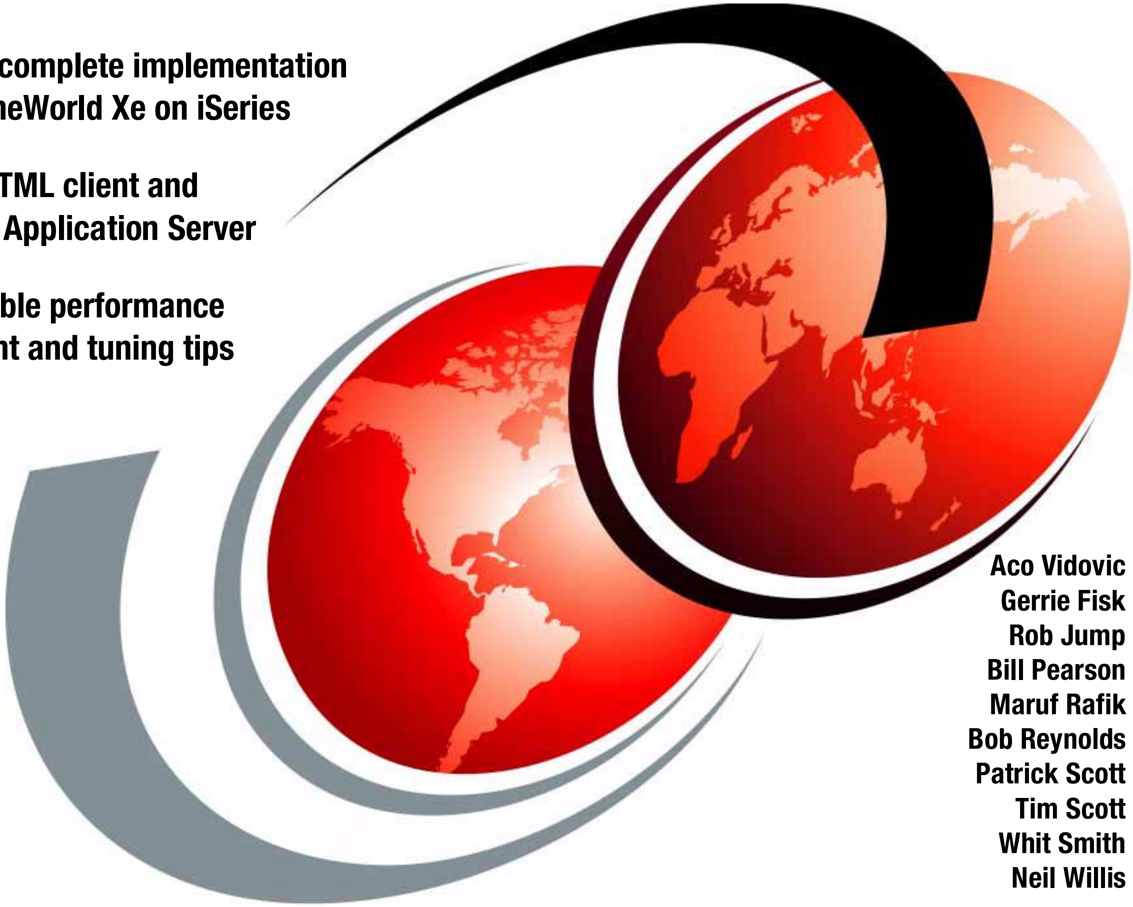
IBM

J.D. Edwards OneWorld Xe

Implementation on IBM eserver iSeries Servers

- Follow this complete implementation guide for OneWorld Xe on iSeries
- Set up an HTML client and WebSphere Application Server
- Learn valuable performance management and tuning tips



Aco Vidovic
Gerrie Fisk
Rob Jump
Bill Pearson
Maruf Rafik
Bob Reynolds
Patrick Scott
Tim Scott
Whit Smith
Neil Willis

Redbooks

ibm.com/redbooks

International Technical Support Organization

J.D. Edwards OneWorld Xe Implementation on IBM @server iSeries Servers

May 2002

SG24-6529-00

Take Note! Before using this information and the product it supports, be sure to read the general information in “Notices” on page ix.

First Edition (May 2002)

This edition applies to J.D. Edwards OneWorld Xe Services Pack 17, for use with the OS/400 Version 5 Release 1.

Contents

Notices ix
Trademarks x

Preface xi
The team that wrote this redbook xi
Special notice xiii
Comments welcome xiv

Chapter 1. OneWorld Xe JAS server overview 1
1.1 JAS server architecture overview 2
1.1.1 HTTP Server 2
1.1.2 Application server 2
1.1.3 OneWorld JAS software 3
1.1.4 Configuring the jas.ini file 7
1.1.5 JDBC drivers 8
1.1.6 J.D. Edwards middleware 8
1.2 Comparison of OneWorld Xe and OneWorld B73.3.2 10
1.2.1 Architecture differences from B73.3.2 to Xe 11
1.2.2 Functionality differences from B73.3.2 to Xe 12
1.3 Generating Java serialized objects 13

Chapter 2. Pre-installation planning, sizing, capacity planning 15
2.1 Pre-installation planning 16
2.1.1 Resources 16
2.1.2 JAS server hardware and software requirements 17
2.1.3 HTML client 18
2.2 Capacity planning 19
2.2.1 Performance Management/400 (PM/400) 20
2.2.2 BEST/1 25
2.2.3 Workload Estimator 49
2.3 Sizing 54
2.3.1 Methodology 54
2.3.2 Benchmark process 56
2.3.3 IBM sizing process 58

Chapter 3. Installation 61
3.1 First-time installation 62
3.1.1 Installing the deployment server 63
3.1.2 Installing the OneWorld databases 84

3.1.3 Planning and configuring OneWorld servers 86
3.1.4 Enterprise server installation 106
3.1.5 Configuring and installing client workstations 122
3.1.6 Completing the installation 127

Chapter 4. Service packs 133
4.1 What a service pack is 134
4.2 Applying a service pack 134
4.2.1 Fundamental steps of applying the service pack 134
4.2.2 System values to verify before applying the service pack 135
4.2.3 Tips for using LINKBSFN when applying the service pack 135
4.2.4 Running PORTTEST after applying the service pack 137
4.2.5 Testing the service pack on the OneWorld client workstation 138
4.3 Rolling back to a previous service pack level 138
4.3.1 Restoring the deployment server 138
4.3.2 Restoring the enterprise server 139
4.4 Running OneWorld on different service pack levels 140
4.4.1 Verifying the current service pack release 141
4.4.2 Creating save files on the iSeries server 141
4.4.3 Transferring service pack files to the iSeries server 141
4.4.4 Verifying the save files residing in the JDETEMP library 142
4.4.5 Installing the service pack 143
4.4.6 Updating the server jde.ini file 145
4.4.7 Creating a subsystem for the newly installed service pack 149
4.4.8 Linking path code business functions to the new service pack 149
4.4.9 Validating the service pack installation 150
4.4.10 Installing the service pack on the deployment server 151
4.5 Configuring environments for the new service pack 152
4.5.1 Adding a new machine record for the new service pack 152
4.5.2 Modifying the existing machine record to remove environments . . 153
4.5.3 Building an update package 154
4.5.4 Deploying client update packages to workstations 171
4.5.5 Updating client workstations and the jde.ini file 171
4.6 Promoting the service pack to production 172
4.6.1 Ending OneWorld and TCP/IP communication services 172
4.6.2 Preparing the iSeries server for service pack promotion 173
4.6.3 Deleting and recreating the subsystem entry 175
4.6.4 Verifying the library list setup 176
4.6.5 Linking path code business functions to the service pack 176
4.6.6 Starting TCP/IP communication and OneWorld services 176
4.6.7 Validating the promoted service pack on the iSeries server 176
4.6.8 Promoting a service pack to production on the deployment server 179
4.6.9 Validating the promoted service pack on the deployment server . . 179

Chapter 5. HTML client and WebSphere setup and tuning 181
5.1 Introduction 182
5.2 Test environment for establishing recommendations 182
5.3 WebSphere Application Server 184
5.3.1 Number of WebSphere Application Server instances (or JVMs) . . 184
5.3.2 Heap size memory settings 185
5.3.3 Garbage collection setting 187
5.3.4 JAS servlet connections 187
5.3.5 Java program optimization 188
5.4 HTTP Server 190
5.4.1 Number of HTTP threads 190
5.4.2 Persistence 192
5.4.3 ACL and Denial of service settings 192
5.4.4 HTTP user profile swapping 193
5.5 OneWorld Xe enterprise server 194
5.5.1 Kernel settings 196
5.5.2 ONEWORLD user profile security setting 196
5.5.3 Checking the kernel setting 197
5.6 OneWorld Xe JAS server 197
5.6.1 JDBC connection pool settings in the jas.ini 198
5.6.2 JDENET connections in the jas.ini 199
5.6.3 Caching Java serialized objects in the jas.ini 199
5.6.4 Asynchronous business function processing in htmlclient.ini 200
5.6.5 Silent post with multi-line edit 200
5.7 JDBC drivers on iSeries servers 201

Chapter 6. XPI on iSeries overview 203
6.1 XPI functional overview 204
6.1.1 XPIe 204
6.1.2 XBP 212
6.1.3 XPIx 213
6.2 Installing XPI on the iSeries server 215
6.2.1 Minimum technical requirements 215
6.2.2 Installing the XPI Foundation code 215
6.2.3 Configuring XPIe 216
6.2.4 Validating the configuration 219
6.2.5 Setting up OneWorld for XPI 220

Chapter 7. SAN technology 223
7.1 Overview of iSeries storage 224
7.2 OneWorld implementation 224
7.3 iSeries and SAN technology considerations 227
7.3.1 Internal versus external storage 227

7.3.2 iSeries server expert cache 227
7.3.3 SAN storage cache 228
7.3.4 SAN configuration: Dedicated versus shared disk 228
7.3.5 Fibre versus SCSI connection 229
7.3.6 Multi-pathing from iSeries to SAN 229
7.3.7 Connectivity of iSeries to SAN 230
7.4 Reference materials 230
7.5 Conclusion 231

Chapter 8. Clustering 233
8.1 Clustering architecture 234
8.1.1 iSeries clustering design 234
8.1.2 iSeries clustering and other platforms 235
8.1.3 iSeries clustering and high availability 235
8.1.4 Other iSeries availability solutions 236
8.2 Typical execution 237
8.2.1 Application to OneWorld Xe 238
8.2.2 Virtual three-tier, two node cluster 238
8.3 Installation process 244
8.3.1 Planning considerations 244
8.3.2 Installation considerations 246
8.3.3 Sample installation plan 247
8.4 Performance tips 249
8.4.1 Primary node performance 249
8.4.2 Secondary node performance 250
8.4.3 Configuration for journaling performance 250
8.5 Operational and maintenance considerations 251
8.5.1 Operational considerations 251
8.5.2 Maintenance considerations 252
8.6 Sources of information 252
8.6.1 Clustering information 252
8.6.2 Installation information 253

Chapter 9. Performance management and tuning 255
9.1 iSeries performance overview 256
9.1.1 Components of performance 256
9.2 Queuing theory and utilization guidelines 259
9.2.1 Queuing theory 260
9.2.2 Utilization guidelines 263
9.3 Setting performance objectives 263
9.4 Basic iSeries server tuning 264
9.4.1 Manual tuning versus automatic tuning 264
9.4.2 Verifying the settings of system values 265

9.4.3 Setting the memory pool sizes and activity levels manually 273
9.4.4 Common WebSphere/Java jobs 277
9.4.5 Working with the prestart jobs. 279
9.5 Performance tools: A top-down approach 285
9.5.1 Which performance tool to use 286
9.6 System level tools 287
9.6.1 Management Central. 287
9.7 WebSphere Application Server level tools 288
9.7.1 WebSphere Resource Analyzer 288
9.8 Java/application level tools 290
9.8.1 DMPJVM 290
9.8.2 Performance Explorer (PEX). 292
9.8.3 Performance Trace Data Visualizer (PTDV) 293
9.8.4 DB Monitor 299
9.8.5 SQL Visual Explain 301
9.9 Communication performance considerations 304
9.9.1 High-speed interconnection between systems 305
9.10 PC performance implications 306
9.11 Optimizing the Java garbage collector 307
9.11.1 Tuning methodology 307
9.11.2 Tuning the garbage collector. 308
9.12 Scaling the OneWorld Xe environment 309
9.12.1 Record locks and index seizes 310
9.12.2 Additional WebSphere Application Servers 311
9.12.3 Additional enterprise application servers 312

Appendix A. Collaborative solutions. 313
Advanced Planning solutions 314
Collaborative CRM 314
Advanced Order Configurator 315
Enterprise Content Manager 315
Additional documentation 316

Related publications 317
IBM Redbooks 317
Other resources 317
Referenced Web sites 318
How to get IBM Redbooks 319
IBM Redbooks collections. 319

Acronyms 321

Index 323

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 illustrates 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. 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 IBM's application programming interfaces.

Trademarks

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

Redbooks(logo)™	DB2 Universal Database™	Perform™
AIX®	Electronic Service Agent™	Redbooks™
AlphaWorks®	Enterprise Storage Server™	SP™
AS/400®	IBM®	SP1®
AS/400e™	iSeries™	TME®
Balance®	Net.Data®	WebSphere®
ClusterProven™	Netfinity®	xSeries™
DB2®	OS/400®	

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

Lotus®	Domino™
Word Pro®	

The following terms are trademarks of other companies:

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

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

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

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

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

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

Preface

This IBM Redbook explains the specific tasks associated with the implementation of OneWorld Xe service pack 17 on the IBM @server iSeries server with OS/400 V5R1 and WebSphere Application Server Version 3.5. It is based on a collection of knowledge from J.D. Edwards consultants and IBM professionals who have experience with J.D. Edwards OneWorld Xe and the iSeries server.

This redbook is designed to assist technical people among J.D. Edwards OneWorld customers, OneWorld consultants, business partners, and IBM service representatives. It targets these professionals who are directly involved with the implementation of a total business solution. Such solutions consist of iSeries server hardware, OS/400, DB2 UDB for iSeries database, OneWorld solutions, and supplemental solution products.

You should use this redbook in conjunction with *J.D. Edwards OneWorld Implementation for AS/400*, SG24-5195. This publication explains detailed concepts and all implementation steps of earlier versions of OneWorld, most of which still apply to OneWorld Xe. You should also use this redbook together with the J.D. Edwards manuals that are provided with the OneWorld software.

The team that wrote this redbook

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

Aco Vidovic is a Senior IT Specialist in IBM ITSO Rochester Center on assignment from IBM Slovenia. Before joining the ITSO in 1999, he worked in the AS/400 area for 10 years in a variety of roles from second-level technical support to marketing. In the ITSO, Aco teaches extensively and writes Redbooks about ERP solutions implementation and WebSphere Application Server on the iSeries server.

Gerrie Fisk is a Senior I/T Specialist and has been with IBM for 25 years. She is a member of the IBM/J.D. Edwards International Competency Center, specializing in the iSeries server. Gerrie is responsible for OneWorld benchmarks and sizing tests on the iSeries server. Her previous assignments were in the midrange arena as a systems engineer installing, supporting, and upgrading over 500 AS/400s and S/3Xs, as well as teaching many AS/400 classes and seminars.

Rob Jump is a Senior I/T Specialist in the IBM/J.D. Edwards International Competency Center. His current work involves benchmarking and sizing tests for J.D. Edwards OneWorld on the iSeries and AS/400 servers. Rob has over ten years experience with the System 36, System 38, AS/400 system, and iSeries server, working with installation, support, and communications. He is also a certified Windows NT product specialist and has worked as a Windows NT and network consultant.

Bill Pearson is the Technology Alliance Manager for J.D. Edwards assigned to IBM. As the technical lead within the partner organization, he communicates and facilitates technology-related activities between IBM and J.D. Edwards' internal organizations. He identifies and prioritizes technical initiatives, assists with marketing strategies, and organizes technical briefings for customers and partners. After joining J.D. Edwards in 1996 as a Technical Consultant for Worldwide Customer Support, Bill became a technical field consultant for Partner Services, and more recently the Genesis Consulting Area. He has a bachelor of science degree in computer information systems from Arizona State University.

Maruf Rafik is a Technical Consultant for the Global Support Services at J.D. Edwards. Prior to joining at J.D. Edwards, he worked as a software test engineer, Web application developer, and AS/400 system administrator. Maruf is currently focused on providing technical support for OneWorld software on the iSeries server platform.

Bob Reynolds is an IBM certified iSeries specialist currently assigned to the IBM/J.D. Edwards International Competency Center in Denver, Colorado. He has over 17 years of iSeries experience across many areas of information technology. These areas include ERP implementation, application development, hardware and software sales, systems engineering, business recovery, capacity planning, and project management. He is also ICCP certified, holds a bachelor's degree in computer science and sociology, and a master's degree in business administration.

Patrick Scott is a Senior Consultant for J.D. Edwards Global Advanced Technology Services. His areas of expertise include UNIX and SAN technology. He has a broad range of technical knowledge in J.D. Edwards software, networking, security, clustering, and Web solutions. He provides technical support for other J.D. Edwards consultants, business partners, and customers.

Tim Scott is an Senior Web Architect for Global Advanced Technology Services. He leads a cross functional team that consists of all technical groups within J.D. Edwards that work on the Web. In addition, he develops technical training courses, manuals, and newsletters for the OneWorld thin client solutions. His current responsibilities focus on the J.D. Edwards HTML, Portal, and Web-based product lines. During his four years with J.D. Edwards, Tim has spent two years with Advanced Technologies and the other two in development as a member of Partners in Development.

Whit Smith has almost twenty years of experience working with IBM mid-range systems. He spent most of that time on the AS/400 system and its predecessor the S/38. Whit currently provides advanced technical support on the AS/400 and iSeries servers for J.D. Edwards OneWorld software in the IBM Americas ERP Advanced Technical Support group.

Neil Willis is a an Advisory Software Engineer in ERP Development at IBM in Rochester, Minnesota. He is the author of many books and papers and has presented and taught classes worldwide. Neil has worked at IBM Rochester for six years in the areas of International Technical Support, Performance Design, and ERP Development. Prior to joining the Rochester team, he was an Advisory Systems Engineer with IBM Australia for 10 years. His current projects include high level scaling of ERP solutions and unicode performance. He holds a bachelor's degree in computer science, a bachelor's degree in electrical engineering (honors) and a master's degree in information systems.

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

Boyd Fenton
IBM Denver

Julie Ransom
IBM Rochester

Special notice

This publication is intended to help technical people to implement J.D. Edwards OneWorld on the iSeries server. The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM or J.D. Edwards.

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. JLU Building 107-2
  3605 Highway 52N
  Rochester, Minnesota 55901-7829

1

OneWorld Xe JAS server overview

The OneWorld JAS server was known originally as *OneWorld Java Application Server*. This name was given before the Internet community settled on what an “application server” is. When used in J.D. Edward’s context today, this phrase can become confused with other similar terms. For example, OneWorld enterprise servers are often referred to as *application servers*. WebSphere Application Server is also known as a Web application server. That is why we have shortened the phrase and refer to it as the *OneWorld JAS server* or *JAS server*.

With the release of OneWorld Xe, this chapter describes how J.D. Edwards migrated 100% of the application functionality to an HTML-based solution. It looks into the architecture of what makes up a OneWorld JAS server. In addition, this chapter describes the tools and applications that are delivered with a OneWorld Xe JAS installation. It compares and contrasts the new features and functionality from OneWorld B73.3.2 and Xe. And finally, this chapter discusses the process of generating Java Serialized Objects from J.D. Edwards TAM specs.

1.1 JAS server architecture overview

This section discusses the architecture of a J.D. Edwards OneWorld JAS server. It explains all the underlying technologies that are installed and how they communicate with one another. It also discusses how the JAS server communicates with other OneWorld servers, both for database and logic. We describe the following topics and components:

- HTTP Server
- Application server
- OneWorld JAS software
- Configuring the jas.ini file
- JDBC drivers
- J.D. Edwards middleware – Logic and BSFNs

1.1.1 HTTP Server

HTTP Server (also known as Web server) is a computer or software that delivers (serves up) HTML pages and images. Every Web server has an IP address and possibly a domain name. The HTTP Server is responsible for receiving requests from the client's Internet browser and returning the requested HTML pages and pictures.

IBM HTTP Server for iSeries

When installing the HTTP Server on an iSeries server, J.D. Edwards supports and recommends using IBM HTTP Server for iSeries (original) or IBM HTTP Server for iSeries (powered by Apache). Both products contain custom modifications that make it integrate more tightly with WebSphere Application Server. IBM has included additional functionality to improve performance. For example, the triggered cache manager (TCM) provides a means to manage dynamic Web pages such as those generated by CGI programs, Net.Data, or Java servlets.

1.1.2 Application server

Application servers extend HTTP Server capabilities to handle application requests. In addition, they provide administrative tools such as tracing and debugging, performance monitoring, and many additional features. HTTP Servers are limited to mostly static content such as HTML pages and pictures. However, they have become more dynamic with their ability to use such technology as CGI.

J.D. Edwards dynamically creates every page as requested. This allows users to see new application modifications, such as header changes and additional grid columns on the fly, as they are generated (see 1.3, “Generating Java serialized objects” on page 13). The OneWorld Xe Web applications are generated by the use of servlets and JavaServer Pages (JSP).

IBM WebSphere Application Server

Currently, J.D. Edwards only supports one application server and that is IBM WebSphere Application Server. Its ability to run on iSeries, Windows NT/2000, and UNIX platforms makes it a perfect fit for J.D. Edwards Web products. Just like OneWorld, it is important to allow customers to use the hardware platform that they are capable of supporting. In addition, the nature of Java is portability, and WebSphere Application Server allows it to run server side Java code such as servlets, JavaServer Pages, and Enterprise JavaBeans (EJBs).

1.1.3 OneWorld JAS software

J.D. Edwards delivers OneWorld JAS software to Web enable their enterprise software package. With the OneWorld JAS software, J.D. Edwards offers:

- An HTML client
- Java client
- Portal solution
- Server Administration Workbench (SAW) application

Once the underlying foundation technology is in place, the J.D. Edwards JAS software delivers all of the necessary code to use any combination of Web clients with its JAS software. The largest portion of the JAS software is the Java Archive files or .jar files. These are compilations of Java classes that include servlets, applets, and JSPs. They also includes other vital objects such as:

- Cascading Style Sheets (.css files)
- Java script files (.js files)
- Images (.gif files)
- Help files

These objects make up the look and feel or the presentation layer of the different Web clients. Let’s discuss the different types of applications that are delivered.

The HTML client

The HTML client is the most common interface delivered with the OneWorld JAS server installation. These servlet-generated pages load Java serialized objects for application runtime information.

All J.D. Edwards applications that run in a traditional “fat” client environment also run on the HTML client. When the HTML client is launched, virtual client sessions are created on the JAS server. Each virtual client represents a validated user who has accessed the HTML client. Each virtual client is made up of four main components:

- State information
- Non-state information
- Web run-time engine
- Rendering agent

The HTML client has much the same look and feel as the Win32 client and almost all of the functionality of OneWorld using the terminal server. Modifying grid colors and sequencing, importing, and exporting into Microsoft Word and Excel are some of the features added to the HTML client since the introduction of OneWorld Xe. The user interface has also gone through many usability changes and has a cleaner look and feel as you can see by the example in Figure 1-1.

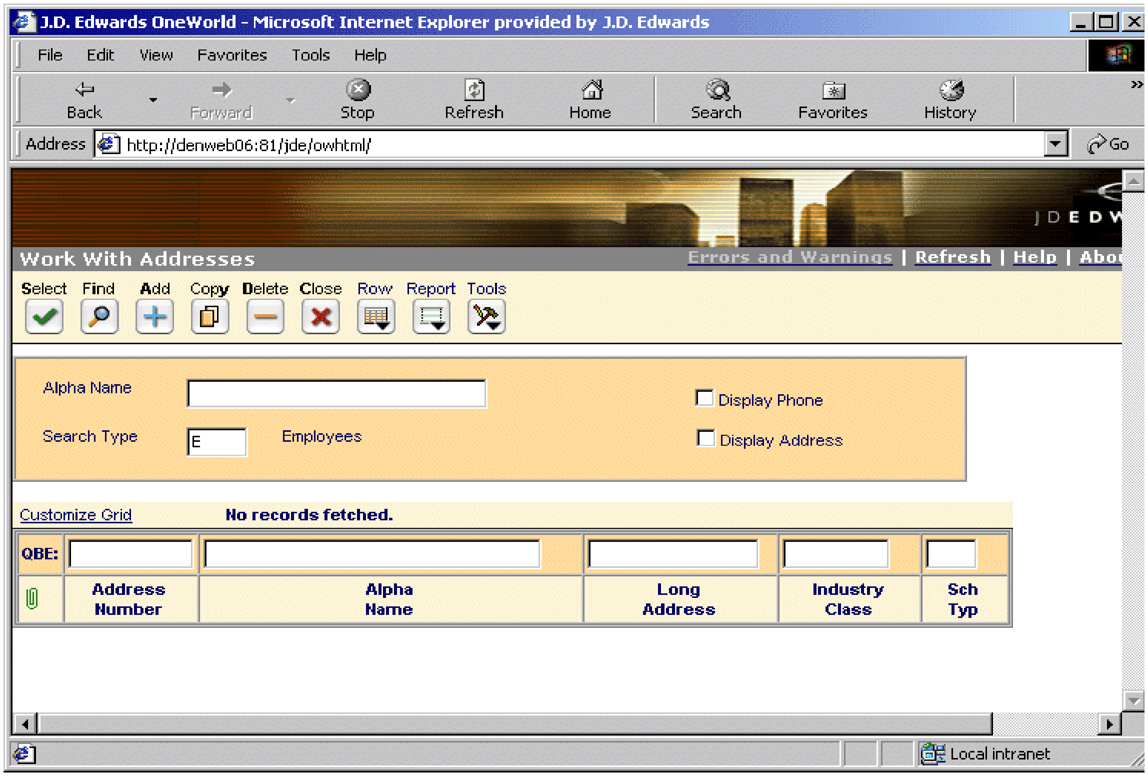


Figure 1-1 HTML client

OneWorld Portal

The OneWorld Portal is J.D. Edwards Enterprise Integrated Portal (EIP) solution. It is a servlet and JSP-based framework that allows users to load the OneWorld HTML client and task explorer as well as any active Web content they desire. Such as Active Server Pages (ASP), Java applets, URIs, etc. You can also plug-in J.D. Edwards integrated applications through the portal such as Knowledge Management/Business Intelligence (KMBI), CRM, and SAW. In addition, external third-party applications can also be used within this framework, such as Outlook, stock tickers, and so on.

The portal integrates closely with OneWorld Solution Explorer functionality. Therefore, new concepts, such as roles and tasks, are used within the portal to control component and workspace permissions and even menu favorites. It is also completely customizable; you can modify the look and feel of the portal to look like any intranet. The color schemes, logos, and toolbars are just some of the external items that you can manipulate for the same look and feel as the rest of the company's infrastructure (Figure 1-2).

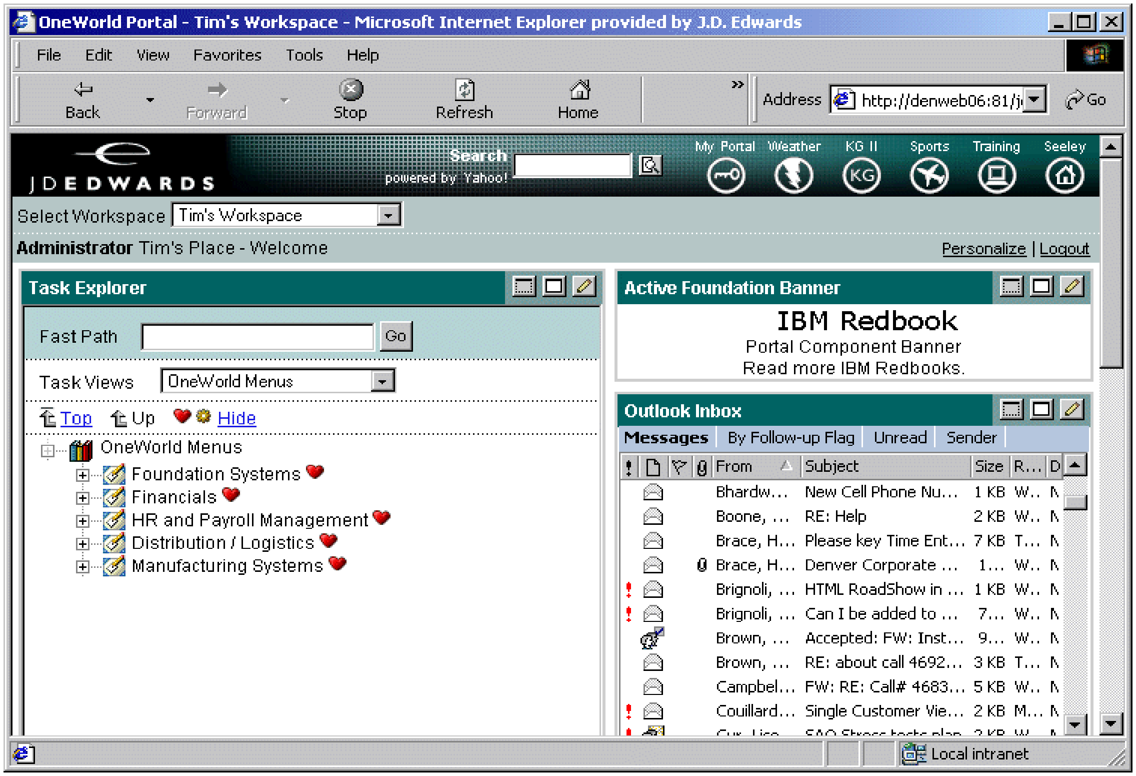


Figure 1-2 OneWorld Portal

Server Administration Workbench (SAW)

Server Administration Workbench has been a traditional part of the J.D. Edwards administrative tool set. The new “Web” SAW tool extends the traditional capabilities of the application and delivers them in a browser-based framework. In its initial Xe release, however, “Web” SAW was used only for the administration and troubleshooting of JAS servers. Its capabilities have been extended to monitor and administrate OneWorld enterprise servers and eXtended Process Integration (XPI) servers.

You can use SAW to collect both logging information and dynamic data such as kernels, threading, and virtual client information. It is a useful tool for trouble shooting issues on both JAS servers and enterprise servers. An example of the Server Administration Workbench is shown in Figure 1-3.

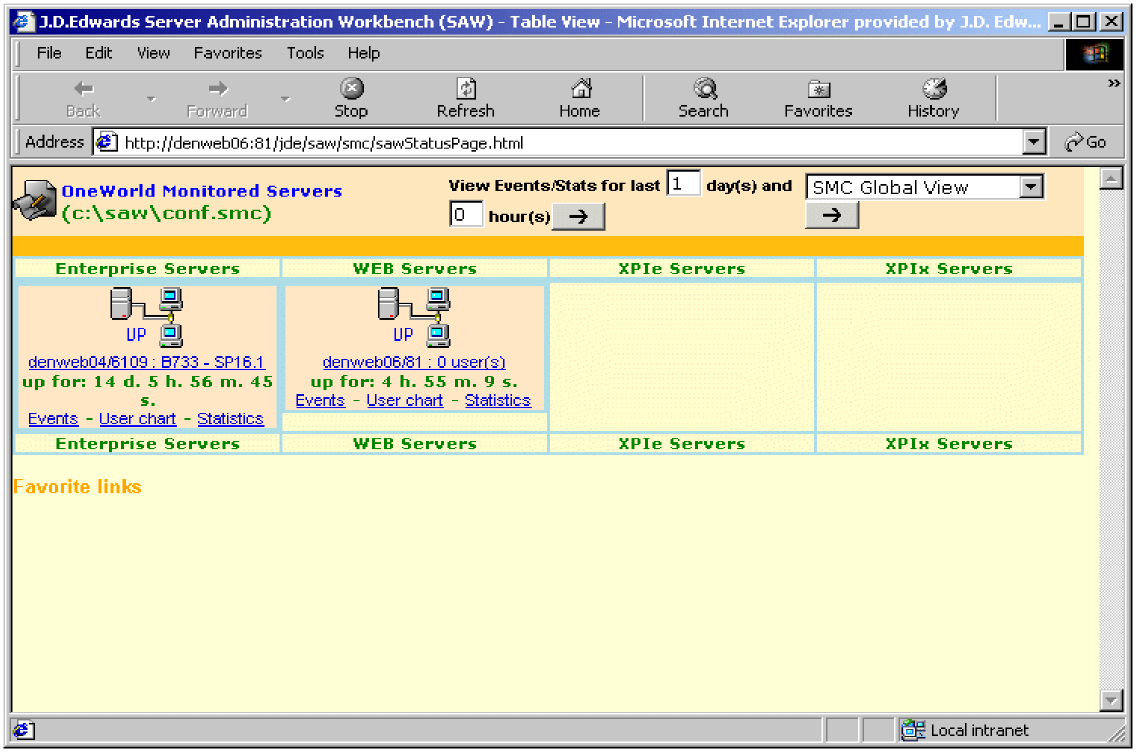


Figure 1-3 Server Administration Workbench

One of the struggles that SAW has with the traditional “Web” was having to catch errors in real-time without any method to store collected data. From the Web, administrators can now set up a Server Monitoring Configuration (SMC) to monitor different instances of OneWorld servers and notify assigned users via e-mail or a pager when an assigned agent detects a faulty event with any of the instances. The Server Administration Workbench provides a unified Web interface for administrators to monitor OneWorld servers (Enterprise, Java, and XPI) remotely through a browser.

1.1.4 Configuring the jas.ini file

The jas.ini file is compilation of settings that dictates how the JAS server will function with its internal components and surrounding environments. When you configure the jas.ini file, the JAS server can talk to the OneWorld enterprise server, the database, the security server, and so on. The OneWorld JAS server communicates to the backend enterprise server in much the same way that a client workstation communicates.

Key sections of the jas.ini file

Some of the key sections of the jas.ini file include:

- [DB SYSTEM SETTINGS]

  This section determines all the essential database information for connectivity. It should be the same as the [DB System Settings] section of a jde.ini file on a working client workstation.

- [SECURITY]

  This section needs a server with an active security kernel. The security setting is a mandatory setting and must be set correctly.

- [OWWEB]

  This section is important because it allows an administrator to hard code the path codes that are available to the Web users. It is also dictates the location of the TNSNAMES information (required for Oracle and SQL only).

- [JDBC DRIVERS]

  This section determines which type of database the JAS server is connecting to and name of JDBC driver to connect with.

- [CONNECTION POOL]

  Connection pooling allows HTML clients to use fewer of server jobs than fat and TSE clients because they can be reused. In this section, you specify the number of connections in the connection pool.

- [CACHE]

  This section specifies the number of milliseconds to hold the cache for Java serialized objects.

- [JDENET] and [SERVER]

  These sections verify that the port number is consistent with what is used on the enterprise servers and the client workstations.
- [LOGS]

  The section describes the location of the JAS and JASDEBUG logs. It also holds the settings for whether to enable logging.

1.1.5 JDBC drivers

Java Database Connectivity (JDBC) is an application program interface (API) specification for connecting programs written in Java to the data in a popular database. JDBC is similar to ODBC in the function it serves, but is not language-independent like ODBC.

JDBC options for the iSeries

When communicating with the database on iSeries server, there are two different JDBC driver options:

- **Native driver that comes with the IBM Developer Kit for Java (part of OS/400)**: It uses *native calls* to access the database, therefore the database must reside on the same server (All In One in our scenario). The native driver runs within QSQSRVR jobs under the QSYSWRK subsystem.
- **IBM Toolbox for Java (IBM licensed program 5722-JC1) driver**: This is a network driver that uses TCP/IP sockets to access the database that can be on the same or on another server. The Toolbox driver uses QZDASOINIT jobs in the QUSRWRK susbsystem.

The driver option you should use depends on your infrastructure. The native driver runs only on the OS/400 Java Virtual Machine (JVM), but performs better than the Toolbox driver when the data is on the same machine as the application. The Toolbox driver runs on any JVM (including the JVM shipped with OS/400).

The current advice is that, in general, if you intend to run your program only on the OS/400 JVM and the data is on the same machine, use the native driver. If you intend your program to run on other JVMs, or the Java program is on one iSeries server and the data is on a different iSeries server, use the Toolbox driver.

1.1.6 J.D. Edwards middleware

J.D. Edwards middleware performs the connections from client to server and server to server, and sends messages for distributed requests. It is a peer-to-peer, message-based, socket-based, multi-process communication middleware solution.

JDENet communication middleware

JDENet is the J.D. Edwards proprietary communication middleware package that provides server-to-server and client-to-server communication. The OneWorld JAS server communicates to the OneWorld Enterprise server for logic and business function calls via this transport.

Logic and business function mappings

The JAS server uses middleware to make business function calls and to submit logic. Unlike the OneWorld terminal server environment, the environments that the Web clients access are *"J" environments* or *Java environment*s. These Java environments completely depend on logic being centralized to the enterprise servers or logic server.

You can access OneWorld business logic in two different ways:

- C business functions
- Named Event Rules (NER)

Naming schemes used by application developers to distinguish the two different types. C business functions are denoted with (BXXXXXX), and NER are indicated by (NXXXXXX).

You can map these business functions to run on the client (type 1) or on both the client and server (type 2). The type 1 C business functions are created this way for a reason. They have been written with local dependencies to a Windows-based Active X control or API. The functions do not run on a UNIX or iSeries logic server. On the other hand, type 1 NER (NXXXXXX objects) are converted to execute at runtime for the Web clients. Therefore, even though NER are sometimes mapped locally, they still run in the Web clients.

Note: You should not remap or modify the business function mappings that are delivered with the software, unless J.D. Edwards specifies you to do so. Changing these mappings can cause applications to run improperly or not function at all.

Figure 1-4 shows the different layers that make a OneWorld JAS server. Each component is separate and unique, but is vital in making a JAS server work.

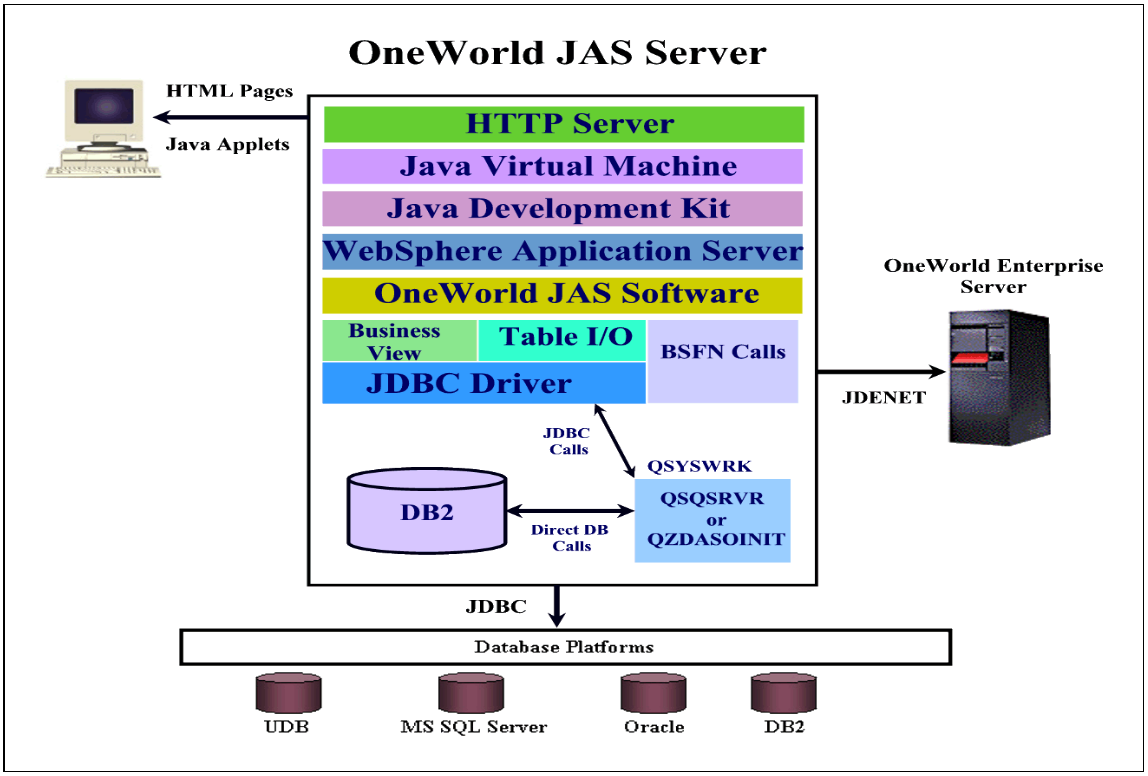


Figure 1-4 JAS server components

1.2 Comparison of OneWorld Xe and OneWorld B73.3.2

This section examines the differences between the OneWorld B7.3.3.2 Web product and the OneWorld Xe model. It also discusses the differences of both the JAS server architecture and the application functionality between the two releases.

Dynamic Web servers, such as the OneWorld JAS server, have become more robust as the underlying technologies that they are built on have matured. HTTP Servers and application server technologies are continually introducing new features and becoming faster and easier to scale. This allows J.D. Edwards to introduce new functionality and tools that capitalize on a better and faster underlying architecture.

1.2.1 Architecture differences from B73.3.2 to Xe

Prior to OneWorld Xe, the only supported JAS server platform was Windows NT 4.0 on Intel processors. With the introduction of Xe, J.D. Edwards now provides support for iSeries, Windows 2000, and UNIX Web servers.

Check the J.D. Edwards Knowledge Garden at `https://knowledge.jdedwards.com` for up-to-date information on the supported Web server platforms and software releases. The Web server architecture for B7.3.3.2 was based on WebSphere Application Server 2.X and has been migrated to the more robust and reliable WebSphere Application Server 3.5.

> **Note:** The J.D. Edwards Knowledge Garden requires a user name and password. If you don't have access to the Knowledge Garden, sign up for access on the J.D. Edwards Online Support site at:
> `http://www.jdedwards.com/onlinesupport/`

Pure Java model over OWProxy

JAS relied on a Win32 client installation on the JAS server to execute OneWorld business functions. With OneWorld Xe, that client piece, known as *OWProxy*, has been removed. JAS now executes business functions on any OneWorld application or enterprise server.

Scalability and stability issues caused by the OWProxy were removed with the pure Java implementation of JAS. Compared with the architecture shown in Figure 1-4, the architecture of the B73.3.2 JAS server was quite different. Figure 1-5 shows the architecture of the old OneWorld JAS server.

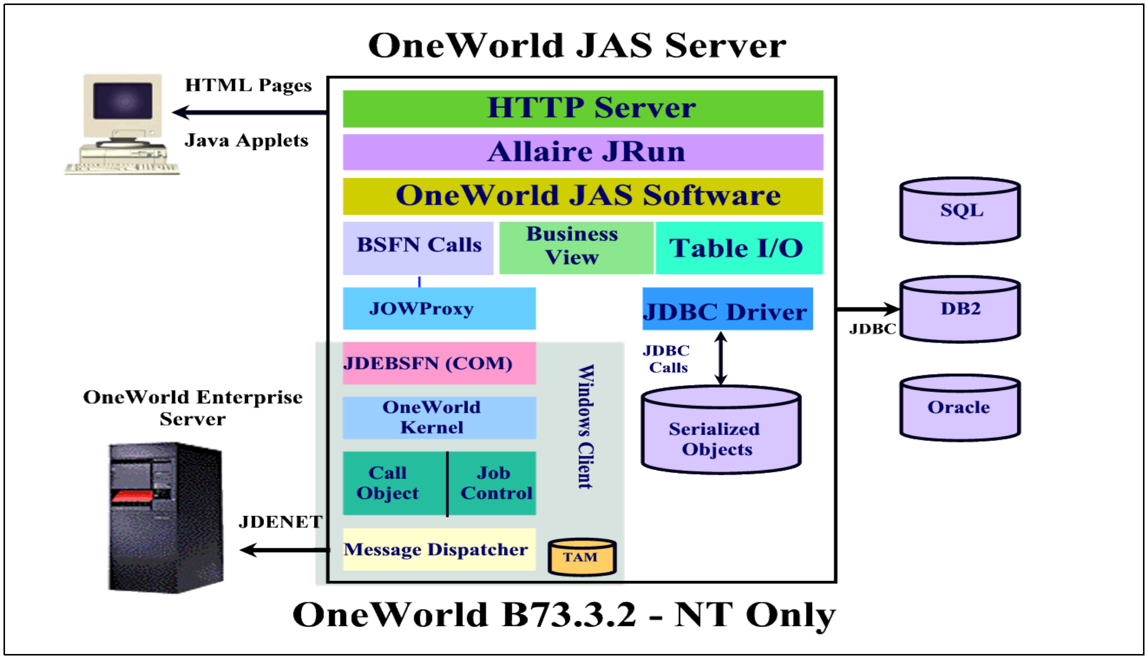


Figure 1-5 OneWorld B73.3.2 JAS server architecture

1.2.2 Functionality differences from B73.3.2 to Xe

This section explains the internal differences of the OneWorld Web application functionality and its maturity in the Xe release.

The OneWorld HTML client had some scalability and functionality issues that stunted its widespread deployment (for instance, parent/child forms were not supported). Furthermore, there were many business functions that had been classified as client only. While they could run on the server, they needed to be tested prior to being converted.

Prior to the Xe release, J.D. Edwards application development analyzed all of the base business functions (BSFN). Then they re-architected all possible client-side BSFNs to run on the OneWorld enterprise server. Even with Xe, some still type 1 BSFNs are still mapped locally and should not be mapped to the enterprise server. See 1.1.6, “J.D. Edwards middleware” on page 8, for a detailed explanation of business function mappings and how they work.

1.3 Generating Java serialized objects

To run the OneWorld Java server, the server must have access to a set of serialized OneWorld JAS objects. These serialized objects need to be generated directly from a complete set of OneWorld TAM specifications. These generated Java serialized objects are inserted into two tables (F989998 and F989999) within the specified database.

Generating OneWorld serialized objects requires a specific machine configuration. While it is possible to configure a Web server to generate OneWorld serialized objects, the recommended method is to dedicate a separate generation machine for this process. See the Minimum Technical Requirements web page on the Knowledge Garden (`https://knowledge.jdedwards.com`) for a list of these requirements. The main reason for this recommendation is note to burden the JAS server with additional workload while users are on-line.

If you upgrade to OneWorld Xe from a previous release and customize your OneWorld objects, test your custom modifications first. Then generate serialized JAS objects from the upgraded path code.

To generate serialized objects for the Java server, your developers must install and use the Java and HTML Generator. This is a OneWorld tool that turns OneWorld specifications into Java serialized objects so you can access OneWorld applications in Java, HTML, or both (see Figure 1-6). The OneWorld objects that are generated are now held in a serialized format. OneWorld stores the objects in a relational database and retrieves them at runtime. By default, there is a set of tables for each pathcode that resides in the Central Objects tablespace for that pathcode. The tables used for holding these records are labeled as:

- F989998 - Java Persistent Objects Cross Reference (improves performance)
- F989999 - Java Persistent Objects Table

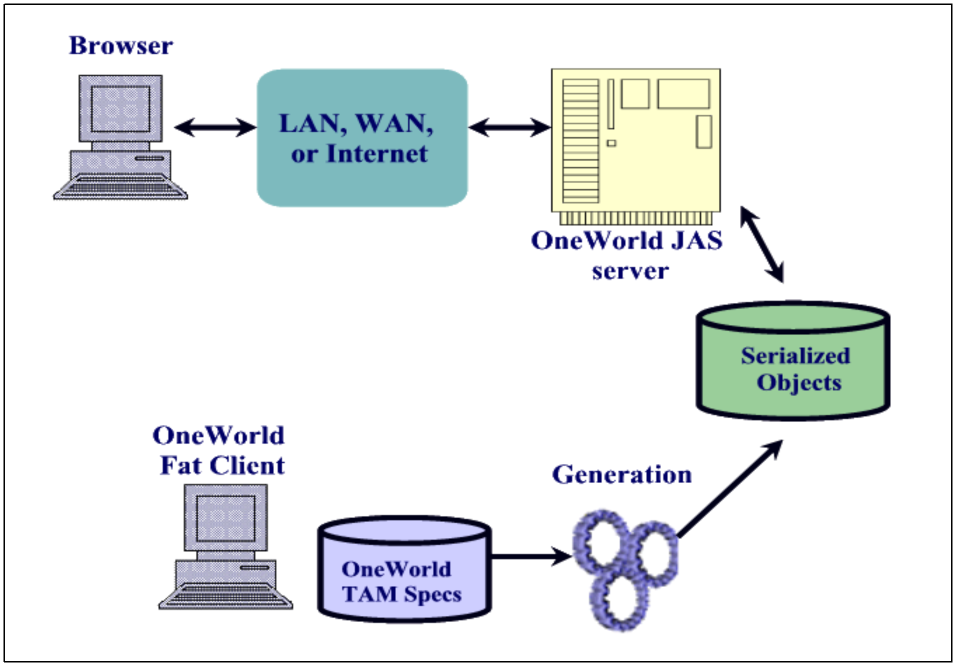


Figure 1-6 Generation process flow

> **Tip:** J.D. Edwards strongly recommends that you generate all objects for each individual pathcode. That way you can be assured that the JAS server is working with the same specifications as your OneWorld Windows clients for those pathcodes. Generating JAS objects is equivalent to building a package for OneWorld fat clients. It is vital that a complete and thorough generation of serialized objects is done to ensure a successful implementation. Coordinating package builds with the generation cycles is highly recommended.

2

Pre-installation planning, sizing, capacity planning

As with any software installation process, pre-installation planning is an important task. A well prepared plan saves considerable time and effort in the later stages of the implementation.

This chapter discusses planning and pre-installation topics relevant to the OneWorld Web environment. Pre-installation planning for a new OneWorld installation is also covered in Chapter 8 of the redbook *J.D. Edwards OneWorld Implementation for AS/400*, SG24-5195, as well as in the OneWorld installation manuals. Use these resources in conjunction with the information provided here to plan the installation. The information in this chapter is specific to the OneWorld Web environment.

Once the software is installed and in use, it is important that you monitor and tune system utilization to provide maximum system performance and the best utilization of system resources. Chapter 9, “Performance management and tuning” on page 255, discusses topics related to tuning and performance. This chapter provides the information necessary for capacity planning to prepare for future growth and business changes.

2.1 Pre-installation planning

Pre-installation planning is important to ensure that the setup, configuration, and deployment of the hardware and software goes smoothly. A number of resources are available.

2.1.1 Resources

The redbook *J.D. Edwards OneWorld Implementation for AS/400*, SG24-5195, covers many topics related to a typical OneWorld environment. Chapter 8, “Pre-installation and Planning”, has a detailed discussion of preparation steps necessary to ensure a smooth implementation. Topics such as hardware minimum requirements, disk space needed and operating system versions supported, as well as necessary software and version levels are discussed. Some of the information no longer pertains to current releases of OneWorld or current environments. This redbook provides updated information that supersedes that provided in the first redbook.

