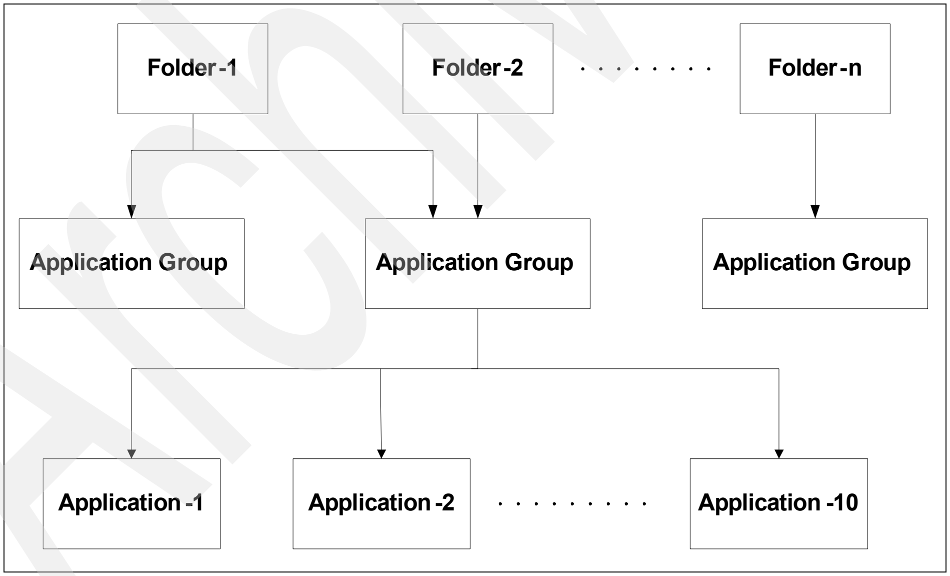


Figure 7-7 Content Manager OnDemand for z/OS case study 3 - Application design for administrative users

Figure 7-8 illustrates the relationship between the application, application group, and folder definitions for the faculty users. Faculty typically defined their data access in one of two ways:

- Project oriented: This method is used for work in progress data storage and access. This is a single or very small number of applications that contain the data specific to a research project or paper that is being worked on, an application group in which all the indexes for these applications are stored, or a single folder that queried that specific application group.
- Research oriented: One or more master folders that accessed multiple application groups. These folders were used when the faculty was looking for archived documents or data that could be located within a specific project or across projects.

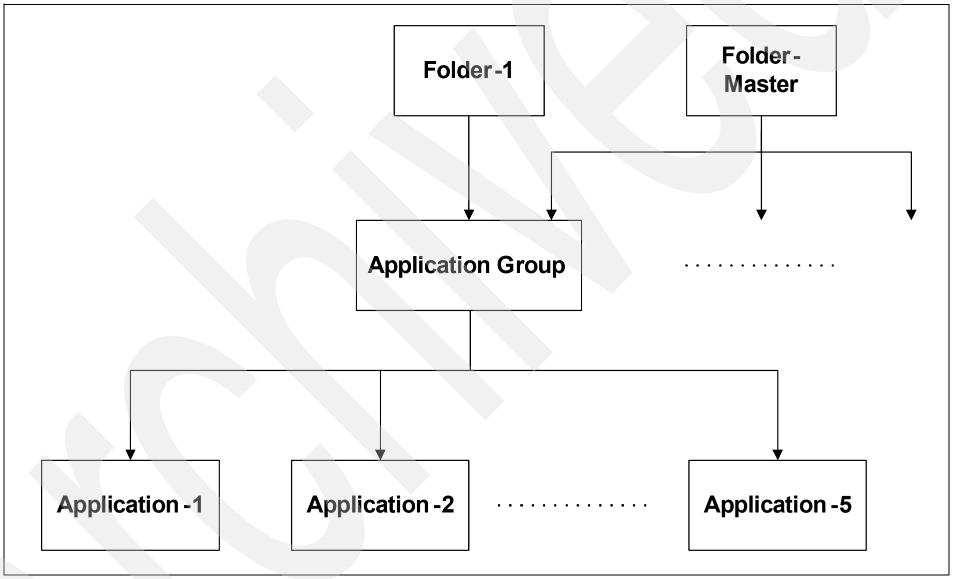


Figure 7-8 Content Manager OnDemand for z/OS case study 3 - Application design for faculty users

Performance testing

The institution performed three separate type of performance tests:

- Load testing: The load tests encompassed running typical daily, monthly, and annual load test cases. Testing revealed that a single load job was sufficient to handle all the report data to be loaded within the given time frame.
- Two-tier document retrieval: Two tier testing was conducted using a script that executed the OnDemand commands. This script simulated the projected number of users and their access intensity.

- Three-tier document retrieval: Three tier document retrieval tests were conducted using Java code that called the Java APIs. It was felt that this would most accurately reflect the projected usage. The predicted *ad hoc* nature of the three-tier retrieval requests would be dealt with by configuring the server to handle a load that would be higher than the maximum expected load. This is simply done by allocating more server threads than the tests called for and by allocating an additional servant in the WebSphere Application Server.

Training

The institution's training requirements were minimal. There were generally three types of skills that are needed:

- System skills: The many years of experience that they had with V2 OnDemand provided them with most of the system skills needed. The additional system skills needed for their implementation, mainly WebSphere Application Server, UNIX System Services, and TCP/IP, had already been acquired as part of being long time WebSphere Application Server users. The main knowledge that they lacked was an understanding of the internals of V7 and the finer tuning points for their implemenatation.They were able to acquire this knowledge from the IBM Content Management Lab Services group during the joint installation and testing phases of the project.
- Report administration skill: Even though the institution's report administrators were highly skilled V2 Content Manager OnDemand administrators, because V2 of Content Manager OnDemand is different from V7 of Content Manager OnDemand, and they were going to expand the usage of OnDemand to new types of users (implying new data types), they decided that there would be significant payoff to acquiring a full understanding of V7 administration features and capabilities. These skills were acquired by first attending training provided by IBM Learning Services. Further implementation specific skills were then transferred from IBM Content Management Lab Services both in a classroom environment and hands on throughout the project life cycle.
  - IM110 - Introduction to IBM Content Manager OnDemand is designed for individuals responsible for creating and loading applications into the Content Manager OnDemand for z/OS system. It provides the basic understanding of all areas of the Content Manager OnDemand for z/OS.
  - OD105 - IBM Content Manager OnDemand System Administration is specifically designed for individuals who are experienced at indexing and loading documents and have a need for a greater in-depth knowledge of the OnDemand system, either for system administration, maintenance, or troubleshooting purposes.
- User training: User training was developed in-house. The OnDemand administrators conducted two in-house training sessions for existing users

and new users. Training sessions for new users will be offered on an annual basis at the IT department's training facility.

Deployment

After all the installation, migration, and performance tuning work was completed, the new Content Manager OnDemand for z/OS was placed in production simply by changing the port number to that of the old Content Manager OnDemand installation port number. This caused all new incoming requests to be routed to the Content Manager OnDemand for z/OS server. The change was transparent to the existing users. Over the next couple of days, the new Windows Client was electronically installed across the campus.

The system is currently fully operational and meeting all users expectations. The system aspect that requires the most work is the help that needs to be provided to new power users when they first start developing their own code to interface with the Java APIs.

Part 3

Appendixes

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 343. Note that some of the documents referenced here may be available in softcopy only.