The OneWorld installation manuals are also valuable resources and should be consulted as the basis of any installation. They contain critical information regarding minimum requirements, software and operating system versions supported as well as step by step installation directions.

This section is not meant to duplicate any of the above sources. The intent is to provide additional insight to make the implementation of the OneWorld Web environment easier and to provide a convenient index of available resources.

Web sites

The J.D. Edwards Global Advanced Technology Services (GATS), formerly Worldwide Advanced Technologies (WWAT) group does extensive testing on new technologies and provides advanced consulting services in a wide variety of environments. Their Web site can be reached by accessing the **Services** tab on the Knowledge Garden Web site at: `https://knowledge.jdedwards.com`

You should consult the Knowledge Garden Web site for the latest information on specific technologies. After you enter your user and password information, use the menu structure on the left-hand side to navigate to the Product subdirectory. After the subdirectories are displayed, select **OneWorld Online** and then **Minimum Technical Requirements**.

You can find further information regarding OneWorld known issues, tips and techniques, and Software Action Requests (SARs) that may affect your installation processes on the J.D. Edwards Solutions Web site at:
`http://www.jdedwards.com/public/0,1921,0%257E533%257E,00.html`

Access to this Web site is available to all J.D. Edwards customers.

You must also ensure that your deployment server hardware supports Windows NT Server 4.0 or Windows 2000. To verify this, check Microsoft's Hardware Compatibility List (HCL) for Windows NT and Windows 2000. You can find this on the Web at: http://www.microsoft.com/isapi/hwtest/hcl.idc

This site contains a detailed list of information regarding tested and approved hardware components, such as disks, network adapters, and so on. Note that the deployment server can be either a stand-alone Windows NT server, such as an IBM @server xSeries Server, an Integrated xSeries Server (formerly the Integrated PC Server and Integrated Netfinity Server) installed on the iSeries server, or an external xSeries server attached via the Integrated xSeries Adapter.

IBM supplies Informational Authorized Problem Analysis Reports (APARs) that contain the details of all PTFs required to facilitate OneWorld support on the iSeries server. You can find links to these Informational APARs on the iSeries Support Web site at:
http://www.ibm.com/servers/eserver/iseries/service/erp/jdesupport.htm

This Web site contains other useful information such as tuning guides and OneWorld related documentation. The APARs are updated regularly and, therefore, should be referenced periodically both during installation planning and post-installation.

Some of these Web sites are only available to J.D. Edwards business partners and customers. Others are available to everyone. If you do not have access to these sites, contact J.D. Edwards Worldwide Customer Support.

2.1.2 JAS server hardware and software requirements

This section covers hardware and software requirements for the JAS server.

Hardware requirements

One of the primary changes with OneWorld Xe is the move to IBM WebSphere Application Server as the platform for the OneWorld Java Application Server. This is discussed in detail in Chapter 1, "OneWorld Xe JAS server overview" on page 1.

Previously this functionality was only provided on Windows NT 4.0, with Microsoft IIS 4.0 and JRun 2.2. With the change to IBM WebSphere Application Server, OneWorld Java Application Server functions can be performed on all supported platforms, including the iSeries and UNIX-based systems. This greatly expands the options available and the scalability of the OneWorld Web environment.

When running on the iSeries server, the JAS server runs natively and can be run in the same partition or in a separate partition as the OneWorld enterprise and database functions. Chapter 5, “HTML client and WebSphere setup and tuning” on page 181, discusses the considerations and tuning recommendations when running the JAS server on the iSeries.

Software requirements

The following software is required on the iSeries JAS server in a OneWorld Xe environment:

- OS/400 V4R5 or higher
- 5769-TC1: TCP/IP Connectivity Utilities
- 5769CX2: ILE/C Compiler for iSeries
- 5733-AS3: IBM WebSphere Application Server Standard Edition Version 3.5 or higher
- 5769-JV1: IBM Developer Kit for Java Version 1.2
- 5769-JC1: IBM Toolbox for Java
- 5769-SS1 opt 30: OS/400 Qshell Interpreter
- 5769-SS1 opt 12: Host Servers
- 5769-DG1: IBM HTTP Server for iSeries

WebSphere Application Server editions

IBM WebSphere Application Server Standard Edition (IBM licensed product 5733-AS3) is required to run the OneWorld JAS server. You may also use the Advanced Edition (IBM licensed product 5733-WA3), but it is not required. A primary enhancement to the Advanced Edition is support for Enterprise JavaBeans (EJBs). OneWorld currently does not use EJBs, so the added functionality of IBM WebSphere Application Server Advanced Edition is not needed. However, other applications or client-developed software that can run on the same server may use EJBs, and therefore, require the Advanced Edition.

To obtain WebSphere Application Server, contact you IBM representative or your hardware business partner representative. Or you can call IBM directly, toll-free in the United States, at 800-426-2255.

2.1.3 HTML client

The OneWorld HTML solution uses a very thin client. Chapter 1, “OneWorld Xe JAS server overview” on page 1, offers a detailed discussion of the HTML client architecture. Most of the OneWorld functions run on the JAS server or the enterprise server, taking the resource requirements off of the client machine. The only requirement is the ability to run a browser such as Microsoft Internet Explorer or Netscape Navigator. These browser packages do not require many resources and can be run on most smaller, slower machines with good performance.

There are a few advanced features within OneWorld, such as Silent Post, which may not be available on all browsers or on older browser versions.

Browser versions

The HTML client requires one of the following browser packages:

- Microsoft Internet Explorer Version 5.5 or Higher
- OneWorld SP16.1 and older supports Internet Explorer 5.0 (SP 2) and Netscape Navigator 4.73

For the latest changes in supported versions, consult the Minimum Technical Requirements as outlined in “Web sites” on page 16.

2.2 Capacity planning

Capacity planning is crucial to the continued success of any environment after the initial installation. To respond to business changes effectively, your system must also change. It is much easier to manage and plan for changes than to simply react to them.

Sometimes good performance just happens. In those cases, the system has plenty of resources to do the job, but there are times when those resources are not in the right place. Maybe systems and clients have been added to the network, or production volume has increased and the workload is significantly higher. Or, more often, workload changes in small, nearly invisible increments, and one day, performance is simply not as good anymore.

Understanding the performance components of your system helps you react quickly when a performance problem occurs at a crucial time. Performance management is a strategy for planning, implementing, controlling, and measuring computer-based tasks to achieve performance that is acceptable.

Chapter 9, “Performance management and tuning” on page 255, provides details on how to analyze the performance of your system, including the enterprise server and the JAS server. This analysis helps you gather the information you need to effectively plan for your future needs.

IBM also provides information and services that can help you plan your capacity needs. Performance Management/400 (PM/400) is a service that is included. It tracks your current system performance and alerts you to possible resource contention. You can use this information in conjunction with your growth plans to help you plan for the future.

This section explains how to use the IBM tools, such as PM/400, BEST/1, and the Workload Estimator, as well as some of the performance metrics to assist you in planning for future system requirements.

2.2.1 Performance Management/400 (PM/400)

PM/400 is a set of automated tools that collect performance data on your iSeries server. It also provides you with reports that summarize system performance, making it easy to track the utilization of system resources. It helps to determine the workload growth and predict when a system upgrade may be needed. In addition, the data can be used in conjunction with the IBM Workload Estimator to help estimate your future needs.

Use Performance Management/400 and the BEST/1 capacity planning tools to plan for future resource demands on system performance. PM/400 allows you to easily submit your utilization and growth data to the Workload Estimator. Use the Workload Estimator to size your next upgrade.

What PM/400 is

PM/400 is an integrated OS/400 function that customers who are under processor warranty or on an IBM maintenance agreement can activate for no additional charge. In return, you receive capacity and performance analysis graphs useful in planning for and managing system growth and performance.

For a full description of PM/400 services and features, see the PM/400e Web site at: http://www.ibm.com/servers/eserver/iseries/pm400/

How PM/400 works

Once PM/400 starts, it collects the system performance data on your iSeries server and downloads the data once a week to IBM for processing. The overhead generated by running PM/400 on your system is minimal.

How PM/400 data is transmitted

PM/400 data can be transferred via modem or via TCP/IP using Electronic Service Agent.

OS/400 Version 5 Release 1 offers an alternative approach to transmit data via TCP/IP. It requires installing the Electronic Service Agent, which is a no charge IBM Licensed Program Offering. The Electronic Service Agent is preloaded on all new hardware shipments with Version 5 Release 1. All customers receive it as part of their software upgrade media.

This method for the transmission of PM/400e performance data can use the integrated asynchronous modem (or other supported modems) and the AT&T Global Network (provided at no charge for PM/400 data transmission). Starting in V5R1, additional connectivity options were added using a virtual private network (VPN) connection over the Internet. This includes using an existing dial-up ISP account (if you already have one) or several direct connect scenarios if your iSeries has Internet access and a globally routable Internet address.

OS/400 Version 4 Release 5 users may also use this support by ordering the IBM Electronic Service Agent CD (SK3T-5141). The PTF for enabling this support is SF64660. Service Agent is designed to be installed and setup by the customer.

For additional information on Electronic Service Agent, V4R5, refer to the Electronic Service Agent User Guide, Release 4.5 on the Electronic Service Agent public site at: `http://publib.boulder.ibm.com/as400_sd/sdsadoc.html`

For details, refer to the Tech Studio article on Electronic Support Access at `http://www.iseries.ibm.com/tstudio/planning/esa/esa.htm` or contact your local IBM Service organization.

For PM/400 FAQs regarding transmission, including using Universal Connection over TCP/IP, refer to:
`http://www.ibm.com/eserver/iseries/pm400/existing_customers/transmission.htm`

The type of information that is returned

PM/400 reports provide information on system resource utilization. There are over 26 different reports available. Figure 2-1 shows an example of an Executive Summary report. Figure 2-2 on page 23 shows an example of a Management Summary report. You can find examples of other reports at the PM/400 Web site at: `http://www.ibm.com/servers/eserver/iseries/pm400/`

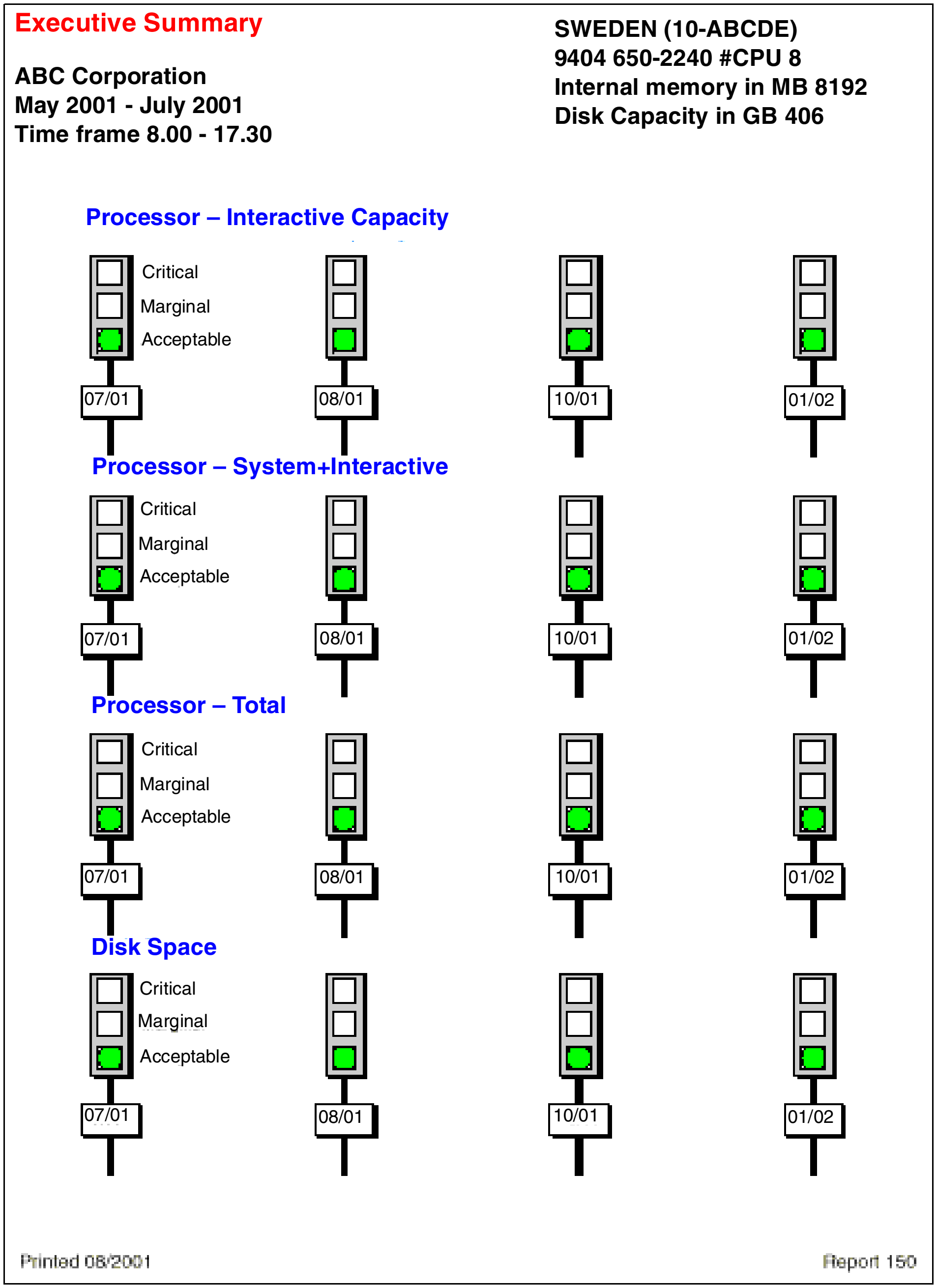


Figure 2-1 PM/400 Executive Summary report

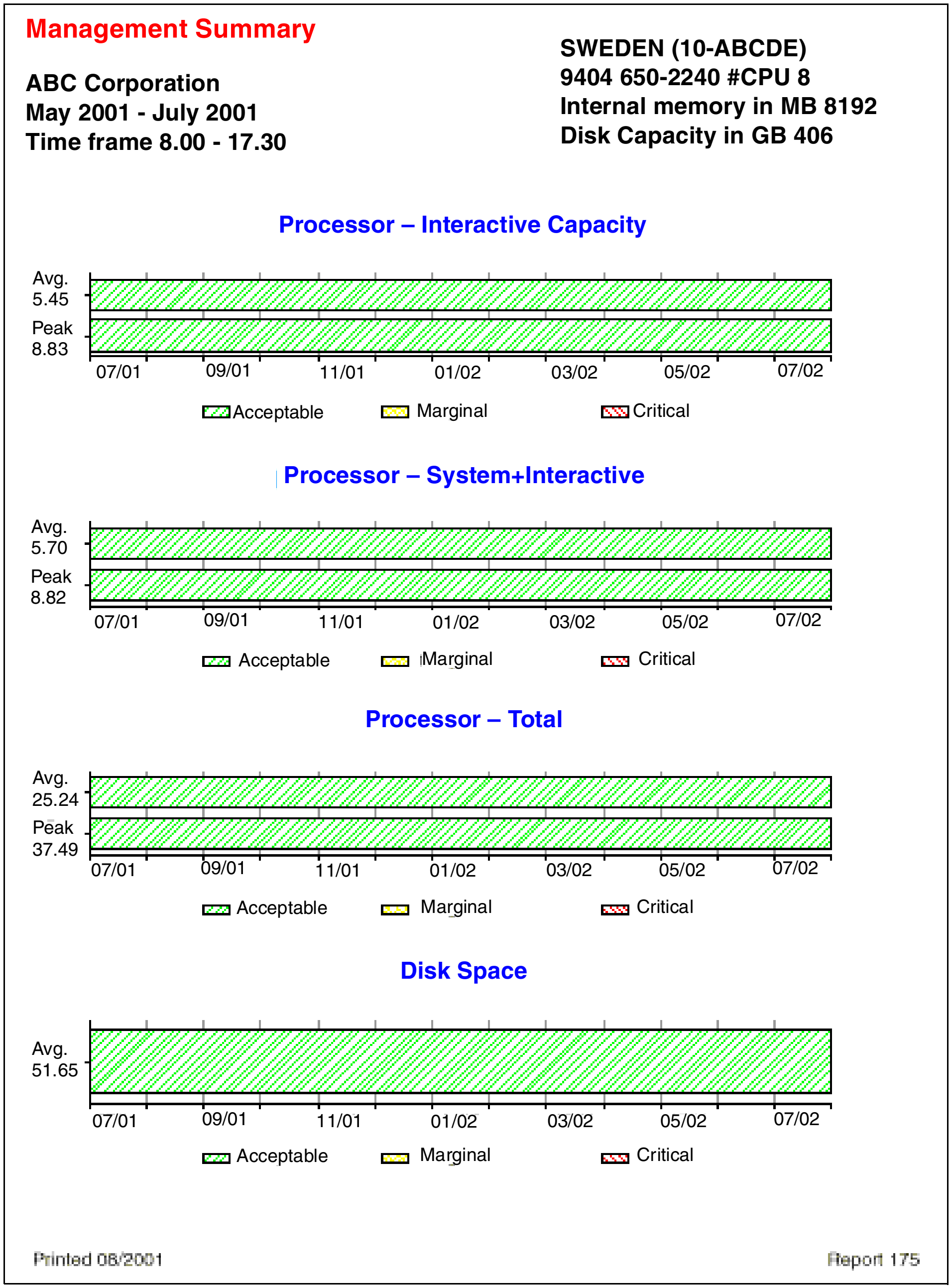


Figure 2-2 PM/400 Management Summary report

Previously, trends and projections were calculated based on historical processor utilization percent. While percent utilized is still shown for the current configuration, the relative Commercial Processing Workload (CPW) is now used for all historical and predicted data. When you use CPW to plot workload demand, you can see an accurate reflection of your workload changes even if a processor upgrade has taken place.

Current iSeries server models have a CPW value. This value represents the relative amount of processing that can be performed by that model. Utilization data is now stored in the PM/400e files in terms of this normalized CPW. This means that reports that show historical data and that use this data to predict future trends will more accurately reflect the workload demand across system upgrades. This is important for customers who regularly upgrade their systems.

These changes are particularly important for users who have implemented logical partitioning (LPAR). Users of LPAR can and do readily change the amount of processing resource allocated to each partition. To properly see the amount of resource being used at a partition level in such a volatile environment, it is essential to use a measure that records the amount of processing power being used across such changes rather than simply the percentage used.

Historical trend reports now reflect the relative amount of CPW used over time, regardless of the iSeries model at any particular time, and predictions are made based on the model of the iSeries currently installed. Predictions for a partition in an LPAR environment are based on the amount of processing power assigned to that partition at the last reporting interval (the default is 15 minutes).

Using PM/400 with IBM Workload Estimator

Workload Estimator and PM/400 have been enhanced to work with one another. Through a Web-based application, the enhancements help customers size upgrades to their existing system based on PM/400 reported utilization, performance, and growth data. As an additional option, upgrade sizings can also include capacity for adding Domino, Java, WebSphere, etc., but they are not a requirement for using the sizer. This assists all customers in planning for future system requirements based on existing utilization data coming from their own system.

When viewing the PM/400 graphs on the Web, you see the Size my next upgrade button. When you click it, all your PM/400 collected data passes right into the Workload Estimator.

With the flexibility to adjust growth rates and time horizons, the output includes an iSeries or AS/400 (170, 7xx, 250, 270, 8xx only) summary-level upgrade recommendations for consideration. The *Selected System* recommendation indicates that your system should be large enough to handle your workload today. The *Consider Growth* recommendation indicates that your system is large enough to handle your workload in the future (as specified in Months to Grow and Growth parameters in the Advanced Growth Options applet).

You can also select between a system on the "upgrade path" or select any replacement system that can handle the projected workloads. The output includes a suggested upgrade for your processor, processor features, memory, disk arms, and disk capacity. Sizings can also be "saved" and later "restored" for later use...either by yourself or for e-mailing to another person for review. This application does not support an LPAR or Processors on Demand environments at this time.

You should not view the output as a capacity planner or modeler; it is a *sizer* that suggests the next system upgrade for consideration. Also you should always review the output with an IBM Representative or IBM Business Partner.

2.2.2 BEST/1

BEST/1 is the iSeries server tool provided to:

- Model current system performance
- Predict hardware upgrades needed to meet current performance objectives
- Predict future hardware needs based on business growth projections
- Predict performance of additional or changed applications

Capacity planning is an ongoing process used to determine current system performance and future data processing needs. In its simplest form, using BEST/1 involves a five-step process:

1. Collect the performance data with the Collection Services part of Operations Navigator.
2. Create the performance data from the data collection with the CRTPFRDTA command.
3. Create a model from the performance data.
4. Analyze the BEST/1 model.
5. Perform a “What-if...?” analysis and continue the analysis process until you are satisfied with the results.

BEST/1 is an analytic modeling tool that provides expert evaluations and configurations. As such, the results depend entirely on the accuracy of the input, the analytic model, and the rule base for evaluations and configurations.

As with most capacity planning, some conditions or assumptions are present. The following list highlights the major assumptions under which BEST/1 performs its modeling work:

- Data is collected on a well-tuned, stable system.
- Data collection is done on a system in which all pool sizes and activity levels are set to reasonable values, secondary paging (thrashing) does not occur, and if excess storage is present, working set sizes are appropriate for the activity level of the pool.
  - When data collection is done, the time slices are large enough to minimize active-to-ineligible transitions (shown as A-I in the Performance Tools Transaction Report Interactive Jobs output).
  - The workload is reasonably homogeneous. That is, one particular program within a workload is not causing a resource limitation.
  - The workload is *steady-state*. This means that the number of interactive users and batch tasks is relatively constant.
  - The proportion of a workload's I/Os to an auxiliary storage pool (ASP) is based on the proportion of total I/Os handled by the disks on that ASP. Read and write operations are distributed on the same basis. There are no severe resource limitations (bottlenecks) in the system. That is, none of the primary devices (processing unit, disk, and memory) are excessively used.
  - All jobs specified as active are performing work.
  - The average service time of each disk request is the same. All disk requests are spread evenly across all disk drives in each ASP.
  - The activity on the local workstation IOPs is spread evenly across all IOPs, and the utilization of each IOP is the same. The service time and response time depend on the application.
  - All controllers are equally distributed across all communications lines for LAN and WAN.
  - All local, LAN, and WAN controllers have the same service time, respectively.

As the system approaches saturation (high utilization) of one or more resources, the accuracy of the analytic model for response time prediction diminishes. An analytic model is best used to predict when a resource will become saturated. Knowing when a resource will become saturated allows you to plan your workloads and hardware upgrades.

Restriction: BEST/1 does not model threads in the manner that they are used by WebSphere Application Server and Java.

The following example explains how to perform each of the following steps. It also includes a sample of the display that appears in each step.

1. Collect the data.
2. Create performance data.
3. Create a model.

   Pay attention to assigning the jobs to a workload.

4. Analyze the model.
5. Play with the workload.
   - Growth in workload
   - Different configurations (CPU/memory)

Creating a model

You should have at least one collection of performance data available before you start to create a model. Because we had multiple data members to choose from, we chose the one that represents a relatively heavy workload. Here are the steps we followed:

1. Enter the STRBEST command. Then the BEST/1 for the AS/400 menu appears as shown in Figure 2-3.

```
                            BEST/1 for the AS/400

 Select one of the following:

     1. Work with BEST/1 models

     5. Create BEST/1 model from performance data

    10. Work with results

    50. General information and tutorial

    60. More BEST/1 options





 Selection or command
 ===> 1

 F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F21=Advanced user level
```

Figure 2-3 Beginning the creation of a model

2. Select option 1 to start creating a model and press Enter. The Work with BEST/1 Models display (Figure 2-4) appears. We chose this option because it is the only one that leads to the display shown in Figure 2-7 on page 32.

3. Enter a name and a meaningful description for the model you are creating, because you cannot change the name or description after the model is generated. Press Enter.

```
                         Work with BEST/1 Models

Library . . . . .   QGPL         Name

Type options, press Enter.
  1=Create   3=Copy   4=Delete   5=Work with   6=Print   7=Rename

Opt  Model       Text                                    Date      Time
 1   YOURMODEL   GIVE A MEANINGFUL NAME
     PETRIDIDIT                                          10/26/01  11:15:21
     SAFEMODEL   Descriptive sample text                 10/25/01  17:26:07
     ACOWASHERE  Sample model                            10/25/01  17:06:44
     BASEMODEL   Descriptive model text here             10/25/01  14:23:54





                                                                     Bottom
Command
===>
F3=Exit   F4=Prompt   F5=Refresh         F9=Retrieve   F12=Cancel
F15=Sort by model     F16=Sort by text   F19=Sort by date and time
```

Figure 2-4 Naming a model

4. On the display shown in Figure 2-5, select the performance data member on which to create the model. You can press F4 to choose from a list of members in a library.

```
                  Create BEST/1 Model from Performance Data

Model . . . . . . . . . . . . . . . :   YOURMODEL

Type choices, press Enter.  Use *SLTHOUR to select an hour-long time period or
  use *SLTITV to select select first and last interval of a one to two hour
  time period.  The time period selected should be representative of your peak
  processing activity.

Text . . . . . . . . . . . . . .   A DESCRIPTIVE TEXT HERE

Performance data:
  Member . . . . . . . . . . . .   TAKEONE     Name, F4 for list
  Library  . . . . . . . . . . .   FROMLIB     Name

Start time . . . . . . . . . . .   *SLTITV     Time, *FIRST, *SLTHOUR, *SLTITV
Start date . . . . . . . . . . .   *FIRST      Date, *FIRST

Stop time  . . . . . . . . . . .   *LAST       Time, *LAST
Stop date  . . . . . . . . . . .   *LAST       Date, *LAST


F3=Exit   F4=Prompt   F12=Cancel
```

Figure 2-5   Selecting the library and member

5. On the Select Time Interval display (Figure 2-6), specify the first and the last interval to be included in the model. Choose data that contains a busy system but not data where the system is suffering from a program loop.

```
                              Select Time Interval

Library  . . . . . . . :   FROMLIB       Performance member . . :   TAKEONE

Type option, press Enter.  Select first and last interval.
  1=Select

                            ----Transaction----   --CPU Util---   I/Os per Sec
Opt   Date       Time          Count   Rsp Time   Total   Inter   Sync   Async
      10/25/01   09:15:00        104         .0       2       0      9       2
      10/25/01   09:20:00         69         .0       6       0     57      18
      10/25/01   09:25:00          0         .0      51       0    181     175
 1    10/25/01   09:30:00          0         .0      91       0    339     300
      10/25/01   09:35:00          0         .0      99       0    277     202
      10/25/01   09:40:00          2         .0      90       0    360     248
      10/25/01   09:45:00         62         .0      94       0    344     234
      10/25/01   09:50:01          0         .0      99       0    274     197
      10/25/01   09:55:00          0         .0      75       0    295     207
 1    10/25/01   10:00:00          0         .0      95       0    346     223
      10/25/01   10:05:00          5         .1       4       0     10       8
                                                                       More...
F3=Exit   F12=Cancel   F15=Sort by interval        F16=Sort by count
F17=Sort by rsp time   F18=Sort by total CPU util  F19=Sort by total I/Os
```

Figure 2-6 Selecting the time interval

6. During collecting the performance data, we kept track of the amount of orders generated and know exactly how many orders were generated between 09:30 and 10:00. We need the number of actual orders generated when predicting the effects of workload growth that is shown in Figure 2-20 on page 42. In this case, we know that we generated 800 orders in 30 minutes.

7. On the Classify Jobs display (Figure 2-7), start the process of creating a model that only includes the data your are interested in when you start increasing the workload. The actual modeling is shown in “Modeling the workload growth” on page 37. If you do not see the display shown in Figure 2-7, start the model creation again. Make sure you start it as shown in Figure 2-3 on page 28.

```
                      Classify Jobs

Select one of the following:

     1. Use default job classification
     2. Classify jobs into workloads
     3. Use existing job classifications





Selection
     2

F3=Exit   F12=Cancel
```

Figure 2-7 Classifying jobs into workloads

8. On the Specify Job Classification Category display (Figure 2-8), specify that you will classify jobs to a workload on a subsystem basis by selecting option **6** (Subsystem).

```
              Specify Job Classification Category

Type choice, press Enter.

  Category . . . . . . . . . . . .   6           1=User ID
                                                 2=Job type
                                                 3=Job name
                                                 4=Account code
                                                 5=Job number
                                                 6=Subsystem
                                                 7=Pool
                                                 8=Control unit
                                                 9=Comm line
                                                 10=Functional area




F3=Exit   F12=Cancel
```

Figure 2-8 Classifying jobs by subsystem

9. The the Edit Job Classification display (Figure 2-9) appears. On this display, press F9 to receive the values from the performance data.

```
                             Edit Job Classifications

 Enter workload names and category values which are assigned to each workload,
   press Enter.  Jobs with unassigned values become part of workload QDEFAULT.

 Workload    Subsystem      Workload    Subsystem      Workload    Subsystem
 ________    _________      ________    __________     _________   __________
 ________    _________      ________    __________     _________   __________
 ________    _________      ________    __________     _________   __________
 ________    _________      ________    __________     _________   __________
 ________    _________      ________    __________     _________   __________
 ________    _________      ________    __________     _________   __________


                                                                        More...
 F3=Exit   F9=Display values from data   F12=Cancel
 To display values from performance data, press F9
```

Figure 2-9 Getting values from the measured data

In this example, we include the work performed in subsystems QEJBSBS and QSYSWRK in the workload that will be named WASANDJAVA (Figure 2-10) later in this exercise. We also pick up the work that is not done in a subsystem that represents the operating system and Licensed Internal Code programs. Then press Enter.

```
                           Assign Jobs to Workloads

Workload . . . . . . . . . . . . . . . . . .   WASANDJAVA

Type options, press Enter.  Unassigned jobs become part of workload QDEFAULT.
  1=Assign to above workload   2=Unassign

                                  Number of        CPU         I/O
Opt  Workload     Subsystem   Transactions     Seconds     Count
 1                [redacted]             0      12.882     14698
                  QBATCH                0        .478       215
                  QCMN                  0        .000         0
 1                QEJBSBS               0    2912.844      8757
                  QHTTPSVR              0      16.507         3
                  QINTER               64        .396       257
                  QSERVER               0        .002         6
 1                QSYSWRK               0     872.728   1146196
                  QUSRWRK               0        .000         0


                                                                       Bottom
F3=Exit   F12=Cancel       F15=Sort by workload      F16=Sort by subsystem
F17=Sort by transactions   F18=Sort by CPU seconds   F19=Sort by I/O count
```

Figure 2-10   Assigning subsystems to a workload

10. You see the Specify Paging Behaviors display (Figure 2-11). Accept the default values.

```
                         Specify Paging Behaviors

 Type choices, press Enter.

               Paging Behavior
 Workload       (F4 for list)
 QDEFAULT         *GENERIC
 WASANDJAVA       *GENERIC







                                                                  Bottom
 F3=Exit   F4=Prompt   F12=Cancel
```

Figure 2-11 Specifying the paging behavior

11. On the Define Non-Interactive Transactions display (Figure 2-12), specify the metrics used to represent non-interactive transactions. Because there are no hooks in our application indicating the transaction start and end boundaries, we assume that the amount of logical I/O generated equals the 800 transactions that we know took place when collecting the performance data.

```
                     Define Non-Interactive Transactions

Job classification category  . . . . . . :   Subsystem


Type choices, press Enter.

              ---Activity Counted as Transaction---     Total Transactions
Workload          Type                 Quantity         when Type = *NONE
QDEFAULT          *LGLIO                 100.0                        0
WASANDJAVA        *LGLIO                 100.0                        0



                                                                  Bottom
Type:  *LGLIO, *CMNIO, *CPUSEC, *PRINT, *NONE

F3=Exit   F12=Cancel
```

Figure 2-12 Defining a non-interactive transaction

12. After you define a non-interactive transaction, press Enter. Then you see the Save Job Classification Member display (Figure 2-13). Save the workload member.

```
                       Save Job Classification Member

 Change values if desired, press Enter.

   Member . . . . . . . . . . . .   WASANDJAVA     Name
     Library  . . . . . . . . . .     QGPL         Name

   Text . . . . . . . . . . . . .  JAVA, WebSphere and System work load

   Replace  . . . . . . . . . . .   Y              Y=Yes, N=No



 F12=Cancel
```

Figure 2-13 Saving the job classification member

13. You see the Confirm Creation of BEST/1 Model display (Figure 2-14). Confirm the creation of the model. The actual creation of the model is done in a batch job that is submitted on this display.

```
                       Confirm Creation of BEST/1 Model

 Type choices, press Enter.

   Model  . . . . . . . . . . . .   YOURMODEL      Name
     Library  . . . . . . . . . .     QGPL         Name

   Text . . . . . . . . . . . . .   A descriptive text here

   Replace  . . . . . . . . . . .   N              Y=Yes, N=No

   Job name . . . . . . . . . . .   CRTBESTMDL     Name, *JOBD
   Job description  . . . . . . .  QPFRJOBD       Name, *NONE, *USRPRF
     Library  . . . . . . . . . .     QPFR         Name, *LIBL, *CURLIB



 F12=Cancel
 Member WASANDJAVA has been saved
```

Figure 2-14 Submitting the creation of model

Modeling the workload growth

Use the BEST/1 tool to predict how your existing hardware would perform if the workload grew. You can also use the BEST/1 to simulate how the same workload would run on a different hardware. In both cases, the steps to follow are the same:

1. Collect performance data.
2. Create performance data.
3. Create a model.
4. Use the model to simulate the changes.

Since we already created the model in the previous section, follow the steps as outlined here:

1. Enter the STRBEST command. Then choose option `1` to work with models. This leads you to the Work with BEST/1 Models display (Figure 2-15). Choose option `5` to work with a specific model and press Enter. This causes the BEST/1 software to read the model to its workspace.

```
                         Work with BEST/1 Models

Library . . . . .   QGPL        Name

Type options, press Enter.
  1=Create   3=Copy   4=Delete   5=Work with   6=Print   7=Rename

Opt  Model       Text                           Date      Time

 5   YOURMODEL   Descriptive text here         10/25/01  14:23:54







                                                                      Bottom
Command
===>
F3=Exit   F4=Prompt   F5=Refresh       F9=Retrieve   F12=Cancel
F15=Sort by model     F16=Sort by text   F19=Sort by date and time
```

Figure 2-15 Starting to work with a model

2. After the model is read, the display shown in Figure 2-16 appears. Choose option 5 to analyze the model.

```
                          Work with BEST/1 Model

Performance data . . . :   QMPGDATA (TAKEONE)
Model/Text . . . . . . :   YOURMODEL   A descriptive text here


Select one of the following:

     1. Work with workloads
     2. Specify objectives and active jobs

     5. Analyze current model
     6. Analyze current model and give recommendations
     7. Specify workload growth and analyze model

    10. Configuration menu
    11. Work with results

                                                                     More...
Selection or command
===> 5
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F15=Save current model
F17=Analyze using ANZBESTMDL       F22=Calibrate model   F24=More keys
Model YOURMODEL has been read
```

Figure 2-16 Starting to analyze a model

3. You now see the Display Analysis Summary (Figure 2-17). Verify that the model resembles the load that was on the system during the time the performance data was collected. Then press Enter.

```
                          Display Analysis Summary

CPU Model / release level  . . . . . :   23F5        V5R1M0
Main Storage . . . . . . . . . . . . :   4096       MB

                                         Quantity    Predicted Util
CPU  . . . . . . . . . . . . . . . . :     2/2           92.4
Database . . . . . . . . . . . . . . :                   92.4
Disk IOPs  . . . . . . . . . . . . . :      0              .0
Disk ctls  . . . . . . . . . . . . . :      0              .0
Disk arms  . . . . . . . . . . . . . :     12             3.3
                                                                  More...
                                                Interactive   Non-interactive
CPU utilization % . . . . . . . . . . . . . :   .0             92.4
Transactions per Hour . . . . . . . . . . . :   110            58135
Local response time (seconds) . . . . . . . :   .0             .8
LAN response time (seconds) . . . . . . . . :   .5             2.0
WAN response time (seconds) . . . . . . . . :   .0             .0


Performance estimates -- Press help to see disclaimer.
F3=Exit   F10=Re-analyze   F11=Measured and predicted comparison   F12=Cancel
F15=Configuration menu     F17=Analyze multiple points   F24=More keys
```

Figure 2-17 Display Analysis Summary

4. You return to the Work with BEST/1 Model display. Choose option 7 (Specify workload growth and analyze model) to start exploring what happens if the workload grows. This is shown in Figure 2-18.

```
                              Work with BEST/1 Model

 Performance data . . . :   QMPGDATA (TAKEONE)
 Model/Text . . . . . . :   YOURMODEL   A descriptive text here


 Select one of the following:

      1. Work with workloads
      2. Specify objectives and active jobs

      5. Analyze current model
      6. Analyze current model and give recommendations
      7. Specify workload growth and analyze model

     10. Configuration menu
     11. Work with results

                                                                       More...
 Selection or command
 ===> 7
 F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F15=Save current model
 F17=Analyze using ANZBESTMDL        F22=Calibrate model   F24=More keys
```

Figure 2-18   Starting to grow the workload

5. When you specify the workload growth, you can grow the workload with a specified percentage for one growth period or multiple growth periods. You can also specify separate growth percentages for various growth periods as shown in Figure 2-19.

```
                     Specify Growth of Workload Activity

 Type information, press Enter to analyze model.
   Determine new configuration . . . . . . . . . .   Y    Y=Yes, N=No, K=Keep
                                                            new configuration
   Periods to analyze  . . . . . . . . . . . . . .   13   1 - 10

   Period 1  . . . . . . . . . . . . . .   Period 1      Name
   Period 2  . . . . . . . . . . . . . .   Period 2      Name
   Period 3  . . . . . . . . . . . . . .   Period 3      Name
   Period 4  . . . . . . . . . . . . . .   Period 4      Name
   Period 5  . . . . . . . . . . . . . .   Period 5      Name

             ------Percent Change in Workload Activity-------
 Workload    Period 1  Period 2  Period 3  Period 4  Period 5
 *ALL             .0      10.0      10.0      10.0      10.0




                                                                           Bottom
 F3=Exit   F11=Specify growth by workload   F12=Cancel
 F13=Display periods 6 to 10   F17=Analyze using ANZBESTMDL
```

Figure 2-19 Specifying the amount of growth per period

The results appear as show in the Display Analysis Summary in Figure 2-20. Pay attention to Period 1, where the workload has not grown, and especially to the number of Non-Interactive transactions, which is 58135 in our case.

```
                           Display Analysis Summary

          CPU         Stor  ------CPU------     DB   -Disk IOPs-  -Disk Arms--
Period    Model       (MB)    Nbr       Util    Util  Nbr    Util    Nbr    Util
Period 1  270 23F5    4096    2/2       92.4    92.4    0      .0     12     3.3
Period 2  820 23B8    4096    4/4       73.4    73.4    0      .0     12     3.6
Period 3  820 23B8    4096    4/4       80.8    80.8    0      .0     12     4.0
Period 4  820 23B8    4096    4/4       88.8    88.8    0      .0     12     4.4
Period 5  820 23B8    4096    4/4       97.7    97.7    0      .0     12     4.8
Period 6  820 24B8    4096    4/4       90.4    90.4    0      .0     12     5.3
Period 7  820 24B8    4096    4/4       99.5    99.5    0      .0     12     5.8
Period 8  830 23D8    4096    8/8       54.7    54.7    0      .0     12     6.4
Period 9  830 23D8    4096    8/8       60.2    60.2    0      .0     12     7.0
Period10  830 23D8    4096    8/8       66.2    66.2    0      .0     12     7.7


          --Non-Inter Rsp Time--  -----Non-Inter------  Release
Period     Local     LAN     WAN  CPU Util    Trans/Hr   Level
Period 1      .8     2.0      .0     92.4        58135  V5R1M0
Period 2      .2     2.7      .0     73.4        63949  V5R1M0
Period 3      .3     2.8      .0     80.7        70344  V5R1M0
Period 4      .4     2.8      .0     88.8        77378  V5R1M0
Period 5     1.8     2.8      .0     97.7        85116  V5R1M0
Period 6      .4     2.3      .0     90.4        93627  V5R1M0
Period 7     6.6     2.3      .0     99.5       102990  V5R1M0
Period 8      .2     2.3      .0     54.7       113289  V5R1M0
Period 9      .2     2.3      .0     60.2       124618  V5R1M0
Period10      .2     2.3      .0     66.2       137080  V5R1M0




                                                                          Bottom
F3=Exit   F10=Re-analyze   F11=Alternative view          F12=Cancel
F15=Configuration menu     F17=Analyze multiple points   F24=More keys
BEST/1 CPU model 270 23F5 refers to IBM CPU model 270 2434-1520
BEST/1 CPU model 820 23B8 refers to IBM CPU model 820 2398-1521
BEST/1 CPU model 820 24B8 refers to IBM CPU model 820 2438-1521
BEST/1 CPU model 830 23D8 refers to IBM CPU model 830 2403-1531
```

Figure 2-20 Display Analysis Summary: The results

Remember where we chose the intervals to be used within the model, and that during those intervals, we created 800 orders as shown in Figure 2-12 on page 35? What do these two have in common? The answer is that the number 58,135 is equivalent to the 800 orders that were generated. When you divide 58,135 with 800, you get the value 72.66875 that we use to divide the rest of the numbers with in the Trns/Hr column.

Table 2-1 Deciphering the real order lines from the BEST/1 transactions

BEST/1 orders generated	Real order lines generated in 30 minutes	Real order lines generated per hour
63949 / 72.66875	880	1760
70344 / 72.66875	968	1936
77378 / 72.66875	1064	2128
85116 / 72.66875	1171	2342
93627 / 72.66875	1288	2576
102990 / 72.66875	1417	2834
113289 / 72.66875	1558	3116
124618 / 72.66875	1714	3428
137080 / 72.66875	1886	3772

Modeling the same workload on a different CPU model

Sometimes there comes a situation where you are not planning for growth, but instead are trying to find the hardware that would support the workload. This scenario is described in the following steps:

1. Start modeling with either the STBEST command or by navigating through the menus. In both cases, you see the Work with BEST/1 menu (Figure 2-18 on page 40). Verify that you are at the Advanced user level, and choose option `10`. Then you see the configuration menu (Figure 2-21).

```
                              Configuration

CPU Model . . . . . . . . :   23F5      Comm IOPs . . . . . . . . :   0
Processors  . . . . . . . :   2/2       LAN lines . . . . . . . . :   2
Main stor (MB)  . . . . . :   4096      WAN lines . . . . . . . . :   0
Main stor pools . . . . . :   6
Disk IOPs . . . . . . . . :   0         Multifunction IOPs  . . . :   2
Disk ctls . . . . . . . . :   0         Disk IOAs . . . . . . . . :   2
Disk arms . . . . . . . . :   12        Comm IOAs . . . . . . . . :   2
ASPs  . . . . . . . . . . :   1         IPCS IOAs . . . . . . . . :   0

Select one of the following:

     1. Change CPU and other resource values
     2. Work with disk resources
     3. Edit ASPs
     4. Edit main storage pools
     5. Work with communications resources

Selection or command
===>
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Check configuration
F17=Correct configuration           F24=More keys
```

Figure 2-21 The BEST/1 Configuration display

2. Choose option 1 to predict the affect of changing the CPU and the amount of main storage on the workload. You may use F4 to prompt for a list of available values or simply enter the values as shown in Figure 2-22.

```
                        Change CPU and Other Resource Values

Model/Text:   YOURMODEL   A descriptive text here

Type choices, press Enter.

  System unit . . . . . . . . . . . .   9406      9402, 9404, 9406
  CPU Model . . . . . . . . . . . . .   24B8      F4 for list
  System storage (MB) . . . . . . . .   8192      F4 for list
  Release level . . . . . . . . . . .   V5R1M0    F4 for list
  Active processors . . . . . . . . .   4
  Unavailable PCI slots . . . . . . .   0



F3=Exit   F4=Prompt   F9=Specify other logical partitions   F12=Cancel
```

Figure 2-22   Specifying the CPU model and Main Storage size for the model

3. Because you change the amount of main storage within the model, you must portion the additional memory between the memory pools. The easiest way to do this is to press F17. This divides the memory evenly between the storage pools in the model. If you decide to specify the new pool sizes manually, you must make sure that the total amount of memory in the pools matches the amount of memory available before you can continue using this model. We recommend that you use the Re-scaling function. The Edit Main Storage Pools display appears (Figure 2-23).

```
                          Edit Main Storage Pools

Model/Text:   YOURMODEL   A descriptive text here

Main storage size  . . . . . . . . . :       8192   MB
                                          8388608   KB

Type changes, press Enter.  To delete a pool, set size to *NOSTG.

                          Activity         Size
Pool ID     Pool Name       Level           (KB)
    1                          0          479188
    2                        240         1005912
    3                        150          614400
    4                          5           41940
    5                          9          209664
    6                        300         1843200



                                                                   Bottom
Activity level:  0=Unrestrained

F3=Exit   F6=Create   F12=Cancel   F17=Re-scale pool sizes

Sum of main storage pool sizes (4194304 KB) must equal system storage size
```

Figure 2-23   Re-scaling the memory pool sizes

4. After the new memory pool sizes are adjusted, you return to the main configuration display shown in Figure 2-24. On this display, you can both check that the configuration is valid and correct the configuration automatically, if needed. Use the F17 key to correct the configuration.

```
                                Configuration

CPU Model . . . . . . . . :   24B8      Comm IOPs . . . . . . . . :   0
Processors  . . . . . . . :   4/4       LAN lines . . . . . . . . :   2
Main stor (MB)  . . . . . :   8192      WAN lines . . . . . . . . :   0
Main stor pools . . . . . :    6
Disk IOPs . . . . . . . . :   0        Multifunction IOPs  . . . :   2
Disk ctls . . . . . . . . :   0         Disk IOAs . . . . . . . . :   2
Disk arms . . . . . . . . :   12        Comm IOAs . . . . . . . . :   2
ASPs  . . . . . . . . . . :   1         IPCS IOAs . . . . . . . . :   0

Select one of the following:

     1. Change CPU and other resource values
     2. Work with disk resources
     3. Edit ASPs
     4. Edit main storage pools
     5. Work with communications resources

Selection or command
===>
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Check configuration
F17=Correct configuration                  F24=More keys
```

Figure 2-24 Checking and correcting the configuration

5. Next you see the Configuration Changes display (Figure 2-25). Enter option 5 to analyze current model on the Work with BEST/1 Model display.

```
                          Configuration Changes

 The following changes have been made to your configuration:

 1 284E IOP(s) deleted
 1 284C IOP(s) created

                                                                    Bottom
 Press Enter to continue.
 F3=Exit   F12=Cancel
```

Figure 2-25 The configuration changes made by BEST/1

6. You see the Display Analysis Summary shown in Figure 2-26. Press the F11 key.

```
                          Display Analysis Summary

CPU Model / release level  . . . . . :   24B8         V5R1M0
Main Storage . . . . . . . . . . . . :   8192       MB

                                         Quantity    Predicted Util
CPU  . . . . . . . . . . . . . . . . :     4/4           56.1
Database . . . . . . . . . . . . . . :                   56.1
Disk IOPs  . . . . . . . . . . . . . :      0              .0
Disk ctls  . . . . . . . . . . . . . :      0              .0
Disk arms  . . . . . . . . . . . . . :     12             3.3
                                                                     More...
                                                Interactive   Non-interactive
CPU utilization % . . . . . . . . . . . . . :   .0              56.1
Transactions per Hour . . . . . . . . . . . :   110             58135
Local response time (seconds) . . . . . . . :   .0              .2
LAN response time (seconds) . . . . . . . . :   .5              2.3
WAN response time (seconds) . . . . . . . . :   .0              .0


Performance estimates -- Press help to see disclaimer.
F3=Exit   F10=Re-analyze   F11=Measured and predicted comparison   F12=Cancel
F15=Configuration menu     F17=Analyze multiple points   F24=More keys
BEST/1 CPU model 820 24B8 refers to IBM CPU model 820 2438-1521
```

Figure 2-26 Displaying the analysis summary with a new CPU

Now you can compare the measured values with the predicted values as shown in Figure 2-27.

```
                       Measured and Predicted Comparison

                                              Measured      Predicted
Total CPU util . . . . . . . . . . . . :          92.4           56.1
Database util  . . . . . . . . . . . . :          92.4           56.1
Disk IOP util  . . . . . . . . . . . . :            .0             .0
Disk arm util  . . . . . . . . . . . . :           3.2            3.3
Disk IOs per second  . . . . . . . . . :         569.1          563.7
Multifunction IOP util . . . . . . . . :           2.3            2.4
Disk IOA util  . . . . . . . . . . . . :           8.6            8.6
LAN IOA util . . . . . . . . . . . . . :                          .0
WAN IOA util . . . . . . . . . . . . . :                          .0
Integrated PC Server IOA util  . . . . :            .0             .0

Interactive:
  CPU util . . . . . . . . . . . . . . :            .0             .0
  Int rsp time (seconds) . . . . . . . :            .0             .0
  Transactions per hour  . . . . . . . :           110            110
Non-interactive thruput  . . . . . . . :         58119          58135

Performance estimates -- Press help to see disclaimer.
F3=Exit   F6=Print   F9=Work with spooled files   F12=Cancel
```

Figure 2-27 Comparing the measured data with the predicted model

2.2.3 Workload Estimator

Workload Estimator is helpful to both IBM and Business Partner sales and marketing people. It provides a comprehensive iSeries sizing tool for new and existing customers that are interested in deploying new emerging workloads standalone or in combination with their current workloads.

The recommendations created with this tool are for the most recent hardware. They project the model, CPU% utilization, memory, disk arms, and capacity.

It is easy to use with less than a dozen questions per application, with default values for the applications and system. You may specify workload type or types and workload details, as well as assess workload complexities before you configure the system. By making simple changes to the information already entered, you evaluate the sensitivity to variations in the workload of the proposed hardware.

A printed report with defaults, input parameters, and results is available.

Workload Estimator is not a capacity planning tool or a configurator, since capacity planning tools offer detail modeling from actual system performance data. Capacity planning tools also project the response time for specific configurations.

Workload Estimator is basically a quick method to create a system from scratch. It does not provide support for LPAR, journaling, response times, or locked resources.

Using Workload Estimator is straightforward. It requires a minimum amount of information and may be completed in a few steps as shown in this section. To launch the Workload Estimator, point your browser to:
`http://www-912.ibm.com/servlet/EstimatorServlet`

Workload type selection

On the Workload Selection page (Figure 2-28), you may specify multiple workloads. Remember that during this process, you must describe each workload separately during the later steps.

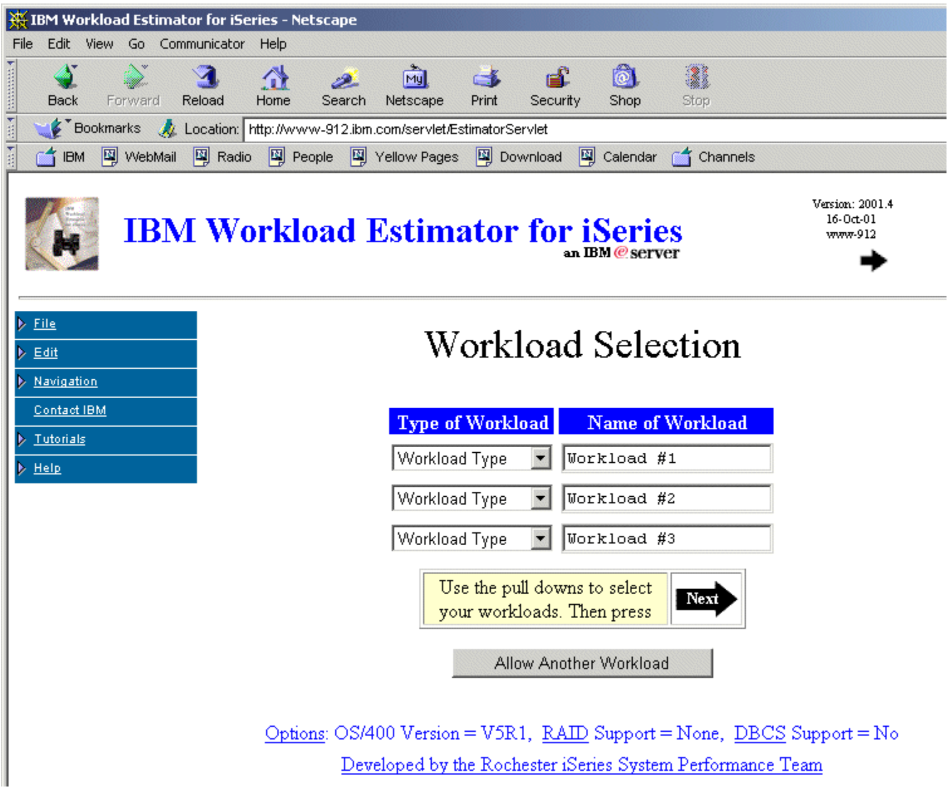


Figure 2-28 IBM Workload Estimator for iSeries

Workload detail entry

Figure 2-29 shows how you enter the specified workload volume and detail distribution. In this example, we only show the WebSphere workload details. Normally you must describe all workloads that you specified on the previous page.

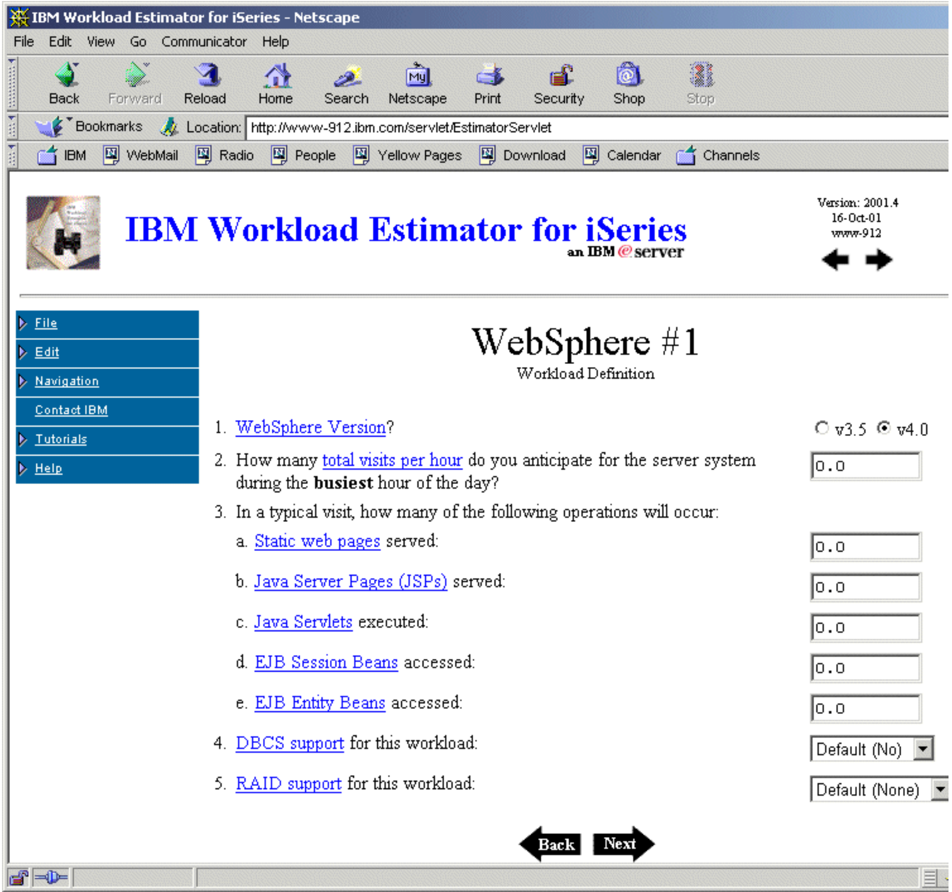


Figure 2-29 Specifying the WebSphere workload

Workload complexity

Figure 2-30 shows how you specify the complexity of the WebSphere workload.

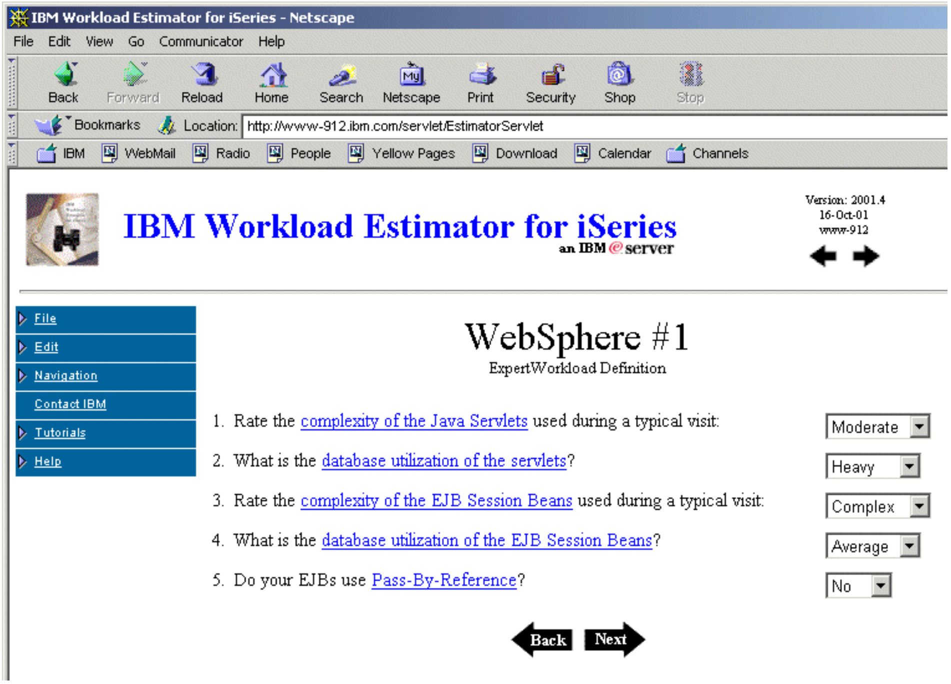


Figure 2-30 Specifying workload complexity

Proposed configuration

Figure 2-31 shows the proposed configuration that is needed to support the previously described workload. Remember, the more accurately you describe the application functions and the workload, the more accurately sized the iSeries configuration will be.

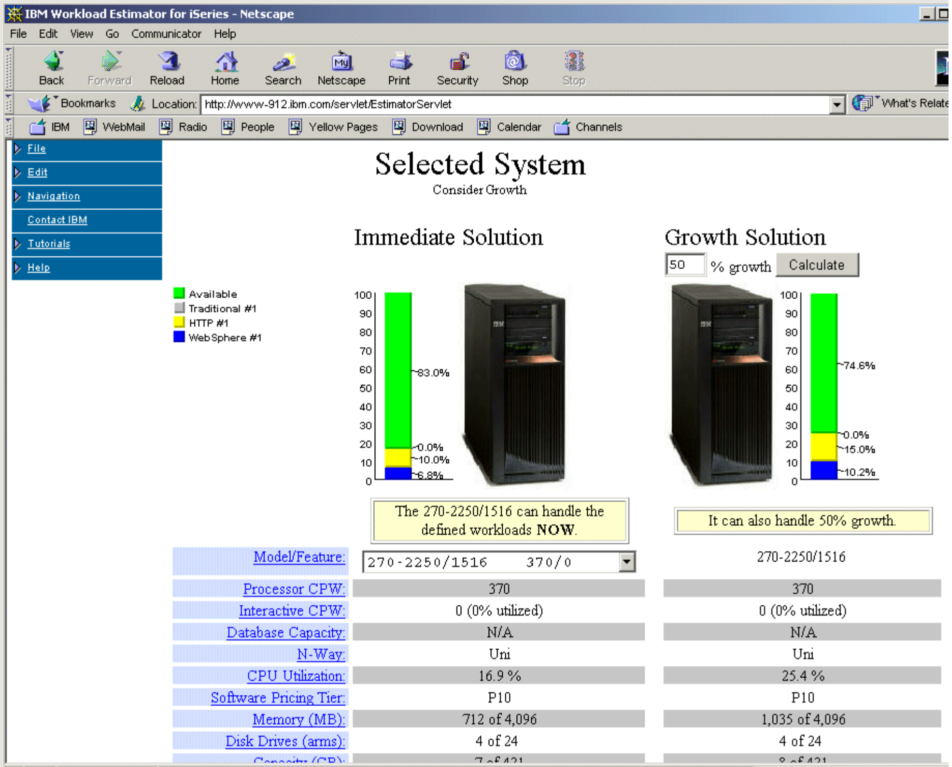


Figure 2-31 Estimated configuration

Sensitivity to variations

By making simple changes to the details already specified, for example workload complexity, or growth, the estimated configuration can be re-calculated to determine the affects of these changes.

Provided the limitations inherent in the specification and calculation process are appreciated, and the model is not pushed too far, then good estimated sizings will result.

Additional information

For more information, consult the *AS/400 Performance Capabilities Reference Version 4 Release 5*. See:

- Chapter 6 for information on Webserving Performance capacity planning
- Chapter 7 for general Java performance information
- Chapter 22 for information on Workload Estimator

2.3 Sizing

There are no easy answers to system sizing. The goal is to maximize performance while minimizing cost and complexity. An unconstrained system will provide optimal performance; however, this is usually not feasible in most business scenarios. Careful planning must be done to assure the best use of resources and provide adequate performance at a reasonable cost.

This section highlights the primary factors that are taken into account in the IBM sizing process. It also outlines the process to obtain a system sizing from IBM.

2.3.1 Methodology

To provide system sizing assistance for OneWorld, IBM has developed the IBM/J.D Edwards International Competency Center. This group is located on site at the J.D. Edward's headquarters, with other sizing-related offices in Philadelphia, London, and Tokyo. The Competency Center is responsible for running J.D. Edwards benchmarks. It also handles other sizing and tuning tests to help develop the tools that are used to size the J.D. Edwards workloads on IBM servers. The goal is to provide a system that performs well with a given workload, at a competitive cost.

The current methodology is to provide user-based sizing. The sizings take into account:

- The peak workload
- Active user counts
- Specific sizing-related testing
- Benchmark tests
- Input provided from business partners and clients from real-world implementations

Peak workload

To avoid poor performance, it is important to consider peak workloads when sizing a system. It would be unacceptable to have acceptable response times most of the time with long response times during the busiest time of the year or during any critical period. Use care in looking at past business history to determine when the peak is for each business.

For example, business A may receive 80% of their orders during the morning period between 9 a.m. and 11 a.m., with lower activity during other periods. Business B may be focused on month-end processes that require long, processor-intensive reporting jobs to be run during a few day period once per month. Business C may be using OneWorld Web interface for employee benefit enrollment and have to handle 50% or more of the employees during the last week of enrollment.

The key to any sizing is defining when the peak period is for a given business and then determining exactly what is happening during that peak. Is there a certain number of transactions that must take place within a certain time window? Is there a given number of employees that must be handled in a given time period?

Active users

IBM bases most sizings on users, although transaction rates can also be used. It is sometimes difficult to determine exactly how many users will actually use the system at a given time. Typically customers know how many licenses they will buy. Or they may know how many people are in each department. But the number of users that are actually logged on to a system isn't quite as straightforward. And of those users that are logged on to the system, it isn't always easy to determine how many are actively using system resources.

To standardize users, the IBM OneWorld sizing questionnaire refers to active users. These are defined as users that are clicking OK or using a function key or some other key that causes processing two to three times per minute. Users that are logged on to the system, but not actively using it are not considered in the sizing.

The ratio of named or licensed users to active users varies greatly. In some instances, almost all of the named users could be using the system at a given time. In other cases, OneWorld may be deployed to thousands of users around the world, but only a small percentage of those users would actively use the system at any given time. When more definite information is not available, a factor of 65% is used to determine active users. This means if there are 100 licensed users, the sizing would be done for 65 active users.

Other jobs, third-party applications

A standard IBM OneWorld sizing does not account for other, non-OneWorld workloads. These should be sized by means listed in the capacity planning section or by the third-party software provider. The requirements for these other applications should then be added to the requirements given for the OneWorld workloads to obtain the overall system requirements.

2.3.2 Benchmark process

To obtain sizing information for each platform, IBM conducts standard J.D. Edwards benchmarks. These benchmarks are standardized across the platform to provide a set amount of work in a given amount of time. J.D. Edwards currently has several types of benchmarking tests available. Some of these tests are user-based and some are batch/transaction-oriented.

Scripts

J.D. Edwards user-based benchmark scripts are designed to simulate real-world users. These scripts run through a variety of applications much as a real user would, doing lookups, pausing between entries, etc. Multiple applications also run concurrently to simulate an actual working environment where multiple users are using different applications.

The tests are not designed to measure maximum throughput of a system. Instead, they are used to provide a more realistic estimate of how many transactions a working OneWorld environment could run on a given system.

Applications run

Depending on the test being done, currently up to 18 applications are run during the user-based standard benchmark tests. The applications are split into three vertical types: manufacturing, distribution, and finance. The standard applications are listed in Table 2-2. Other applications are used for more specific testing.

Table 2-2 Standard applications

Finance	Distribution	Manufacturing
Work Order Entry	Inventory Adjustments	Voucher Entry
Bill of Materials Inquiry	Summary Inventory Availability	Supplier Inquiry
Work Order Materials issues	Purchase Order Entry	Journal Inquiry

Finance	Distribution	Manufacturing
Work Order Partial Completion	Purchase Order Receipts Match	Trial Balance
Supply/Demand Inquiry	Sales Order Entry	Address Book Name Search
	Customer Service Inquiry	
	Shipment Confirmation	
	Open Receipts Inquiry	
	Address Book Name Search	

Disk sizing

To accurately determine the size of a final, production database, detailed knowledge of business processes and environment specific information is needed. This includes number of items, years and type of history, etc. During the pre-sales process, many of the specifics are not yet known. Because of this, IBM has elected to provide a general planning estimate of disk space required, but the final amount of disk needed may vary.

When sizing disks, the amount of raw disk space, as well as the type of RAID protection and the percentage utilization should be taken into account. A standard iSeries sizing from IBM assumes disks are 70% full and plans for RAID5. If it is a heavy journaling environment, then we recommend that you create a second ASP with RAID1 protection and place journal receivers in that ASP.

Performance of a system can also vary depending on the number of disk arms available. Much like a SAN or disk farm, the iSeries spreads data over all available disks. This provides very fast disk I/O subsystem performance when there are plenty of disks. However, the converse is also true; if there are not enough disk arms to handle the I/O load, performance can suffer.

An online tool is available tat can help determine the appropriate number of disks for a given iSeries model. This tool, the Online Disk Arms Calculator (ODAC), is available at:
http://www-1.ibm.com/servers/eserver/iseries/perfmgmt/odac.htm

To determine the minimum number of disks for adequate OneWorld performance, the current recommendation is to follow the ODAC “light” workload. This should be suitable for most OneWorld environments. Some applications are more I/O intensive, so disk system performance should be monitored to assure performance is satisfactory.

Batch sizing

Many OneWorld reports and update processes run as background tasks, called Universal Batch Engines (UBEs). These batch processes are generally very processor intensive and result in relatively high CPU utilizations. These tasks usually process at a lower priority than the user workload and, therefore, should not affect the response time of the users significantly.

Since UBEs are processor intensive, running too many streams of UBEs concurrently can quickly consume available processor resources, resulting in poor performance. Rules of thumb, such as one UBE per processor, evolved with the previous versions. Current iSeries models have faster, more powerful processors, with multiple registers. OS/400 is a multi-tasking operating system, designed to handle multiple workloads concurrently. Because of this, in contrast to the older rules of thumb, multiple UBEs can be run per processor. Performance will vary depending on the actual UBE in question. But generally running multiple UBEs per processor will provide suitable performance and make the best use of available system resources.

There are cases when running multiple processes per processor is not advised. Since most UBEs process at the same priority, if the CPU is highly utilized, when running multiple UBEs per processor, they compete for constrained resources. As a result, the length of time for both UBEs to process will lengthen. In general, this degradation should be acceptable, but there are times when the processes need to complete in the shortest time possible. If these processes are mission critical, they should be run at a higher priority, or a processor should be allocated for each concurrent critical UBE stream. This helps to eliminate contention for processor resources.

A standard iSeries sizing from IBM permits multiple UBEs to be run per processor. If clients designate that there are critical batch processes that must run as fast as possible, the sizing will account for this by assuring that there are at least as many processors as there are concurrent critical UBE streams.

2.3.3 IBM sizing process

IBM has a process to assist Business Partners and clients with new pre-sales sizing needs. Sizings for upgrades should be handled through the capacity planning processes. IBM Global Services or other J.D. Edwards business partners can also assist with capacity planning on a fee basis.

To obtain a system sizing from IBM, follow these steps:

1. Contact your IBM Business Partner, IBM representative, or J.D. Edwards representative. If you do not know any of these contacts, you can submit a sizing questionnaire as explained in the following section. An appropriate contact will be made for you.
2. Fill out an IBM OneWorld sizing and planning questionnaire. This contains all of the information specific to your environment, such as number of users, desired architecture, and applications being used.
3. A sizing response document is generated by the IBM Sizing team and returned to the Business Partner, IBM representative, or J.D. Edwards representative, who then delivers it to the customer.

Remember that sizing is generally an iterative process. It may take several tries to determine the best solution for your environment.

Questionnaire

To obtain sizing from IBM, you can complete the OneWorld sizing questionnaire. You can find this questionnaire at the IBM/J.D. Edwards Alliance Web site at:
`http://www.ibm.com/erp/jdedwards`

The first page lists the directions for submitting the completed questionnaire to obtain a sizing.

Contacts for more information

- ICC Alliance Web site: `http://www.ibm.com/erp/jdedwards`
- Sizing center

  E-mail: `eSizings@us.ibm.com`
  Phone: 1-800-462-0222 or 770-835-6690
  Fax: 770-639-5245
- Sizing Web site: `http://www.ibm.com/erp/sizing`
- iSeries support Web site:
  `http://www.ibm.com/servers/eserver/iseries/service/erp/jdesupport.htm`
- EMEA Questionnaire submittal:
  - `JDEdwards_EMEA_Presales@uk.ibm.com`
  - `IBMJDE@uk.ibm.com`

3

Installation

This chapter provides a sample step-by-step approach for the basic installation of OneWorld Xe when using:

- An iSeries server with OS/400 V5R1 as an enterprise server
- The Integrated xSeries Server with Windows 2000 as a deployment server

You should use the information provided in this chapter in conjunction with the *Installation Guide (AS/400 Systems) Xe*, which is supplied with the OneWorld software. The *Installation Guide* is unique for each release of OneWorld due to the continuing evolution of the installation process. Where necessary, detailed supplemental information pertaining to the software installation is supplied in this chapter that is not currently available in the *Installation Guide (AS/400 Systems) Xe*.

OneWorld installation is a complex process, and as such, should be performed by a qualified OneWorld Installer by following the *Installation Guide (AS/400 Systems) Xe*.

This chapter was created from the perspective of giving you a general feel for the type of tasks that you will perform and the type of problems you may encounter during the execution of your OneWorld installation. The majority of these tasks are common to all types of OneWorld installations where an iSeries server serves as the enterprise server. The many hardware, software, and configuration options available when implementing OneWorld dictate that it is not possible to include all potential scenarios in this book. It is very likely that your particular installation will differ greatly from any other installation. The more you understand the configurable nature of OneWorld, the more sense you can make of the specific tasks required of you.

Following the completion of the installation processes, your Conference Room Pilot (CRP) phase may commence. During this phase, you test all aspects of your client/server environment, including performance (both server and network), security, and reliability. You may need to make changes to your configuration based on your findings during this testing. For example, it is possible that you may amend Object Configuration Manager (OCM) mappings within a WAN environment to allow certain static data tables to be located on a server closer to client systems remote from the enterprise server. The possibilities are numerous. The completion of your installation and initial configuration tasks is only the beginning of what is likely to be an in-depth implementation process.

3.1 First-time installation

The information in this section is presented as a series of notes relating to each of the main installation processes. The main processes are defined in the *Installation Guide Xe (AS/400 Systems)* as:

- Installing the deployment server
- Planning the installation
- Running Installation Workbench
- Installing the enterprise server
- Installing the workstations
- Completing the installation

The first step in the OneWorld installation process is to load the software onto the deployment server. From this central point, the software is distributed throughout the enterprise to all servers and clients. On this system, the configuration of OneWorld is planned, including the determination of what objects to install and where to install them. The software is then loaded onto all required servers through a combination of direct loads from CD-ROM and file transfers from the deployment server. Deployment packages are created for each user to determine which OneWorld objects to install on the user's workstation. Finally, several post-installation tasks are performed to complete the OneWorld installation.

The following sections describe each of these processes in more detail and provide additional information where appropriate.

3.1.1 Installing the deployment server

The deployment server acts as the focal point for the OneWorld installation process and ongoing creation and deployment of OneWorld packages to client workstations. J.D. Edwards requires the deployment server to be an Intel-based server running Windows NT or 2000. The Integrated xSeries Server for iSeries meets these requirements and is certified by Microsoft to run Windows 2000. The installation tasks described in this chapter are based on using the Integrated xSeries Server for iSeries as the deployment server.

Note: Terminal Services must not be installed on the deployment server.

The software is loaded onto the deployment server by running the setup program on the first installation CD-ROM. The execution of the setup.exe program walks you through selecting the OneWorld components and path codes you are going to install. Path code directory structures are created based on the installation options selected.

The Planner Path Code and environment are created on the deployment server for use during the Installation Planner and Installation Workbench phases. Together with the JDEPLAN.MDB Access database, the Planner environment serves as a “staging” area for the OneWorld installation process.

Starting and using the OneWorld Planner environment require that the software to be authenticated with a security code. The deployment server installation process asks you to input an authorization code that can be obtained from the J.D. Edwards Contracts Department by calling 1-800-289-2999 and then following these steps:

1. Select option 5 for SPC code authorization.
2. When prompted, enter your customer number on your telephone keypad followed by the # key.
3. Select the option for OneWorld to reach the Authorization code department.

Prior to V4R3M0 of OS/400, OneWorld Central Object tables had to reside in an Oracle or Microsoft SQL database, due to a restriction in the size of binary large object (BLOB) table fields that could be stored in the DB2 Universal Database (UDB) for iSeries – the iSeries server integrated database.

J.D. Edwards has created a workaround to overcome this restriction. This workaround involves dividing up BLOBs into 32 KB pieces; it works with OS/400 V4R3 and later. Although OS/400 V4R4 can natively support larger BLOBs, J.D. Edwards is not expecting to take advantage of this native support until a future release of OneWorld. Scripts are supplied with the OneWorld software that are used to create the databases, and the tables that are required on the deployment server. Once these are created, Central Object specifications and other (BLOBs) can be loaded into the databases. Two enterprise server activities are also performed during the deployment server phase:

- The installation of the OneWorld Data Dictionary
- The configuration of TCP/IP that enables communication between the enterprise and the deployment servers

Configuring the Integrated xSeries Server

Before you start the deployment server installation processes, the Windows 2000 Server must be configured. The installation consists of three steps:

1. Install the Integration for Windows 2000 software on the iSeries server.
2. Install Windows 2000 on the Integrated xSeries Server.
3. Complete Windows 2000 post installation activities.

The process begins on the iSeries server and completes on the Windows 2000 server. The iSeries server controls the installation process and supplies default values for the Windows 2000 installation portion. The redbook *Consolidating Windows 2000 Servers in iSeries: An Implementation Guide for the IBM Integrated xSeries Server for iSeries*, SG24-6056, contains detailed installation and configuration information.

> **Note:** Typically there are at least three Network Server Storage spaces associated with the Integrated xSeries Server used for the OneWorld deployment server:
>
> - One for the Windows 2000 system (at least 1024 MB)
> - One for the installation source (at least 360 MB)
> - One for the OneWorld code (based on the number of path codes to be supported, but typically at least 16 GB)
>
> The Windows 2000 system and installation source spaces are created and linked to the network server description during the INSWNTSVR command. The OneWorld space needs to be created and linked to the network server description after the INSWNTSVR command and Windows 2000 installation completes. See the redbook *Consolidating Windows 2000 Servers in iSeries: An Implementation Guide for the IBM Integrated xSeries Server for iSeries*, SG24-6056, for details on creating and linking storage spaces.

Windows 2000 post installation activities

There are a number of steps that you need to perform after you install, configure, and connect the iSeries server with the Windows 2000 server. Follow these steps:

1. Verify successful communications. From an MS-DOS command screen, ping the adapter card, the iSeries server, and other systems on the network using PING or TRACERT followed by the IP address or system name.
2. Install Client Access Express. For OneWorld on the deployment server to communicate successfully with the iSeries Host Server, the Client Access ODBC driver must be installed on the deployment server.

   > **Note**: Use the Client Access Express ODBC and *not* the ODBC drivers from previous Client Access clients such as Client Access Windows 95/NT. Previous ODBC drivers are not supported with OneWorld.

   a. Load the Client Access Express CD in the iSeries CD-ROM drive.
   b. Using Windows 2000 Explorer, find the CD drive and find the **Setup.exe** file. Double-click **Setup.exe** to run it.
   c. Click **Next** on the Client Access Welcome display.
   d. Select **Yes** to agree to the conditions on the Client Access Express License Information display.
   e. For Type of Installation, select **Typical** if Data Transfer and 5250 Display and Printer Emulation are required (both of these options require a Client Access License for each concurrent display on which they are used). Otherwise, select **Custom** and select only the Client Access Express components that are required (at least Client Access ODBC *must* be selected).
   f. Select the default for Destination Folder (or select **Browse** to choose an alternate directory). Click **Next**.
   g. Click **Next** for Programs Menu Shortcut.
   h. Click **Next** for Copying Files.
   i. Deselect the **README** file. Click **Next** on the Install Completed display.
   j. Click **Finish** on the Setup Complete display to restart your computer now.
   k. Click **Yes** on the Shutdown Confirmation display.
   l. Windows 2000 re-boots.
3. Install the latest Client Access service pack. Obtain the latest Client Access Express service pack from the Client Access Web site `http://www.as400.ibm.com/clientaccess` and place it on a shared network drive or order the service pack using the Send PTF Order (SNDPTFORD) command.

a. Using Windows 2000 Explorer, locate the Client Access Express service pack **Setup.exe** file (either on CD or a shared network drive). Double-click the **Setup.exe** file to run it.

b. Deselect **View the README file**. Click **Next** on the Welcome display.

c. Click **Next** on the Copying Files display.

d. Click **Finish** on the Setup Complete display.

4. Allocate the iSeries server disk storage for use by the deployment server. To do this, use the CRTNWSSTG command on the iSeries server, for example:

```
CRTNWSSTG NWSSTG(B2BARIB23) NWSSIZE(24000) FROMNWSSTG(*NONE) FORMAT(*NTFS)
ASP(1) TEXT('Stg Space for OneWorld on the deployment server')
```

5. Assign drive space to the network server card using the ADDNWSSTGL command on the iSeries server, for example:

```
ADDNWSSTGL NWSSTG(B2BARIB23) NWSD(B2BARIB2) DRVSEQNBR(*CALC)
ACCESS(*UPDATE)
```

6. Format allocated disk space to Windows 2000.

   Once you allocate the space in OS/400, you can now format this space in Windows 2000. To format the storage space for use with Windows 2000 server, follow these steps:

   a. Go to the Windows 2000 Disk Administrator. Click **Start-> Programs-> Administrative Tools-> Computer Management**.

   b. Expand **Storage**.

   c. Click **Disk Management**.

   d. The disk drive that appears as Unknown needs to be formatted to be usable by Windows 2000. Perform the following steps:

      i. Right-click the disk drive, select **Format**, and then follow the prompts.

      ii. Click **OK** on the warning since there is no data on the disk to erase. The format should proceed to completion automatically. If you see a warning that an application is using the drive, verify that this is the newly created and linked drive, and then click **Yes** to force the format.

7. Install the C++ Compiler. We installed Visual Studio 6.0, which includes the C++ compiler.

8. Install the latest Visual Studio 6.0 service pack.

Running Environment Checker

Environment Checker is a standalone application that you run before you install OneWorld. Environment Checker diagnoses any OneWorld configuration or setup issues that you may have at the operating system level. For example, it verifies that you have enough disk space to install OneWorld to various machines.

When you run Environment Checker, the interface prompts you for information about your system. Once the application finishes, Environment Checker creates an output report that contains information about your system and any warning messages that need attention.

The current version of Environment Checker supports Microsoft Windows and Intel platforms only, and therefore can only be used for checking the deployment server and client workstations.

Here are the steps for running the Environment Checker:

1. Insert the OneWorld Setup CD into the iSeries server CD-ROM drive.
2. On the deployment server, using Windows 2000 Explorer, click the CD drive. Open the CD Templates folder, and click **Environment Checker**. Double-click **ec_winintel.exe** to run the Environment Checker program.
3. Type the directory where you want the results created. Press the Enter key.
4. Type `N` in reply to the prompt for using the previous set of answers. Press the Enter key.
5. Type `B73.3` in reply to the prompt for the OneWorld version being used. Press the Enter key.
6. Type `1` to indicate that this is the deployment server. Press the Enter key.
7. Type `3` to indicate that the deployment server will run in an iSeries server environment. Press the Enter key.
8. Type `D` in reply to the prompt for which database product will be used on the deployment server. Press the Enter key. Even though no database product will be used on the deployment server, a valid answer is required.
9. Type the number of path codes (that is the number of OneWorld environments) that you will load. Press the Enter key.
10. Type the name of the iSeries enterprise server. Press the Enter key.
11. Type `3` in reply to the prompt for the most detail on the Environment Checker report. Press the Enter key.
12. The Environment Checker runs. Press the Enter key to view the ec_rpt.txt file for warnings.

13. Ensure that you resolve all warnings before you proceed with the OneWorld deployment server installation.

Client Access TCP/IP configuration tasks

On the iSeries server, use the CFGTCP command, and enter option 1 (Work with TCP/IP interfaces) to identify both the virtual *TRLAN and the physical LAN IP addresses. You use these IP addresses to define the Client Access sessions. The Integrated xSeries Server needs to be configured using the IP address assigned to the virtual *TRLAN. For all other client machines, use the actual IP address of the external LAN.

Setting up Client Access Express to use the internal LAN

You can set up Client Access Express for Windows to use the internal LAN. This minimizes the risk of connection loss due to an external network failure. If you are using a low speed LAN or have heavy network traffic on your LAN, it may also improve the performance of ODBC communications between Windows 2000 on the Integrated xSeries Server and the iSeries server. To do so, complete these steps:

1. Find the iSeries server's internal LAN TCP/IP interface. On an OS/400 command line, type:

   ```
   NETSTAT *IFC
   ```

   Press Enter. Search for an entry with a line description with the same name as your network server description but ending in 00. The TCP/IP address of that entry should have the form 192.168.x.y. Write down this address.

2. Create a new Client Access Express for Windows connection. Follow these steps:

   a. Use Operations Navigator to create a new connection.

   b. Right-click the new connection and select **Properties**.

   c. Click the **Connection** tab.

   d. In the IP address lookup frequency window, select **Never - Specify IP address**.

   e. Type the IP address of the internal network address in the IP Address window.

   f. Click **OK**.

After you create that connection, you can use Client Access Express for Windows functions the same as you can on any other PC.

Setting the deployment server paging file size

Before you start the installation processes for the deployment server, you should set the Windows 2000 Server paging file size to optimize system performance. To achieve this, perform these steps:

1. Select **Start-> Settings-> Control Panel**.
2. Double-click **System**.
3. Select the **Advanced** tab.
4. Click **Performance Options**.
5. Click the **Change** button in the Virtual Memory section.
6. Select the disk drive or drives that will be used for the paging file.
7. Set the initial size for the Paging File Size field to a number 1.5 times the amount of RAM available on the system.
8. Set the Maximum size to around double the initial size. Do *not* to set the maximum number too high on systems where there is limited overall disk space, since this will have a negative affect on system performance. The display appears similar to the example in Figure 3-1.

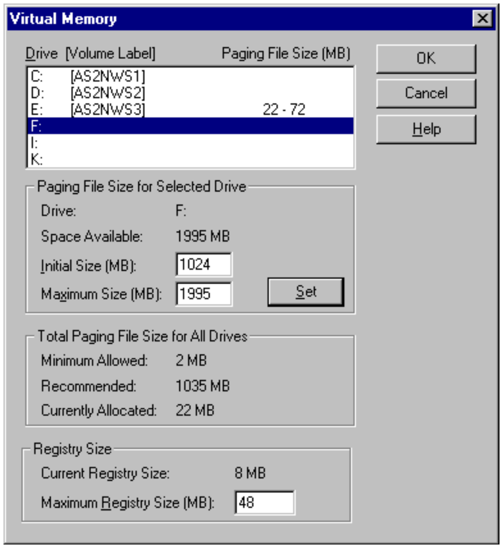


Figure 3-1 Setting the paging file size on the deployment server

9. Click **Set** to complete the settings.
10. Click **OK** to save the settings.
11. If prompted to do so, re-boot Windows 2000.

Configuring the OneWorld deployment server

Before you install OneWorld, you must configure the deployment server. To do so, complete these steps:

1. Create the user `JDE` as a system administrator on the deployment server. Sign on the deployment server as `Administrator`.
   a. Select **Start-> Programs-> Administrative Tools-> Computer Management**.
   b. Expand **Local Users and Groups**.
   c. Right-click **Users** and select **New User...**.
   d. Right-click **Users** and select **New User...** to enter a new user. The display looks similar to the example in Figure 3-2. Enter the user `JDE` with the password `JDE`.

      Note: Passwords are case sensitive.

   e. Deselect **User must change password at next logon**.

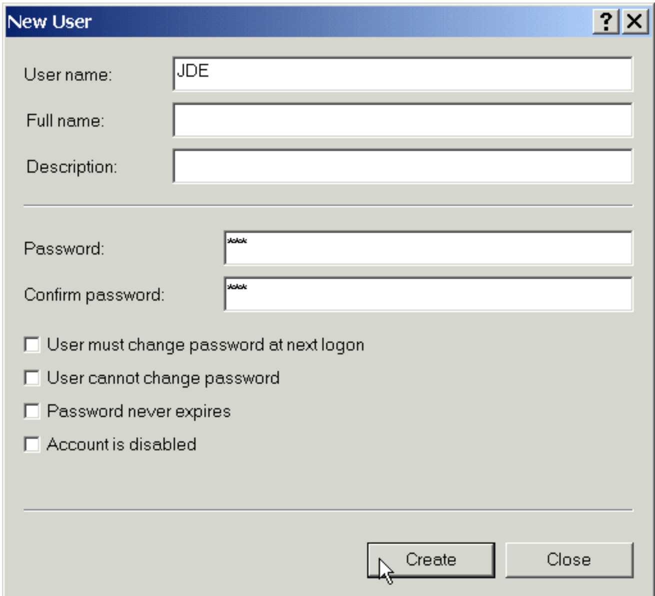


Figure 3-2 Creating a new Windows 2000 user

   f. Click **Create** to create this new JDE user, and then click **Close**.
   g. Right-click the new JDE user, select **Properties**, and then click the **Member Of** tab.

h. Click **Add** to see a list of groups, and select **Administrator**.

i. Click **Add** to add the user to the group, and then click **OK**.

j. Click **OK** to close the Properties panel, and then close the Computer Management panel.

k. Select **Start-> Shut Down**.

l. Select **Log off Administrator**.

m. Log on again as user `JDE`.

> **Note:** If the deployment server is part of a domain, you have to select **Start-> Programs-> Administrative Tools-> Active Directory Users and Computers** to add the JDE user and add it to the Administrators group.

2. Create a printer device for use by the deployment server:

   a. Right-click the **My Network Places** icon on the desktop and select **Explore**.

   b. Expand **Entire Network**.

   c. Expand **Microsoft Windows Network**.

   d. Double-click the workgroup that the deployment server was made a member of with the INSWNTSVR command.

   e. Double-click the print server.

   f. Enter an appropriate user ID (for example, `Administrator`) and password to connect to the print server. Click **OK**.

   g. Double-click the printer that should be added for the deployment server.

   h. If prompted for Windows to set up the printer, click **OK**. Windows adds this printer to the selectable printers for the deployment server.

   i. As an alternative method for creating the printer device, perform the following steps:

      i. Select **Start-> Settings-> Printers**.

      ii. Double-click **Add Printer** to open the Add Printer Wizard and click **Next**.

      iii. Select **Network Printer** and click **Next**.

      iv. Click **Next** to browse for a network printer.

      v. Expand the workgroup that the deployment server was made a member of with the INSWNTSVR command.

      vi. Expand the print server.

vii. Click the printer to highlight it, and click **Next**.

viii. Select **Yes** or **No** based on whether this printer should be used as the default Windows printer, and click **Next**.

ix. Click **Finish** in reply to the message that the network printer has been successfully installed.

3. Install file transfer protocol (FTP) on the deployment server:

> **Note:** By default, the Microsoft Internet Information Server code is installed as part of the Windows 2000 Server installation, but the FTP component is not. You will need to add the FTP component.

   a. Select **Start-> Settings-> Control Panel**.
   b. Double-click **Add/Remove Programs**.
   c. Click **Add/Remove Windows Components**.
   d. Select **Internet Information Services (IIS)** and click **Details...**
   e. Double-click **Microsoft Internet Information Server**.
   f. Select the **File Transfer Protocol (FTP) Server** component.
   g. Click **OK** to return to the Windows Components Wizard.
   h. Click **Next**.
   i. When the Files Needed panel appears, type the path to the installation source (normally `D:\I386` on the Integrated xSeries Server), and click **OK**.
   j. Click **Finish** on the Completing the Windows Components Wizard panel when it appears.

4. Configure FTP:

   a. Select **Start-> Programs-> Administrative Tools-> Internet Services Manager**.
   b. If necessary, expand the site tree.
   c. Right-click **Default FTP Site** and select **Properties**.
   d. Click the **Security Accounts** tab.
   e. On the panel similar to the example in Figure 3-3, select **Allow Anonymous Connection**. Deselect **Allow only anonymous connections**. Click **OK**.

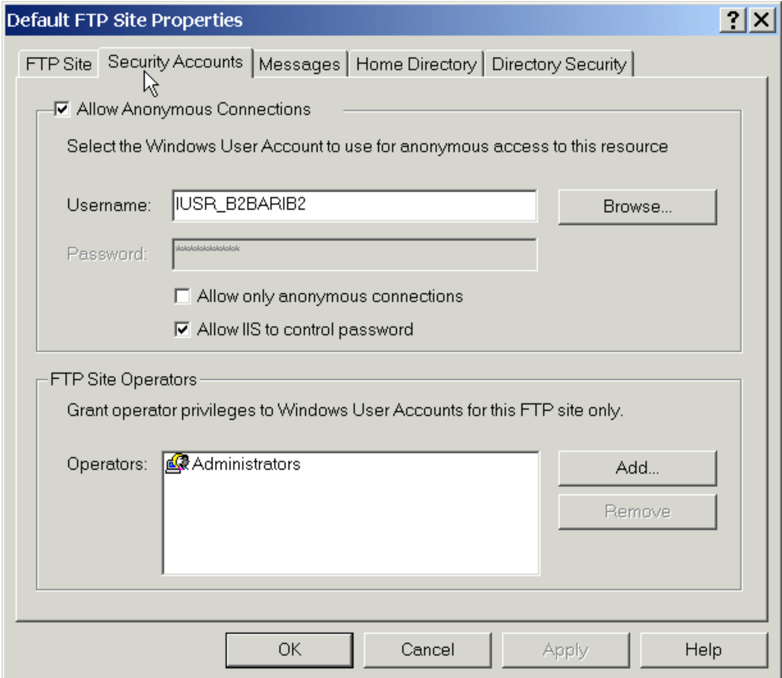


Figure 3-3 FTP Site Properties

f. Right-click **Default FTP Site** and select **New-> Virtual Directory**.

g. Click **Next** on the Virtual Directory Creation Wizard.

h. Enter the virtual directory alias (similar to Figure 3-4) based on the drive on which you will install OneWorld. Click **Next**.

i. Enter the drive path (similar to Figure 3-5) where you will install OneWorld. Click **Next**.

j. Select **Read** and **Write** access permissions, and then click **Next**.

k. Click **Finish** to exit the Virtual Directory Creation Wizard.

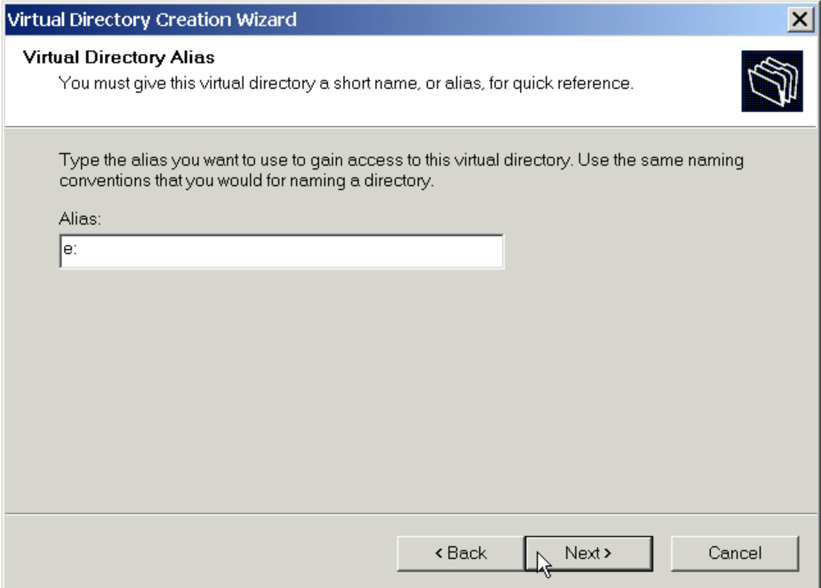


Figure 3-4 Virtual Directory Alias

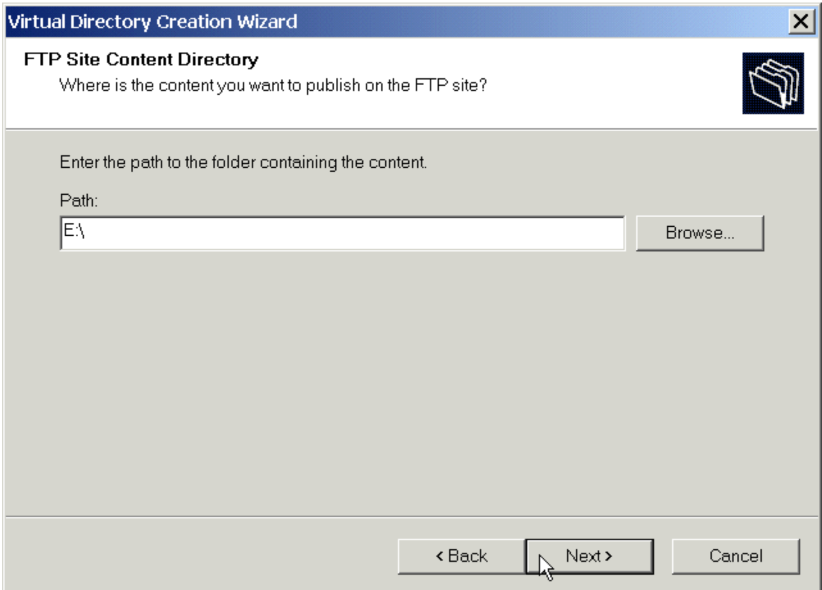


Figure 3-5 FTP Site Content Directory

Installing platform-independent objects

The first OneWorld installation CD is used at this point to load the platform independent objects onto the deployment server. As part of the load, the installer is given the option to install Adobe Acrobat Reader and Microsoft Internet Explorer, if compatible versions are not already installed. These are both required on the deployment server to enable certain aspects of OneWorld functionality.

Upon completion of the setup program, verify the OneWorld directory structure. The expected structure is listed in Chapter 3 of the *OneWorld Installation Reference Guide (All Systems)*. This includes directory paths for your selected path codes.

The second installation CD contains the setup scripts for each relational database management system (RDBMS) compatible with OneWorld. These are used later in the deployment server installation process to create the databases necessary to contain the Central Object tables and other BLOBs tables. The installer is prompted to select the chosen RDBMS, either Microsoft SQL Server or Oracle Workgroup Server. Since we plan to install the Central Objects tables in DB2 Universal Database for iSeries, we do not use these scripts here.

1. Sign on the deployment server as `JDE`.
2. Insert the first OneWorld setup CD into the iSeries CD-ROM drive. Setup starts automatically.
3. Click **Xe Deployment Server Install**.
4. Click **Next** on the OneWorld installation program Welcome display.
5. When the Welcome to J.D. Edwards OneWorld Setup! display appears, record the serial number and follow these steps:
    a. Contact the J.D. Edwards Contracts Department (call 1-800-289-2999 or 1-303-334-4000) and provide the serial number and your customer number.
    b. You are provided an Expiration Date, Number of Licenses, and Authorization Code. Enter these in the corresponding fields, and click **Next**.

    If a new display appears, the correct authorization code has been entered.

    > **Attention:** If you cancel the OneWorld Installation program before it completes successfully, you need to obtain a new authorization code.

6. You may see a display to install the Adobe Acrobat Reader or Internet Explorer, if compatible versions of those applications are not already installed.

7. Select **Custom** for the J.D. Edwards OneWorld Deployment Server Setup Type (Figure 3-6). This option is required to install the Central Objects on the iSeries server. Enter the directory where the OneWorld files will be installed on the deployment server (for example, `K:\JDEdwardsOneWorld\B7333`). If this directory does not already exist, the setup program creates it. Click **Next**.

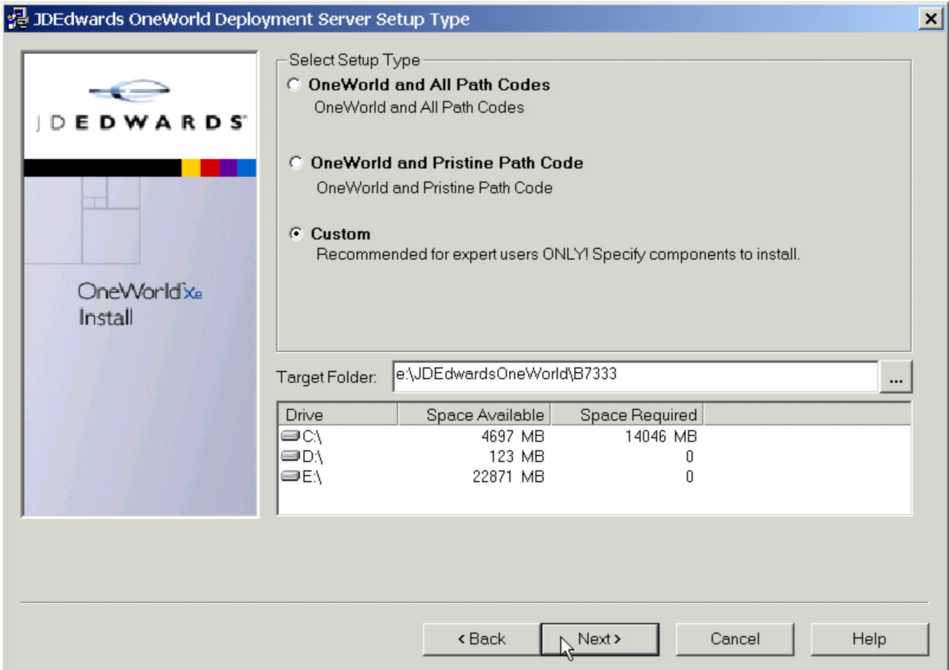


Figure 3-6 Setup Type

8. The setup program verifies that there is sufficient disk space for installing OneWorld on the disk drive requested. If there is sufficient disk space, click **Next** to continue. Otherwise, choose another disk drive, click **Next**, and repeat this step.

9. Select the path codes corresponding to the OneWorld environments that you plan to install (Figure 3-7). If all path codes are automatically selected, deselect those that you do not want. Click **Finish**.

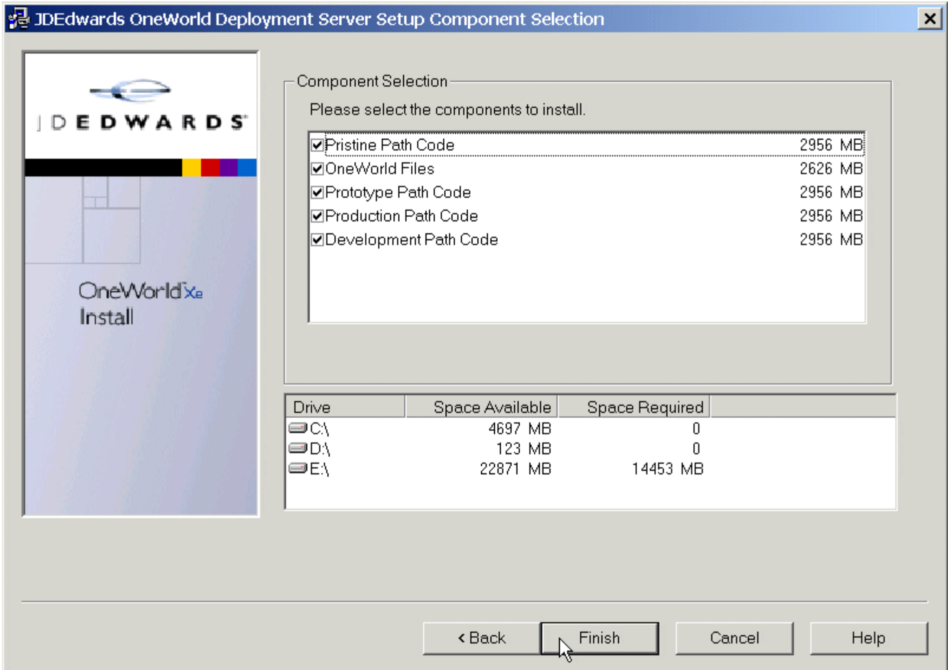


Figure 3-7 Component Selection

10. Click **OK** when you see the message stating that the J.D. Edwards OneWorld Deployment Server Setup succeeded.
11. Cancel out of the OneWorld Installation Manager, because the installation is finished. Since we are installing Central Objects on the iSeries server, we do not use the second CD.