- *Content Manager OnDemand Guide*, SG24-6915
- *Content Manager OnDemand Backup, Recovery, and High Availability*, SG24-6444
- *IBM System Storage DR550 Setup and Implementation*, SG24-7091
- *Integrating IBM Tivoli Workload Scheduler and Content Manager OnDemand to Provide Centralized Job Log Processing*, SG24-6629

Other publications

These publications are also relevant as further information sources:

- *DB2 Universal Database for z/OS: SQL Reference*, SC18-7426
- *IBM Content Manager OnDemand Distribution Facility Installation and Reference Guide*, SC27-1377
- *IBM Content Manager OnDemand for i5/OS Common Server Planing and Installation Guide,* SC27-1158
- *IBM Content Manager OnDemand: Messages and Codes*, SC27-1379
- *IBM Content Manager OnDemand: User's Guide*, SC27-0836
- *IBM Content Manager OnDemand for Multiplatforms Installation and Configuration Guide,* SC18-9232
- *IBM Content Manager OnDemand for z/OS and OS/390: Administration Guide*, SC27-1374
- *IBM Content Manager OnDemand for z/OS and OS/390: Configuration Guide*, GC27–1373

- *IBM Content Manager OnDemand for z/OS and OS/390: Indexing Reference*, SC27-1375
- *IBM Content Manager OnDemand for z/OS and OS/390: Introduction and Planning Guide*, GC27–1438
- *IBM Content Manager OnDemand for z/OS and OS/390: OnDemand Distribution Facility Installation and Reference Guide*, SC27-1377
- *IBM Content Manager OnDemand for z/OS and OS/390, V7.1 Migration Guide*, LY37-3746
- *IBM Content Manager OnDemand for z/OS and OS/390: WEK Implementation Guide*, SC27-1376
- *Kofax Ascent Capture Release Script Guide*, SC09-7602

Online resources

These Web sites are also relevant as further information sources:

- Content Manager OnDemand V8.4 Information Center:

  `http://publib.boulder.ibm.com/infocenter/cmod/v8r4m0/index.jsp`

- *Create customized views for line data reports*, found at:

  `http://www.ibm.com/developerworks/db2/library/techarticle/dm-0607wagner/`

- IBM Content Manager OnDemand product main Web site:

  `http://www.ibm.com/software/data/ondemand`

  Go to the specific product page by selecting the product (either OnDemand for Multiplatforms, for i5/OS, or for z/OS and OS/390) and click **Go**. From the specific product page, you can:

  – Click the **Information Center** link to get the online Information Center.

  – Click the **Product manual** link to obtain all manuals (in different languages) for the specific product.

  – Click the **Product support** link to get to the IBM Redbooks publications, Technotes, and white papers.

  – Click other links, such as **Demos**, **Developer resources**, and **Web casts** to get other information.

- *IBM Tivoli Storage Manager: Quick Start*, found at:

  `http://publib.boulder.ibm.com/tividd/td/tdprodlist.html`

- Quickstart Guide (listed under Related Resource in the web site)

  http://www.ibm.com/software/data/ondemand/400/support.html

How to get IBM Redbooks publications

You can search for, view, or download IBM Redbooks publications, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy IBM Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM

Redbooks

Implementing IBM Content Manager OnDemand Solutions with Case Studies

(0.5" spine)
0.475"<->0.875"
250 <-> 459 pages

Implementing IBM Content Manager OnDemand Solutions with Case Studies

Product philosophy and history

Platform specific implementation guidelines

Multiple case studies for various platforms

This IBM Redbooks publication will help you implement an IBM Content Manager OnDemand solution from the beginning to the end. We discuss the various stages of the Content Manager OnDemand solution implementation, including planning, software installation and configuration, design, application setup and verification, functional testing, performance testing, training, and finally deployment into production.

The book is intended to provide end-to-end implementation guidelines to audiences who already have general Content Manager OnDemand product knowledge. To really help you understand the implementation process, we provide case studies drawn from real-life implementations. We cover all platforms of the Content Manager OnDemand products, which include Multiplatforms, i5/OS, and z/OS.

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

SG24-7511-00 ISBN 0738485551