Sharing the directory \JDEdwardsOneWorld\B7333

Using Windows 2000 Explorer, find the \B7333 directory in the J.D. Edwards OneWorld directory on the deployment server disk drive where you installed OneWorld. Then complete the following steps:

1. Right-click the **\B7333** directory.
2. Select **Sharing**.
3. Select **Shared As**.
4. Type `B7333` as the Share Name.
5. Click **OK**.

Installing libraries and Data Dictionary on the iSeries server

At this stage, the OneWorld Data Dictionary must be created on the iSeries enterprise server. At Release B73.3 and later, the Data Dictionary library must *always* be loaded from the installation CD-ROMs. For previous releases of OneWorld, this was necessary for coexistence installations only.

This process also creates all iSeries server libraries required by each of the OneWorld environments supplied by J.D. Edwards. Verify that the following libraries are created:

- SVM7333
- DD7333: Contains Data Dictionary tables for OneWorld
- PathCode7333
- PathCode7333DNT: Contains the versions tables for the OneWorld path code
- PathCodeCTL: Contains menu tables for OneWorld
- PathCodeDTA: Contains data tables for OneWorld, and WorldSoftware in a co-existence environment, with the exception of PRISTDTA. PRISTDTA only contains data tables for OneWorld, because WorldSoftware does not have a Pristine environment.
- JDEOW
- OL7333
- SYS7333

Here, *PathCode* represents the PathCode identifier (CRP or PY for Conference Room Pilot, PRIST or JD for Pristine, PROD or PD for production, DEV or DV and TEST or TS for test) for each of the OneWorld environments that you are installing.

There is *no* DEVDTA library, because the development and test environments share the same data. There is no DEVCTL library since the development and test environments share the same menu tables. There is no TS7333 or TS7333DNT since the test and CRP environments share the same version tables.

To install the iSeries server libraries and the Data Dictionary, complete the following steps:

1. Sign on to the iSeries server as security officer (user ID `QSECOFR`).
2. Insert the iSeries server Direct Enterprise Server Install 1 of 7 CD into the iSeries CD-ROM drive.
3. On an OS/400 command line, enter the command:

`LODRUN OPTxx`

Here *OPTxx* is the name of the CD-ROM drive. This creates all of the libraries listed earlier, except the Data Dictionary. Then the A98OWMNU menu (Figure 3-8) appears.

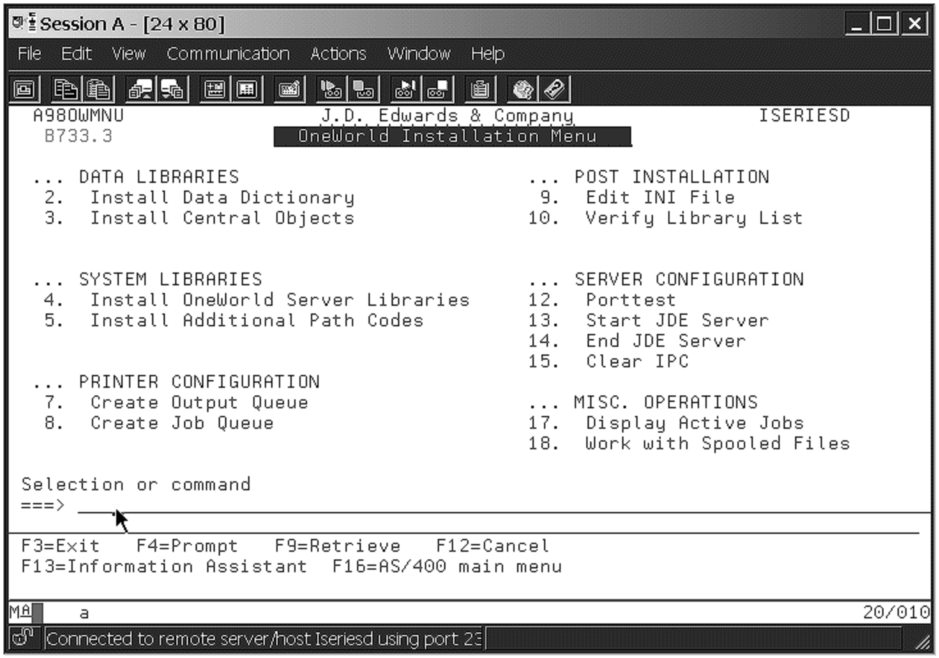


Figure 3-8 OneWorld Installation Menu

4. Enter option `2`, and press Enter to install the Data Dictionary. The Install Data Dictionary display (Figure 3-9) appears.

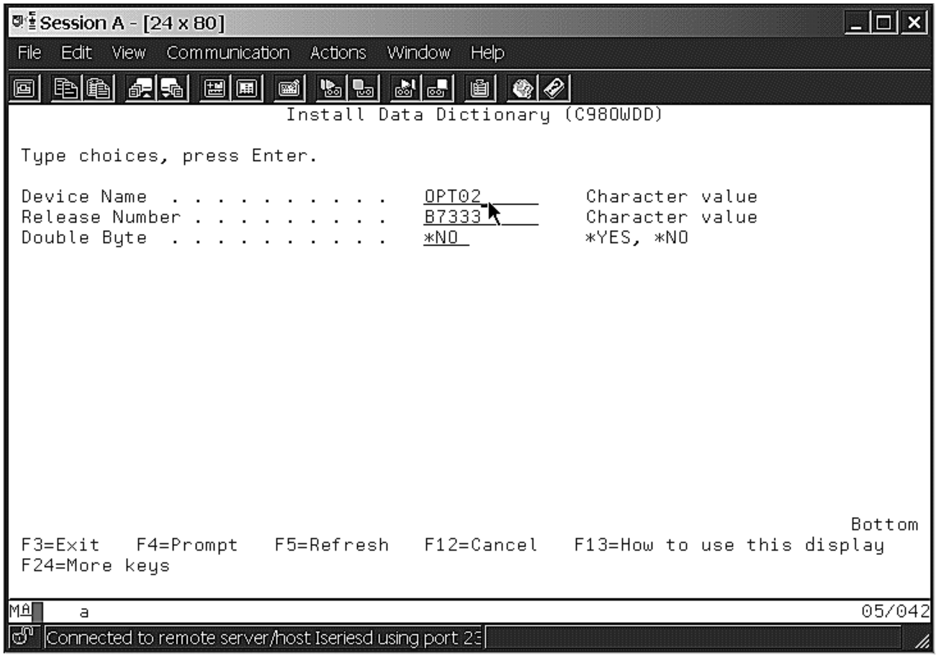


Figure 3-9 Installing OneWorld Data Dictionary on the iSeries server

5. If necessary, change the Device name to that of your iSeries CD-ROM drive (this is usually `OPT01`). Press Enter.

> **Note:** OS/400 libraries may be created for environments that have not been selected for installation. They are not automatically deleted. If you do not want them on your iSeries server, you must delete them manually (use the DLTLIB command).

Setting up the TCP/IP protocol for the enterprise server

TCP/IP should already be configured on the iSeries server (when the infrastructure is prepared). It is necessary at this stage of the installation process to perform further configuration tasks that allow the deployment server to communicate with the iSeries enterprise server.

1. Sign on to the iSeries server as security officer (`QSECOFR`).
2. Enter the CFGTCP command and press Enter to access the Configure TCP/IP menu.
3. Choose option `10` and press Enter to work with the TCP/IP host table entries.

For more information on TCP/IP configuration on AS/400 system, refer to Appendix B in *J.D. Edwards OneWorld Implementation for AS/400*, SG24-5195.

Creating an enterprise server host table entry

The iSeries enterprise server must have an entry in its TCP/IP host table in the format `hostname.domain name`. To create this entry, follow these steps:

1. Find the IP address for the enterprise server. If it exists, choose option `5` and press Enter to verify that the format is correct as shown in Figure 3-10.

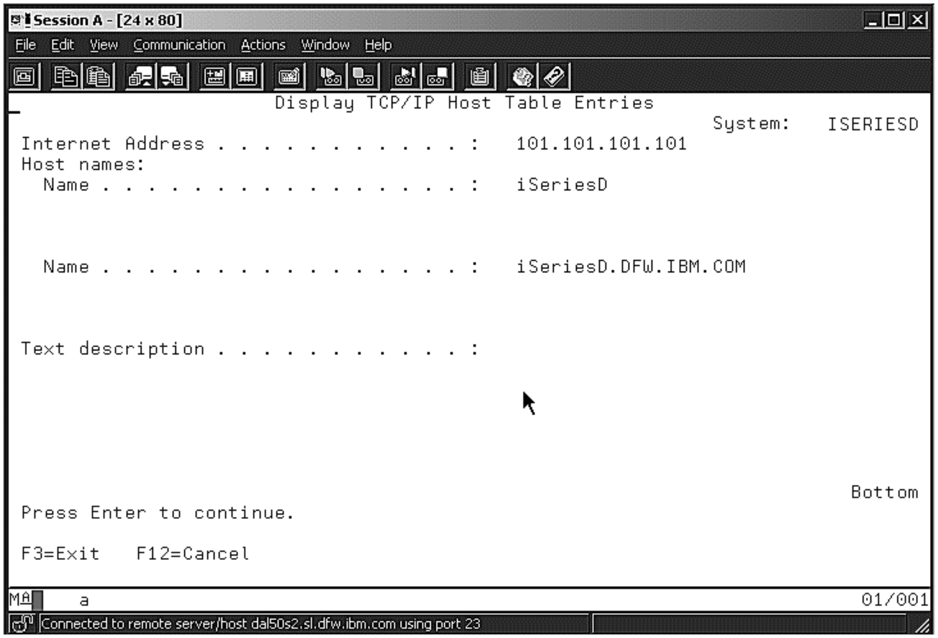


Figure 3-10 Display TCP/IP Host Table Entries

2. If the entry does not exist, enter option `1` and press Enter to create one.
3. If there is an entry, but it is in another format, an alias can be added to it. To do this, choose option `2` and press Enter.

> **Note:** The TCP/IP host name for the enterprise server is the same as the current system name. Use the Display Network Attributes (DSPNETA) command to verify this, as shown in Figure 3-11.

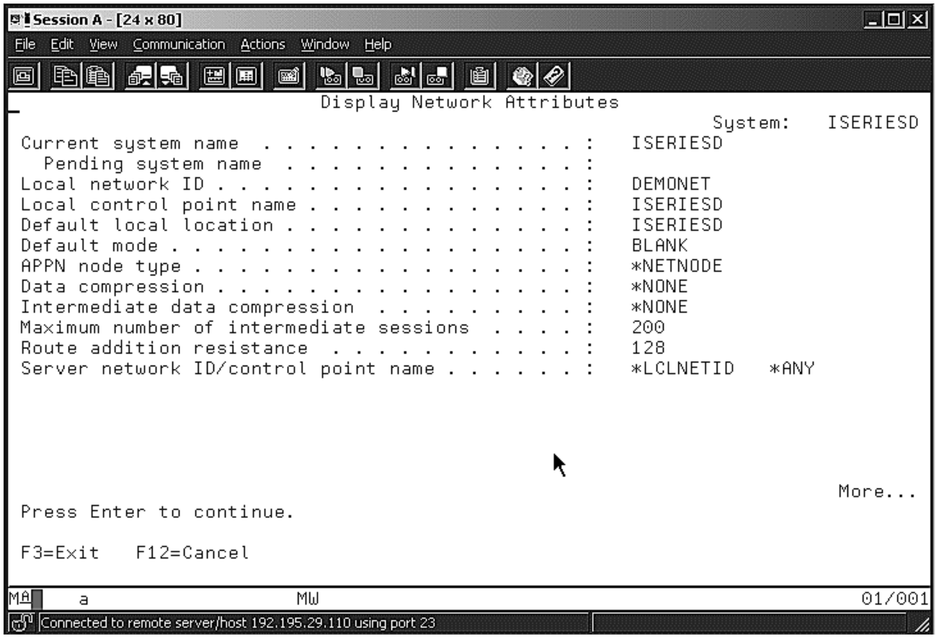


Figure 3-11 Display Network Attrributes

Creating a deployment server host table entry

A host table entry must also be created for the deployment server on the iSeries enterprise server. To create this entry, follow these steps:

1. Enter option `1` and press Enter to add a new host table entry for the deployment server.
2. Type the IP address of the deployment server in the Internet Address field. Enter the name of the deployment server in the Host name field. Press Enter.

After adding this entry, issue a Verify TCP/IP Connection (`PING`) command from the iSeries server using the system name, not the TCP/IP address, of the deployment server. This ensures that successful communications can be established between the two systems.

Deployment server FTP configuration verification

The File Transfer Protocol (FTP) is used during the Installation Workbench and host load processes to transfer the OneWorld system data source and software from the deployment server to the iSeries enterprise server. To ensure that FTP is configured correctly on the deployment server, an FTP session should be established from the iSeries server to the deployment server, as well as from the deployment server to the iSeries server, using the JDE account to sign on.

Perform the following steps on the iSeries server to test FTP functionality:

1. On an OS/400 command line, type:

   `FTP NTserver`

   Here *NTserver* is the name of your deployment server on the Integrated xSeries Server. Press Enter to start FTP.

2. Type `JDE` in reply to the FTP user ID prompt. Press Enter.
3. Type the password for the JDE user ID on the deployment server. Press Enter.
4. If the connection is successful, type `QUIT`. Press Enter to end the FTP connection.
5. Perform the following steps on the Windows 2000 Server on the Integrated xSeries Server to test the FTP functionality:

   a. In an MS-DOS session, type:

      `FTP servername`

      Here *servername* is the name of your iSeries server. Press Enter to start FTP.

   b. Type `JDE` in reply to the FTP user ID prompt. Press Enter.

   c. Type the password for the JDE user ID on the deployment server. Press Enter.

   d. If the connection is successful, type `QUIT`. Press Enter to end the FTP connection.

These commands are also issued by the system during the host load process. By verifying their success now, you should avoid potential connectivity problems later.

Copying your specifications and installation plan files

To protect your deployment server installation in case of failure, J.D. Edwards recommends that you copy your JDEKRNL.XDB and JDEKRNL.DDB specifications files and your JDEPLAN.MDB installation plan file to a temporary directory. If the original files are corrupted later, you will have backup copies of these files.

1. Create a temporary directory for these backups if you do not already have one.
2. Using Windows 2000 Explorer, copy the JDEKRNL.XDB and JDEKRNL.DDB files from the *n*:\JDEdwardsOneWorld\B7333\System\Bin32 directory to the temporary directory, where *n* is the drive location of the OneWorld software on the deployment server.

3. Using Windows 2000 Explorer, copy the JDEPLAN.MDB file from the JDEPLAN.MDB file from the *n*:\JDEdwardsOneWorld\B7333\Planner\Data directory to the temporary directory, where *n* is the drive location of the OneWorld software on the deployment server.

Configuring Microsoft Access ODBC

J.D. Edwards recommends that the buffer size for the Microsoft Access ODBC driver on the deployment server is set at 2048 Kb per processor on the deployment server. To do this, follow these steps:

1. To set this allocation, click **Start-> Programs-> Administrative Tools-> Data Sources (ODBC)-> User DSN**.
2. If you do not see the User Data Source MS Access 97 Database, click **Add**. Select **Microsoft Access Drive (*.mdb)**. Click **Finish**.
3. Double-click **MS Access 97 Database**. From the ODBC Microsoft Access Setup display, select **Options**. Update the Buffer Size field as displayed in Figure 3-12, if necessary. Click **OK**.

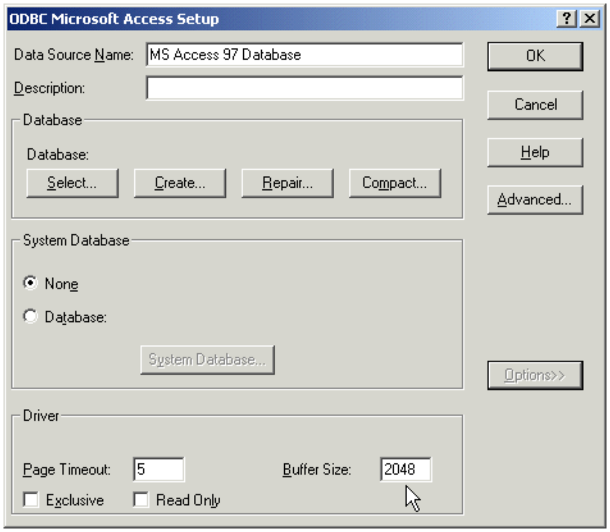


Figure 3-12 ODBC Setup display

3.1.2 Installing the OneWorld databases

Due to limitations in the size (maximum of 32 KB) of BLOBs in the iSeries database in OS/400 releases *prior to V4R4*, J.D. Edwards required that you store the OneWorld Central Objects either in an Oracle or Microsoft SQL Server database.

Even though DB2 UDB for iSeries no longer has this restriction, J.D. Edwards still does not support the *native* BLOB implementation of DB2 UDB for iSeries for OneWorld Release Xe. However, J.D. Edwards has developed a workaround technique that breaks the BLOBs into 32 KB pieces and stores them in DB2 UDB for iSeries. The workaround makes it possible to use DB2 UDB for iSeries for Central Objects. This is supported for OneWorld Xe on OS/400 V4R4 and later.

Since the OneWorld Central Objects database will be stored on the iSeries server, you do not need to tailor creation scripts to meet the individual requirements of your installation processes.

> **Note:** Central Objects on the iSeries server require approximately 2 GB of disk space for each path code that you plan to install. You should verify that this has been included in your initial sizing.
>
> If you plan to install Central Objects on the iSeries server, let J.D. Edwards know this when you order OneWorld. This way, an extra OneWorld Central Objects CD will be shipped with the other OneWorld CDs. This CD contains the special code required to load Central Objects on the iSeries server and is required to complete this step.

Central Objects are loaded on the iSeries server into Central Object libraries, as shown in Table 3-1.

Table 3-1 OneWorld Central Objects data sources on the iSeries server

Central Objects data source	Library name
Central Objects - PY7333	COPY7333
Central Objects - DV7333	CODV7333
Central Objects - PD7333	COPD7333
Central Objects - JD7333	COJD7333

Loading Central Objects in DB2 UDB for iSeries

To load Central Objects in DB2 UDB for iSeries, complete the following steps:

1. Sign on to the iSeries server as security officer (`QSECOFR`).
2. Insert the OneWorld Central Objects CD into the iSeries CD-ROM drive.
3. On an OS/400 command line, type:

   ```
   ADDLIBLE JDEOW
   ```

   Press Enter.
4. On an OS/400 command line, type:

```
GO JDEOW/A98OWMNU
```

Press Enter.

5. Enter option `3`. Press Enter.
6. Type the names of each of the Central Objects libraries (`COPY7333`, `CODV7333`, `COPD7333`, `COJD7333`) corresponding to the OneWorld environments that you are installing. Press Enter.

3.1.3 Planning and configuring OneWorld servers

During this phase of the OneWorld installation, the Installation Planner and Installation Workbench applications are used to define and install the configuration of all required servers. Aside from the enterprise server, these systems may optionally include Java Application Servers, Windows Terminal Servers, or Workgroup Servers. These additional servers may be included in the plan at this stage or added to the configuration through the creation of other plans at a later date. The choice is yours.

Environment data is entered into the Installation Planner database as defined on the pre-installation Detail Worksheet. ODBC data sources are created for each OneWorld data source to facilitate the required inter-system connectivity. The completed plan must be validated. In B73.3, the plan validation process is incorporated into the Installation Planner application. Once the validation process confirms that the plan is complete and that all entries are valid, it may be released. This updates the Release Master tables (in iSeries server library SYS7333) and the setup.inf file. The plan is then ready for Installation Workbench to use it.

Creating an RDBMS directory entry on the iSeries server

You should verify the existence of a relational database directory entry on the iSeries server. The entry name should be the same as the iSeries server name and have a remote location name of *LOCAL. To verify whether this directory entry exists already, on an OS/400 command line, enter the following command and press Enter:

```
WRKRDBDIRE *ALL
```

If it does not exist, a relational database directory entry must be created on the iSeries enterprise server using the Add RDB Directory Entry (ADDRDBDIRE) command. This is used to establish ODBC connections to the OneWorld data tables residing on the iSeries server. If the directory entry is not on this list, enter the following command on an OS/400 command line, and press Enter:

```
ADDRDBDIRE RDB(sysname) RMTLOCNAME(*LOCAL) TYPE(*IP)
```

Here, *sysname* is the name of your iSeries server.

Installation Planner

Installation Planner is a OneWorld application that runs on the deployment server, automating much of the OneWorld configuration. It is comprised of a number of separate workbenches, used to collect information that detail host system configuration, OneWorld data source, and environment configuration and deployment packages.

You access Installation Planner by signing on to the OneWorld Planner environment (JDEPLAN) on the deployment server using the JDE user. Installation Planner defines the following information:

- **Location setup**: Defines the use of multiple deployment locations to OneWorld.
- **Deployment server setup**: Defines the deployment server to OneWorld, its network name, and characteristics.
- **Enterprise server setup**: Defines the enterprise server to OneWorld, its name, type, and characteristics.
- **Setup of other servers**: Defines the use of other servers such as a workgroup server, application, or database servers.
- **Data source setup**: Configures the required data sources to support your configuration.
- **Environment setup**: Defines the environments and path codes needed to support your implementation plan.
- **Coexistent merges**: Defines which merges to run in a coexistent installation, such as the Data Dictionary.
- **Package setup and builds**: Defines which client workstation deployment packages are to be used. J.D. Edwards delivers pre-built standard packages that can be used during a new installation.

All of this information is stored in the Microsoft Access database JDEPLAN.MDB on the deployment server for use by the Installation Workbench application. After the plan is complete, you go through steps to verify and release the plan. These steps populate the appropriate tables with the configuration information. Releasing the plan creates all of the system tables on the iSeries server in the specified system library SYS7333. At this time, only selected tables are populated.

Plan types

Two types of plans are available to the installer: a Typical Plan or a Custom Plan. A *Typical Plan* automatically creates OneWorld data sources based on server information, but allows for little flexibility. A *Custom Plan* allows the installer to amend OneWorld data source definitions if necessary.

Creating a custom installation plan

J.D. Edwards recommends that you *always* select the option to create a Custom Plan for iSeries server installations where co-existence with J.D. Edwards' WorldSoftware is required, or Central Objects are stored on the iSeries server. The flexibility of the Custom Plan is needed to make the required changes to the ODBC data sources.

Starting Installation Planner

To start Installation Planner, complete the following steps:

1. Sign on to OneWorld on the deployment server with user `JDE` and password `JDE` in the JDEPLAN (planner) environment.
2. Enter `GH961` in the Fast Path field. Press Enter to go to the System Installation Tools Menu.
3. Double-click **Custom Installation Plan**.
4. Click **Add** on the Work with Installation Plans display.

Entering information into Installation Planner

To enter information into Installation Planner, complete the following steps:

1. Type the name you want to assign to your installation plan in the Name field on the Installation Planner form. Write down this name, because you will need it when you run the Planner Validation report.
2. Type a description of your plan.
3. Click **Install**.
4. Click **Coexistent with WorldSoftware** *only* if this is a OneWorld with WorldSoftware co-existence installation.
5. Verify `B7333` in the To Release field.
6. Click **OK**.
7. Click **OK** when prompted to Enter Location Information.

The Location Revisions form appears.

Entering plan location information

To enter plan location information, complete the following steps:

1. Type the name of your location in the Location field on the Location Revisions form.
2. Type a description for this location.
3. Since this is the primary deployment server, leave the Location Code and Parent Location blank.

4. Click **OK**.
5. Click **OK** when prompted to enter deployment server information.

The Deployment Server Revisions form appears.

Entering deployment server information

To enter deployment server information, complete the following steps:

1. Type the name of your deployment server in the machine name field on the Deployment Server Revisions form.
2. Type a description of your deployment server.
3. Verify `B7333` for the Release.
4. Type `JDE` for the Primary User.
5. Type `1` for Primary Deployment Server.
6. Type `\B7333` for the Server Share Name.
7. Click **OK**.
8. Click **No** when prompted to add another deployment server.
9. Click **OK** when prompted to enter enterprise server information.

The Enterprise Server Revisions form appears.

Entering enterprise server information

To enter enterprise server information, complete the following steps:

1. Type the name of your enterprise server in the machine name field on the Enterprise Server Revisions form.
2. Type a description of your enterprise server.
3. Verify `B7333` for the Release.
4. Type `JDE` for the Primary User.
5. Type `10` for the Host Type (AS/400).
6. Type `6009` for the Port Number if it is not correct.
7. Type the system name of your iSeries server for the Logical Machine Name if it is not correct.
8. Type `I` for the Database Type (Client Access).
9. Type *`machinename`* `- B7333 Server Map` for the Server Map Data Source if it is not correct, where *machinename* is the name of your iSeries server.
10. Type `B7333SYS` for Installation Path if it is not correct.
11. Type the deployment server name using the visual assist button.

12. Click **OK**.
13. Click **OK** when prompted to enter data source information for the enterprise server.

The Data Source Setup form appears for B7333 Server Map.

OneWorld data sources

During the creation of the installation plan, you are required to customize the OneWorld data sources. The parameters required for customizing these data sources are described in the following list and shown in Table 3-2.

> **Note:** The ODBC data sources that correspond to the OneWorld data sources are automatically created for you.

Table 3-2 shows the variable portions of the OneWorld data sources. The Library Name for each data source is the recommendation, based on using the J.D. Edwards standard naming convention, which is recommended. The rows are listed in alphabetical sequence. The following items list the constants for each of the OneWorld data sources:

- Data Source Use is always DB, except for *machine* * and *machine* - Logic *, which are SVR.
- Data Source Type is always I (Client Access).
- Object Owner ID is always blank.
- Library List Name is always the name of the iSeries enterprise server.
- DLL Name is always the default JDBODBC.DLL.
- Server Name is always the name of the iSeries enterprise server and is always the same as the Library List Name.
- Platform is always AS/400 or iSeries.
- You always choose the default to deselect Use Table Owner.
- You always select the default Use Julian Date.
- You always select the default Decimal Shift.
- You always select the default Support for Updates.
- Database Name is always the same as the Data Source Name, except for *machine* * and *machine* - Logic *, which are *machine* - B7333 Server Map*.

Table 3-2 OneWorld data sources

Data source name	Library name	Database name	OCM data source	iSeries BLOB data source
iSeries Common	CLTCOM **	iSeries Common	No	No
iSeries Common - CRP	CLTCOM **	iSeries Common - CRP	No	No
iSeries Common - Production	CLTCOM **	iSeries Common - Production	No	No
iSeries Common - Test	CLTCOM **	iSeries Common - Test	No	No
Business Data - CRP	CRPDTA	Business Data - CRP	No	No
Business Data - CRP - DNT	CRPDTA	Business Data - CRP - DNT	No	No
Business Data - JDE	PRISTDTA	Business Data - JDE	No	No
Business Data - JDE - DNT	PRISTDTA	Business Data - JDE - DNT	No	No
Business Data - PROD	PRODDTA ***	Business Data - PROD	No	No
Business Data - PROD - DNT	PRODDTA ***	Business Data - PROD - DNT	No	No
Business Data - TEST	TESTDTA	Business Data - TEST	No	No
Business Data - TEST - DNT	TESTDTA	Business Data - TEST - DNT	No	No
Central Objects - DV7333	CODV7333	Central Objects - DV7333	No	Yes
Central Objects - JD7333	COJD7333	Central Objects - JD7333	No	Yes
Central Objects - PD7333	COPD7333	Central Objects - PD7333	No	Yes
Central Objects - PY7333	COPY7333	Central Objects - PY7333	No	Yes
Control Tables - CRP	CRPCTL	Control Tables - CRP	No	No
Control Tables - CRP - DNT	CRPCTL	Control Tables - CRP - DNT	No	No
Control Tables - JDE	PRISTCTL	Control Tables - JDE	No	No

Data source name	Library name	Database name	OCM data source	iSeries BLOB data source
Control Tables - JDE - DNT	PRISTL	Control Tables - JDE - DNT	No	No
Control Tables - Prod	PRODCTL	Control Tables - Prod	No	No
Control Tables - Prod - DNT	PRODCTL	Control Tables - Prod - DNT	No	No
Control Tables - Test	TESTCTL	Control Tables - Test	No	No
Control Tables - Test - DNT	TESTCTL	Control Tables - Test - DNT	No	No
Data Dictionary - B7333	DD7333	Data Dictionary - B7333	No	No
machine *	SVM7333	*machine* - B7333 Server Map *	No	No
machine - Logic *	SVM7333	*machine* - B7333 Server Map *	No	No
machine - B7333 Server Map *	SVM7333	*machine* - B7333 Server Map *	No	No
Object Librarian - B7333	OL7333	Object Librarian - B7333	No	No
OWJRNL	OWJRNL	OWJRNL	No	No
System - B7333	SYS7333	System - B7333	Yes	No
System - B7333 - DNT	SYS7333	System - B7333 - DNT	No	No
Versions - PY7333	PY7333DNT	Versions - PY7333	No	No
Versions - PY7333 - DNT	PY7333DNT	Versions - PY7333 - DNT	No	Yes
Versions - DV7333	DV7333DNT	Versions - DV7333	No	No
Versions - DV7333 - DNT	DV7333DNT	Versions - DV7333 - DNT	No	Yes
Versions - JD7333	JD7333DNT	Versions - JD7333	No	No
Versions - JD7333 - DNT	JD7333DNT	Versions - JD7333 - DNT	No	Yes
Versions - PD7333	PD7333DNT	Versions - PD7333	No	No
Versions - PD7333 - DNT	PD7333DNT	Versions - PD7333 - DNT	No	Yes

Data source name	Library name	Database name	OCM data source	iSeries BLOB data source

* *machine* is the name of your iSeries enterprise server.

** CLTCOM is the J.D. Edwards WorldSoftware default name of the common library. (If you are not using the J.D. Edwards WorldSoftware defaults, substitute the name of your WorldSoftware common library.)

*** In a co-existence environment, Data Sources Business Data - PROD and Business Data - PROD - DNT map to your current WorldSoftware production library, which is CLTDTA, if you are using the J.D. Edwards recommended naming convention for WorldSoftware.

Complete the following steps:

1. Verify the B7333 Server Map source information on the Data Source Setup form. It should look similar to the example in Figure 3-13, except that your iSeries server name appears. Make any necessary changes, and click the **Advanced** button.

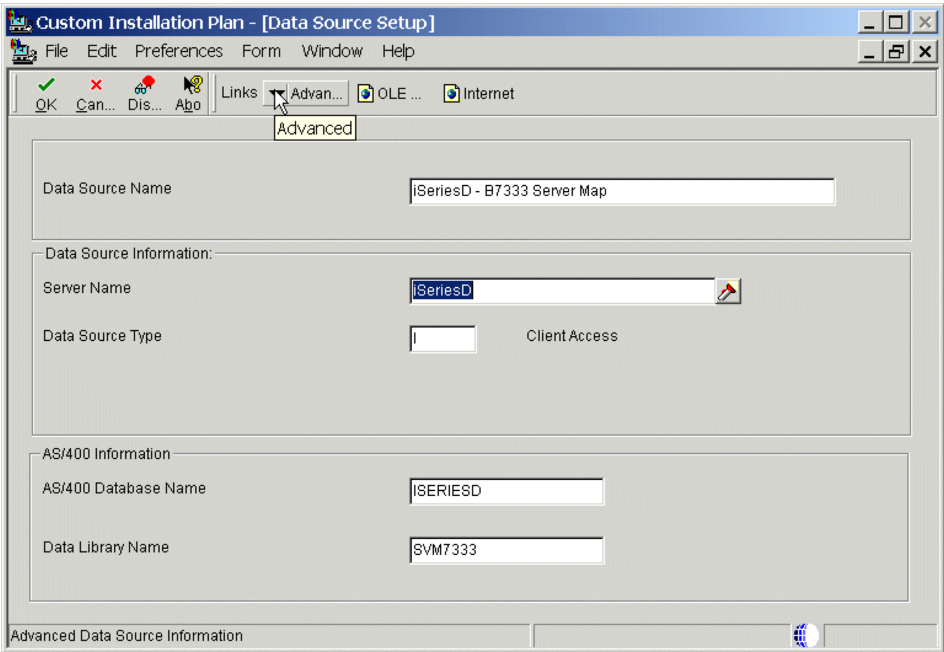


Figure 3-13 OneWorld Data Source Setup

2. Verify the B7333 Server Map advanced settings on the Data Source Revisions form. It should look similar to the example in Figure 3-14, except that your iSeries server name appears. Make any necessary changes, and click **OK**.

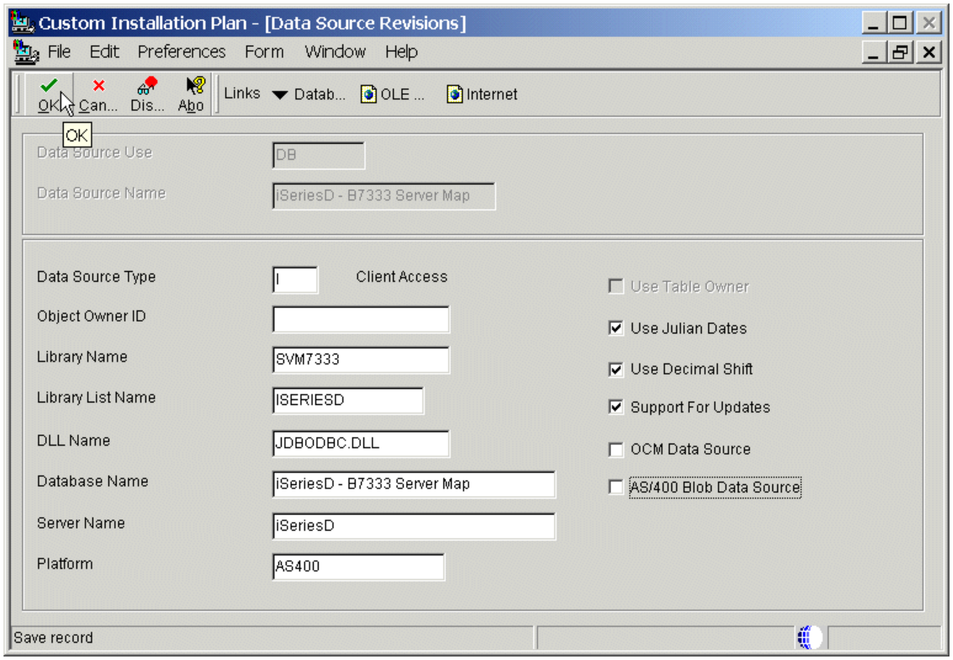


Figure 3-14 OneWorld Data Source Revisions

3. Click **No** when prompted to add another enterprise server. The Machines/Servers Types form appears.

Defining machine/server types

To specify machine/server types, complete the following steps.

1. Deselect **Data Server**.

> **Note:** This prompt states that a data server must be defined for your Central Objects when your enterprise server is an iSeries server. However, this is only the case when the Central Objects are *not* installed in DB2 UDB for iSeries as you are doing.

2. Deselect **JAS Server**, because you are not configuring a Java Application Server.
3. Deselect **WTS Server**, because you are not configuring a Windows Terminal Server.

   Your form should appear similar to the example in Figure 3-15.

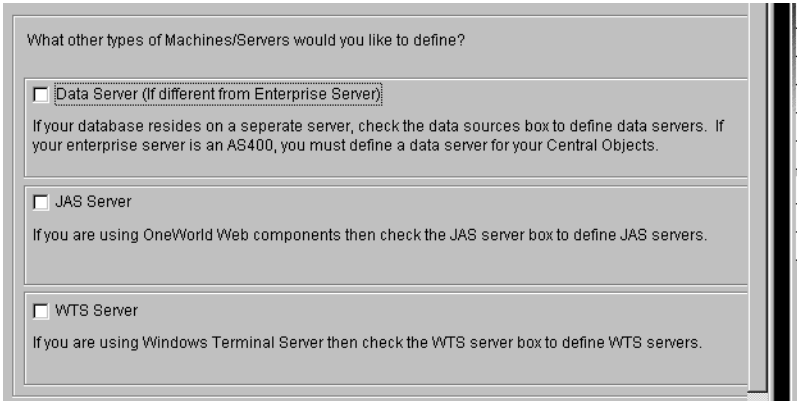


Figure 3-15 OneWorld machine/server types

4. Click **OK** to continue.

Entering shared data source setup information

To specify the shared data source setup information, complete the following steps:

1. Select **OK** when prompted to enter Data Source Information for the shared data sources. The Data Source Setup form for Data Source System - B7333 appears.
2. Verify the System - B7333 source information on the Data Source Setup form. It should look similar to the example in Figure 3-13 on page 93, except the Data Source Name is now System - B7333 and Data Library Name is now SYS7333. Make any necessary changes, and click the **Advanced** button.
3. Verify the System - B7333 advanced settings on the Data Source Revisions form. It should look similar to the example in Figure 3-14, except the Library Name is now SYS7333 and the Database Name is now System - B7333. Select **OCM Data Source**. Make any other necessary changes, based on Table 3-2 on page 91. Click **OK**.
4. The Data Source Setup form for System - B7333 re-appears, similar to the example in Figure 3-13 on page 93. Click **OK** to continue. The Data Source Setup form for Data Source Object Librarian - B7333 appears.
5. Verify the Object Librarian - B7333 source information on the Data Source Setup form. It should look similar to the example in Figure 3-13 on page 93, except the Data Source Name is now Object Librarian - B7333 and Data Library Name is now OL7333. Make any necessary changes, and click the **Advanced** button.
6. Verify the Object Librarian - B7333 advanced settings on the Data Source Revisions form. It should look similar to the example in Figure 3-14, except

the Library Name is now OL7333 and the Database Name is now Object Librarian - B7333. Make any necessary changes, and click **OK**. The Data Source Setup form for Object Librarian - B7333 re-appears, similar to the example in Figure 3-13 on page 93. Click **OK** to continue. The Data Source Setup form for Data Source Data Dictionary - B7333 appears.

7. Verify the Data Dictionary - B7333 source information on the Data Source Setup form. It should look similar to the example in Figure 3-13 on page 93, except the Data Source Name is now Data Dictionary - B7333 and Data Library Name is now DD7333. Make any necessary changes, based on Table 3-2, “OneWorld data sources” on page 91. Click the **Advanced** button.
8. Verify the Data Dictionary - B7333 advanced settings on the Data Source Revisions form. It should look similar to the example in Figure 3-14 on page 94, except the Library Name is now DD7333 and the Database Name is now Data Dictionary - B7333. Make any necessary changes, and click **OK**. The Data Source Setup form for Data Dictionary - B7333 re-appears, similar to what you see in Figure 3-13 on page 93. Click **OK** to continue.

Setting up environments and environment data sources

To specify the environment parameters, complete the following steps:

1. After the Environment Selection form appears, deselect **Default Environments**.
2. Deselect **Default Data Load**.
3. Deselect **Default Advanced Parameters**.

   The form should look similar to the example in Figure 3-16.

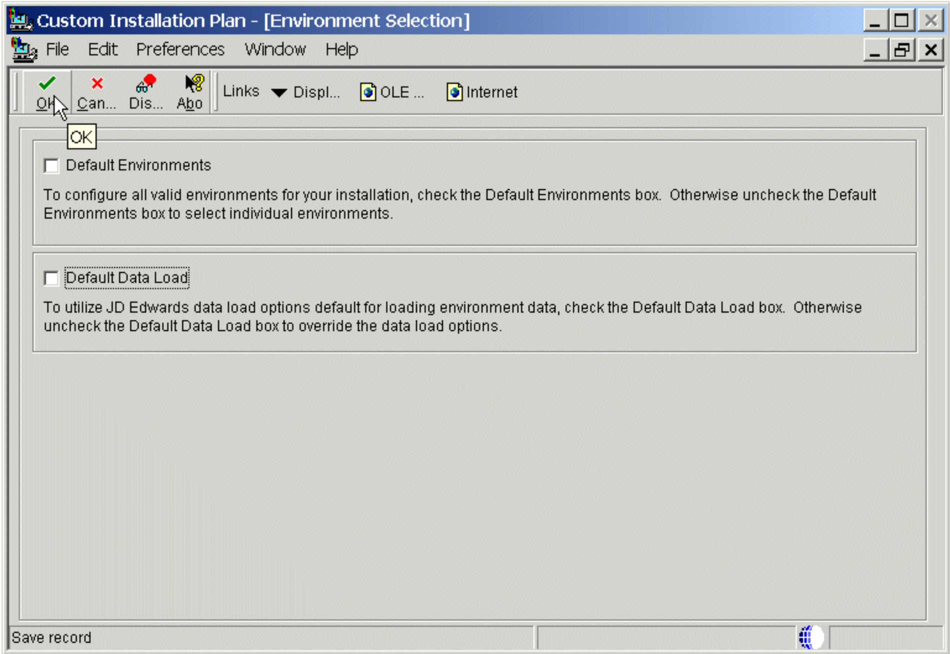


Figure 3-16 OneWorld Environment Selection

4. Click **OK**. The Select Environments form appears.
5. Repeat the following steps for each environment that you are defining:
   a. Double-click the row corresponding to the environment you want to define. The Data Load Parameters form appears.
   b. Select **Load Production Data** for all environments except the Pristine environment.
   c. Select **Load Demo Data** for the Pristine environment only.
   d. The form should look similar to the example in Figure 3-17. Click **OK**. The Advanced Parameters form appears.

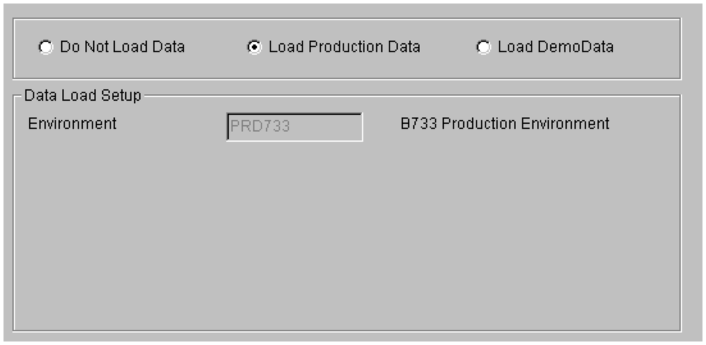


Figure 3-17 OneWorld Advanced Parameters

e. If you are *only* installing OneWorld in a co-existence environment with WorldSoftware, select **Coexistent Environment**, and select **Coexistent Merges**. Deselect **Coexistent Environment**, and deselect **Coexistent Merges** *always* for the Pristine environment.

f. If necessary, deselect **Upgrade Environment** because this is a new installation.

g. Click **OK**.

h. Click **OK** when the enter Data Source Information prompt appears. J.D. Edwards recommends that you do *not* select defaults for the iSeries server. Depending on the environment you select and whether you select co-existence, you now set up the environment data sources for any or all of the data sources: OWJRNL, Control Tables, iSeries Common, Central Objects, Business Data, and Versions. Complete the following steps to set up the environment data sources for each environment:

   i. After the Data Source Setup form for the next data source appears, verify the source information on the Data Source Setup form. It should look similar to the example in Figure 3-13 on page 93, except for the Data Source Name and Data Library Name. Make any necessary changes, based on Table 3-2 on page 91. Click the **Advanced** button.

   ii. Verify the advanced settings on the Data Source Revisions form. Make any necessary changes and click **OK**.

6. The Select Environments form re-appears, showing *Selected* for the environment you set up. If you have another environment that you want to set up, repeat this series of steps for the next environment you want to define.
7. When you are finished defining all of your OneWorld environments, click **Close** on the Select Environments form.
8. Click **No** when prompted to add another location.

9. Click **OK** when you see the message indicating that the plan has been finalized. The Work With Batch Versions - Available Versions form appears for Batch Application R9840B (Planner Validation Report).

Plan Validation

The Plan Validation process is included as an integral part of the Installation Planner application in Xe, and is initiated automatically. This process ensures the plan contains all of the necessary information. The Installation Planner Validation Report (R9840B) should display “Record Validated” in the far right-hand column for each record. If this is not the case, revise the plan until all records are validated prior to finalization.

Running the Planner Validation report

Normally the Planner Validation report is automatically initiated when the plan is finalized. To run the Planner Validation report if it is not automatically initiated, on the tree view of your plan, click **Validate** to initiate the Planner Validation report.

> **Note:** In the Machines section of the R9840B Validation Report, a Validation Result – `FTP Server has not been setup` – may appear for Machine Description iSeries enterprise server. If you can successfully use FTP in both directions between the deployment server and AS/400 enterprise server, you can ignore this error. It does not cause any further issues that the Detail Status remains at 20. You can find this information in the Breaking News item from the J.D. Edwards Knowledge Garden (`http://www.jdedwards.com`).
>
> In the Specifications Table Merge section of the R9840B Validation Report, a Validation Result – `Cnv.Plan Match` – may appear for Environment TS7333. This is an incorrect message, since TS7333 and PY7333 share a common data path. If this same object (for example, F98750) and same target data source (for example Central Objects - PY7333) appear with a Validation Result – `Record Validated for PY7333`, you can ignore this message. It does not cause any further issues so the Detail Status remains at 20.

JDE user profile on the iSeries server

Before you proceed with the next phase of installation, Installation Workbench, you must create a user profile JDE on the iSeries server with at least the special authorities of *JOBCTL and *ALLOBJ. These special authorities are required so the Installation Workbench can run successfully. To create this JDE user profile on the iSeries server, complete the following steps:

1. Sign on to the deployment server as `Administrator`.

2. Click **Start-> Programs-> IBM AS/400 Client Access Express-> AS/400 Operations Navigator**. This starts Operations Navigator for accessing the iSeries server from the deployment server.
3. If necessary, expand **My AS/400 Connections** to see your iSeries enterprise server.
4. Expand your iSeries enterprise server.
5. When prompted for Sign on to the iSeries server, type `QSECOFR` for AS/400 User ID and the corresponding password for Password. Click **OK**.
6. Expand **Users and Groups**.
7. Right-click **All Users**.
8. Click **New User**.
9. Type `JDE` for User name and an appropriate description.
10. For Password, type the same password used for user JDE on the deployment server.
11. Deselect **User must change password at next logon**.
12. Click **Capabilities**.
13. If necessary, select the **Privileges** tab.
14. Select **All object access**.
15. Select **Job control**.
16. Click **OK**.
17. Click **Add** on the New User display.

Installation Workbench

The Installation Workbench application executes the plan created in the Installation Planner phase, by copying tables from the Planner environment into the data sources of the defined environments on the appropriate servers.

Installation Workbench consists of several separate workbench functions that include:

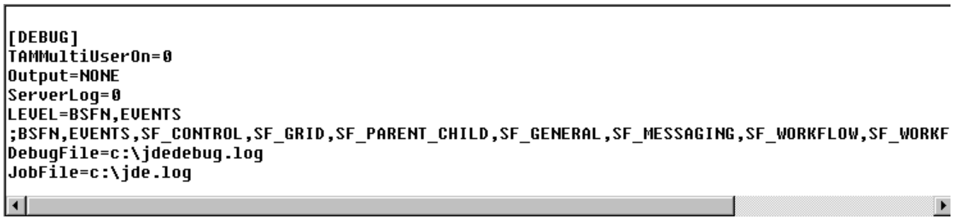
- **Additional Workbench Tasks**: You can complete the following preliminary tasks before the individual workbenches begin:
  - Copy Data Dictionary (upgrade only)
  - Copy system tables (upgrade only)
  - Security copy (upgrade only)
  - Release Master (installation and upgrade)
- **Location Workbench**: Updates the System - B7333 data source based on the deployment locations defined in the plan.

- **Data Source Workbench**: Copies all data sources defined in the plan from the F98611 Data Source Master and F986115 Data Source Sizing tables in the planner data source into the System - B7333 data source.
- **Environment Workbench**: Copies the F0094, F00941, and F00942 environment information tables and the F986101 Object Configuration Manager table into the System - B7333 data source. Depending on the plan's settings, it also runs a batch application that creates tables and indexes for the required environments. Also, depending upon your plan's setting, coexistent merges may be launched at this time.
- **Machine Workbench**: Copies the Server Configuration tables (F9650 and F9651) from the Planner data source to the System - B7333 data source.
- **Package Workbench**: Copies the Package Information tables (F9603 and F9631) from the Planner data source to the System - B7333 data source.
- **Remote Location Workbench**: Transfers the F9885, F9886, and F9887 package information tables to the System - B7333 data source.

Before commencing the Installation Workbench procedure, perform the tasks described in the following sections.

Switching off debugging

We recommend that you switch off debugging during the execution of the Installation Workbench. If it is switched on, it degrades the performance of the process and creates a large log file that wastes disk space. To ensure that this is off, check the jde.ini file located in the \winnt directory. The OUTPUT statement in the DEBUG section should equal NONE (see Figure 3-18).



```
[DEBUG]
TAMMultiUserOn=0
Output=NONE
ServerLog=0
LEVEL=BSFN,EVENTS
;BSFN,EVENTS,SF_CONTROL,SF_GRID,SF_PARENT_CHILD,SF_GENERAL,SF_MESSAGING,SF_WORKFLOW,SF_WORKF
DebugFile=c:\jdedebug.log
JobFile=c:\jde.log
```

Figure 3-18 The jde.ini file (DEBUG section)

Verifying JDE passwords

You should verify that user profile JDE is on your iSeries server. If it is not, you should create it now so that the Installation Workbench can run successfully. You should also verify that it has the same password as you gave to the user profile JDE on the deployment server.

The JDE user profile password is cached on the deployment server. This password is used to access the JDE user profile on the iSeries server during the Installation Workbench process. Therefore, the passwords must be synchronized on both systems to allow successful ODBC connections to be established. If the passwords do not match on the two systems, each attempt to initialize an ODBC connection causes a database sign on window to be issued that asks for the correct iSeries server password to be entered.

Verify that the iSeries server user profile JDE has special authority *JOBCTL and *ALLOBJ. These special authorities are required for the Installation Workbench to run successfully.

Verify that the object authority *CHANGE is assigned to *PUBLIC for the iSeries server user profile JDE. This is required later for OneWorld client workstation users to successfully submit Universal Batch Engines (UBEs) to run on the iSeries server. If necessary, you can change the object authority by using the Edit Object Authority (EDTOBJAUT) command.

Verifying CCSID

> **Attention:** Currently, *only* CCSIDs 37 and 65535 are suitable for use during the installation phase. Therefore, the JDE iSeries user profile must be checked and amended if necessary.

Starting Installation Workbench and configuring your locations

To start the Installation Workbench, complete the following steps:

1. Log onto J.D. Edwards OneWorld on the deployment server as user `JDE` and planner environment `JDEPLAN`.
2. If necessary, type `GH961` on Fastpath and press Enter to display the System Installation Tools menu.
3. Right-click **Installation Workbench.**
4. Choose **Prompt For** and then **Values**. The Processing Options panel appears.
5. In the appropriate field, enter `1` for unattended workbench mode or `0` for attended workbench mode.

   > **Note:** We used attended workbench mode for this example. In unattended mode, the various workbenches proceed in sequence automatically. You must check that each individual task ran successfully at the conclusion of processing. If a task did not complete, you must re-run that task.

6. Click **OK**, and the Work with Installation Plan windows opens.

7. Double-click your installation plan. The Additional Workbench Tasks form appears.

8. Click **Configure**. The display should look similar to the example in Figure 3-19.

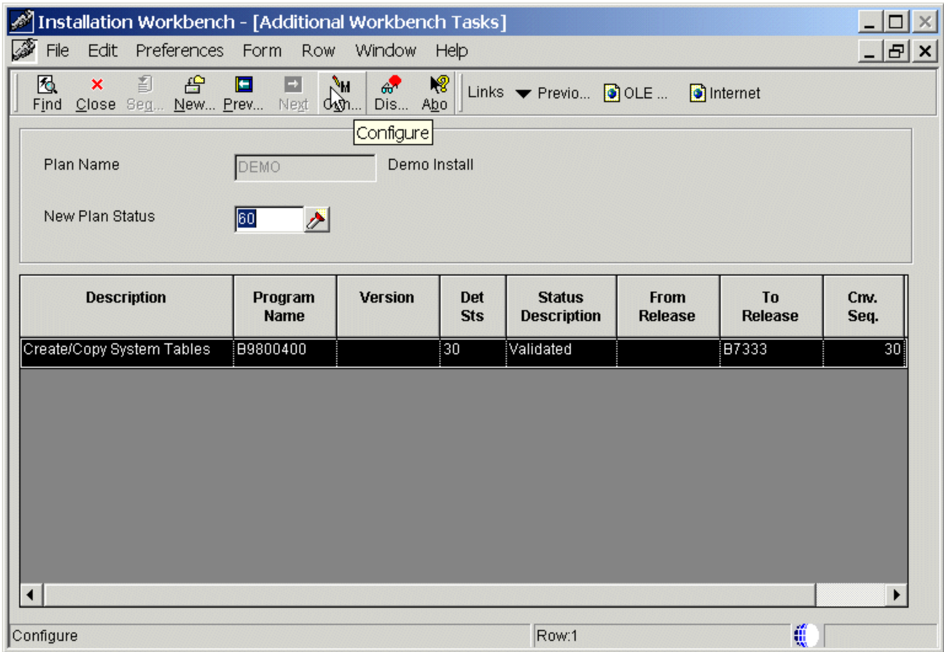


Figure 3-19 Additional Workbench Tasks

9. The Location Workbench form appears. Verify that all of the locations for your plan are listed before continuing.

10. Click **Configure**. Your display should look similar to the example in Figure 3-20.

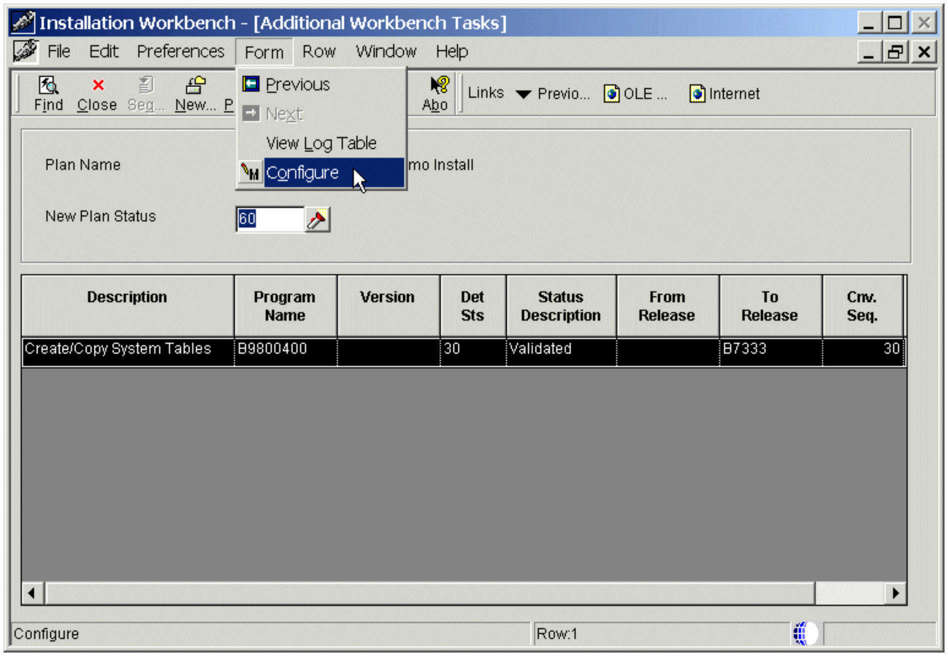


Figure 3-20 Location Workbench

11. The Data Source Workbench form appears listing the required OneWorld data sources for your installation plan.
12. Click **Configure**.
13. The Environment Workbench form appears. Verify that each of your configured OneWorld environments is listed.
14. Click **Configure**. This step may take hours to complete.

> **Note:** It is possible that the Windows 2000 Task Manager function may display a status of "not responding" for OneWorld during the execution of the Installation Workbench. This is a normal condition and may be ignored.

The Machine Workbench appears.

Verifying Installation Workbench

An Environment Database Creation report (R98403) is generated during the execution of the Installation Workbench for each environment defined in the plan. Review these reports for any instances of failure messages.

Note: See Chapter 7 of the *Installation Reference Guide (All Systems)* that provides additional details of acceptable errors.

If your R98403 report shows errors, you need to review and resolve them before you continue. What needs to be done to resolve the problem depends on the type of error.

Configuring your machines

To configure your machines, perform the following steps:

1. Click **Next** on the Environment Workbench form. The Machine Workbench form appears.
2. Click **Configure** on the Form menu.

   When OneWorld configures your servers, the Detail Status becomes 60 and the Status Description becomes *Installed*.
3. Click **Next**.

Configuring your packages

To configure your packages, perform following steps:

1. The Control Table Workbench form appears, showing a list of your packages.
2. Click **Configure** on the Form menu.

   When OneWorld configures your packages, the Detail Status becomes 60 and the Status Description becomes Installed.
3. Click **Next**.

Completing the Installation Workbench

To complete the Installation Workbench, perform the following steps:

1. A message like the one shown in Figure 3-21 appears, indicating the successful completion of the Installation Workbench.
2. Click **Finish** on the Form menu.

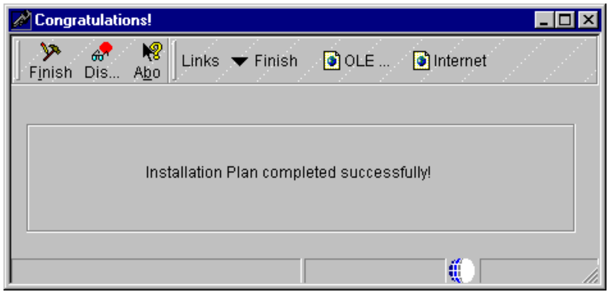


Figure 3-21 Installation Plan completion message

3. The Installation Workbench form re-appears, showing your package with an updated Plan Status of 60 and Status Description of *Installed* (Figure 3-22).

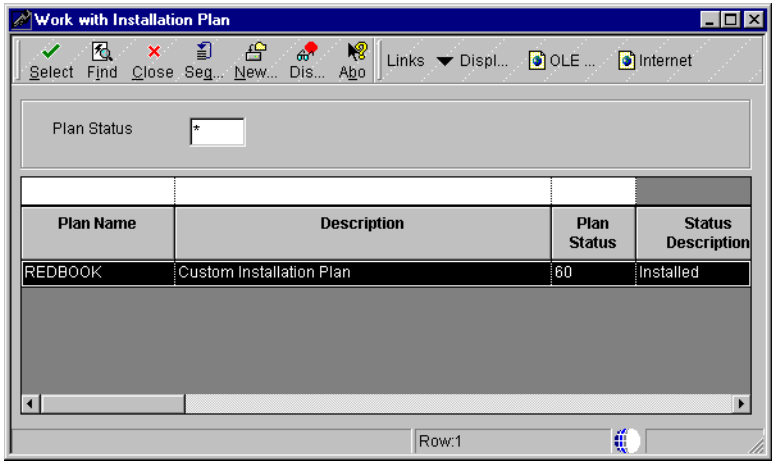


Figure 3-22 The finished Installation Plan

4. Click **Close**.

3.1.4 Enterprise server installation

During this phase, the iSeries enterprise server is configured and OneWorld objects are created along with associated IFS object links. Many of these installation tasks are also required on any other server that you plan to include in your client/server environment to process OneWorld logic.

At Release B73.3, the majority of OneWorld objects are installed onto the enterprise server directly from the installation CD-ROMs. This reduces the volume of network traffic generated during the installation phase in comparison to earlier releases and subsequently reduces installation time. Objects containing OneWorld configuration information are still transferred from the deployment server to the enterprise server using FTP.

The host installation process loads the OneWorld services programs and the appropriate path code libraries. Each path code on the iSeries server is comprised of specification files located in the Integrated File System (for example /PD7333/SPECFILE/*.*) along with business function programs located in a path code library, for example PD7333. OneWorld services programs are restored into the SYS7333 library.

Before the iSeries enterprise server installation phase is started, several configuration tasks must be performed on the iSeries server and the deployment server. These are described in the following sections.

Setting the iSeries system values

A number of iSeries system values must be verified on the iSeries server and amended as necessary. This can be done by typing the Work with System Values (WRKSYSVAL) command on an OS/400 command line followed by pressing Enter. Then enter option `2` to change an individual system value. The values are:

- **QSECURITY**: J.D. Edwards recommends that you do not specify an OS/400 security level of 50 on a system running OneWorld, because the application will not function correctly. Therefore, if the QSECURITY system value is currently set to 50, it should be lowered to `40`. An IPL is required for this change to take affect.

  > **Note:** The security level of your system should be ascertained during the initial configuration planning. If it is found to be set at 50, a review of your system security policy should be performed to verify that lowering it to 40 would not compromise the security requirements of your system.

- **QJOBMSGQFL**: This system value dictates what action is taken when a job's message queue becomes full. We recommend that you set this value to `*WRAP` or `*PRTWRAP` to ensure that the batch jobs submitted do not fail due to insufficient message queue space.

  > **Note:** This setting must be retained permanently.

- **QALWOBJRST**: This value must be set to `*ALL` for the duration of the installation process to allow programs adopting QSECOFR authority to be restored onto the system. This system value can be reverted to its original setting after the installation has completed successfully.
- **QQRYDEGREE**: The QQRYDEGREE system value controls the system-wide usage of parallel processing. Set this system value to `*OPTIMIZE`. This divides many database functions into separate tasks for simultaneous processing by one or more processors.

> **Note:** The DB2 Symmetric Multiprocessing for iSeries option of OS/400 is a prerequisite for the activation of SMP.

Creating OneWorld user profiles

The *Installation Guide (AS/400 Systems) Xe* states that an iSeries server user profile with *JOBCTL authority should be created at this time for each OneWorld user that will access the OneWorld iSeries server. Although this needs to be done, it can wait until the completion of the enterprise server OneWorld installation. A user profile ONEWORLD is created during the enterprise server installation process. It can be copied to these new OneWorld user profiles if it is done after the completion of the enterprise server installation.

Starting TCP/IP on the iSeries server

Once TCP/IP is correctly configured on the iSeries server, it is necessary to start it. You can do this using the Start TCP/IP (STRTCP) command. The required TCP/IP servers should automatically start as a result of the STRTCP command. If they do not, you can use the Start Host Servers (STRHOSTSVR) command to do this. For more information, refer to Chapter 7 in *J.D. Edwards OneWorld Implementation for AS/400*, SG24-5195.

These tasks need to be accomplished each time your iSeries server is started (IPLed). You can do this manually, write a Control Language (CL) program, or modify the iSeries server Startup Program (as specified in system value QSTRUPPGM).

Amending the iSeries server startup program

If you want the OneWorld services to start automatically at IPL time, the iSeries server startup program must be amended to include the STRTCP *ALL and STRNET commands.

> **Note:** In V4R3 and later releases, the STRHOSTSVR *ALL command is integrated into the STRTCP command.

We recommend that you modify the iSeries server startup program as explained here:

1. Use the Display System Value (DSPSYSVAL) command for system value QSTRUPPGM to determine the name of the startup program for your iSeries server:

   ```
   DSPSYSVAL QSTRUPPGM
   ```

2. Use the Retrieve Control Language Source (RTVCLSRC) command to obtain the current source code for your iSeries server startup program:

   ```
   RTVCLSRC PGM(QGPL/QSTRUP) SRCFILE(QGPL/QCLSRC)
   ```

3. Use the Edit File (EDTF) command to modify your iSeries server startup program by adding the STRTCP, STRHOSTSVR, and STRNET commands. Your resulting startup program source should include Control Language statements similar to those shown in Figure 3-23.

4. Recompile your iSeries server startup program using the Create Control Language Program (CRTCLPGM) command:

   ```
   CRTCLPGM PGM(QGPL/QSTRUP) SRCFILE(QGPL/QCLSRC) SRCMBR(QSTRUP) TEXT('iSeries
   server Startup Program')
   ```

```
 PGM
     DCL VAR(&STRWTRS) TYPE(*CHAR) LEN(1)
     DCL VAR(&CTLSBSD) TYPE(*CHAR) LEN(20)
     DCL VAR(&CPYR) TYPE(*CHAR) LEN(90) VALUE('5716-SS1 (C) COPYRIGHT-
 IBM CORP 1980, 1996. LICENSED MATERIAL - PROGRAM PROPERTY OF IBM')
     QSYS/STRSBS SBSD(QSPL)
     MONMSG MSGID(CPF0000)
     QSYS/STRSBS SBSD(QUSRWRK)
     MONMSG MSGID(CPF0000)
     QSYS/RLSJOBQ JOBQ(QGPL/QS36MRT)
     MONMSG MSGID(CPF0000)
     QSYS/RLSJOBQ JOBQ(QGPL/QS36EVOKE)
     MONMSG MSGID(CPF0000)
     QSYS/STRCLNUP
     MONMSG MSGID(CPF0000)
     QSYS/RTVSYSVAL SYSVAL(QCTLSBSD) RTNVAR(&CTLSBSD)
     IF COND((&CTLSBSD *NE 'QCTL      QSYS      ') *AND (&CTLSBSD *NE-
 'QCTL      QGPL      ')) THEN(GOTO CMDLBL(DONE))
     QSYS/STRSBS SBSD(QINTER)
     MONMSG MSGID(CPF0000)
     QSYS/STRSBS SBSD(QBATCH)
     MONMSG MSGID(CPF0000)
     QSYS/STRSBS SBSD(QCMN)
     MONMSG MSGID(CPF0000)
     QSYS/STRSBS SBSD(QSNADS)
     MONMSG MSGID(CPF0000)
     STRTCP
     MONMSG MSGID(CPF0000)
     STRHOSTSVR SERVER(*ALL)
     MONMSG MSGID(CPF0000)
     ADDLIBLE JDEOW
     STRNET
     MONMSG MSGID(CPF0000)
DONE:
     QSYS/RTVSYSVAL SYSVAL(QSTRPRTWTR) RTNVAR(&STRWTRS)
     IF COND(&STRWTRS = '0') THEN(GOTO CMDLBL(NOWTRS))
     CALL PGM(QSYS/QWCSWTRS)
     MONMSG MSGID(CPF0000)
NOWTRS:
     RETURN
     CHGVAR VAR(&CPYR) VALUE(&CPYR)
     ENDPGM
```

Figure 3-23 iSeries server startup program source

Changing the host requirement for Client Access Express

Subsystem QUSRWRK runs the host server job for the Client Access Express file serving and database serving functions. To ensure better performance of Client Access Express, perform the following steps:

1. Ensure that the automatic performance adjuster has been turned on. On an OS/400 command line, type:

   ```
   WRKSYSVAL QPFRADJ
   ```

   Press Enter. Type option 5 to display the value. If the value of QPFRADJ is either 2 or 3, no changes are required. Otherwise, enter option 2 to change this system value to either 2 or 3.

2. Put the QUSRWRK subsystem in its own memory pool:

   a. Using the Work with Shared Storage Pools (WRKSHRPOOL) command, determine a shared pool that has not been defined yet. Select an unused shared pool (for example, *SHRPOOL6).

   b. Assign some memory (for example, 50 MB) by typing this number in the Defined Size (M) field. For each active OneWorld user, allocate approximately 15 MB of memory.

   c. Assign an activity level (for example, 10) in the Max Active field. Press Enter. The amount of memory assigned and the setting for the activity level are not critical since the performance adjuster has been turned on. Adjustments will be made by the iSeries server as required.

   d. Change the QUSRWRK subsystem description to use the shared pool that you just defined. For example, on an OS/400 command line, type the following command, and press Enter:

      ```
      CHGSBSD SBSD(QUSRWRK) POOLS((1 *SHRPOOL6))
      ```

   All new jobs started in the QUSRWRK subsystem will now use the new memory pool.

3. Change the QPWFSERVER class (used by Client Access Express jobs) Eligible to Purge setting to *YES.

   This should only be done if there is limited memory on your iSeries server. IBM ships QPWFSERVER with Eligible to Purge set to *NO. The maximum CPU time and Maximum temporary storage options should already be set to the recommended *NOMAX. However, if you need to make these changes, on an OS/400 command line, type:

   ```
   CHGCLS CLS(QPWFSERVER) PURGE(*YES) CPUTIME(*NOMAX) MAXTMPSTG(*NOMAX)
   ```

   Press Enter.

   For more information on these activities, refer to Chapter 7 in *J.D. Edwards OneWorld Implementation for AS/400*, SG24-5195.

Setting the number of prestart jobs in QUSRWRK subsystem

The number of prestart jobs for program QZDAINIT depends on the number of concurrent and active ODBC user connections. If you do not have enough prestart jobs, your system may be slow to make connections. Each OneWorld user typically uses three to five ODBC connections. Each ODBC connection uses one QZDAINIT program. Therefore, the initial value for QZDAINIT should be set to approximately four times the number of expected concurrent connected users. The programs QZSCSRVS and QZSOSIGN are used for making the connection. Therefore, the initial value for QZSCSRVS and QZSOSIGN should be the number of expected concurrent connected users.

Since the determination of the correct number to specify involves a degree of approximation, we recommend that you review this setting at a time when a typical number of users are accessing the system.

It is important to keep in mind that the higher the number of prestart jobs specified, the greater the performance impact is on the system at IPL time or QUSRWRK subsystem startup time.

For in-depth information regarding ODBC performance, refer to Section 21.2.1.1 in *J.D. Edwards OneWorld Implementation for AS/400*, SG24-5195.

Complete the following steps to make these changes for QZDAINIT. Repeat the same steps, while changing the program name, to make the changes for QZSCSRVS and QZSOSIGN:

1. To see the current settings, type the following command and press Enter:

   ```
   DSPSBSD QUSRWRK
   ```

   Enter option `10` to view the Prestart job entries. Enter option `5` for program QSYS/QZDAINIT to display the details to determine the current settings.
2. To change the current settings, on an OS/400 command line, type:

   ```
   CHGPJE SBSD(QSYS/QUSRWRK) PGM(QSYS/QZDAINIT)
   ```

   Press F4 to prompt. Make the appropriate changes and press Enter.

For the above changes to take effect, the QUSRWRK subsystem needs to be restarted. This can be accomplished by using the following steps:

1. To end the QUSRWRK subsystem, on an OS/400 command line, type the following command and press Enter:

   ```
   ENDSBS QUSRWRK *IMMED
   ```
2. To restart the QUSRWRK subsystem, type the following command on an OS/400 command line and press Enter:

   ```
   STRSBS QUSRWRK
   ```

3. To restart TCP/IP, type the following command on an OS/400 command line and press Enter:

```
STRTCP
```

4. To restart the required iSeries server host server programs, type the following command on an OS/400 command line and press Enter:

```
STRHOSTSVR
```

Modifying the enterprise server jde.ini file

We recommend that you make the following modifications to the enterprise server's jde.ini file on the deployment server. This file is located in the \JDEdwardsOneWorld\B7333\hosts\as400*system name*\ directory, where *system name* is the name of your iSeries server.

- **Security**: Amend the DefaultEnvironment statement in the SECURITY section of the jde.ini file to specify a valid environment value (for example, PD7333). Deactivate the SecurityServer statement by adding a semicolon (;) in front of the statement for the duration of all installation and configuration processes. Activate it again when you are certain that the installation is successfully completed.
- **Remove unwanted path codes**: Any statements in the jde.ini file relating to path codes that were not installed must be altered to reflect installed path code values. This avoids the production of error messages.
- **Update Server Name**: Any statements in the jde.ini file relating to the name of the server must be changed, if necessary, to a valid server name. The name of the server should not be left blank, even if the statement has been commented with a semicolon (;) in front of the statement. This avoids the production of error messages.

Checking the LIBRARY file

The LIBRARY file located in the \JDEdwardsOneWorld\B7333\hosts\as400*system name*\ directory should include all the path codes that are to be installed. This file is used during the enterprise server installation to determine which path codes are to be installed on the enterprise server.

> **Note:** The first entry in this file should *always* be B7333SYS.

Installing OneWorld on the iSeries enterprise server

Installing the iSeries enterprise server populates the libraries on the iSeries server. To install the enterprise server, perform the following steps:

1. Sign on the iSeries server as user `QSECOFR`.
2. Insert the OneWorld iSeries Server Direct Enterprise Server Install 1 of 7 CD in the iSeries CD-ROM drive.
3. On an OS/400 command line, type the following command and press Enter:

   ```
   ADDLIBLE JDEOW
   ```

4. On an OS/400 command line, type the following command and press Enter:

   ```
   GO JDEOW/A98OWMNU
   ```

5. The menu shown in Figure 3-8 on page 79 appears. Type option `4` and press Enter to install the OneWorld server libraries. This includes library SYS7333, the path code (PD7333, PY7333, DV7333, JD7333) libraries as required, and the appropriate Integrated File System (IFS) directories. The Install OneWorld Libraries display (Figure 3-24) appears.

```
                Install OneWorld Libraries (C98OWINST)

 Type choices, press Enter.

 Device Name  . . . . . . . . . .   OPT01        Character value
 Release Number . . . . . . . . .    B7333
 Deployment Server Name . . . . .
 OneWorld Deployment Path . . . .   '?:\JDEdwardsOneWorld\B7333'
 Deployment Server User ID  . . .   JDE          Character value
 Deployment Server Password . . .                Character value
 Change Messages to Break Mode  .   *YES         *YES *NO










                                                                   Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 3-24 Installing OneWorld server libraries on the iSeries server

6. If necessary, change the Device name to the same name as your iSeries server CD-ROM drive (this is usually `OPT01`).
7. Type `B7333` for Release Number.
8. Type the name of your deployment server.
9. Type the drive letter (for example, `K`) where OneWorld is installed on your deployment server, replacing the question mark in OneWorld Deployment Path.
10. Type `JDE` for the Deployment Server User ID.
11. Type the password for user ID JDE in Deployment Server Password.
12. Type `*YES` in Change Messages to Break Mode.
13. Press Enter.

    The installation process submits the J98OW20 job to batch. Installation status messages may appear on the display.

14. When you are prompted for the next CD in the series, insert the CD in the iSeries server CD-ROM drive. When the iSeries server CD-ROM light stops flashing, type `G` in reply to the message, and press Enter.

    The fourth CD may require a long time to process (may be hours depending on the number of OneWorld path codes that you are installing). While the installation is running, you can perform the steps outlined in “Creating output queues” on page 115 and in “Creating iSeries server job queues” on page 116.

15. To monitor the status of the installation, on an OS/400 command line, type the following command and press Enter:

    ```
    WRKSBMJOB
    ```

    Find the job **J98OW20**. When the status of the job changes to OUTQ, the installation is complete. To confirm that the installation is successful, enter option `8` to view the spooled files. Find the spooled file QPJOBLOG, and enter option `5` to display it.

Creating output queues

The OneWorld output queues are created at this stage by entering option `7` from the JDEOW/A98OWMNU menu and pressing Enter. The Create OneWorld Outqueue display (Figure 3-25) appears.

```
                        Create OneWorld Outqueue (C98OWOUT)

 Type choices, press Enter.

 Outqueue Name  . . . . . . . . .                Character value
 Printer IP Address . . . . . . .
 Remote Printer Queue . . . . . .   ' '















Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 3-25 Creating a OneWorld output queue on the iSeries server

Type your desired output queue name, the appropriate printer IP address for your printer and, if necessary, remote printer queue. Then, press Enter.

This creates a remote output queue with a TCP/IP connection. The value to be entered for the Remote printer queue parameter is specific to the make and model of printer to which the output queue is associated. This information should be available in the printer's handbook or alternatively from the manufacturer.

Creating iSeries server job queues

Separate job queues can be created for each OneWorld module. For example, the following job queues can be created:

- JDEFIN
- JDEHRM
- JDEDST
- JDEMFG
- JDETECH

The actual names for these job queues should adhere to any naming conventions in place on your system. These are added to the QBATCH subsystem, or the alternative subsystem you chose for OneWorld batch processing.

Note: J.D. Edwards recommends that these job queues be single-threaded. That is to say the MAXACT parameter in a job queue entry should be set to 1. Adopting this setting removes the possibility of resource contention during the execution of concurrent UBEs within the same vertical OneWorld module.

Attention: At Release B73.3, the allocation of enterprise server job queues for UBE processing was changed to bring it in line with all other supported platforms. The software no longer relies solely on OS/400 work management concepts for the allocation of job queues. Instead, it now refers to a statement in the jde.ini file on the client system for details of which queue to submit UBEs into.

The OneWorld job queues are created at this stage by entering option 8 from the JDEOW/A98OWMNU menu and pressing Enter. Then the Create OneWorld Job Queue display (Figure 3-26) appears.

```
                    Create OneWorld Job Queue (C98OWJOB)

Type choices, press Enter.

Job queue  . . . . . . . . . . .   QB7333 Character value
Sequence number  . . . . . . . .

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 3-26 Create OneWorld Job Queue on the iSeries server

Type your desired job queue name and sequence number and press Enter.

You need to determine beforehand the value to be entered for the Sequence number parameter. Enter option `6` on the JDEOW/A98OWMNU menu and press Enter. This shows you a list of sequence numbers that are already used for the job queue so you can select another number.

Verifying SETUP.INF files on workstations

Installation Guide (AS/400 Systems) Xe suggests that you modify the SETUP.INF file by adding a line `AllowDataCompression = 1` for each Client Access data source listed in the ODBCDatasource.inf file in the \JDEdwardswsOneWorld\B7333\client directory. We found that this entry is automatically added during the installation process. You should verify that this line has been added to each Client Access data source in your ODBCDatasource.inf file. If it is not, add it.

Verifying the enterprise server installation

There are several tasks that you should perform to ascertain the success of the OneWorld enterprise server process. These tasks are described in the following sections.

Reviewing job logs and objects

When all of the batch jobs submitted by the host load process are completed, check the job logs for any errors.

Checking object counts

View the OneWorld libraries on the iSeries server to ensure that they contain the correct number of objects. These numbers should be equal for all path codes. The number of IFS object links should also be checked and again should be equal for all path codes.

The number of objects and IFS object links may vary between releases. Contact J.D. Edwards for the exact numbers that are relevant to the release of OneWorld that you are installing.

Performing PORTTEST

PORTTEST is a J.D. Edwards-supplied application that performs environment initialization and verification. It may be run on the enterprise server as a post-installation verification process for all installed environments:

1. Sign on the iSeries server as user `ONEWORLD`.
2. Enter option `1` for the version of OneWorld that you want to verify.
3. Type the following command, and press Enter:

   ```
   GO JDEOW/A98OWMNU
   ```

   The OneWorld Installation Menu appears as shown in Figure 3-8 on page 79.

4. For each OneWorld environment that you want to verify, complete the following steps:
   a. Enter option `12` to run PORTTEST.
   b. Type `JDE` for user ID, the password for JDE, and the environment that you want to test (for example, `PY7333`, `DV7333`, `PD7333`, `JD7333`, `TS7333`). Press Enter.

Any failures during the processing of this application may suggest a problem with the configuration of the selected OneWorld environment. There are a number of conditions that may cause PORTTEST to fail. These are listed in the PORTTEST Checklist at the end of Chapter 8 of the *Installation Guide (AS/400 Systems) Xe*.

As a general rule, before re-attempting PORTTEST after a failure, perform the following procedure on the enterprise server:

1. End OneWorld services with the command:

   ```
   ENDNET
   ```

2. Clear inter-process communications:

   ```
   CLRIPC
   ```

3. Sign off the iSeries enterprise server.
4. Sign on as user `ONEWORLD`.
5. Start OneWorld services with the command:

   ```
   STRNET
   ```

Starting the OneWorld iSeries enterprise server

Start OneWorld services to ensure that all functions begin as expected. Complete the following steps:

1. Sign on the iSeries server as user `ONEWORLD`.
2. Enter option `1` for the version of OneWorld that you want to start.
3. Type the following command and press Enter:

   ```
   GO JDEOW/A98OWMNU
   ```

   The OneWorld Installation Menu appears as shown in the example in Figure 3-8 on page 79.
4. Enter option `14` to end the JDE server.
5. Enter option `15` to clear the IPC.
6. Enter option `13` to start the JDE server.
7. Enter option `17` to display the active jobs.

8. Verify that there is an entry NETWORK with a program PGM-JDENET_N and status SELW.
9. Verify that there is an entry SENTINEL with a program PGM-MONITOR and status SIGW.

This starts the JDEB7333 subsystem and the associated SENTINEL, NETWORK, and JDENET autostart jobs. These OneWorld services must be active before PORTTEST can be executed, if the OneWorld Security Server or JDBNet processes have been configured for use.

Setting up a OneWorld printer

When the enterprise server installation is successfully completed, it is necessary to configure OneWorld printers. Perform the following steps:

1. Sign on the deployment server as user `JDE` in the deployment (`DEP7333`) environment.
2. Type `GH9013` in Fast Path, and press Enter to go to the OneWorld System Administration Menu.
3. Double-click **Printers** to view the printers menu as shown in Figure 3-27.

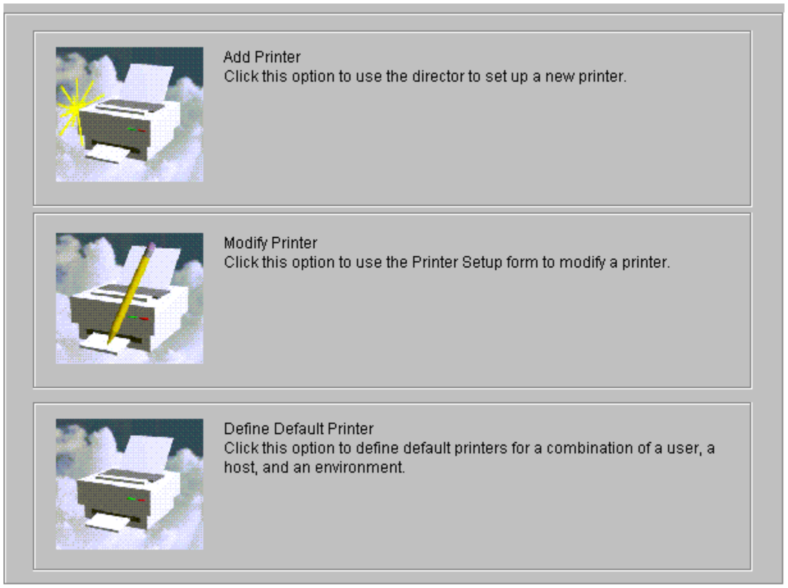


Figure 3-27 OneWorld printers menu

4. Click the Add Printer icon (left of the description).
5. Click **OK** on the Printer Welcome form.

6. Type `AS400` when prompted for Platform type. Click **Next**.
7. When prompted for printer information, type `QUSRSYS` for Library Name. Type the name of the output queue for output queue name corresponding to the output queue you created. Click **Next**. You are prompted for the location and model of the printer.
8. Use the visual assist to select your printer model (for example, Laser Printer, Line Printer). Type the location of the printer.
9. Double-click each row corresponding to supported paper types for this printer (you see a pink check mark next to the row).
10. Type `1` in Default Type for the paper type that is the default for this printer. This needs to be done even if there is only one supported paper type. Your display should look similar to the example in Figure 3-28.

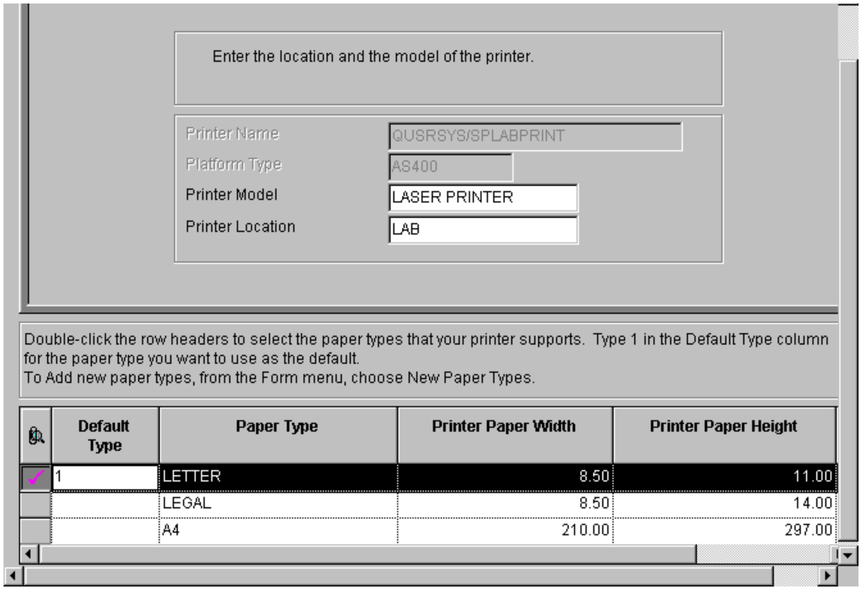


Figure 3-28 OneWorld printer model and location form (General tab)

11. Click the **Details** tab.
12. Click the appropriate Printer Definition Language for your printer. Your display should appear similar to the example in Figure 3-29.

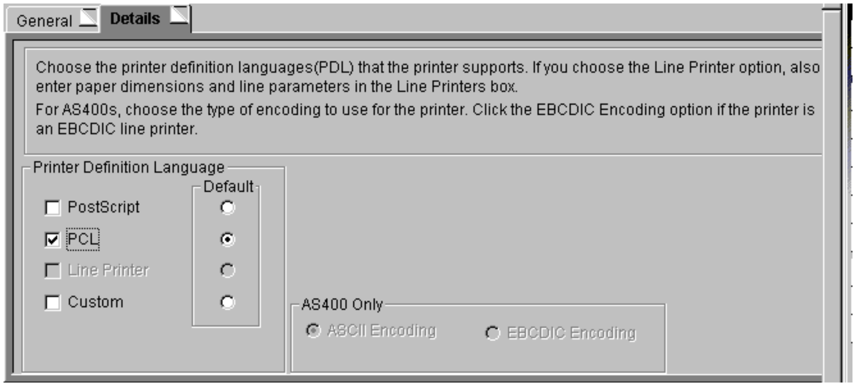


Figure 3-29 OneWorld printer model and location form (Details tab)

13. Click **End**.

3.1.5 Configuring and installing client workstations

During this phase, OneWorld user profiles are created. OneWorld objects and client configuration files are copied to client workstations.

The workstation installation and package deployment topic is very involved. It is covered in great detail in the *OneWorld Package Management Guide.*

The following sections provide a high-level overview of the related tasks required at this stage.

Creating user profiles and group profiles

OneWorld group profiles and user profiles must be created to allow environment and package deployment characteristics to be assigned prior to software deployment to client workstations. Each individual user can have access to their required environments. As an alternative, their valid environments can be determined from the group profile they are a member of, if any.

Note: Assignments made on individual profiles always override those made on the group.

Installing OneWorld objects on client workstations

After OneWorld user profiles are created and assigned one or more deployment packages, they can be used to perform the workstation installation of OneWorld objects.

J.D. Edwards recommends that the initial installation of OneWorld objects on each client system is performed using the “Interactive” mode of installation. This allows the installer to select which deployment package to use for installation from those available to the user.

Running the setup.exe program located in the \client directory on the deployment server at a workstation initiates the OneWorld client installation process. This process must be repeated for each package (environment) that you want to install on each workstation, since only one package at a time is installed.

Before you install any packages on a new workstation, you must to sign on the workstation as `Administrator`. Then you create a new user JDE to be used for installing the client workstation. This is the same process as described in “Configuring the OneWorld deployment server” on page 70.

Sharing the B7333 directory on the deployment server

We use the shared drive approach to installing the client workstation. To use a shared drive, complete the following steps on the deployment server before you install the first client workstation:

1. Using Windows 2000 Explorer, locate and right-click the **\JDEdwardsOneWorld\B7333** directory.
2. Click **Properties**.
3. Select the **Sharing** tab.
4. Click **Shared As**. Type `B7333` for Share Name. Click **Maximum Allowed** for User Limit. Your display should look similar to the example in Figure 3-30.

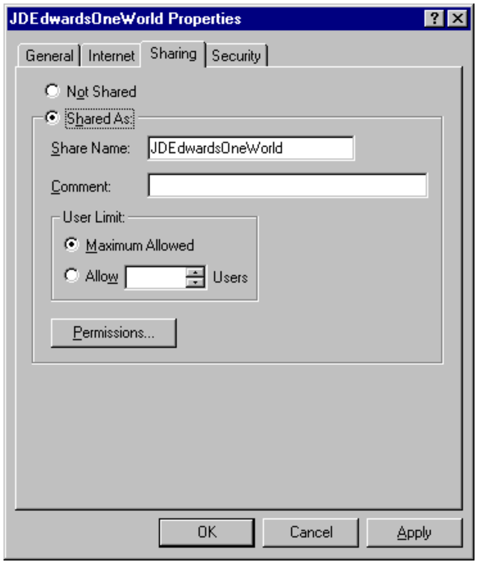


Figure 3-30 Deployment server directory sharing

5. Click **OK**.

Installing OneWorld in the interactive mode

The OneWorld client installation process may not run correctly if other applications are open. Therefore, ensure that all other applications (including virus checker, if applicable) are closed and disabled on the workstation.

Since we are using the shared drive approach to install the client workstation, it is necessary to map a network drive on the workstation. Complete the following steps:

1. On the Windows Explorer display on the workstation, click **Tools**.
2. Click **Map Network Drive**.
3. Select a drive letter on your workstation that you want to use to refer to the deployment server OneWorld directory.
4. For Path Name, type:

   ```
   \\server name\JDEdwardsOneWorld\B7333
   ```

   Here *server name* is the name of your deployment server.
5. Select **Reconnect at Logon**. Your display should appear as shown in Figure 3-31.

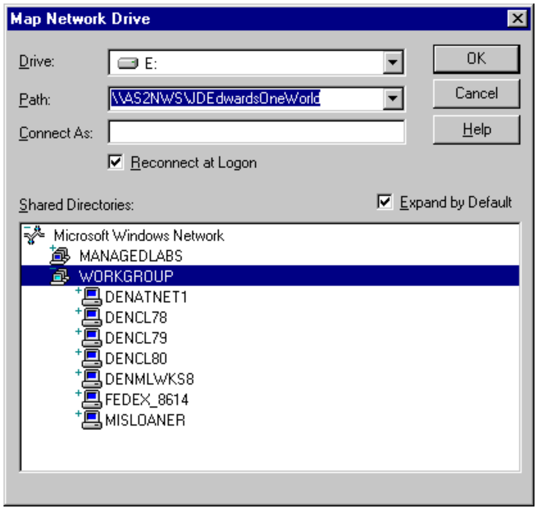


Figure 3-31 Map Network Drive on workstation

6. Click **OK**.
7. Expand **B7333** on the mapped network drive.
8. Select **OneWorld Client Install** directory.
9. Double-click **InstallManager.exe**.
10. Click **Workstation Install** on the OneWorld Installation Manager form.
11. Click **Next** on the Client Workstation Setup form.
12. If you do not have the latest versions of Microsoft Internet Explorer or Adobe Acrobat Reader installed, the OneWorld Client Workstation Setup Third Party Application form appears, similar to the example in Figure 3-32. Complete the following steps:
    a. Verify that the third-party application is selected. Click **Next**.
    b. If you receive a warning during the installation of Acrobat 4.0 similar to Figure 3-33, click **OK**. Then exit the installation, remove the old release of Acrobat, and restart the client OneWorld installation.
    c. Click **Next** on the Acrobat Reader 4.0 Setup form.
    d. Click **Accept** on the Software License Agreement form.
    e. On the Choose Destination Location form, if you want to change the location where Acrobat Reader 4.0 will be installed, select **Browse** and select the desired location.
    f. Click **Next**.

g. Click **OK** when the `Thank you for choosing Acrobat Reader` message is displayed.

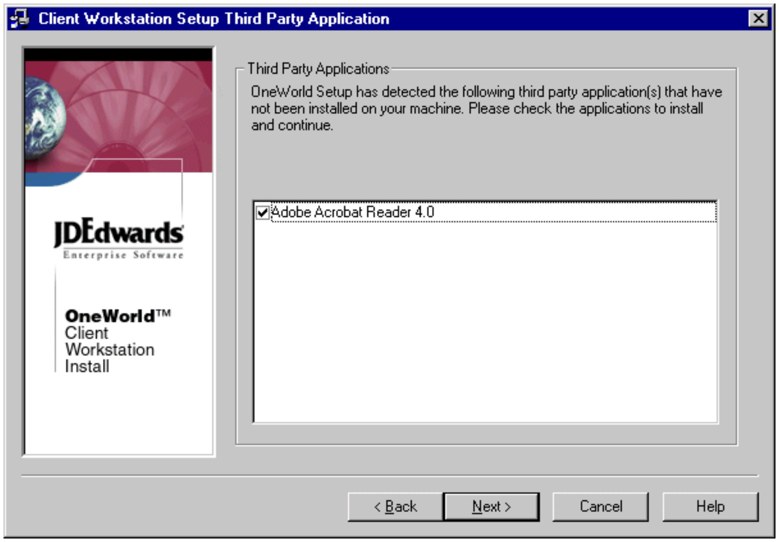


Figure 3-32 Client Workstation Setup Third Party Application installation

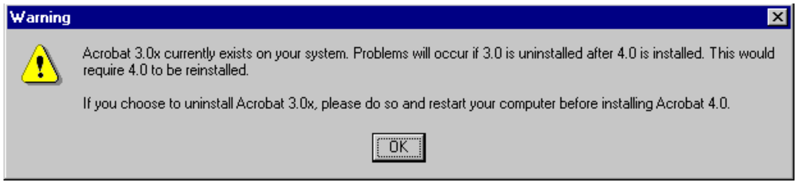


Figure 3-33 Acrobat 4.0 installation warning

13. Select the package that you want to install on the Client Workstation Setup Package Selection form. Click **Next**.
14. If you plan to develop applications at this workstation, click **Development** on the Client Workstation Setup Type form. Otherwise, click **Production**.
15. To choose the disk drive on your client workstation where OneWorld is installed, click the ellipsis (**...**). The directory path *must* be \B7 on whichever disk drive you choose.
16. Click **Finish**.

When OneWorld is finished installing, you see a display like the example in Figure 3-34.

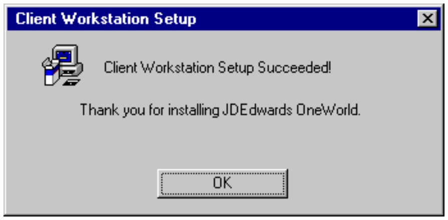


Figure 3-34 OneWorld Client Workstation Setup completion

Windows Terminal Server installation

OneWorld Xe includes support for Microsoft Windows Terminal Server, which allows a number of client workstations to connect to the enterprise server through a single client connection.

3.1.6 Completing the installation

This section describes the tasks that you should perform following the main software installation procedures to ensure correct OneWorld functionality. These are necessary to remedy several known issues with Release B73.3.2 of OneWorld. It is anticipated that these will be resolved in future releases.

Amending Server Map Object Configuration Manager mappings

Following the installation of OneWorld on the enterprise server, amend the default setting for UBEs in the Server Map OCM (Figure 3-35) to *not active*, only if necessary. When left *active* in previous releases of OneWorld, this has caused problems for UBEs that call other UBEs. However, in Release B73.3.2, this should no longer be required.

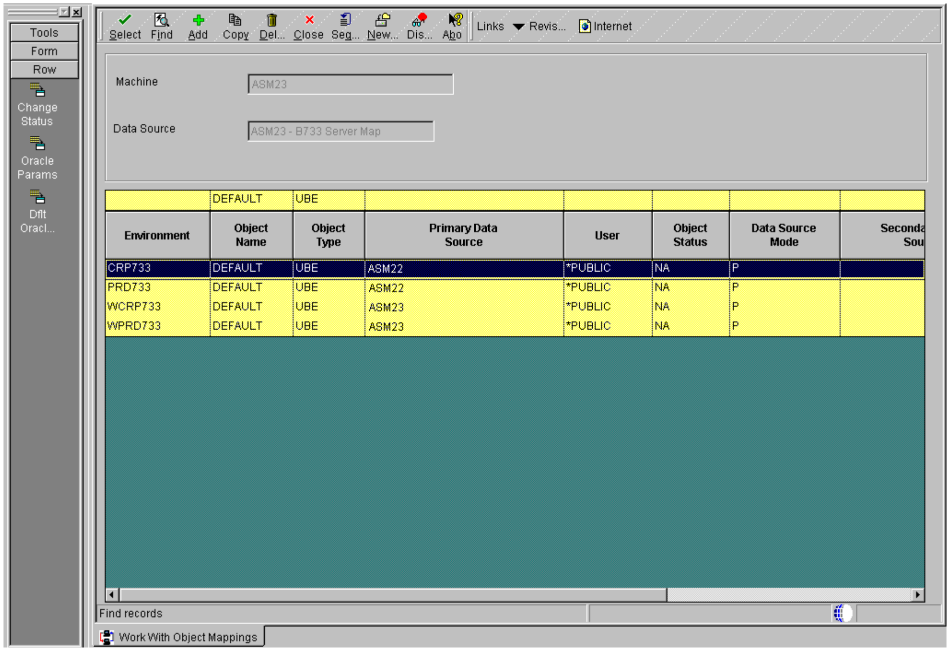


Figure 3-35 Default Server Map OCM entries for UBEs

Amending B7333SYS library authority

There is a possibility that UBEs may fail immediately on submission due to authority problems. When a UBE is submitted, an attempt is made to delete a temporary password file owned by the user profile ONEWORLD in the B7333SYS library. If the user profile submitting the UBE has insufficient authority to this file, the UBE fails.

To circumvent this problem, only if necessary, change the create authority for the B7333SYS library to *ALL. This allows any user with a minimum of *USE authority to the library to delete new objects created in the library. Issue the following command:

```
CHGLIB LIB(B7333SYS) CRTAUT(*ALL)
```

Since it is likely that the temporary password file is the only object created in this library after initial installation, this should not pose a great security exposure.

Library B7333SYS is created with the CRTAUT value of *CHANGE, so this should be unnecessary.

Adding OCM mappings for Business Function B9861101

In some scenarios, users may not be able to view the output from previously submitted jobs. This problem can occur in any environment where the default OCM mapping for Business Functions is not set to LOCAL, typically environments set up for Windows Terminal Servers. This causes a failure in the Work With Servers application, P986110B.

To circumvent this, if necessary, create an OCM mapping for Business Function B9861101 in all environments for user *PUBLIC. Primary Data Source for this OCM entry should be set to “LOCAL” as shown in Figure 3-36.

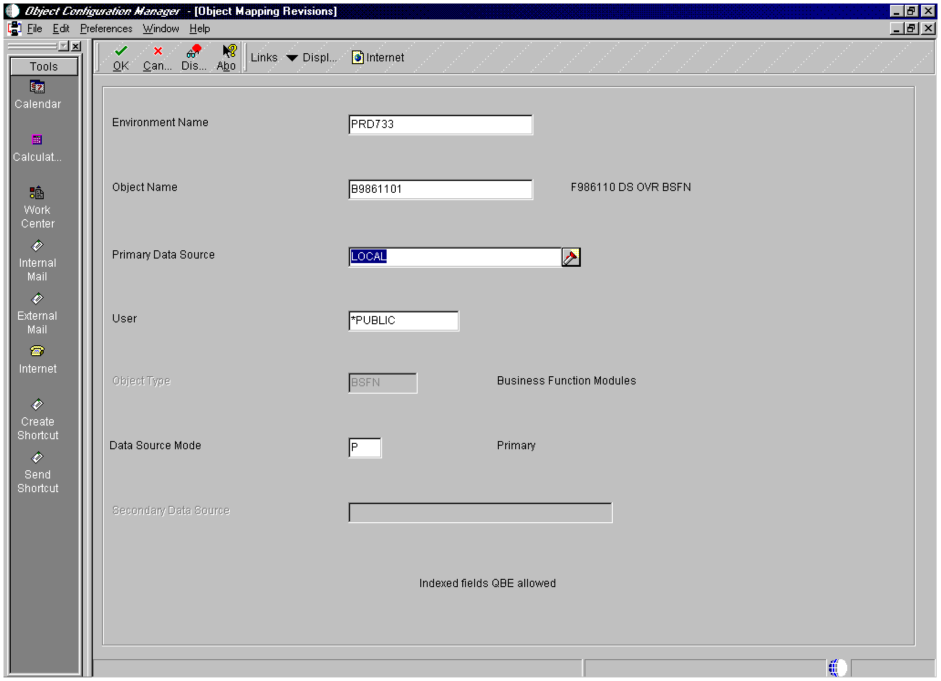


Figure 3-36 OCM Mapping Revisions display

Modifying media object queues

A media object queue must be defined to identify the pointer to the location where actual image files or OLE objects reside. For OLE objects, the name of the queue must be OLEQUE. The media object queues must reside on a Windows 2000 machine. If you want, you can use the deployment server (which is installed on an Integrated xSeries Server) or another Integrated xSeries Server, if you have multiple Integrated xSeries Servers installed on your iSeries server.

Media object queue paths are shipped with generic server names. To enable media object queues at your site, you must enter the deployment server name in the media object queue path. To do this, complete the following steps:

1. Sign on the client workstation with user `JDE` into the specific OneWorld environment where you want to make these changes.
2. Type `GH9011` in Fast Path, and press Enter to display the System Administration Tools menu.
3. Double-click **Media Object Queues**.
4. Click **Find** on the Work With Media Object Queues form.
5. Repeat the following steps for each media object queue:
   a. Double-click a media object queue.
   b. Type `\\`*`server name`*`\B7333\`*`queue name`* for Path, where *server name* is the name of your deployment server and *queue name* is the name of the media object queue.
   c. Click **OK**.

Changing *PUBLIC authority for the JDE user profile

To submit UBEs to execute on the iSeries enterprise server, it is necessary that the OneWorld users have *CHANGE authority to the user profile JDE on the iSeries server. Although *CHANGE authority can be granted for each individual iSeries server user profile, it is easier to provide it for *PUBLIC. If this violates your security policy, provide this authority just for the iSeries server user profiles corresponding to the OneWorld users. To grant *CHANGE authority to user profile JDE for *PUBLIC, complete the following steps:

1. Sign on to the deployment server as `Administrator`.
2. Click **Start-> Programs-> IBM AS/400 Client Access Express-> AS/400 Operations Navigator**. This starts Operations Navigator for accessing the iSeries server from the deployment server.
3. If necessary, expand **My AS/400 Connections** to see your iSeries enterprise server.
4. Expand your iSeries enterprise server.
5. When prompted for Sign on to iSeries server, type `QSECOFR` for the iSeries server User ID and the corresponding password for Password. Click **OK**.
6. Expand **File Systems**.
7. Expand **Integrated File System**.
8. Click **QSYS.LIB** to list the objects in this directory on the iSeries server.
9. Find and right-click **JDE.USRPRF**.

10. Click **Permissions**.
11. Select **Change** for (Public).
12. Click **OK**.

Running the R4094A batch application

If you use OneWorld Inventory Management (system 41) or OneWorld Sales Order Management (system 42), you must run the R4094A batch application, version XJDE0001. This application can be run from any client either locally or on the server once for each environment where these systems will be used.

To run the R4094A application, complete the following process for each environment:

1. Sign on the workstation as user `JDE`.
2. Type `GH9111` in Fast Path and press Enter to display the Report Writer menu.
3. Double-click **Batch Versions**.
4. Type `R4094A` for Batch Application and click **Find**.
5. Select **XJDE0001**, and click **Submit**.

Building Word Search tables

The Word Search Build program populates the Word Search table (F91013) from information in the menu tables (F00821 and F0083). Word Search helps you locate a OneWorld menu if you do not know its name. The installation of these tables is optional.

To build the Word Search Tables, complete the following steps from any workstation in any environment, since this completes the word search build for all environments:

1. Sign on as user `JDE` in the Pristine (`JD7333`) environment at any workstation where Pristine has been installed.
2. Type `GH9011` in Fast Path, and press Enter to display the System Administration Tools menu.
3. Double-click **Batch Versions**.
4. Type `R95012` for Batch Application, and click **Find**.
5. Select **XJDE0001**, and click **Submit**.
6. Type `1` for Option 1 on the Processing Options form to clear the Menu Word Search table before you rebuild it. Ensure that you have a backup copy of this table before you clear it.

4

Service packs

This chapter defines and discusses the process of the OneWorld service pack related to the iSeries environment. You should use the information provided in this chapter in conjunction with the *OneWorld Service Pack Installation Guide (iSeries server)*, which is supplied with the OneWorld service pack software. This chapter contains supplemental information, where appropriate, to define best practices and to facilitate the installation process. A qualified OneWorld installer should perform the service pack installation in an iSeries environment.

This chapter introduces the concept of the OneWorld service pack and then moves into the brief discussion of applying and rolling back the service pack. Next, you go step-by-step through the approach to apply a new service pack to a test environment, while keeping the production environment running on the existing foundation or service pack.

4.1 What a service pack is

A service pack can be defined as the foundation code of OneWorld. This foundation code resides in the system library of the enterprise server, the system directory of deployment server, and the client workstation. A service pack is the mechanism to distribute system foundation or tool-level fixes and enhancements, independent of the cumulative update or upgrade to the existing OneWorld installations.

A service pack contains the system foundation-level update that handles the communication and kernel processes between OneWorld workstations and the server. It is not the same as a cumulative update, which is why it does not contain any application code change or enhancements.

You can easily obtain the specific fixes each service pack contains via the J.D. Edwards Knowledge Garden (via the KG Login link at `http://www.jdedwards.com/`). These specific fixes are tracked through Software Action Requests (SAR), which is the term used by J.D. Edwards to track issues or enhancements for the software.

4.2 Applying a service pack

As mentioned in the previous section, a service pack contains fixes for specific issue that are tracked through SARs. If you experience issues that are fixed by SARs and a specific service pack contains those SARs, you would then definitely want to obtain that service pack. Again, by reviewing the SAR list for a specific service pack, you can also make a decision as to whether you want to apply that service pack to achieve the benefit of any enhancements that have been done in the foundation level.

The detailed instruction on applying the service pack comes with the service pack CD. We recommend that you print out the instructions and then follow them during the installation process. The following discussion outlines the fundamental steps of applying the service pack and what you need to verify during and after the process for a successful installation.

4.2.1 Fundamental steps of applying the service pack

To apply the new service pack, you need to follow these main steps:

1. Back up the existing service pack.
2. Run the installation program contained in the service pack CD on the deployment server.

3. Apply the new foundation code to the iSeries server.
4. Run the OneWorld program LINKBSFN to link the OneWorld path code library service programs to the OneWorld system kernel library service programs.
5. Build and deploy the client update package to OneWorld client workstations.

4.2.2 System values to verify before applying the service pack

We recommend that you verify the following iSeries system values prior to applying a new service pack:

- **QALWOBJRST** (Allow object restore option): Should be set to `*ALL`.
- **QALWUSRDMN** (Allow user domain objects in libraries): Should be set to `*ALL`.
- **QJOBMSGQFL** (Job message queue full action): Should be set to `*WRAP`.

4.2.3 Tips for using LINKBSFN when applying the service pack

As mentioned in 4.2.1, “Fundamental steps of applying the service pack” on page 134, during the installation process, you need to run the OneWorld LINKBSFN program that is located in the OneWorld system library (for example, B7333SYS). This program creates new OneWorld “path code” service programs (object type *SRVPGM), and therefore, updates the signatures of those corresponding OneWorld system kernel service programs. Since each service program has a unique program signature that is tracked by the iSeries server, failure of LINKBSFN results in the error message `Signature violation found` in the job log. If these service programs are found to be out of sync and result in this error message, OneWorld will not function properly.

To ensure the success of running the LINKBSFN program, here are some tips and techniques you should use through the step-by-step approach:

1. Sign on to the iSeries server as `ONEWORLD`.
2. Make sure OneWorld services have are already ended by issuing the ENDNET command.
3. Verify the ONEWORLD user profile’s authority:

   ```
   DSPUSRPRF USRPRF(ONEWORLD)
   ```

   Make sure the user profile ONEWORLD has special authorities *ALLOBJ, *JOBCTL, and *SPLCTL. If not, change the user profile accordingly.
4. Execute the command:

   ```
   DSPOBJAUT OBJ(ONEWORLD) OBJTYPE(*USRPRF)
   ```

   Verify that *PUBLIC authority is set to either `*ALL` or `*CHANGE`.

5. Re-verify that the system value QJOBMSGQFL is set to `*WRAP`. Otherwise the LINKBSFN command may fail with messages, such as `Message queue full`, in the job log.
6. Run the LINKBSFN command against the path code's full package. From the OS/400 command line, type the LINKBSFN command and then press F4 to display the system prompt. You need to enter the package library name for the first parameter (for example, `PY7333FA`) and the path code for the second parameter (for example, `PY7333`).

   Or, type this command on the command prompt:

   ```
   LINKBSFN PARENT(PY7333FA) PATHCODE(PY733)
   ```

   Here *PY7333FA* is the OneWorld Package Library and *PY733* is the path code.
7. The jobs are submitted to the job queue associated with the user ONEWORLD. You can verify the job queue associated with ONEWORLD user profile by using the command:

   ```
   DSPJOBD ONEWORLD
   ```

   See the display in Figure 4-1.

```
                    DSPJOBD ONEWORLD
Job description: ONEWORLD                Library: QGPL
User profile . . . . . . . . . . . . . . . . . . . . : *RQD
CL syntax check . . . . . . . . . . . . . . . . . : *NOCHK
Hold on job queue . . . . . . . . . . . . . . . . : *NO
End severity . . . . . . . . . . . . . . . . . . . . : 30
Job date . . . . . . . . . . . . . . . . . . . . . . : *SYSVAL
Job switches . . . . . . . . . . . . . . . . . . . . : 00000000
Inquiry message reply . . . . . . . . . . . . . . : *RQD
Job priority (on job queue) . . . . . . . . . . . : 5
Job queue . . . . . . . . . . . . . . . . . . . . . . : QBATCH
```

Figure 4-1 DSPJOBD ONEWORLD

8. After you locate the job queue, issue the following command to verify the progress of the LINKBSFN process:

   ```
   WRKJOBQ jobq
   ```

   Here *jobq* is the job queue associated with the ONEWORLD user profile. See Figure 4-2.

```
                          Work with Job Queue

 Queue:   QBATCH        Library:   QGPL         Status:   RLS/SBS

 Type options, press Enter.
   2=Change   3=Hold   4=End   5=Work with   6=Release

 Opt     Job            User           Number     Priority     Status
         CDESIGN        ONEWORLD       032976        5           RLS
         CDIST          ONEWORLD       032977        5           RLS
         CFIN           ONEWORLD       032978        5           RLS
         CFND           ONEWORLD       032979        5           RLS
         CHRM           ONEWORLD       032980        5           RLS
         CINSTALL       ONEWORLD       032981        5           RLS
         CINV           ONEWORLD       032982        5           RLS
         CLOC           ONEWORLD       032983        5           RLS
         CLOG           ONEWORLD       032984        5           RLS
         CMFG           ONEWORLD       032985        5           RLS
More...
```

Figure 4-2 Work with Job Queue

The LINKBSFN process usually takes about 30 to 45 minutes.

4.2.4 Running PORTTEST after applying the service pack

We strongly recommend that you run PORTTEST after you apply the new service pack. To do that, perform the following steps:

1. Sign on to the iSeries server as `ONEWORLD`.
2. On current OneWorld versions, choose the instance of OneWorld on which you applied the new service pack.
3. Clear the Inter Process Communications (IPC) for the three types including queues, semaphores, and shared memory by running the command:

   ```
   CLRIPC
   ```

4. Start the OneWorld subsystem by using the command:

   `STRNET SBS(`*`subsystem`*`)`

   Here *subsystem* is the name of the subsystem for the current service pack (for example, `JDEB733`).

5. Verify that the both NETWORK and SENTINEL are running by using the command:

   ```
   WRKACTJOB SBS(JDEB7333)
   ```

Verify that the NETWORK process is running with a PGM-MONITOR in SELW status and that the entry SENTINEL is running with a PGM-MONITOR in SIGW status. Until a net request is performed, the CPU will remain at 0.

6. On the command prompt, type `PORTTEST` and then press F4 to display the system prompt. Enter the user profile, password, and the environment (for example, `PD7333`) that are needed for verification to ensure that OneWorld is running properly.

> **Note**: If Security Server is on, then OneWorld services must be running before you execute the PORTTEST command.

4.2.5 Testing the service pack on the OneWorld client workstation

Once you apply the service pack to the client workstation, you need to test the service pack for issues that should have been fixed as mentioned in the SAR list for the specific service pack. Again, there are some common things that you should verify, including running any UBEs locally and on the server, printing from OneWorld client, and so on.

4.3 Rolling back to a previous service pack level

The following discussion shows you the steps to roll back to the previous service pack level, should the need arise. To do so, you need to follow these steps:

1. **Restore the deployment server**: Restore the system, system computer, and OneWorld Client Installation directory on the deployment server from the previous service pack level.
2. **Restore the enterprise server**: Restore the system library and system directory (resides in IFS) from the previous service pack level on your enterprise server. Then run the OneWorld program LINKBSFN.

Each of these steps is explained in more detail in the following sections.

4.3.1 Restoring the deployment server

To restore the deployment server, perform the following steps:

1. Sign on to the deployment server as an administrator to the system. We recommend that you use a JDE user profile, because it should have local administrator authority on the PC.
2. Ensure that you are logged out of OneWorld on the deployment server.
3. Using Windows Explorer, navigate to the \OneWorld\B7333 directory.

4. Rename the current SYSTEM directory to the `SYSTEMSPX` (for example, `SYSTEMSP17.1`) directory, where *X* is the level of the service pack. In the same way, rename the SYSTEMCOMP and OneWorld Client Installation directory.
5. Restore the previous service pack directories by renaming the SYSTEM BACKUP and SYSTEMCOMP BACKUP, OneWorld Client InstalBACKUP (these directory names have been assumed for illustration) directories, to SYSTEM and SYSTEMCOMP, OneWorld Client Install respectively.
6. Verify the ptf.log file residing on \JDEdwardsOneWorld\B7333\SYSTEM\Bin32 to ensure that your deployment server is now at the old service pack level.

4.3.2 Restoring the enterprise server

To restore the system library and system directory residing on the IFS of your enterprise server, you need to follow these steps:

1. Sign on to the iSeries server as `ONEWORLD`.
2. On the Current OneWorld Versions menu, choose the instance of OneWorld for which you want to back off the service pack.
3. End OneWorld on the iSeries server by entering the commands ENDNET and CLRIPC.
4. Sign on to your iSeries server as `QSECOFR` and rename the OneWorld SYSTEM library by using the following command:

   ```
   RNMOBJ OBJ(QSYS/syslib) OBJTYPE(*LIB) NEWOBJ(syslibBKSPX)
   ```

   Here *syslib* is the name of your OneWorld system library (for example, `B733XSYS`), and *syslibBKSPX* (for example, `syslibBKSP17`) is the suggested library name in which you would want to hold the current system library.
5. Restore your old system library that should reside on your system as SYSLIBBK (if the service pack guide has been followed) by renaming to original name (for example, `B733XSYS`).

   ```
   RNMOBJ OBJ(QSYS/syslibBK) OBJTYPE(*LIB) NEWOBJ(B733XSYS)
   ```
6. Rename the current OneWorld IFS directory:

   ```
   RNM OBJ('/B733XSYS') NEWOBJ('/B733XSYSBK')
   ```

   Here *X* is the Cumulative Update number of the OneWorld release (for example, `B7333SYS`).
7. Restore the system directory on the IFS from the service pack CD or from any existing backup.

8. To ensure that the rollback of the service is successful, run the following command:

   ```
   DSPPFM FILE(syslib/LOG)
   ```

   Here *syslib* is the name of the system library (for example, B733XSYS).

9. Now that you have restored your system library and system directory, you need to run the LINKBSFN program:

   a. Log on as `ONEWORLD` on the iSeries server.

   b. Type the LINKBSFN command from the command prompt and prompt it by pressing F4.

   c. Enter the package library name for the first parameter (for example, `PY7333FA`) and the path code for the second parameter (for example, `PY7333`).

   For details on the LINKBSFN program, see 4.2.3, “Tips for using LINKBSFN when applying the service pack” on page 135.

10. We recommend that you run PORTTEST. This performs environment initialization and verification to make sure OneWorld installation is operational.

4.4 Running OneWorld on different service pack levels

It is possible to apply a new service pack to a test environment, while keeping the production environment running on the existing foundation/service pack. This allows testing of the new service pack release before promoting it to the production environment.

This multiple foundation installation makes it possible to have one path code at one service pack release while another at a different service pack release. This section shows the multiple foundation/service pack installation process through a step-by-step approach:

1. Verify the current service pack release.
2. Create save files on the server.
3. Transfer the service pack files to the server.
4. Verify the save files.
5. Install the service pack.
6. Update the jde.ini file.
7. Delete and create a subsystem for the service pack.
8. Link the path code business functions to the service pack.
9. Validate the service pack installation.
10. Install the service pack on the deployment server.

4.4.1 Verifying the current service pack release

The new service pack is going to be applied on the test/prototype environment. Prior to the install of the new service pack release, verify the release of the current service pack that is running on the iSeries server by using the following command:

```
DSPPFM FILE (B733XSYS/LOG)
```

Here, *X* is the Cumulative Update number of the current release (for example, `B7333SYS`).

4.4.2 Creating save files on the iSeries server

Perform the following steps:

1. Sign onto the iSeries server as `QSECOFR`.
2. Create a temporary library for the service pack:

   ```
   CRTLIB JDETEMP
   ```

   > **Note**: If this library already exists, delete the library before you proceed with the service pack installation. This library can be removed after the installation is complete.

3. Create a SYSTEM save file in JDETEMP.

   ```
   CRTSAVF JDETEMP/SYSTEM
   ```

4. Create a KRNLSPEC save file in JDETEMP.

   ```
   CRTSAVF JDETEMP/KRNLSPEC
   ```

4.4.3 Transferring service pack files to the iSeries server

Perform the following steps:

1. Insert the service pack CD into a workstation.
2. Open a DOS command prompt.
3. Start an FTP session to connect to the iSeries server as follows:

   ```
   C:\>ftp iSeries server Name
   ```

   Here *iSeries server Name* is the name of the enterprise server.

4. Change the directory to the JDETEMP library on the iSeries server:

   ```
   ftp> cd jdetemp
   ```

5. Change the local directory to the CD directory containing the iSeries server save files:

   ```
   ftp> lcd E:\hosts\AS400
   ```

   Here, E: is the drive that contains the service pack CD.

6. Change the mode of transfer to binary:

   ```
   ftp> bin
   ```

7. Upload the SYSTEM save files to the iSeries server:

   ```
   ftp> put SYSTEM
   ```

8. Upload the KRNLSPEC save files to the iSeries server:

   ```
   ftp> put KRNLSPEC
   ```

9. End the FTP session and exit the command prompt:

   ```
   ftp> quit
   ```

4.4.4 Verifying the save files residing in the JDETEMP library

Perform the following steps:

1. Verify that the KRNLSPEC save file was transferred to the iSeries server successfully by running the following command:

   ```
   DSPSAVF FILE(JDETEMP/KRNLSPEC)
   ```

2. Verify that the SYSTEM save file was transferred to the iSeries server successfully by running the following command:

   ```
   DSPSAVF FILE(JDETEMP/SYSTEM)
   ```

   See Figure 4-3.

```
                      Display Saved Objects - Save File

Library saved . . . :   B7333SYS            Release level  . . . :
V4R3M0
ASP  . . . . . . . . :  1                   Data compressed  . . :   Yes
Save file  . . . . . :  SYSTEM              Objects displayed  . :   6
  Library  . . . . . :    JDETEMP           Objects saved  . . . :   157
Records  . . . . . . :  249467              Access paths . . . . :   0
Save command . . . . :  SAVLIB
Save active  . . . . :  *NO
Save date/time . . . :  10/29/01   13:54:31

Type options, press Enter.
  5=Display saved data base file members

Opt  Object            Type     Attribute     Owner           Size   Data
     B7333SYS          *LIB     PROD          ONEWORLD      229376   YES
     BUILDDLL          *PGM     CLE           ONEWORLD      188416   YES
     BUILDMSTR         *PGM     CLE           ONEWORLD       69632   YES
     CHGLIBOWN         *PGM     CLE           ONEWORLD       81920   YES
     CLEANUP           *PGM     CLE           ONEWORLD      106496   YES
     CLRDMGIPC         *PGM     CLE           ONEWORLD       61440   YES
+

F3=Exit        F12=Cancel
```

Figure 4-3 Display Saved Objects - Save File

Note: There are approximately 151 objects in the SYSTEM save file, but it may vary in different service pack releases. After you run the command to display the SYSTEM save file (SAVF), make note of the object B7333SYS. This object name is used in the following section on the SAVLIB parameter.

4.4.5 Installing the service pack

To create the new system library and install the service pack software that was transferred from CD to the iSeries server, follow these steps:

1. Sign on to the iSeries server as `QSECOFR`.
2. Create a new system library. The library name cannot exceed eight characters in length and it *must* start with “B73” and end with “SYS”:

   `RSTLIB SAVLIB(B733`***X***`SYS) DEV(*SAVF) SAVF(JDETEMP/SYSTEM) RSTLIB(`*syslib*`)`

   Here *X* is the Cumulative Update number of the current OneWorld release and *syslib* is the name of the new system library (for example, `B73SPSYS`).

3. Copy the PRINTQUEUE file from the existing system library to the newly created system library:

```
CPYF FROMFILE(B733XSYS/PRINTQUEUE) TOFILE(syslib/PRINTQUEUE) FROMMBR(*ALL)
TOMBR(*FROMMBR) MBROPT(*REPLACE) FROMRCD(1)
```

   Here *X* is the Cumulative Update number of the current OneWorld release and *syslib* is the name of the new system library (for example, `B73SPSYS`).

   > **Note:** You can improve considerably the performance of the Copy File (CPYF) command by specifying the value 1 for the command's From Record (FROMRCD) parameter. By default, CPYF copies records in key sequence. Specifying FROMRCD(1) forces CPYF to copy records in arrival sequence, which is faster than using a keyed sequence.

4. Copy the jde.ini file from the existing system library to the newly created SYSTEM library:

```
CRTDUPOBJ OBJ(INI) FROMLIB(B733XSYS) OBJTYPE(*FILE) TOLIB(syslib)
DATA(*YES)
```

   Here *X* is the Cumulative Update number of the current OneWorld release and *syslib* is the name of the new system library (for example, `B73SPSYS`).

5. Install objects in the IFS file system:

```
RST DEV('/QSYS.LIB/JDETEMP.LIB/KRNLSPEC.FILE') OBJ(('/krnb733x' *INCLUDE
syslib))
```

   Here *X* is the Cumulative Update number of the current OneWorld release and *syslib* is the name of the new system library (for example, `B73SPSYS`). See Figure 4-4.

```
MAIN                              AS/400 Main Menu
                                                                  System:
JDECS8
 Select one of the following:

      1. User tasks
      2. Office tasks
      3. General system tasks
      4. Files, libraries, and folders
      5. Programming
      6. Communications
      7. Define or change the system
      8. Problem handling
      9. Display a menu
     10. Information Assistant options
     11. Client Access/400 tasks

     90. Sign off
Selection or command
 ===> RST DEV('/QSYS.LIB/JDETEMP.LIB/KRNLSPEC.FILE')

-------------------------------------------------------------------------
      OBJ(('/krnb733x' *INCLUDE B73SPSYS))

----------------------------------------------------------------------------
--
 F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Information Assistant
 F23=Set initial menu
```

Figure 4-4 Installing objects on IFS

6. Create a new IFS directory for the new service pack J.D. Edwards logs:

   `CRTDIR` *`logdirectory`*

   Here *logdirectory* is the name of the IFS directory for storing the J.D. Edwards logs (for example, `JDEB733SP`).

4.4.6 Updating the server jde.ini file

Perform the following steps:

1. Edit the newly copied jde.ini by running the following command:

   `WRKMBRPDM FILE(`*`syslib`*`/INI)`

   Here *syslib* is the name of the new system library (for example, `B73SPSYS`).

2. Enter option `2` to edit the jde.ini file.

3. Change the path under the [DEBUG] section of the server jde.ini to point to the new IFS directory as follows:

   ```
   DebugFile=logdirectory/JDEDEBUG
   JobFile=logdirectory/JDE.LOG
   JDETSFile=logdirectory/JDETS.LOG
   ```

   Here *logdirectory* is the name of the IFS directory for storing the J.D. Edwards logs (for example, `JDEB733SP`) as shown in Figure 4-5.

```
      Columns . . . :    1  71           Edit
      B73SPSYS/INI
       SEU==>
      JDE
       FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6
      ...+... 7
              *************** Beginning of data
      *************************************
      0001.00 ; OneWorld initialization file INI(JDE)
      0002.00 ; AS/400 specific version - B73.3.3
      0003.00
      0004.00 [DEBUG]
      0005.00 Output=NONE
      0006.00 Trace=FALSE
      0007.00 DebugFile=JDEB733SP/jdedebug
      0008.00 JobFile=JDEB733SP/jde.log
      0009.00 JDETSFile=JDEB733SP/JDETS.LOG
0010.00 ClientLog=0
```

Figure 4-5 Updating jde.ini

4. Change the path under the [INSTALL] section of the server jde.ini to point to the new library:

   ```
   DefaultSystem=syslib
   ```

   Here *syslib* is the name of the new system library (for example, `B73SPSYS`). See Figure 4-6.

```
Columns . . . :    1  71            Edit
B73SPSYS/INI
 SEU==>
JDE
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+...
7

0031.00
0032.00 [INSTALL]
0033.00 ;POSTSCRIPT_ONLY=1
0034.00 DefaultSystem=B73SPSYS
0035.00 ClientPath=B7333APP
0036.00 B733=
0037.00 Double_Byte=0
0038.00 LocalCodeSet=US_EBCDIC
```

Figure 4-6 Updating jde.ini

5. Change the startIPCKeyValue under the [JDEIPC] section of the server jde.ini to a key value that is not being used by another OneWorld instance's IPC range. For example, if another OneWorld instance's IPC value is 2101, then set the startIPCkey value to 5101:

   `startIPCKeyValue=5101.`

   See Figure 4-7.

```
Columns . . . :    1  71            Edit
B73SPSYS/INI
SEU==>
JDE
FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+...
7

0040.00 [JDEIPC]
0041.00 maxNumberOfResources=1000
0042.00 startIPCKeyValue=5101
0043.00 avgResourceNameLength=15
```

Figure 4-7 Updating jde.ini

6. Increase the PORT number for *serviceNameListen* and *serviceNameConnect* by 100 under the [JDENET] section of the server jde.ini. This value should be an unused PORT number. For example, if the original setting was 6009, then change the PORT number to be 6109. See Figure 4-8.

```
Columns . . . :    1  71           Edit
B73SPSYS/INI
 SEU==>
JDE
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+...
7

0048.00 [JDENET]
0049.00 serviceNameListen=6109
0050.00 serviceNameConnect=6109
```

Figure 4-8 Updating jde.ini

7. Under the [SECURITY] section of the server jde.ini, either turn off the Security server or modify the Default Environment to reference a valid environment that will access the new service pack release. See Figure 4-9.

```
Columns . . . :    1  71           Edit
B73SPSYS/INI
 SEU==>
JDE
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6
...+... 7


0210.00 [SECURITY]
0211.00 DataSource=System - B7333 - DNT
0212.00 User=JDE
0213.00 Password=JDE
0214.00 DefaultEnvironment=PY7333
0215.00 SecurityServer=
```

Figure 4-9 Updating jde.ini

8. Under the [DB SYSTEM SETTINGS], [SECURITY], and [SVR] sections of the server jde.ini, replace all references to a valid environment that will access the new service pack release.
9. Save and then exit the jde.ini file.

You should end OneWorld and then restart it for the INI file changes to take effect.

4.4.7 Creating a subsystem for the newly installed service pack

Perform the following steps:

1. Sign on to the iSeries server as `QSECOFR`.
2. Type the following command:

   ```
   WRKSBSD SBSD(syslib/*ALL)
   ```

   Here *syslib* is the name of the new system library (for example, `B73SPSYS`).
3. Enter option `4` next to the entry that includes the new *syslib* (for example, `B73SPSYS`) and delete the entry.
4. Add the OneWorld SYSTEM library to QSECOFR's library list by typing the following command on the command prompt:

   ```
   ADDLIBLE syslib
   ```

   Here *syslib* is the name of the new system library (for example, `B73SPSYS`).
5. Enter the following commands to create a new subsystem for the newly installed service pack (assuming the service pack is 15 or later):

   ```
   CHGCMD CMD(CRTOWSBS) PGM(*LIBL/CRTOWSBS)
   CRTOWSBS SUBSYSTEM(subsystem) SYSLIB(syslib)
   ```

   Here *subsystem* is the name of the new subsystem for the newly installed service pack (for example, `JDEB733SP`) and *syslib* is the name of the new system library (for example, `B73SPSYS`).

4.4.8 Linking path code business functions to the new service pack

For each OneWorld path code associated with the new service pack, the business functions need to be relinked.

1. Log on as `OneWorld` on the iSeries server.
2. Ensure that the newly created OneWorld system library is in library list.
3. Type the LINKBSFN command on the command prompt and prompt it by pressing F4.

   For details on the LINKBSFN process and to ensure its success, see 4.2.3, “Tips for using LINKBSFN when applying the service pack” on page 135.
4. Enter the package library name for the first parameter (for example, `PY7333FA`) and the path code for the second parameter (for example, `PY7333`).

4.4.9 Validating the service pack installation

To ensure that the new service pack is properly installed on the enterprise server and to review the service pack release of the new system library, run PORTTEST, and start the OneWorld services.

If the Security server is not turned on, complete these steps:

1. Sign on the iSeries server as `ONEWORLD`.
2. On the Current OneWorld Versions menu, choose the newly installed instance of OneWorld:

   ```
   DSPPFM FILE(syslib/LOG)
   ```

   Here *syslib* is the name of the new system library (for example, `B73SPSYS`).
3. On a command line, type `PORTTEST` and press F4 to display the system prompt.
4. Enter `JDE` as the user profile and its password.
5. Enter the environment that the service pack was applied to (for example, `PY7333`).

If the Security server is turned on, then OneWorld services must be running for PORTTEST to run. In this case, follow these steps:

1. Sign on the iSeries server as `ONEWORLD`.
2. On the Current OneWorld Versions menu, choose the newly installed instance of OneWorld:

   ```
   DSPPFM FILE(syslib/LOG)
   ```

   Here *syslib* is the name of the new system library (for example, `B73SPSYS`).
3. End the OneWorld subsystem:

   ```
   ENDNET SBS(subsystem)
   ```

   Here *subsystem* is the name of the new subsystem for the newly installed service pack (for example, `JDEB733SP`).
4. Clear the Inter Process Communications:

   ```
   CLRIPC
   ```

5. Start the OneWorld subsystem:

   ```
   STRNET SBS(subsystem)
   ```

   Here *subsystem* is the name of the new subsystem for the newly installed service pack (for example, `JDEB733SP`).
6. Verify that the both the NETWORK and SENTINEL are running:

   ```
   WRKACTJOB
   ```

Verify that the NETWORK process is running with a PGM-MONITOR in SELW status. Verify that the SENTINEL the is running with a PGM-MONITOR in SIGW status. Until a net request is performed, the CPU will remain at 0.

7. On the command line, type `PORTTEST` and press F4 to display the system prompt.
8. Enter `JDE` as the user profile and its password. Then enter the environment that the service pack was applied to (for example, `PY7333`).

4.4.10 Installing the service pack on the deployment server

Perform the following steps:

1. Sign on to the deployment server as `Administrator` to the local PC.
2. Using Windows Explorer, navigate to the \JDEdwardsOneWorld\B7333 directory.
3. Select the **B7333** directory, and underneath it, create a directory called `System SP`*y*, where *y* is the release level of the service pack being installed (for example, `System SP18`). See Figure 4-10.

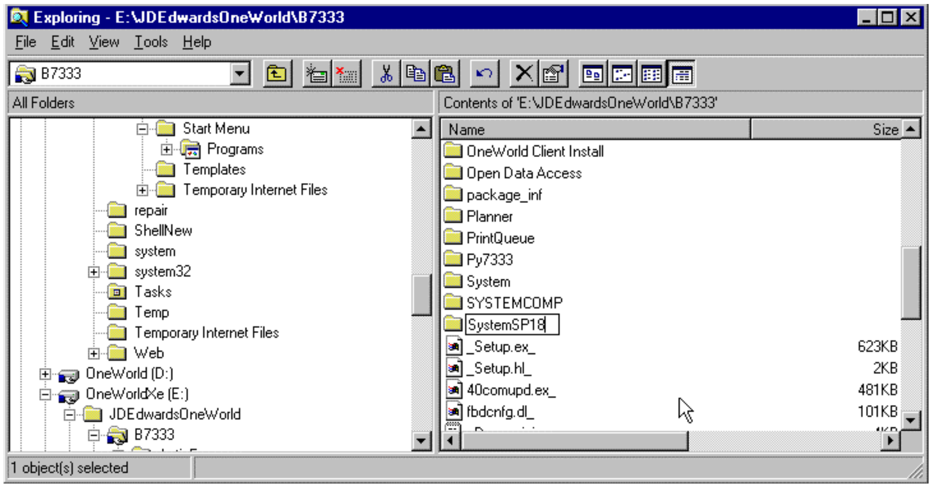


Figure 4-10 Creating a directory

4. Navigate to the \JDEdwardsOneWorld\B7333 directory and rename the System and SYSTEMCOMP directories as `System Backup` and `SYSTEMCOMP Backup` respectively. Click **OK** on any warning messages.
5. Select the **\B7333** directory, and underneath it, create a subdirectory called `OneWorld Client Install Backup`. Copy the OneWorld Client Install directory to the OneWorld Client Install Backup directory.

6. Insert the service pack CD into the deployment server CD drive and run the installation program. This program copies the new system, SYSTEMCOMP, and OneWorld Client Install directories onto the deployment server under \JDEdwardsOneWorld\B7333 directory.
7. Move the newly installed system, SYSTEMCOMP, and OneWorld Client Install directories into the \JDEdwardsOneWorld\B7333\System SPy directory, where *y* is the release number of the service pack being installed (for example, `System SP18`).
8. Restore the original directories by renaming the System Backup and SYSTEMCOMP Backup directories to `System` and `SYSTEMCOMP`, respectively.

4.5 Configuring environments for the new service pack

This section discusses the process to add or modify the machine record to configure environments. It explains the process of creating a package with a non-default foundation. The following steps are performed:

1. Add a new machine record for the new service pack.
2. Modify the existing machine record to remove environments.
3. Build an update package with a non-default foundation.
4. Deploy a client update package to workstations.
5. Update client workstations and the jde.ini file.

4.5.1 Adding a new machine record for the new service pack

Perform the following steps:

1. On the deployment server, log in to the deployment environment in OneWorld.
2. From the fast path menu, run the P9654A application.
3. On the Work with Locations and Machines form, click **Find** and expand the location tree.
4. Click **Enterprise Servers** and then click **Add**.
5. On the Enterprise Server Revisions form, complete the following fields:
   – **Machine Name**: Enter the name of your existing primary enterprise server.
   – **Description**: Enter a description for the machine.
   – **Release**: Enter `B733x` (where *x* is the cumulative number of the release).
   – **Host Type**: Click the **visual assist** button and choose the type of enterprise server.

- **Port Number**: Enter a number that should match the port number defined in the new service pack jde.ini.
- **Database Type**: Enter the type of database being used on the enterprise server.
- **Installation Path**: Enter the library in which the new service pack is installed on the enterprise server, for example, `B73SPSYS`.

6. Click **OK**.
7. On the Work with Locations and Machines form, click **Find** and expand the tree.
8. Expand the enterprise server, click the newly added machine, and click **Select**.
9. On Enterprise Server Revisions, click **Environments** in the Form Exit menu or row.
10. On the Machine Environments Revision form, add the environments for which you want to enable access to the new service pack. Do this either by:
    - Typing them into the grid
    - Using visual assist to select them
11. Click **OK** twice. This takes you back to Work with Locations and Machines.

4.5.2 Modifying the existing machine record to remove environments

Perform the following steps:

1. On the deployment server, log in to the deployment environment in OneWorld.
2. From the fast path menu, run the P9654A application.
3. On the Work with Locations and Machines form, click **Find** and expand the location tree.
4. Expand the enterprise servers, click the initial enterprise server machine record, and click **Select**.
5. On the Enterprise Server Revisions form, click **Environments** in the Form Exit menu or row.
6. Delete all the environments that you added in the previous section. This should leave you with only those environments that you want to access the production service pack (for example, `PD7333`).
7. Click **OK** twice.

4.5.3 Building an update package

Once you apply the new foundation code or service pack on the deployment server and configure OneWorld clients to access it, you need to build an update package that contains the new system directory and deploy to the workstations.

Since the full package takes long time to build, we suggest that you use an update package instead of a full package to move the foundation code to a OneWorld client workstation to save time. This section shows a step-by-step approach for creating and deploying a client update package with a specific service pack.

1. Sign on to the deployment server as an administrator to the system.
2. Start OneWorld in the DEPB733 environment as the `JDE` user.
3. On the Fast Path, type `GH9083` to access the Package and Deployment Tools menu (GH9083).
4. Define the package to be built by double-clicking **Package Assembly** program (P9601) under the GH9083 menu.
5. The Work with Packages form (Figure 4-11) appears. Click **Add**.

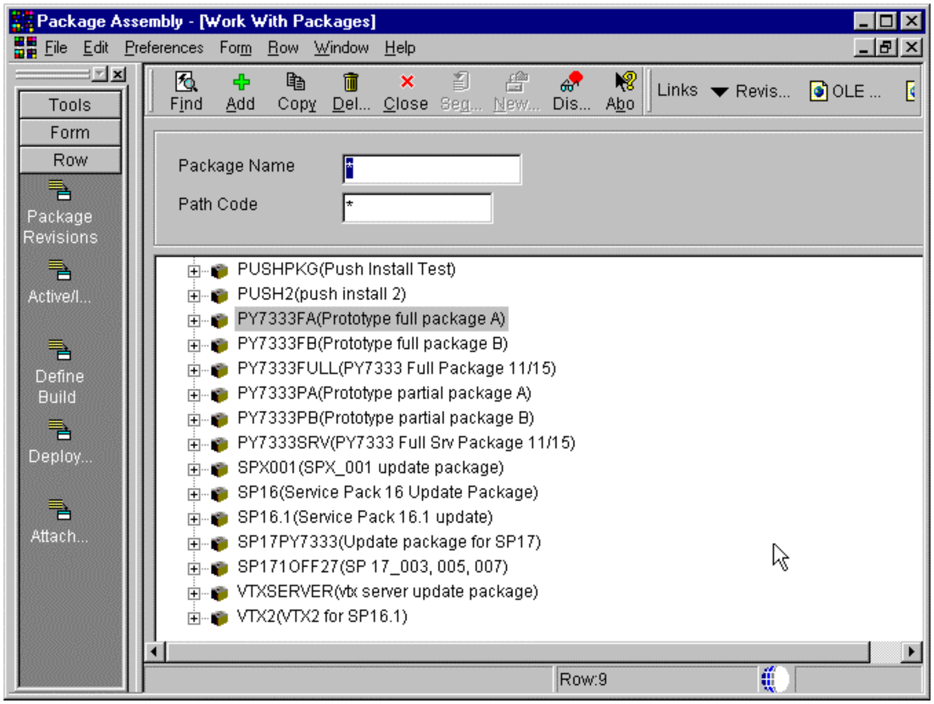


Figure 4-11 Package Assembly display

6. Now you go to the Package Assembly Director display (Figure 4-12). Click the **Next** button.

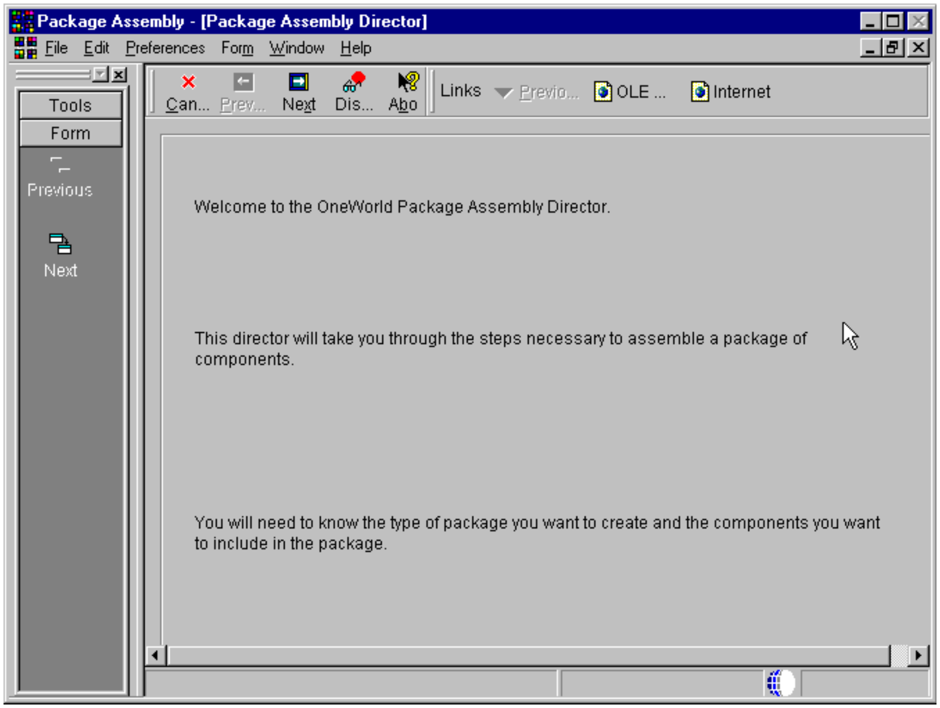


Figure 4-12 Package assembly director

7. You see the Package Information display (Figure 4-13). Enter the information and click the **Next** button.

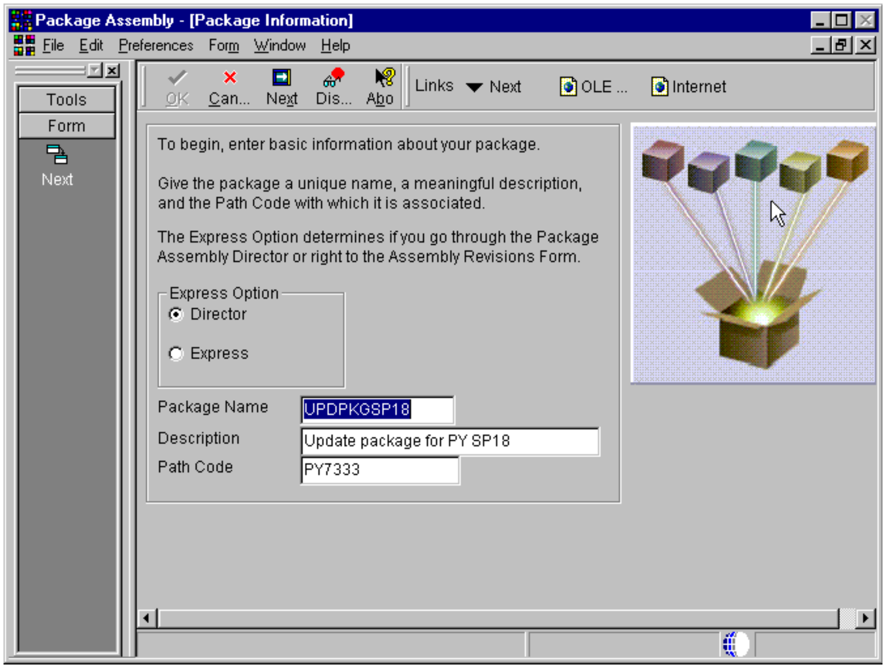


Figure 4-13 Package Information display

8. You see the Package Type Selection display (Figure 4-14). Complete these tasks:

   a. Build an update package by clicking the **Update** radio button.

   b. Select the **Include Object Specification** check box to include all the specifications for the objects that you define in the update package.

   c. Select a parent package (in our example, we select **PY7333FA**) for your update package. The parent package you choose in this field needs to be a full package that exits in the path code for which you are building the update package.

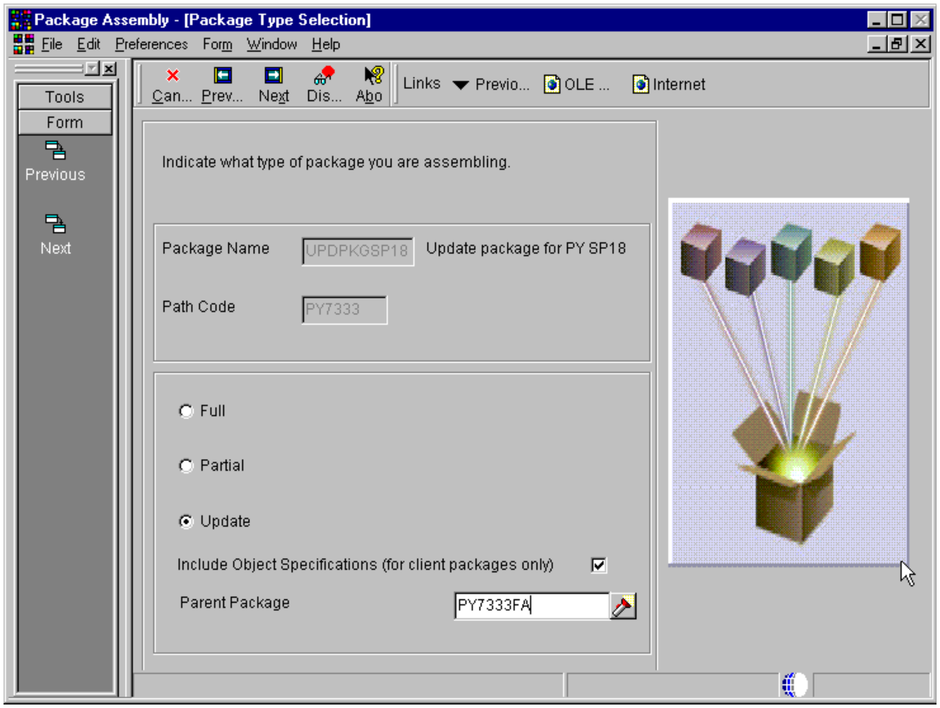


Figure 4-14 Package Type Selection display

Click the **Next** button.

9. The Foundation Component display (Figure 4-15) appears. Choose a specific foundation to build the update package. Click the **Browse** button at the bottom of the display.

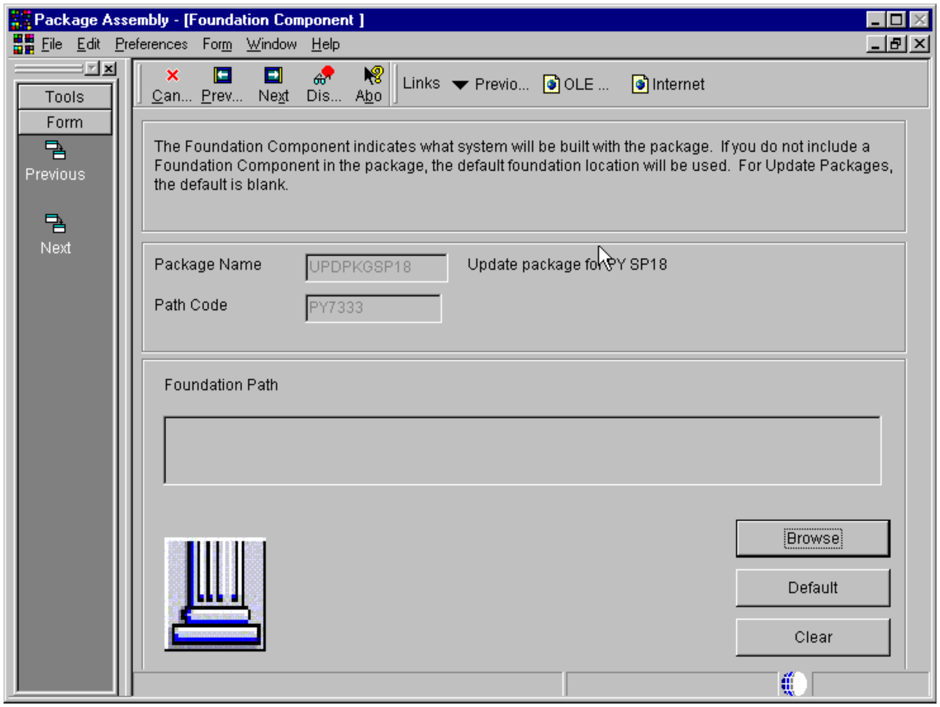


Figure 4-15 Foundation Component display

10. You see the Foundation Component Selection display (Figure 4-16). To add the new foundation, click the **Add** button.

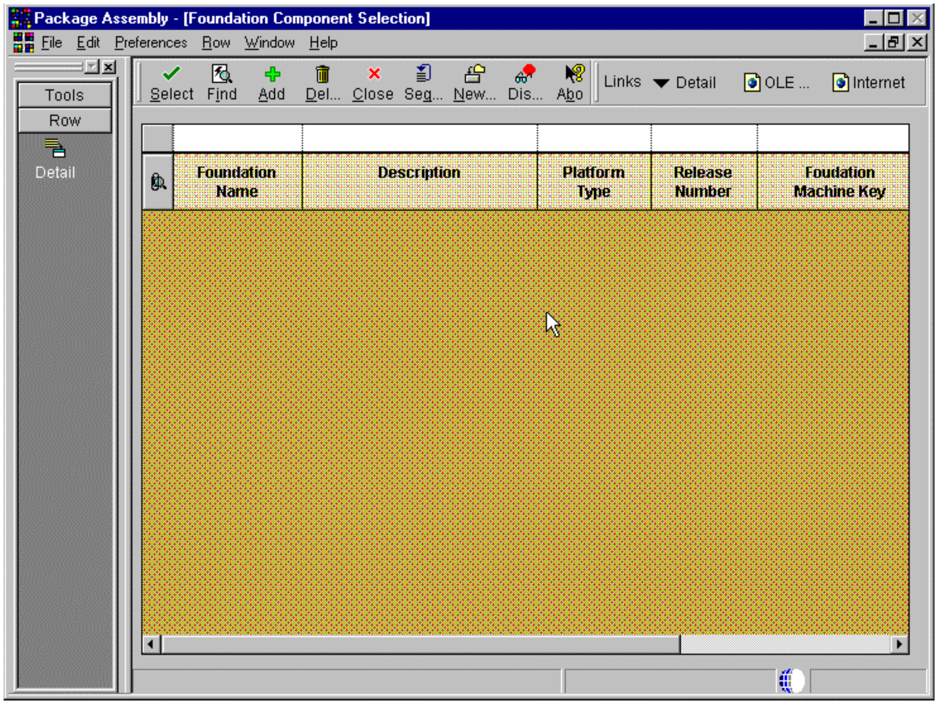


Figure 4-16 Foundation Component Selection display

11. The Foundation Item Revision display (Figure 4-17) appears. Click the **Browse...** button.

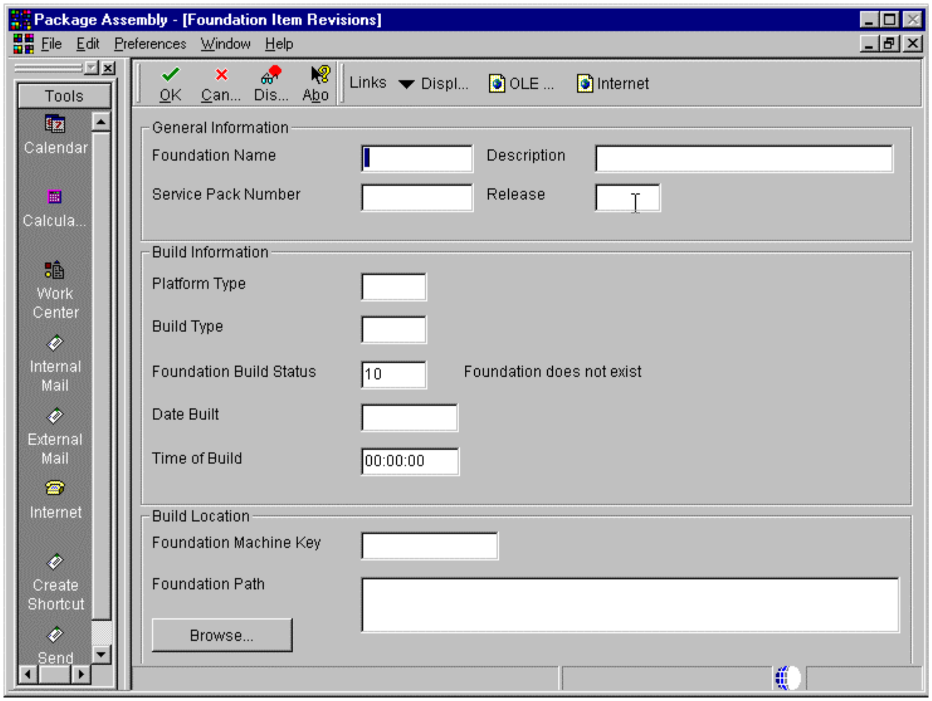


Figure 4-17 Foundation Items Revision display

12.The Select Directory panel appears as shown in Figure 4-18. Click **Network** to navigate to the system directory you want to use. Make sure not to directly select the system directory via the tree displayed in the Select Directory window.

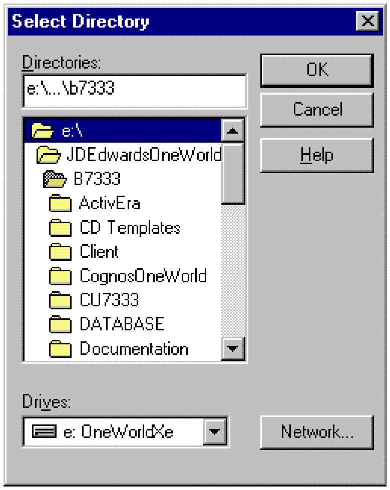


Figure 4-18 Select Directory

13.The Map Network Drive panel appears as shown in Figure 4-19. Expand the directory to the machine name of the deployment server in the Shared Directories window. Choose the shared **B733x** directory and click **OK**.

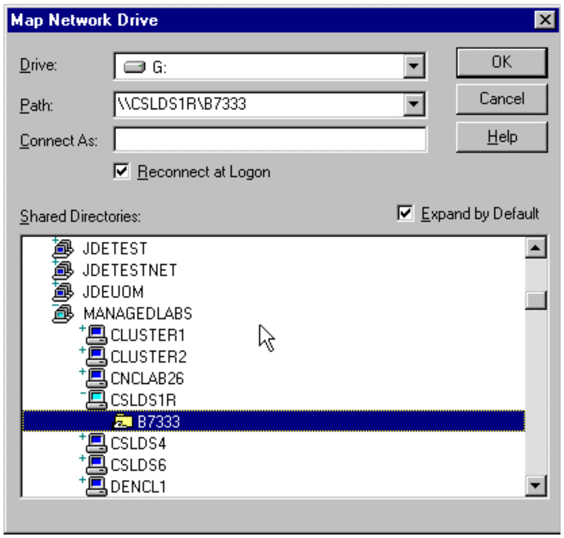


Figure 4-19 Map Network Drive

14. You return to the Select Directory display as shown in Figure 4-20. Navigate to the \System SPx\System directory and click **OK**.

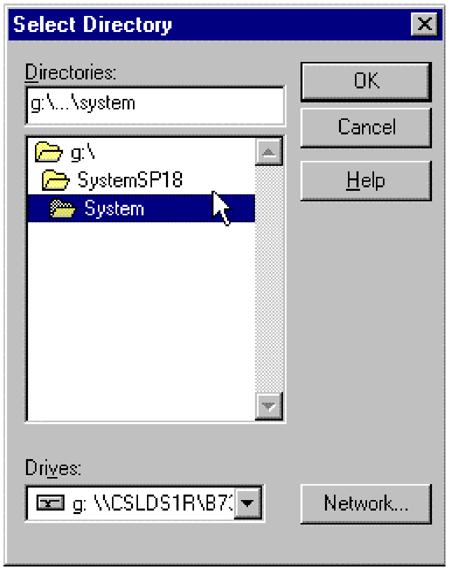


Figure 4-20 Select Directory

15. The program returns to the Foundation Item Revisions form, with the System directory you selected into the Foundation Path field. See Figure 4-21.

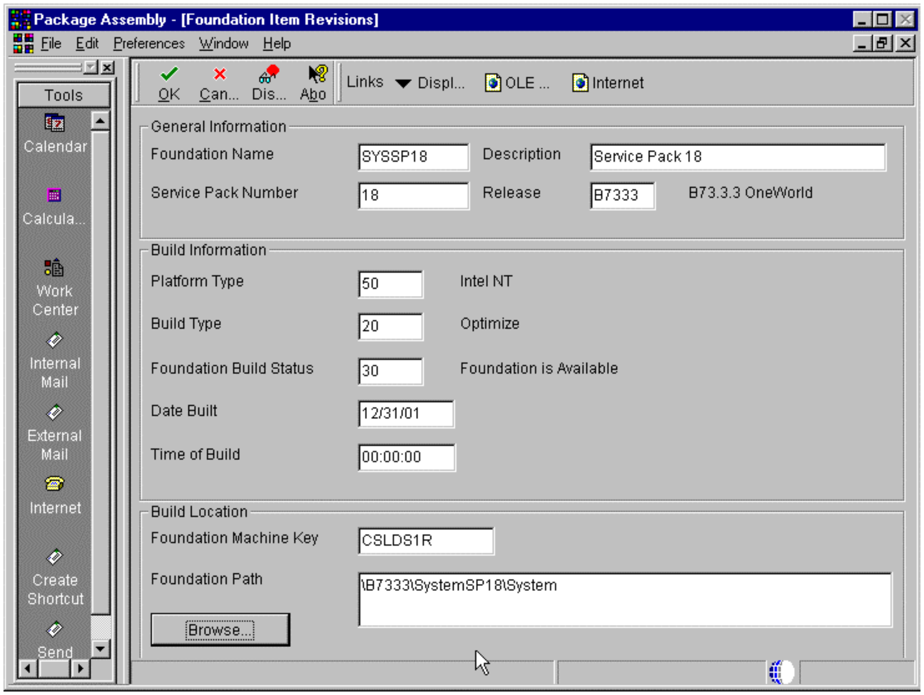


Figure 4-21 Foundation Item Revisions

16.On the Foundation Component Selection display (Figure 4-22), click the **Find** button and the foundation you entered should be listed in the grid. Choose the foundation you entered and click **Select**.

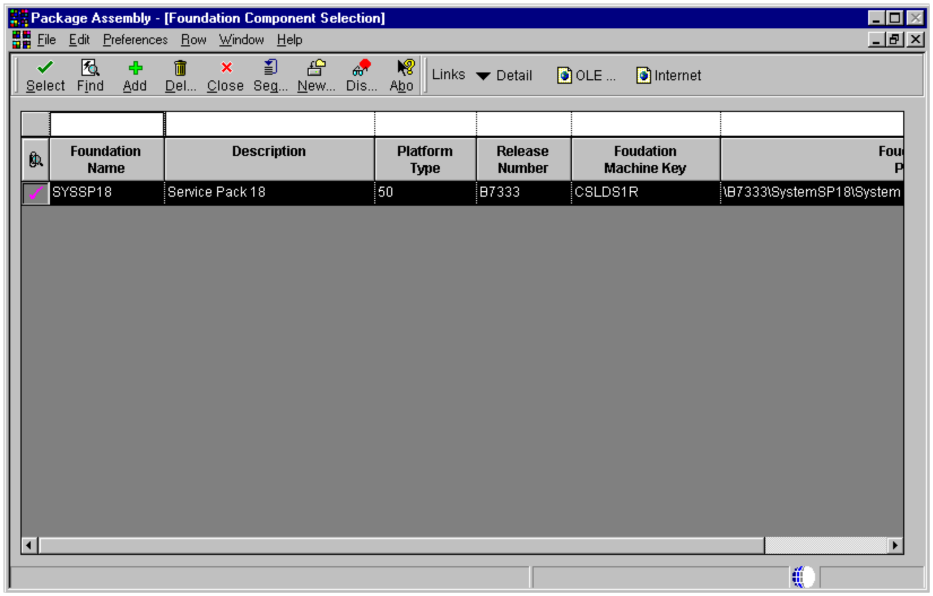


Figure 4-22 Foundation Component Selection

17.On the Foundation Component display (Figure 4-23), verify that the Foundation Path window lists the correct path to the foundation and then click **Next**.

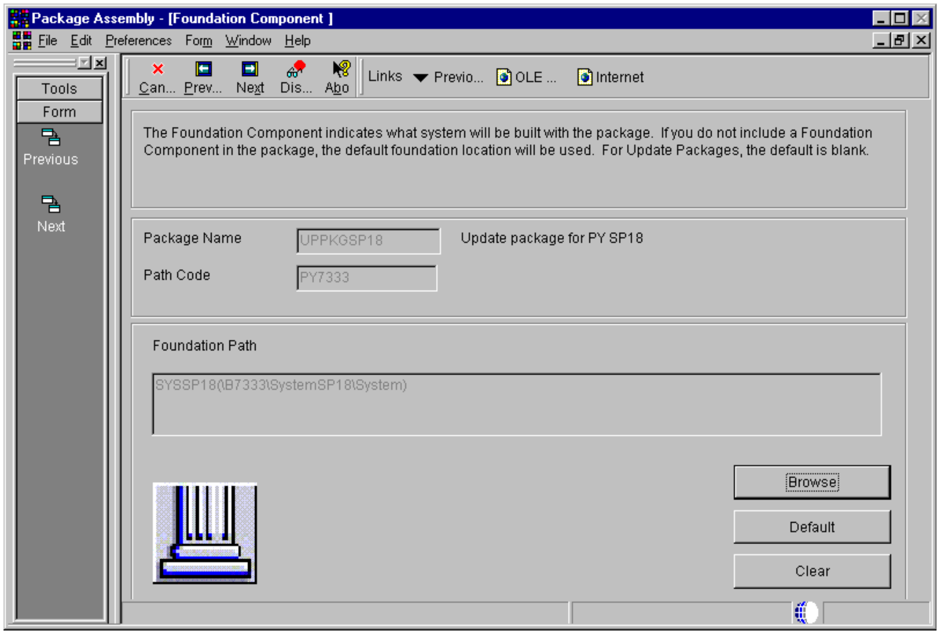


Figure 4-23 Foundation Component

18.On the Help Component display, accept the displayed default location by clicking **Next** or clicking **Default**, unless you want to point to custom help files.

19.On the Database Component display, accept the default location by clicking **Next** or clicking **Default**.

20.On the Object Component display, click **Next**.

21.On the Features Component display, click **Next**, which brings up the Language Component display. Since we are working on an installation in English, click **Next**. English is selected automatically because it is the default value for language.

22. This takes you to the Package Component Revisions display (Figure 4-24). Here, you can review every selection at a glance. Click **End**.

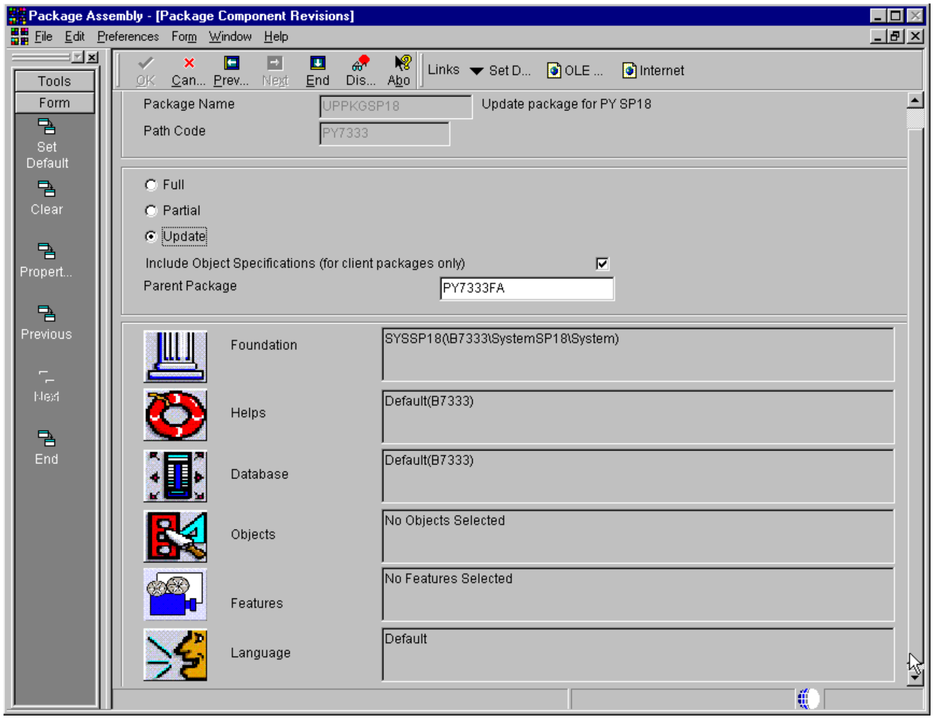


Figure 4-24 Package Component Revisions

23. You return to the Work with Packages display (Figure 4-25).

   a. Click **Find**. Then you should be able to see the package that is represented through an open box.

   b. Highlight the package and click **Activate/Inactivate** in the Row Exit Bar.

   c. Highlight the package again and click **Define Build** on the Row Exit Bar.

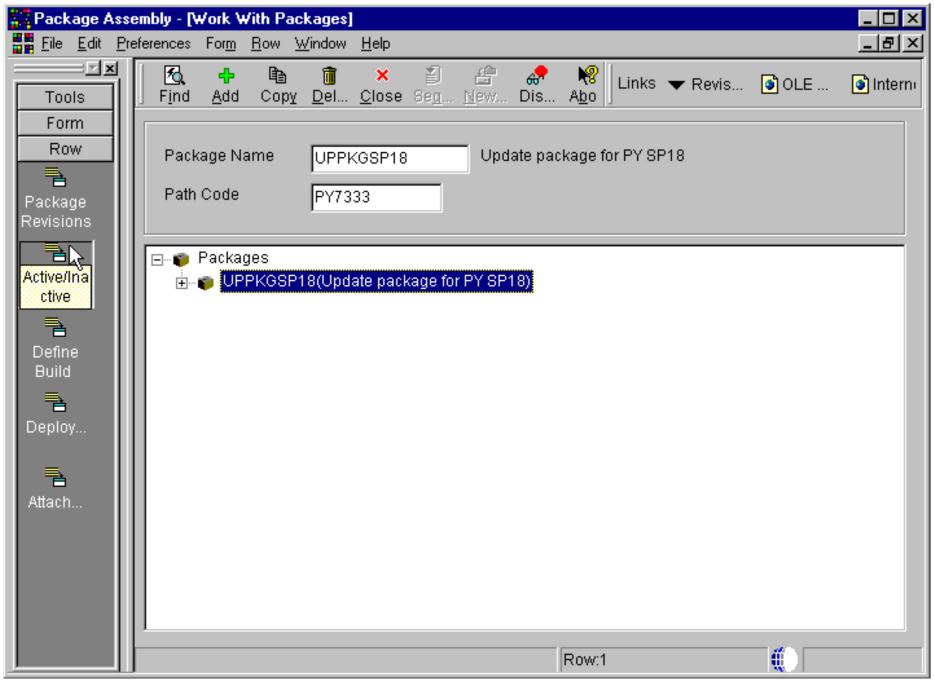


Figure 4-25 Work with Packages

24. Now you see the Package Build Definition Director display (Figure 4-26). Click **Next**.

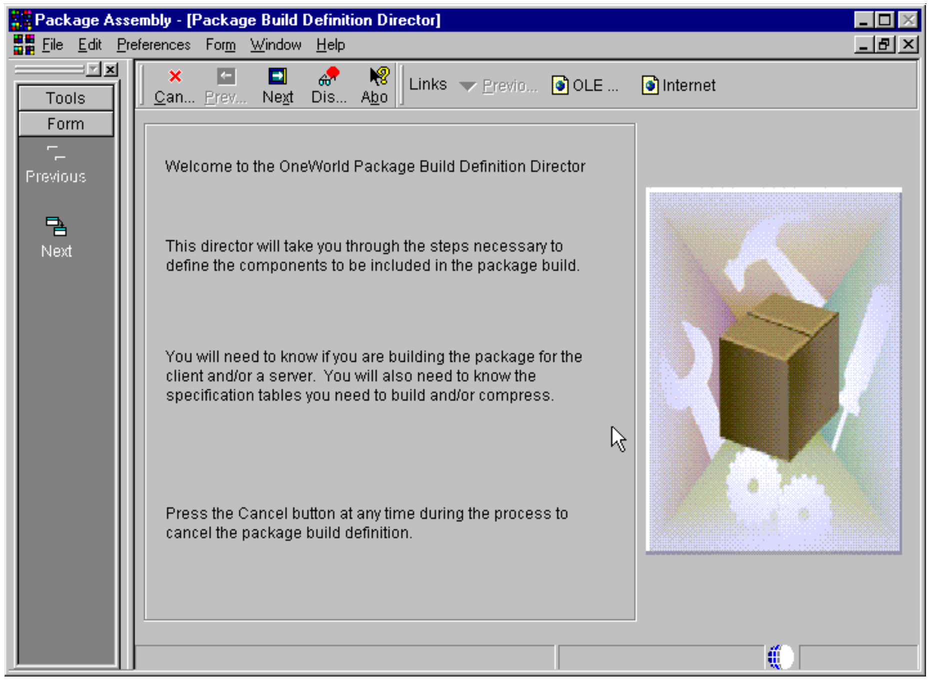


Figure 4-26 Package Build Definition Director

25. The Package Build Location display (Figure 4-27) appears. Make sure that **Client** box is selected since we are building client package to deploy the new service pack to the workstations. Click **Next**.

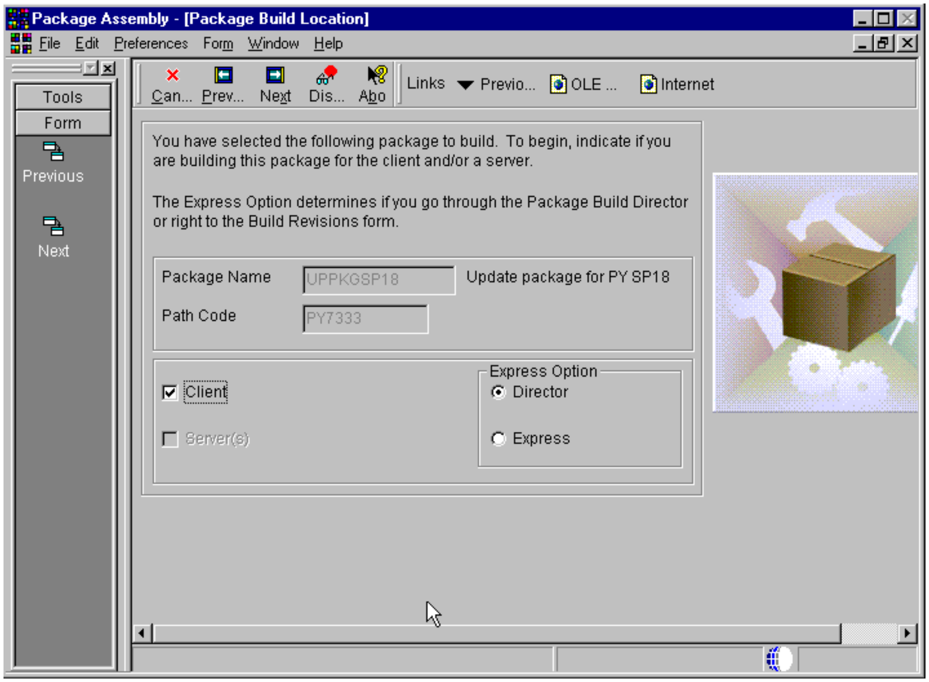


Figure 4-27 Package Build Location

26. You see the Build Specification Options display. Click **Next**.

27. On the Compression Options display (Figure 4-28), make sure that **Compress Options** is selected. Select the **Compress Foundation** check box. Click **Next**.

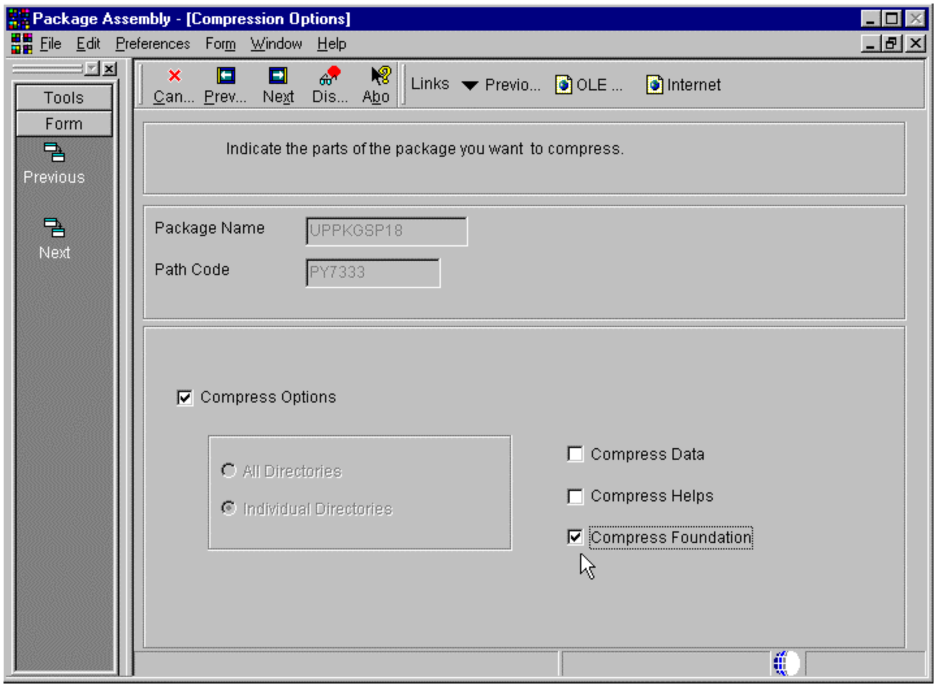


Figure 4-28 Compress Options

28. The Package Build Revisions display appears. Click **End**.

29. On the Work With Package Build Definition display (Figure 4-29), click **Activate/Inactivate** from the Row Exit bar. Then click **Submit Build** from the Row Exit Bar as well.

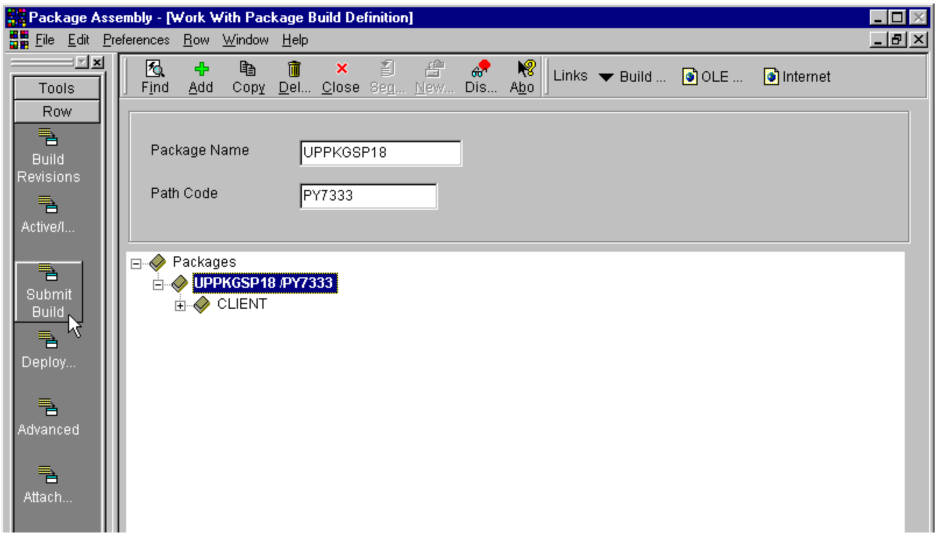


Figure 4-29 Work with package build definition

After successful completion of the UBE (R9621), the new foundation is ready to be deployed to client workstations.

4.5.4 Deploying client update packages to workstations

To deploy the package, you need to activate the package from the Package Deployment (P9631) application and assign the update package to specific client workstations.

4.5.5 Updating client workstations and the jde.ini file

In order for the OneWorld client workstations to access the new service pack, you must edit the jde.ini file in client workstations to point to the port number that matches the INI in the new system on the enterprise server. You need to:

1. On the OneWorld workstation, open Windows Explorer. Navigate to the c:\WINNT directory and open the jde.ini file in Notepad.
2. Locate the [JDENET] section of the file, and edit the port settings for *serviceNameListen* and *serviceNameConnect*. Change these entries to match those of the server jde.ini file for the new service pack:

```
[JDENET]
serviceNameList=portNumber
serviceNameConnect=portNumber
```

Here *portNumber* is the same port number you entered in the jde.ini file on the enterprise server.

4.6 Promoting the service pack to production

After the service pack is thoroughly tested, the new foundation code may be promoted to the production environment on iSeries server. Perform the following steps:

1. End OneWorld and TCP/IP Communication Services.
2. Prepare the iSeries server for service pack promotion.
3. Delete and recreate the subsystem entry.
4. Verify the library list setup.
5. Link the path code business functions to the service pack.
6. Start TCP/IP and OneWorld services.
7. Validate the promoted service pack on the iSeries server.
8. Promote the service pack on the deployment server.
9. Validate the promoted service pack on the deployment server.

The following sections explain these steps in detail.

4.6.1 Ending OneWorld and TCP/IP communication services

Perform the following steps:

1. Sign on to the iSeries server as `ONEWORLD`.
2. Select the current production release.
3. End OneWorld services with the command:

   ```
   ENDNET
   ```

4. Clear Inter Process Communications with the command:

   ```
   CLRIPC
   ```

5. Sign off from the iSeries server and then sign back on as `ONEWORLD`.
6. Select the OneWorld release that the service pack was applied to.
7. End the OneWorld services with the command:

   ```
   ENDNET
   ```

8. Clear Inter Process Communications with command:

   ```
   CLRIPC
   ```

9. Sign off from the iSeries server and then sign back on as QSECOFR. To make sure that no object lock exists in any system library (for example, `B7333SYS`) or any other OneWorld library, perform these steps:
   a. End TCP/IP using the ENDTCP command.
   b. End all host services with the command:
      ```
      ENDHOSTSVR *ALL
      ```
   c. End the subsystem QUSRWRK using the command:
      ```
      ENDSBS SBS(QUSRWRK)
      ```
      Note: Starting with OS/400 V5R1, the host server job for the Client Access Express database serving functions runs in the QUSRWRK subsystem instead of the QSERVER subsystem as in previous releases.

4.6.2 Preparing the iSeries server for service pack promotion

Perform the following steps:

1. Delete all locks from the system library using the command:
   ```
   WRKOBJLCK OBJ(QSYS/B733XSYS) OBJTYPE(*LIB)
   ```
   Here *X* is the Cumulative Update number of the OneWorld release (for example, `B7333SYS`).

   If you find any object locks, end the job by typing option `4` (End job).
2. Make sure the system library (for example, `B733XSYS`) does not contain any *SQLPKG, *JRN, or *JRNRCV object types. In case of *SQLPKG, you can use the DLTSQLPKG command to delete those SQL packages. Here are a couple of examples:
   – Example 1:
     ```
     DLTSQLPKG SQLPKG(systemlib/R*)
     ```
     This command deletes all SQL packages that start with “R” (for example, `R0006p`) in the system library (for example, `B733XSYS`).
   – Example 2:
     ```
     DLTSQLPKG SQLPKG(systemlib/O*)
     ```
     This command deletes all SQL packages that start with “O” (for example, `OWPKG`) in the system library (for example, `B733XSYS`). It is mandatory to delete all SQL packages in the system library before you proceed with the installation, because a library cannot be renamed with an SQL package associated with it.

SQL packages are always recreated whenever necessary, so it is all right to delete them.

Again, the system library is not supposed to contain any object type *JRN or *JRNRCV; if they exist, make sure to delete them before you proceed with the installation.

3. End journaling on the iSeries server:

```
ENDJRNPF FILE(*ALL) JRN(OWJRNL/OW_JRNL)
```

4. Delete all SQL packages from the OneWorld system library:

```
WRKOBJ OBJ(B733XSYS/*ALL) OBJTYPE(*SQLPKG)
```

Here *X* is the Cumulative Update number of the OneWorld release (for example, `B7333SYS`).

Use option `4` to delete all SQL packages *except* for ones that start with “Q”.

> **Attention:** Do not delete any system packages in libraries QIWS and QSYS. Delete *only* the JDE owned SQL packages.

5. Enter the following command to clear any locks in the system library (for example, `B73SPSYS`):

```
WRKOBJLCK OBJ(QSYS/syslib) OBJTYPE(*LIB)
```

If you find any object locks, make sure to end the job by typing option `4` (End job).

6. Change the authority of *PUBLIC for the ONEWORLD user profile to `*USE`:

```
GRTOBJAUT OBJ(ONEWORLD) OBJTYPE(*USRPRF) USER(*PUBLIC) AUT(*USE)
```

7. Rename the system library that needs to be promoted:

```
RNMOBJ OBJ(syslib) OBJTYPE(*LIB) NEWOBJ(B733XSYS)
```

Here *syslib* is the name of the system library being promoted to production and *X* is the Cumulative Update number of the current OneWorld release.

8. Rename the IFS directory that needs to be promoted:

```
RNM OBJ('/syslib') NEWOBJ('/B733XSYS')
```

Here *syslib* is the name of the system directory being promoted to production and *X* is the Cumulative Update number of the current OneWorld release.

9. Copy the PRINTQUEUE files from the SYSTEM library backup to the newly restored system library:

```
CPYF FROMFILE(B733XSYSBK/PRINTQUEUE) TOFILE(B733XSYS/PRINTQUEUE)
FROMMBR(*ALL) TOMBR(*FROMMBR)    MBROPT(*REPLACE) CRTFILE(*YES) FROMRCD(1)
```

Here *X* is the Cumulative Update number of the OneWorld release (for example, B7333SYS).

10. Rename the jde.ini file by issuing the following command:

```
RNMOBJ OBJ(B733XSYS/INI) OBJTYPE(*FILE) NEWOBJ(OLD_INI)
```

Here *X* is the Cumulative Update number of the OneWorld release (for example, B7333SYS).

11. Copy the jde.ini file from the system backup library to the newly restored system library:

```
CRTDUPOBJ OBJ(INI) FROMLIB(B73XSYSBK) OBJTYPE(*FILE) TOLIB(B733XSYS)
DATA(*YES)
```

Here *X* is the Cumulative Update number of the OneWorld release (for example, B7333SYS).

4.6.3 Deleting and recreating the subsystem entry

Perform the following steps:

1. Sign onto the iSeries server as QSECOFR.
2. If the subsystem library still exists, delete it:

```
WRKSBSD SBSD(systemlib/*ALL)
```

Here *systemlib* is the name of the OneWorld system library (for example, B733SYS).

3. Enter option 4 next to the entry that includes the subsystem (for example, JDEB733SP) for the service pack being promoted.
4. Add the OneWorld SYSTEM library to the QSECOFR library list:

```
ADDLIBLE B733XSYS
```

Here *X* is the Cumulative Update number of the OneWorld release (for example, B7333SYS).

5. Enter the following commands (assume service pack release 15 or later):

```
CHGCMD CMD(CRTOWSBS) PGM(*LIBL/CRTOWSBS)
CRTOWSBS SUBSYSTEM(newsubsystem) SYSLIB(B733XSYS)
```

Here *newsubsystem* is the name of the new subsystem for the newly installed service pack (for example, JDEB733*X*), and *X* is the Cumulative Update number of the OneWorld release (for example, B7333SYS).

4.6.4 Verifying the library list setup

Perform the following steps:

1. Sign onto the iSeries server as `ONEWORLD`.
2. To ensure the library list is set up correctly, verify that it contains the newly created OneWorld system library (for example, `B7333SYS`).

   ```
   DSPLIBL
   ```

4.6.5 Linking path code business functions to the service pack

For each OneWorld path code associated with the new service pack, the business functions need to be relinked:

1. On the iSeries server, sign on as `ONEWORLD`.
2. Type the LINKBSFN command from the command prompt and press F4.
3. Enter the package library name for the first parameter (for example, `PY7333FA`) and the path code for the second parameter (for example, `PY7333`).

4.6.6 Starting TCP/IP communication and OneWorld services

Sign on as QSECOFR on the iSeries server console and then execute the following commands:

1. Start TCP/IP using the STRTCP command.
2. Start all host servers with the command:

   ```
   STRHOSTSVR *all
   ```

3. Start the subsystem QUSRWRK with the command:

   ```
   SRTSBS QUSRWRK
   ```

4.6.7 Validating the promoted service pack on the iSeries server

To ensure that the service pack is properly promoted on the enterprise server, review the service pack release of the system library by starting the OneWorld services, and then run PORTTEST.

> **Note:** If the Security server is on, then OneWorld services must be running before you execute PORTTEST.

Perform the following steps:

1. Sign on to the iSeries server as `ONEWORLD`.

2. On the Current OneWorld versions, choose the newly installed instance of **OneWorld**.
3. Run the following command from the command prompt:

   ```
   DSPPFM FILE(B7333SYS/LOG)
   ```

   You see the Display Physical File Member screen as shown in Figure 4-30.

```
                         Display Physical File Member
File . . . . . . :   LOG                 Library  . . . . :   B7333SYS
Member . . . . . :   PTF      Record . . . . . :   1
Control  . . . . .                       Column . . . . . :   1
Find . . . . . . .
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
000100000000B733
000200000000SP18
000300000000
                          ****** END OF DATA ***
```

Figure 4-30 Display Physical File Member

4. Clear Inter Process Communications:

   ```
   CLRIPC
   ```

5. Start OneWorld subsystem using the STRNET command that is supplied in OneWorld system library (for example, B7333SYS):

   STRNET SBS(*subsystem*)

   Here *subsystem* is the name of the new subsystem for the newly installed service pack (for example, JDEB733).

6. Verify that both the NETWORK and SENTINEL are running using the WRKACTJOB command:

   ```
   WRKACTJOB  SBS(JDEB7333)
   ```

   The Work with Active Job display appears as shown in Figure 4-31. Verify that the NETWORK process is running with a PGM-MONITOR in SELW status and that the entry SENTINEL is running with a PGM-MONITOR in SIGW status. Until a net request is performed, the CPU remains at 0.

```
                         Work with Active Jobs                    JDECS8
                                                            11/28/01
14:32:10
 CPU %:     .3     Elapsed time:   00:00:28     Active jobs:   243

 Type options, press Enter.
   2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
   8=Work with spooled files   13=Disconnect ...

 Opt  Subsystem/Job  User       Type  CPU %  Function       Status
      JDEB7333       QSYS       SBS      .0                  DEQW
        JDENET_K     ONEWORLD   BCI      .0                  DEQW
        JDENET_K     ONEWORLD   BCI      .0                  DEQW
        JDENET_K     ONEWORLD   BCI      .0                  DEQW
        NETWORK      ONEWORLD   BCH      .0  PGM-JDENET_N    SELW
        SENTINEL     ONEWORLD   ASJ      .0  PGM-MONITOR     SIGW
```

Figure 4-31 Work with Active Jobs

7. On the command prompt type `PORTTEST`, and then press F4 to display the system prompt (Figure 4-32). Enter the User Profile, User Password, and Environment (for example, `PD7333`) that are needed to verify that OneWorld is running properly.

```
                      RUN PORTTEST (PORTTEST)

 Type choices, press Enter.

 User Profile . . . . . . . . . .   JDE           Name
 User Password  . . . . . . . . .   JDE           Character value
 Environment  . . . . . . . . . .   PD7333        Name





Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 4-32 PORTTEST

4.6.8 Promoting a service pack to production on the deployment server

Perform the following steps.

1. Sign on to the deployment server as an administrator on the PC.
2. Make sure no user is logged on to the OneWorld on the deployment server.
3. Navigate to the \JDEdwardsOneWorld\B7333 directory, and create a directory called `System Backup`.
4. Copy the System and SYSTEMCOMP directories into the System Backup directory.
5. Navigate to the \JDEdwardsOneWorld\System SPy directory, and move the System and SYSTEMCOMP directories into the \JDEdwardsOneWorld\B7333 directory.

4.6.9 Validating the promoted service pack on the deployment server

To ensure that the correct service pack has been applied, verify that the PTF.LOG, located in the \system\bin32 folder, displays the promoted service pack release.

5

HTML client and WebSphere setup and tuning

This chapter describes the HTML client and WebSphere Application Server settings that we found during testing. These settings provide the best overall end-user response time, while keeping the system load as low as possible.

We tested with V5R1 of OS/400 on the iSeries server with WebSphere 3.5.4 and OneWorld Xe SP14.1_WEBP, using two different hardware configurations.

> **Note:** While the rest of the redbook covers OneWorld Xe SP17, the tests in this chapter were done with SP14.1, since SP17 was not available. However, most of the recommendations provided in this chapter should apply to both SP14.1 and SP17 versions.

5.1 Introduction

Since the HTML client has quite a small footprint, requiring only a Web browser, it has become a very popular client choice for OneWorld. For the best performance of the HTML client, parameters in the WebSphere Application Server, the HTTP Server, and the OneWorld JAS and enterprise servers need to be set. Many of these configurable parameters are based on the number of users.

For background or additional information for this chapter, refer to the *OneWorld Java Server OneWorld Installation Guide* and the IBM Web site:
http://www.as400.ibm.com/products/websphere/docs/as400v35std/docs/index.html

Consider these parameter guidelines as a starting point. Your environment will be different from our lab and your results may vary. For example, if you have fat or TSE clients, run a different mix of batch jobs (UBEs), or if you have other applications running on your system, additional iterations may be required to optimize the performance of your implementation. Keep in mind that there are steady improvements and fixes in the iSeries server software and OneWorld code. Therefore, over time, the guidelines established by our ongoing testing will continue to evolve as well.

For each configurable parameter, we include:

- An explanation.
- Where practical, the actual settings for 100 users (50 users running distribution applications, 25 users running manufacturing applications, and 25 users running financial applications). You should extrapolate these settings for your environment based on your number of users.
- The recommendation summarized in general terms.

5.2 Test environment for establishing recommendations

For the first test configuration, All in One (AIO), everything was installed on a single iSeries server (OneWorld JAS server, WebSphere, OneWorld application, and database servers). The second hardware configuration, Virtual 3-Tier (V3T), had one iSeries server setup as the OneWorld JAS server with WebSphere, and another iSeries server setup as the OneWorld enterprise (application and database) server. You can see a diagram of the two hardware configurations in Figure 5-1 and Figure 5-2.

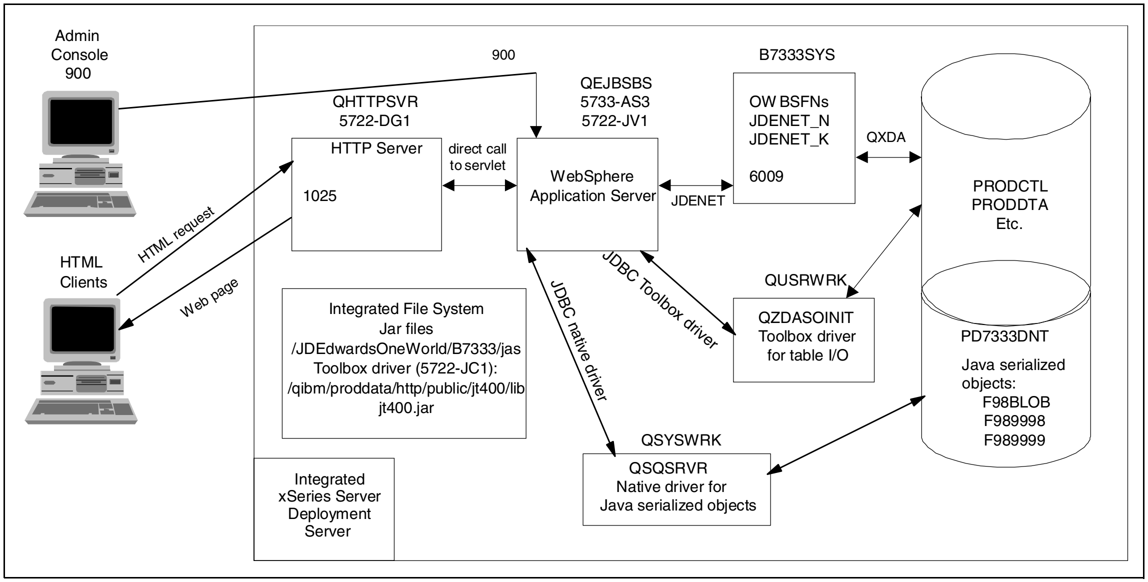


Figure 5-1 All in One (AIO) configuration

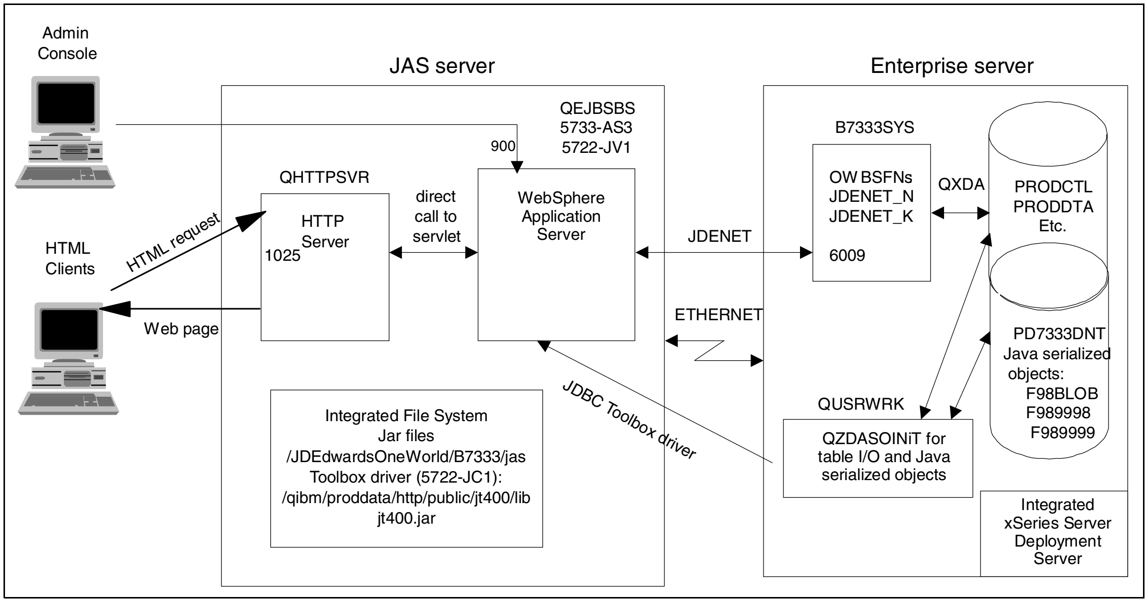


Figure 5-2 Virtual 3-Tier (V3T) configuration

Both iSeries servers used in this testing had the level of software as shown in Table 5-1. This combination of software offered the latest fixes available at the time of the testing and is included for your reference. Keep in mind that software changes and improves over time, so we generally recommend that customers stay current on service packs.

Table 5-1 Test environment service pack levels

Software for test environment	Description	Service pack
OS/400	V5R1	Cum C1254510
Database	SF99501	Service pack #4
OneWorld	Xe	SP14.1_WEBP
HTTP	QHTTPSVR/SF99156	Service pack #5
WebSphere Standard Edition 3.5.4	QEJB/SF99146	Service pack #4
Java	QJAVA/SF99069	Service pack #5
Client Access Express V5R1 (only needed on a fat client to do the Java generation, but not required for OneWorld HTML clients)	SI01907	Service pack #2

We followed the PTF recommendations in Informational APAR II12878 for V5R1. You can find the APAR on the Web at:

http://www-1.ibm.com/servers/eserver/iseries/service/bms/jde-support.htm

This Web site includes Informational APARs for other releases as well as other documentation pertinent to running J.D. Edwards software on the iSeries server.

5.3 WebSphere Application Server

Most of the settings described in this section are set using the WebSphere Administrative Console on the workstation (typically a PC).

5.3.1 Number of WebSphere Application Server instances (or JVMs)

HTML clients use a Web browser running Internet Explorer 5.5, which connects through the HTTP Server running on the iSeries server to pass requests to WebSphere (5733-AS3) running OneWorld JAS servlets and JSPs. We followed the JAS installation instructions for creating the JAS server instance (referred to as a JVM because each instance runs in its own Java Virtual Machine (JVM)). We found that each JVM provided the best performance if it supported up to 400 users. Running 400 or less users per JVM reduces the memory contention within WebSphere and therefore reduces the response time for the HTML clients. If you have more than 400 users, you need to create additional instances to provide for the additional users, as needed.

> **100 users example:** We created one JVMs to accommodate our 100 users.

> **Note:** Follow the JAS installation instructions and create one or more instances to run the JAS code for up to 400 users running in each instance.

When more than one WebSphere Application Server instance is required, you can set up OneWorld's JAS redirector or have your users run to separate HTTP ports. We chose the latter because our test environment required this.

5.3.2 Heap size memory settings

You can control how much of the iSeries server's total memory is available for the JAS server by varying the heap size. You vary the heap size for each instance via the command line arguments within each JVM. WebSphere allows for an initial heap setting and a maximum heap setting.

We found that WebSphere on the iSeries server performed best when using only an initial heap size setting and allowing the system to grow the heap to the maximum size needed. The iSeries server is unique in managing the maximum heap size; both AIX and Windows implementations require the maximum parameter.

In our test, we varied our initial heap size between 6 MB and 10 MB per user and found that having an initial heap size of 7.5 MB memory per user was the optimal setting. Our command line parameter setting was `–Xms750m`, which indicates 750 MB of initial memory allocated (7.5 MB x 100 users). We found that the largest memory utilization stabilized at slightly more than double the initial heap size so our effective utilization was about 15 MB of memory per user.

Depending on the application mix, you may need to raise or lower your initial heap size. You can review your heap size by using OneWorld's "Web" SAW.

> **100 user example:** In the JVM command line, we added the parameter –Xms750m, which sets the initial heap size to 750 MB.

To adjust the initial heap size, follow these steps:

1. From your WebSphere administrative console, expand your node (DENAS3.MLAB.JDEWARDS.COM), and double-click your JVM (JDEJAS1). Then you see a display like the example in Figure 5-3.
2. Review the command line arguments by scrolling left and right within them.
3. Either change or add the initial memory setting to reflect the size you chose.
4. Remove the maximum heap argument (-mx) if it is there.
5. Click **Apply** and watch for the console message below that says `Command "JDEJAS1.ModfiyAttributes" completed successfully`.
6. You must stop and restart your JVM for the changes to take effect.

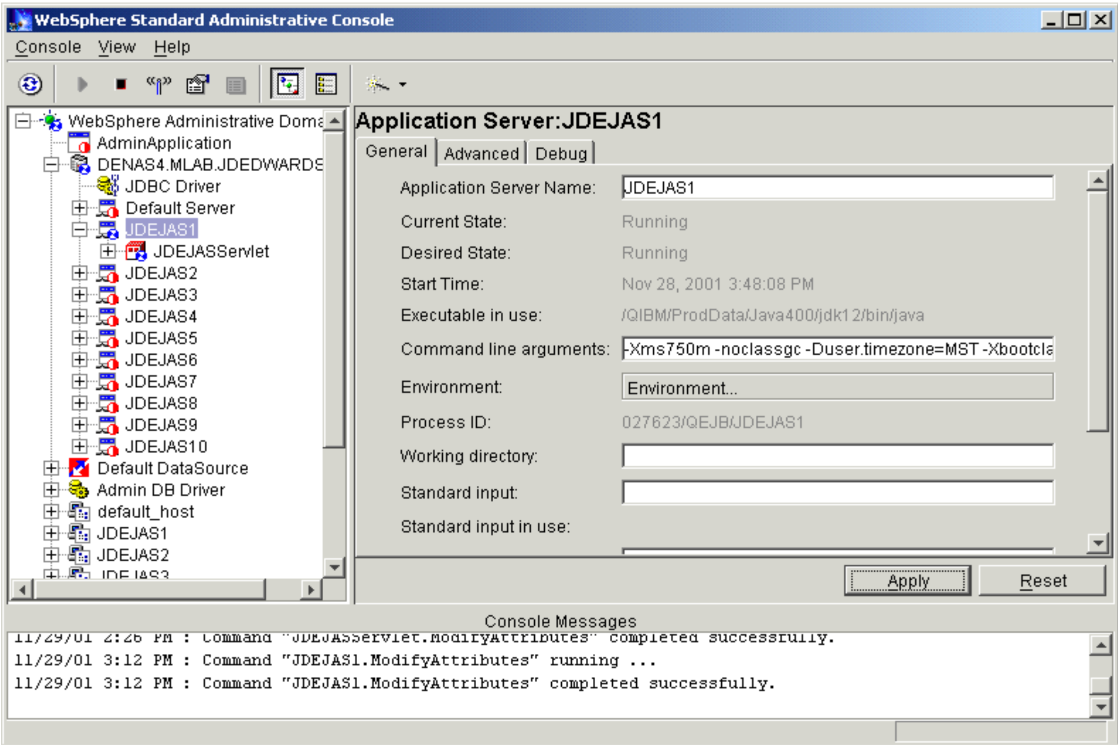


Figure 5-3 Heap and garbage collection example

> **Recommendation:** For the heap size in each JVM, on the command line, set an initial heap of 7.5 MB memory per user and no maximum heap setting. Total memory utilization is effectively 15 MB per user at this setting.

5.3.3 Garbage collection setting

Java manages its memory usage by periodically cleaning up storage that is not in use. This is referred to as *garbage collection*. WebSphere loads the OneWorld Java classes and garbage collects them as needed. Because these classes are constantly used in this environment, we set the command line arguments to prevent the classes from being garbage collected by using the `–noclassgc` setting.

Referring to Figure 5-3, you follow these steps:

1. Review the command line arguments by scrolling left and right within them.
2. Add the `-noclassgc` argument, if it is not there already.
3. Click **Apply** and watch for the console message that says `Command “JDEJAS1.ModfiyAttributes” completed successfully`.
4. You must then stop and restart your JVM for this change to take effect.

> **Recommendation:** Use the `–noclassgc` setting in the WebSphere instance command line arguments.

5.3.4 JAS servlet connections

Within the instance, we configured the servlet running the OneWorld JAS code to control how many users can run concurrently within the JVM. Because we ran 100 users in our JVM, we set the maximum number of connections for the servlet to be 100. The maximum should be equal to the number of the users. This setting should not exceed 400 per instance because each user uses a thread, and we only want to allow 400 threads per instance. Because more of a user's time is spent on data input than running the servlet, changing this setting did not significantly affect performance.

> **100 user example:** We set the maximum number of connections for the servlet to 100.

To change the maximum connections, follow these steps:

1. From your WebSphere administrative console, expand your node (DENAS3.MLAB.JDEDWARDS.COM) and expand your JVM (JDEJAS1). Click **JDEJASServlet**.
2. Click the **Advanced** tab.
3. You see a window like the example in Figure 5-4. Change the maximum connections. Click **Apply** and watch for the console message that says `Command “JDEJASServlet.ModifyAttributes” completed successfully`.

4. Stop and restart your JVM for this change to take effect.

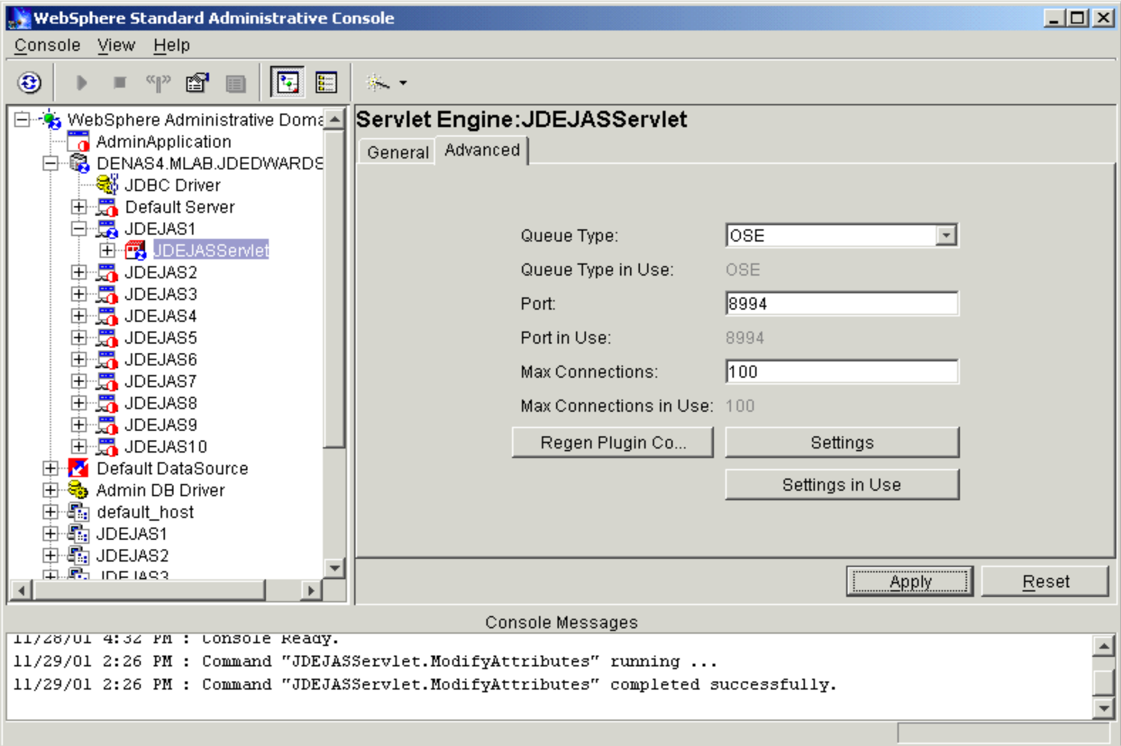


Figure 5-4 JAS servlet connections

Recommendation: Set the number of OneWorld servlet connections to equal the number of users connecting to the WebSphere instance, with a maximum of 400.

5.3.5 Java program optimization

We looked at optimizing the OneWorld Java code that runs within WebSphere. Java is an interpreted language where byte code is interpreted into directly executable machine instructions. To improve performance, the iSeries JVM can create a directly executable program for a Java .class or .jar file. This iSeries feature gives you the option to create a persistent Java program that is associated with the OneWorld JAS.jar file. In some cases, creating the persistent Java program provides better performance. If a Java program is not created, the iSeries JVM (V4R5 or later) will use its Just In Time (JIT) compiler to create machine instructions prior to executing the .class or .jar file.

In our tests, we created Java programs for the JAS.jar file at two optimization levels (30 and 40) and compared the performance of JIT. Optimization 30 ran slower than JIT. Optimization level 40 resulted in errors. These errors forced us to use the Licensed Internal Code option (LICOPT) `–nopreresolveextref` parameter. As a result, optimization 40 also ran slower than JIT. We found the best performance was achieved by using JIT and not creating the Java program.

Recommendation: Do not create a Java program for the OneWorld JAS.jar file. (This is the default, so no action is required.)

Follow these steps:

1. To verify your setting, type the following command on an OS/400 command line:

   ```
   DSPJVAPGM CLSF('/jdedwardsoneworld/b7333/jas/jas.jar')
   ```

   It should say `OPTIMIZATION(*INTERPRET)` as shown in the example in Figure 5-5, or it should say "`no Jvapgm associated with this class file`".

```
                    Display Java Program Information
 File name  . . . . . . . . . . . . . . . : wardsoneworld/b7333/jas/jas.jar
 Owner  . . . . . . . . . . . . . . . . . :   GUEST




 Java program creation information:
   File change date/time  . . . . . . . . . . . . :   09/18/01  08:09:24
   Java program creation date/time  . . . . . . . . :   11/13/01  15:05:19
   Java programs  . . . . . . . . . . . . . . . . . :   239
   Total classes in source  . . . . . . . . . . . . :   1549
   Classes with current Java programs . . . . . . . :   239
   Classes without current Java programs  . . . . . :   1310
   Classes containing errors  . . . . . . . . . . . :   0
   Optimization . . . . . . . . . . . . . . . . . . :   *INTERPRET
   Enable performance collection  . . . . . . . . . :   *NONE
   Use adopted authority  . . . . . . . . . . . . . :   *NO
   User profile . . . . . . . . . . . . . . . . . . :   *USER

More...
 Press Enter to continue.

 F3=Exit   F12=Cancel
```

Figure 5-5 DSPJVAPGM

2. If the optimization is 30 or 40, type the following command to delete the Java program:

```
DLTJVAPGM CLSF('/jdedwardsoneworld/b7333/jas/jas.jar')
```

3. You must then stop and restart your JVM for these changes to take effect.

5.4 HTTP Server

The IBM HTTP Server for iSeries (5722-DG1) offers two HTTP Servers on the iSeries server:

- HTTP Server (original)
- HTTP Server (powered by Apache)

We used the original HTTP Server for testing because the HTTP Server (powered by Apache) was not available when we started testing.

All of the tuning in this section for the HTTP Server involves changing the HTTP configuration file. There are two ways to access the HTTP configuration file:

- From your browser
- Via the WRKHTTPCFG command (green screen method)

The browser gives you GUI screens and extensive help text. With the green screen method, you can easily view, modify, and add to your configuration entries. We use both methods in this section.

To start HTTP Server Administration, follow these steps:

1. On the OS/400 command line, type the command:

```
STRTCPSVR *HTTP *ADMIN
```

2. Press Enter. Open an Internet Explorer browser session and type the following the command line in the browser:

```
http://server-name:2001
```

Here *server-name* is the host name of your iSeries server.

5.4.1 Number of HTTP threads

In our test environment, each HTTP Server instance, capable of supporting up to 400 users, was attached to a separate port. The HTTP Server job runs threads for each of the users that connect to the system. Setting the minimum number of threads equal to the number of users that connect to the port and the maximum number of threads to be twice that number performed best. If you use JAS redirector, its load balancing characteristics achieve a similar effect. Changing this setting improved the overall response time slightly, but it did not adversely impact the CPU utilization or other system resources.

To set the number of threads, follow these steps:

1. On your browser screen, click **IBM HTTP Server for iSeries**.
2. Click **Configuration and Administration** and then click **Configuration** on the toolbar.
3. Open the pull-down menu in the left window pane and click your JVM. Scroll down to and expand the section **System Management**. Double-click **Performance**.
4. Change your minimum and maximum threads as indicated in Figure 5-6. Then scroll down and click **Apply**.
5. You must then stop and restart your HTTP Server instance for these changes to take effect.

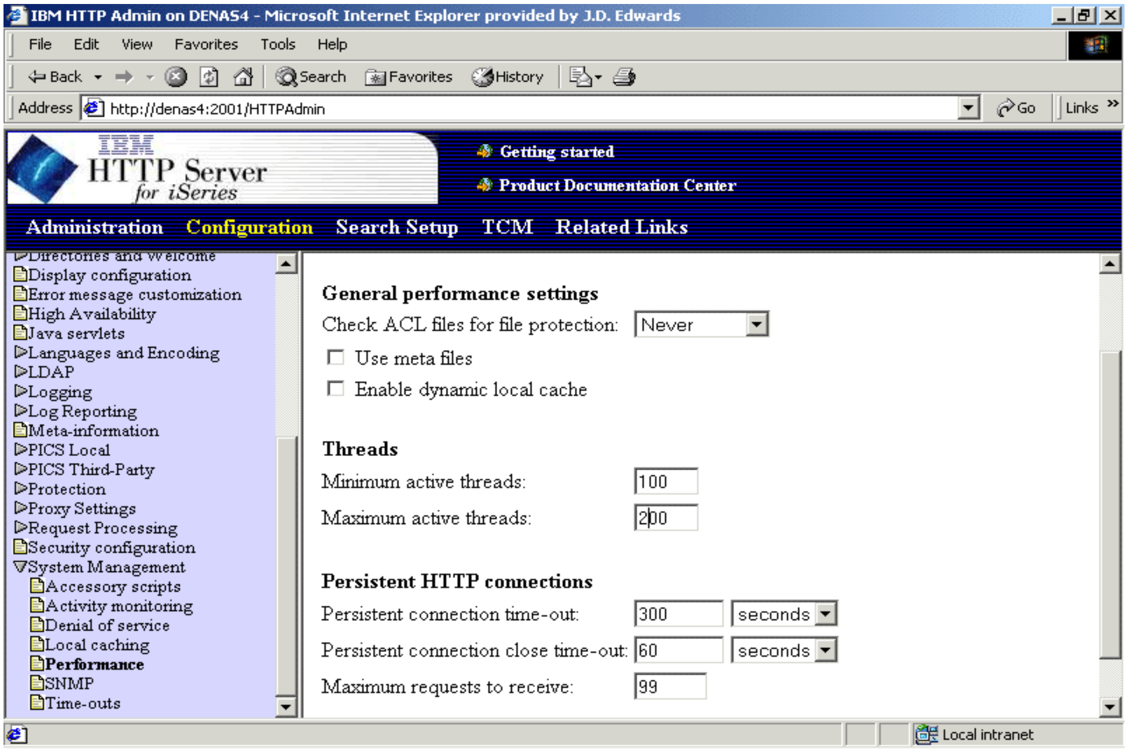


Figure 5-6 HTTP performance configuration settings

100 user example: Set Minimum active threads to 100 and Maximum active threads to 200.

Recommendation: For the minimum and maximum active threads for the HTTP Server instance, set the minimum to the number of expected users (maximum of 400) and the maximum to be about twice that amount.

For each additional WebSphere instance that you have, you need to create an additional HTTP Server instance, whether you point them to separate ports as we did, or you choose to use the redirector code. You need to have an equal number of WebSphere instances and HTTP Server instances.

5.4.2 Persistence

The HTTP Server maintains a persistent connection for any currently connected user, referred to as *persistence*. The HTTP Server drops the connection if it is not used. As shipped, the defaults for the persistent values are set very low with a four-second timeout and persisting only one request.

Because the average user does not continually key and we know that the OneWorld application sends more than one HTTP request at a time, we increased the settings. We set the time-out to 300 seconds, or five minutes, because our test users respond within 5 minutes. You may want to increase this value even higher for your environment. We also set the maximum number of requests to 99 to ensure that no single Web page needs to perform multiple receives. See Figure 5-6 for an example.

Recommendation: Set the persist settings high enough for your environment (you could start with 300 seconds and 99 requests), or your users will experience timeouts.

5.4.3 ACL and Denial of service settings

Denial of service and access control lists (ACL) checking are enabled by default within the HTTP Server. Because our test only had internal users, that is, the HTTP Server was not running outside the company firewall, we did not need these functions. We changed the Check ACLs files for file protection setting from the default of *Protect only* to **Never**, and we increased the denial of service settings. These changes did not significantly improve the performance. See Figure 5-6 for an example of the Check ACLs files for file protection setting. The denial of service settings are shown in Figure 5-7.

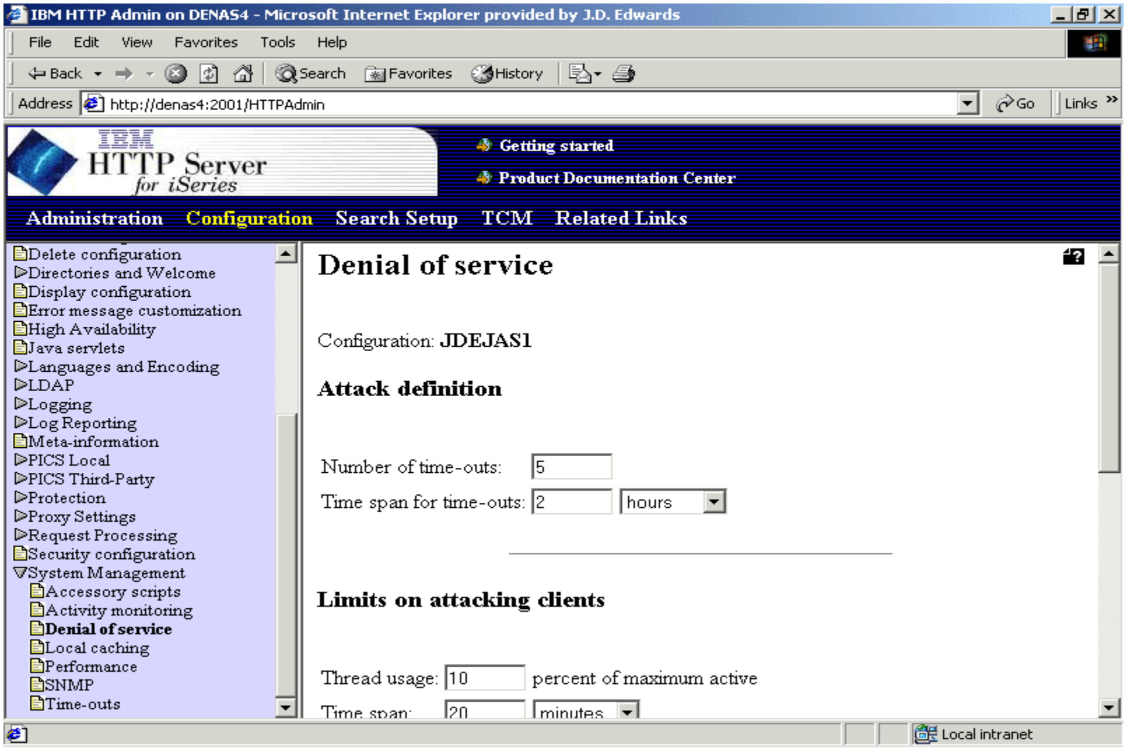


Figure 5-7 HTTP configuration of the denial of service settings

If you want to view these settings from the OS/400 WRKHTTPCFG command discussed in the following section, the pertinent entries look like this:

```
UseACLs never
DenialOfServicePenalty 10 20 Minutes
DenialOfServiceThreshold 5 2 Hours
```

Recommendation: Do not use ACLs and check for denial of service attacks very infrequently, unless they are required for your environment.

5.4.4 HTTP user profile swapping

The HTTP Server swaps between user profiles to perform some of its internal functions. If a large number of users are running on the system, this swapping can cause resource contention. OS/400 V5R1 ships with swapping turned on but offers a circumvention to the swapping. To circumvent or disable swapping, you need to add a line to the HTTP Server instance configuration. If your system is running a large number of HTML users, this change considerably reduces the contention on the user profile and improve the performance.

To turn off user profile swapping, follow these steps:

1. On an OS/400 command line, type:

   ```
   WRKHTTPCFG configuration-name
   ```

   Press Enter.

2. Select option 1 to add a line to your HTTP configuration file. Then type the content of the line as shown in (Figure 5-8), which is:

   ```
   StatSwap Off
   ```

```
                        Work with HTTP Configuration
                                                             System:   DENAS4
  Configuration name . . . . . . . :   JDEJAS1

Type options, press Enter.
  1=Add   2=Change   3=Copy   4=Remove   5=Display   13=Insert

      Sequence
Opt    Number    Entry
       00300     StatSwap Off
       00010     HostName DENAS4
       00020     Port 1025
       00030     UseACLs never
       00040     MinActiveThreads 100
       00050     MaxActiveThreads 200
       00060     PersistTimeout 300 Seconds
       00070     MaxPersistRequest 99
       00080     DenialOfServicePenalty 10 20 Minutes
       00090     DenialOfServiceThreshold 5 2 Hours
       00100     # IBM HTTP Server configuration file
```

Figure 5-8 HTTP configuration to turn off user profile swapping

3. You must then stop and restart your HTTP Server instance for these changes to take effect.

Recommendation: Disable the user profile swapping within the HTTP Server by making the configuration change.

5.5 OneWorld Xe enterprise server

Within OneWorld Xe, there are two areas for tuning:

- The enterprise server
- The JAS server

The J.D. Edwards OneWorld Xe enterprise server provides configurable parameters through the J.D. Edwards member of the ini file in the system library (in our example, B7333SYS).

1. On the OS/400 command line, type:

   ```
   STRSEU system-library-name/INI
   ```

   Press Enter.

2. Select option 2 to edit the J.D. Edwards member. On the Edit menu (Figure 5-9), scroll down to each item (discussed in the following sections) and either change the default or add an extra line as indicated.

```
Edit                                                     B7333SYS/INI
 SEU==>
JDE
 FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+...
7 ...+... 8 ...+... 9 ...+... 0 ...+... 1 ...+... 2
0040.00 [JDEIPC]
0041.00 maxNumberOfResources=1000
0042.00 startIPCKeyValue=2101
0043.00 avgResourceNameLength=15
0044.00 maxMsgqEntries=1024
0045.00 maxMsgqBytes=65536
0046.00 ipcTrace=0
0047.00
0048.00 [JDENET]
0049.00 serviceNameListen=6009
0050.00 serviceNameConnect=6009
0051.00 maxNetProcesses=1
0052.00 maxNetConnections=800
0053.00 netShutdownInterval=15
0054.00 maxKernelProcesses=50
0055.00 maxKernelRanges=13
0056.00 netTrace=0
0057.00
0058.00 [JDENET_KERNEL_DEF1]
0059.00 krnlName=JDENET RESERVED KERNEL

 F3=Exit F4=Prompt F5=Refresh F9=Retrieve F10=Cursor F11=Toggle
  F16=Repeat find       F17=Repeat change          F24=More keys
```

Figure 5-9 Editing the ini file's JDE member

3. When you are finished, press F3 to exit and save your changes.

4. You need to stop and restart OneWorld, your JVM, and your HTTP Server for the changes to take effect.

5.5.1 Kernel settings

We set the number of call object, security, and network kernel jobs based on the number of users. In our range of tests, we found that 7.5 users per call object kernel offered the best performance for the workload that we were supporting. Your ratio may vary if your call object kernels are running a different mix of business functions.

The rule of thumb for network kernel jobs is to have one for every 20 call object kernel jobs. Network kernel jobs should have a minimum of two to avoid potential bottlenecks because the first kernel acts as a traffic cop to pass off the requests, as well as perform network requests itself. One security server for every 100 users is recommended. For our 100 users, we set 14 call object kernels, one security kernel, and two network kernel jobs.

In the 100 users example, consider:

```
[JDENET]
maxNetProcesses=2 (default is 1)

[JDENET_KERNEL_DEF4]
maxNumberOfProcesses=1 (default is 1)

[JDENET_KERNEL_DEF6]
maxNumberOfProcesses=14 (default is 6)
```

Recommendation: Allow one call object kernel for every 5 to 10 users (7.5 is a good place to start), one security kernel job for every 100 users, and one network job for every 20 call object kernel jobs (with a minimum of two kernels).

5.5.2 ONEWORLD user profile security setting

The OneWorld Xe enterprise server runs under the ONEWORLD user profile. When a business function is performed by a call object kernel, the job swaps to the user profile of the user that is running the business function. A change was introduced eliminating these extra swaps, for example, the security model changed to use a single user profile (ONEWORLD) for accessing iSeries server objects. (The security server still validates the user ID and password passed from the client.)

You need to decide whether this security model change is acceptable in your environment. This change was introduced with OneWorld® Xe SP 14.1_WEBP. It significantly reduces the amount of swapping and improves the overall response time. In the [AS400] section, add the following new line:

```
[AS400]
useAS400Security=FALSE
```

You must then stop and restart OneWorld to have this change take effect. (Whenever, you recycle OneWorld, you must always also recycle the HTTP Server and JAS server.)

> **Recommendation:** Eliminate the extra user profile swaps by the call object kernels with the new setting in the ini file.

5.5.3 Checking the kernel setting

While the enterprise server is running, it passes messages between the network jobs and the kernel jobs. By default, the network job verifies that each of the call object kernels is running, every tenth time it sends or receives a message.

J.D. Edwards released an enhancement in SAR #5128281 with SP16.1 that enables you to customize this verification to occur less frequently. This change reduces the amount of overhead within the enterprise server and improves the overall response time. You can choose to check a kernel's health every 1,000th time there is a message or for example, every 5,000 times there is a message; the trade-off is performance versus prompt knowledge of a kernel problem. While you do want to continue to monitor your kernels, if you have a heavy load of transactions, you may consider increasing this number to have it check for kernel health less frequently. In the [JDENET] section, add the new line, checkKrnlHealth with the value that you choose, as in the 100 user example here:

```
[JDENET]
checkKrnlHealth=1000
```

> **Recommendation:** Enable the check kernel health setting in the ini file. Setting the network job to check every 1,000 times it sends or receives a message is a reasonable place to start, or you might set it higher if you have several transactions.

5.6 OneWorld Xe JAS server

The JAS server also provides some opportunities for adjustments, most of which are made through the jas.ini, the htmlclient.ini, and the htmloverrides.ini files. The default location for these files in the IFS is /JDEdwardsOneWorld/B7333/jas/. These files can be modified through your normal PC editor (Notepad).

5.6.1 JDBC connection pool settings in the jas.ini

JDBC is the database interface used by the OneWorld JAS code running on the WebSphere Application Server to access the database on the enterprise server. For performance reasons, it makes sense to put and use the database connections in the connection pools.

We ran with various numbers of initial, minimum, maximum, and pool growth settings. These settings affect the number of QZDASOINIT and/or QSQSRVR jobs that get used. Connection pooling allows HTML clients to use less of these jobs than fat clients and TSE clients because they can be reused.

While running with 100 users, we achieved the best performance, with a small number of minimum connections (5), a small growth rate (5), and the maximum should be equal to the number of users (100).

In a 100 user example, consider:

```
[CONNECTION POOL]
MaxConnection=100
MinConnection=5
PoolGrowth=5
InitialConnection=10
```

> **Recommendation:** If the J.D. Edwards user IDs are all mapped to a single iSeries server user profile (proxy user ID), set the initial connections to 1 for every 10 users, minimum connections to 5, pool growth to 5, and maximum connections equal to the number of users, up to a maximum of 400 per instance.

Alternatively, if JDE user IDs are mapped to separate iSeries server user profiles (proxy user IDs), each user will have its own JDBC connection pool. In this scenario, each user profile uses its own connection pool. Therefore, the best performance would be achieved by smaller connection pool settings.

Consider the various user count example:

```
[CONNECTION POOL]
MaxConnection=2
MinConnection=1
PoolGrowth=1
InitialConnection=1
```

> **Recommendation:** If the JDE user IDs are all mapped to separate iSeries server user profiles, set initial connections to 1, minimum connections to 1, pool growth to 1, and maximum connections to 2. (This method for connection pooling may be eliminated for future releases of OneWorld.)

5.6.2 JDENET connections in the jas.ini

The connections between the OneWorld JAS server and the enterprise server are JDENET connections, the same as any fat client or TSE user. The number of JDENET connections should be the same as the number of users that are being run in the JVM (maximum of 400). This is controlled via a setting in the jas.ini file. This setting did not significantly improve our performance.

Consider the 100 user example:

```
[JDENET]
maxPoolSize=100
```

> **Recommendation:** Set the same number of JDENET connections, maxPoolSize, because there are users running in the JVM (maximum of 400).

5.6.3 Caching Java serialized objects in the jas.ini

With SP 14.1_WEBP, code was introduced into the OneWorld JAS server to cache Java serialized objects. At that time, we recommended that you move the Java serialized objects to the system running WebSphere. With SP 16 and later, this is no longer necessary; however, while all of the Java serialized objects can be cached with SP16, caching needs to be turned on and set with the caching time limit.

> **Note:** Be aware that the first user in each application will experience slow performance because the Java serialized objects are being loaded into cache.

In the [CACHE] section of the jas.ini file, you need to add a line that states DDItem=*x* (where *x* is the number of milliseconds to hold the cache).

```
[CACHE]
DDItem=6000000
```

> **Recommendation:** Add the line DDItem=6000000 to the [CACHE] section of the jas.ini file if you want the cache preserved for 100 minutes. If you want to guarantee the cache will be held overnight, for example, 12 hours, the value would be 43200000.

5.6.4 Asynchronous business function processing in htmlclient.ini

OneWorld fat clients and TSE clients perform business functions asynchronously, for example, while a user is typing edit lines, previously entered edit lines are processed in the background. A change was introduced with OneWorld Xe SP 14.1_WEBP that enabled asynchronous business function processing for the HTML client. Enabling this function significantly reduces response time.

```
[General]
AsyncThreadPoolSize=50
AsyncThreadPoolSizeIncrement=5
AsynchBSFNEnabled=TRUE
```

Recommendation: Enable asynchronous business function processing by setting it to TRUE in the htmlclient.ini file.

5.6.5 Silent post with multi-line edit

Note: As of SP 17, the silent post feature is no longer enabled officially through the htmloverrides.ini file, but rather by selection within the OneWorld design tool Form Design Aid (FDA). The document "SP 17 Multiple Line Edit" in the Knowledge Garden outlines the procedure to make a form of an application enabled for multiple line edit. Search the Knowledge Garden under Services for Advanced Technology for "SP 17 Multiple Line Edit".

Fat clients and Terminal Server Edition (TSE) users use another type of background processing. As edit lines are entered into forms, each line is processed. When the user clicks OK on the form, only form processing needs to be completed. A change was introduced with OneWorld Xe SP14.1_WEBP that provides the same function for HTML clients. This function, referred to as *silent post*, is controlled by making the following two changes:

1. Turn on multi-line edit in the htmloverrides.ini file for each application that will use silent post. Since this is read during the Java Generation process of the JAS installation instructions, you must generate Java serialized objects again for it to take effect. (If this is the only change you make, your system will begin taking advantage of silent post, with the default set for processing two lines at a time.)

   Multi-line edit significantly improved our performance. The following example turns on multi-line edit in the htmloverrides.ini file for sales order entry and standard voucher entry:

```
[HTMLAppOverrides]
P4210_W4210A
P0411_W0411K

[P4210_W4210A]
multiLineEdit=True
[P0411_W0411K]
multiLineEdit=True
```

2. If you want more than two lines to be processed at a time, change the JavaScript file, form.js, which is located in the directory /JDEdwardsOneWorld/B7333/jdewww/js. The rowCriticalSize variable (default is 2) can be changed to the number of edit lines that should be processed at one time.

   While the setting can be as low as 1, it improved the end user response time when changed to 3 in our environment. You need to evaluate how the majority of your users operate. For example, if many-line forms is the norm, 3 would be better than 2. Since our forms typically had 10 lines, we set our variable to 3. Background processing occurred three times, and then when the user clicked OK, only one more line was left to edit along with the final form processing.

   After the comment line, “// asynchronous row processing”, change the rowCriticalSize variable to 3, or a number that suits your environment:

   ```
   // asynchronous row processing
   var rowCriticalSize = 3;
   ```

You must then regenerate your Java serialized objects and stop and restart your JVM for these changes to take effect.

> **Recommendation:** Turn on multi-line edit in each of your applications that would benefit from silent post and regenerate your Java serialized objects. In addition, you can optionally increase the number of rows processed at a time beyond the default of 2 if your environment warrants it, by updating the form.js file.

5.7 JDBC drivers on iSeries servers

The OneWorld JAS server performs table I/O against the OneWorld database using a JDBC driver. It also accesses the Java serialized objects through JDBC. (Actually, the OneWorld enterprise server jobs also perform database access through the QXDAPRCED interface, but that interface is automatic.)

IBM offers two JDBC drivers on iSeries servers (see “JDBC options for the iSeries” on page 8). You have to decide which JDBC driver to use for the two types of data access that occur in the JAS server.

We tested both drivers with iSeries server and OneWorld Xe SP14.1_WEBP. In the second hardware configuration (Virtual 3-Tier), we used the Mod 4 version of the Toolbox called JTOPEN. This code, or later versions, can be downloaded from the IBM Toolbox for Java Web site at:
http://www-1.ibm.com/servers/eserver/iseries/toolbox/

These two drivers are identified in the [JDBC DRIVERS] section of the jas.ini file. In the following example "I" is the Toolbox JDBC driver and "4" is the native driver:

```
[JDBC DRIVERS]
I=com.ibm.as400.access.AS400JDBCDriver
4=com.ibm.db2.jdbc.app.DB2Driver
```

For the All in One configuration example, if the JAS server and enterprise are all running on a single iSeries server, we recommend that you use the Toolbox driver for database access. The set up looks like this:

```
[DB SYSTEM SETTINGS]
Type=I
```

The Java serialized object access should be set up to access the Native JDBC Driver. The setup looks like this:

```
[JDBC URL]
DEFAULT=jdbc:db2://DENAS4|4|PD7333DNT|JDE|JDE
```

Recommendation: For the All In One configuration, use both the Toolbox and native drivers.

For V3T, if your configuration has the JAS server set up on a different machine than the enterprise server, we recommend that you use the JTOpen (Mod 4) version of the IBM Toolbox for Java:

```
DEFAULT=as400:db2://DENAS4|I|PD7333DNT|JDE|JDE
```

Recommendation: For a Virtual 3-Tier environment, use the Toolbox driver.

6

XPI on iSeries overview

World and OneWorld software provide a large suite of applications for running your business. However, most enterprises have other software within the enterprise that is required to inter-operate with World or OneWorld. Also, most enterprises have either implemented a way to, or are investigating ways to, do business electronically with other companies.

eXtended Process Integration (XPI) provides the technology and the business processes for collaborating within the four walls of the enterprise and beyond. This chapter explains the basic concepts of XPI technology and walks you through the installation process on the iSeries server.

6.1 XPI functional overview

XPI is the suite of technologies that allows OneWorld and World software to collaborate with other software products. XPI is composed of two parts:

- **XPIe (e for enterprise)**: Provides the infrastructure for internal collaboration within the enterprise walls. XPIe could be used to integrate a company's internal warehouse software with OneWorld Inventory and Financials applications.
- **XPIx (x for extended)**: Provides the infrastructure for B2B collaboration beyond the enterprise walls with supplier, trading partners, or any other entity with which an enterprise conducts business. XPIx could be used to link a company's use of OneWorld's Sales Order application with another company, its supplier, to request products, for example.

eXtended Business Process (XBP) is the term used to describe integrations created with the XPI toolset. While XPIe and XPIx provide the infrastructure for connecting applications or companies, XBPs are the executable business processes that tie the applications or enterprises together. XBPs can be internal business processes, created using XPIe or external business processes created using XPIx.

6.1.1 XPIe

XPI accomplishes collaboration by transforming information from one application into a message that is flowed to another application where it is transformed into necessary information for it:

- The heart of XPI is the flow of messages that is performed by the *broker server* or *information broker*.
- The mechanisms that act on behalf of the applications by sending and receiving these messages are the *intelligent adapters*.
- The reformatting of data so that different adapters can access the same data is done by *agents*.
- These messages flow and content is created and monitored through a set of *tools*.

This section looks at each of these components and discusses them. Similar to OneWorld, the components of the XPIe architecture are flexible and can be located on a single server or split among multiple systems. Information on deciding where to locate the various components is discussed in 6.2, "Installing XPI on the iSeries server" on page 215.

Information broker

XPI uses a component-based message broker technology to integrate the various applications (referred to as *resources*) and enable internal and external collaboration. This message broker is referred to as the *broker server* or simply the *broker*. While it is possible to have multiple brokers configured and running, we discuss a single broker configuration throughout this chapter.

An integration between two applications occurs through the broker. Figure 6-1 shows the broker as the hub of an XPI transaction. An action occurs in Application A that causes Adapter A to send a message (referred to as an *event*). The broker receives the event from the adapter. Internally, the broker uses an agent to reformat the data as necessary. Then the broker sends the message to Adapter B, which generates a corresponding action in Application B.

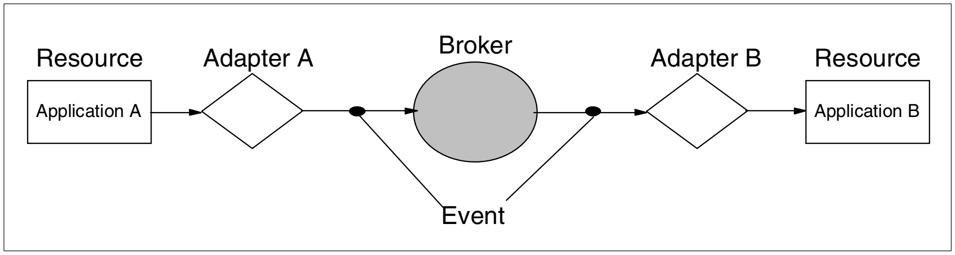


Figure 6-1 The broker routing events

In the above example, Application A may be an application within the enterprise, but not part of the OneWorld application suite, for example, Ariba. When a purchase order is created in Ariba, Adapter A sends an XPI message to the broker. The broker converts the message to a OneWorld purchase order and sends it to Adapter B. In this case, Application B is OneWorld and a purchase order is created within OneWorld. The PO number and any other necessary information is returned to Ariba following the reverse route.

Adapters

The broker is the hub of the XPI infrastructure, and the adapters are the spokes connecting the broker to OneWorld and any external applications. An adapter is configured for each application that needs to interoperate with World or OneWorld. Each adapter is capable of performing a specific action on a resource. In this example, the action is receiving information from Ariba when an order is created. An adapter can subscribe to input events or publish output events. The adapter is written in Java and runs on the XPI server (an Integrated xSeries Adapter).

As you can see in Figure 6-2, Adapter A is responsible for connecting Application A to the broker. For example, we use Ariba as Application A. As such, it is composed of two halves. One half is capable of dealing with information from Ariba, and the other half sends and receives messages with the broker. Adapter B also sends and receives messages with the broker. However, it is capable of communicating information to the OneWorld application.

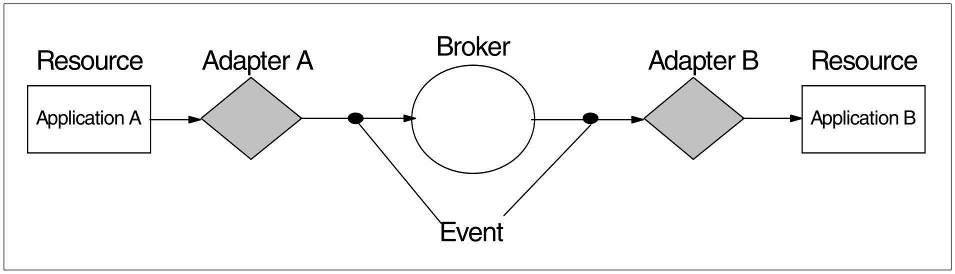


Figure 6-2 Adapters connecting events to resources

Let's take a closer look at the left-hand side of the diagram in Figure 6-3.

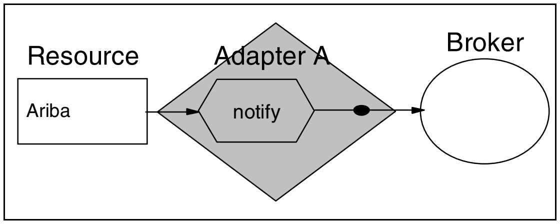


Figure 6-3 Adapter sending information to the broker

When a purchase order is created in Ariba, the adapter takes this information and sends a message to the broker. In this example, a notification message is sent. A *notification event* is a one-way event and no reply is expected. This is because after a purchase order is created in OneWorld, Ariba does not keep track of the OneWorld purchase order number, so no reply is necessary. The other event types are discussed later in this chapter.

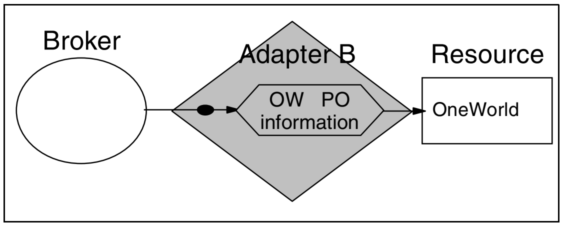


Figure 6-4 Adapter receiving information from the broker

Expanding the right-hand side of the diagram in Figure 6-4, you see that this adapter receives an event from the broker and, in turn, calls OneWorld to run the purchase order entry business functions. This adapter is capable of dealing directly with OneWorld kernel jobs. In addition to using call object kernel jobs, OneWorld uses three additional kernel job types to deal with XPI. These are discussed later in this chapter.

J.D. Edwards delivers a large number of adapters. Some of these are included with XPIe such as the OneWorld adapter, a database adapter for DB2, and several generic language adapters. Other adapters are sold separately and allow integration with outside applications such as Ariba or other ERP applications.

Agents

As mentioned earlier, the broker uses an agent to reformat messages. When the broker receives an event, the message contents are specific to the originating resource. In our example, Ariba has an adapter, but the adapter can be used to integrate with OneWorld, World, or some other application. The adapter contains no information about the application that ultimately receives the message. The broker is the messaging infrastructure and knows nothing about the contents or format of the messages.

Agents are needed to reformat incoming message content from a native event to a generic message, referred to as a *canonical event*. Agents are also used to reformat messages from a canonical to the outgoing native message format.

Figure 6-5 shows how an incoming native event from Ariba would get converted to a generic canonical purchase order event. This canonical event would then be sent through another agent to be reformatted to a native OneWorld event.

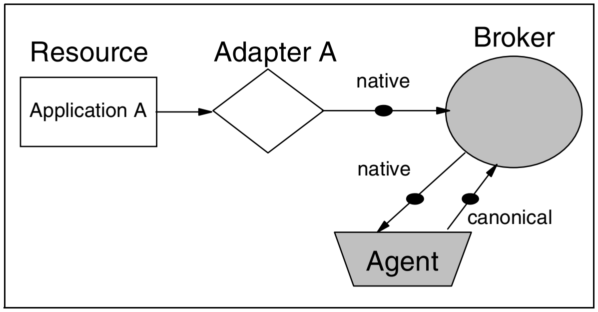


Figure 6-5 Agents reformatting messages

The need for an agent to reformat the message content may seem unnecessary at first. It may seem possible to have one or the other adapter perform the necessary reformatting and save the step of going through the generic canonical event. However, if the example is expanded, you can easily see the need for the agent. If the Ariba integration is needed for World and OneWorld, you can easily configure the broker to send a message to multiple receivers, but the adapter would need to convert to multiple message formats. The agent (or agents) is used by the broker to send native messages out to one or more recipient applications.

Agents also to keep track of information that is different between the collaborating applications. In our example, OneWorld does not keep track of the Ariba purchase order number, and Ariba does not keep track of the OneWorld purchase order number, yet the two applications need to interoperate. Agents use a *cross reference table* when transforming native events to canonical events and vice versa. Figure 6-6 shows how agents would use a cross reference table.

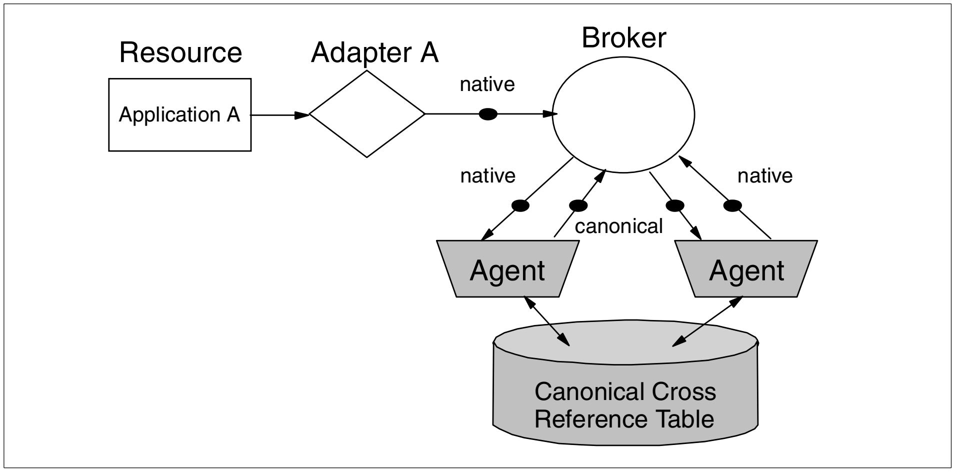


Figure 6-6 Agents using a cross reference table

Putting it all together, you see the diagram in Figure 6-7. Although Application A runs independently of and without communicating to Application B, they interoperate effectively via XPI. The adapters communicate with the applications and the broker, while the broker uses agents to transform messages between the applications.

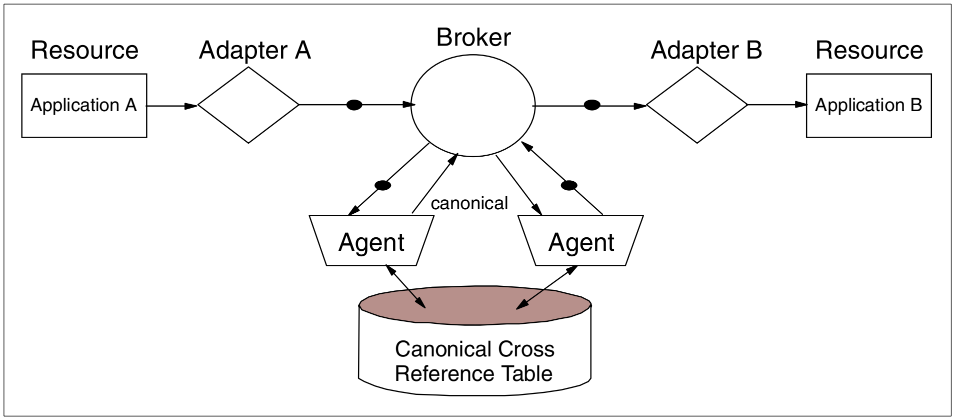


Figure 6-7 XPI data flow

XPIe tools

J.D. Edwards provides a comprehensive toolset for XPI. It covers the initial configuration tools for brokers and adapters, the implementation of tools for developing collaborations, and the management tools for tracking activity of a deployed solution.

Adapter Configuration Tool

The Adapter Configuration Tool is used to set an adapter's runtime information so that it can be used in an interoperability scenario. Figure 6-8 shows the Adapter Configuration Tool as it would look setting up the OneWorld adapter. An adapter of a particular type is attached to a specific broker. The other tabs in the adapter configuration are specific to the adapter type. For an intelligent OneWorld adapter, login information, the environment, and the supported events also need to be configured.

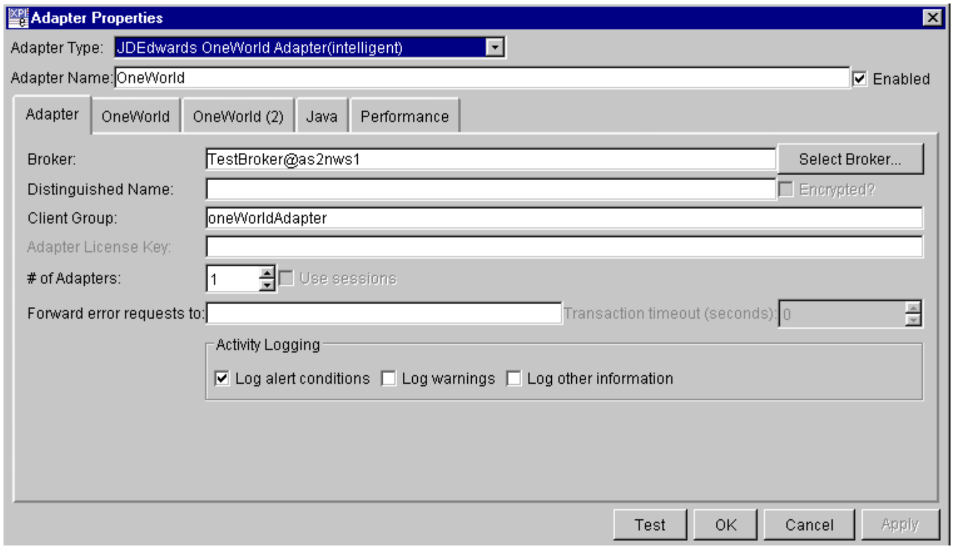


Figure 6-8 Adapter Configuration Tool

Visual Integrator

The Visual Integrator is one of the main development tools. It is used to build the steps that are necessary to implement an integration between two resources. Figure 6-9 shows the Visual Integrator as it would be used to develop an integration for the Purchase Order application. For a OneWorld integration, the steps would include the business functions that are called and any logic that is needed. Integrations are associated with the broker that runs them. The Visual Integrator also keeps track of the adapters and event types that are associated with a broker.

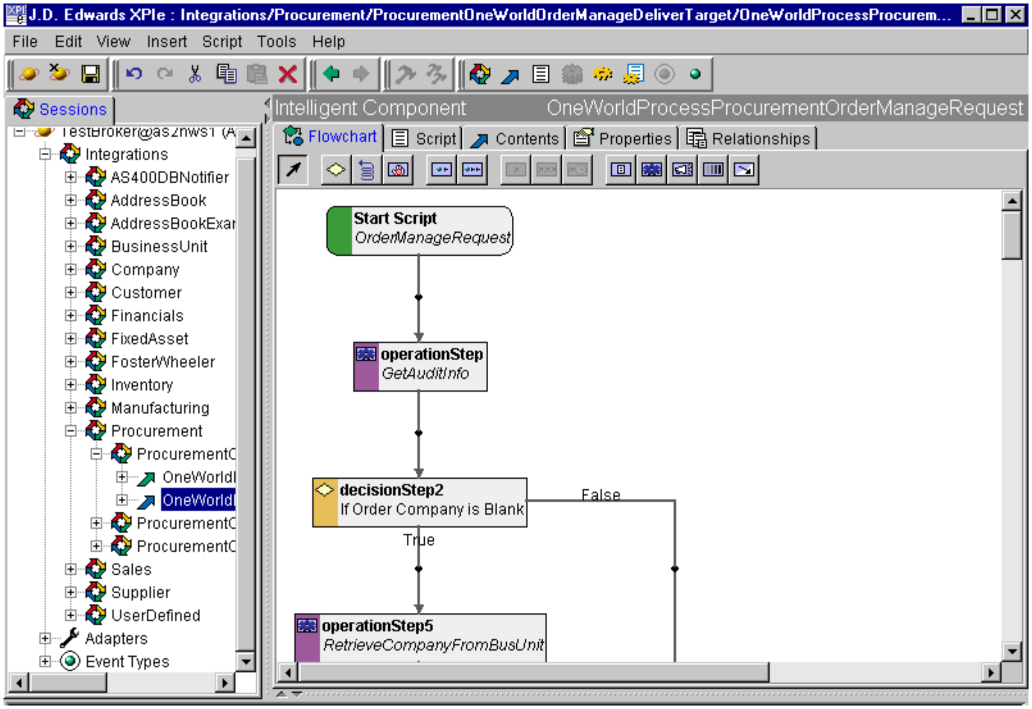


Figure 6-9 Visual Integrator

Event Tracker

The Event Tracker is used to monitor events when a completed installation is being tested or is running in production. The Event Tracker can be used to monitor both outgoing and incoming events. Figure 6-10 shows monitoring the incoming Purchase Order events as they occur.

In the example, we are looking at the detail of the fifth of ten Purchase Order delivery events. You can examine the various inputs and outputs for the event. In our example, we are looking at the supplier information contained in the purchase order header.

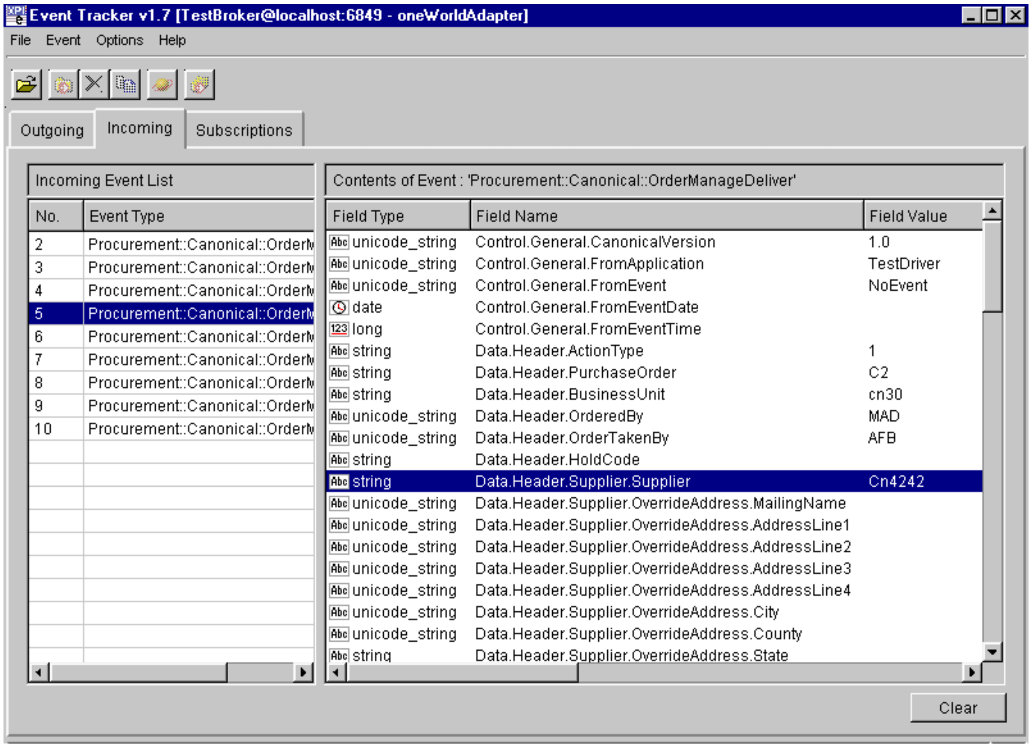


Figure 6-10 Event Tracker monitoring

The Event Tracker is first setup to subscribe to the various events that we want to track. Figure 6-11 shows how the subscriptions can be set up by simply selecting (see check marks) the events to be monitored. As the events occur, they are displayed under the incoming or outgoing tab.

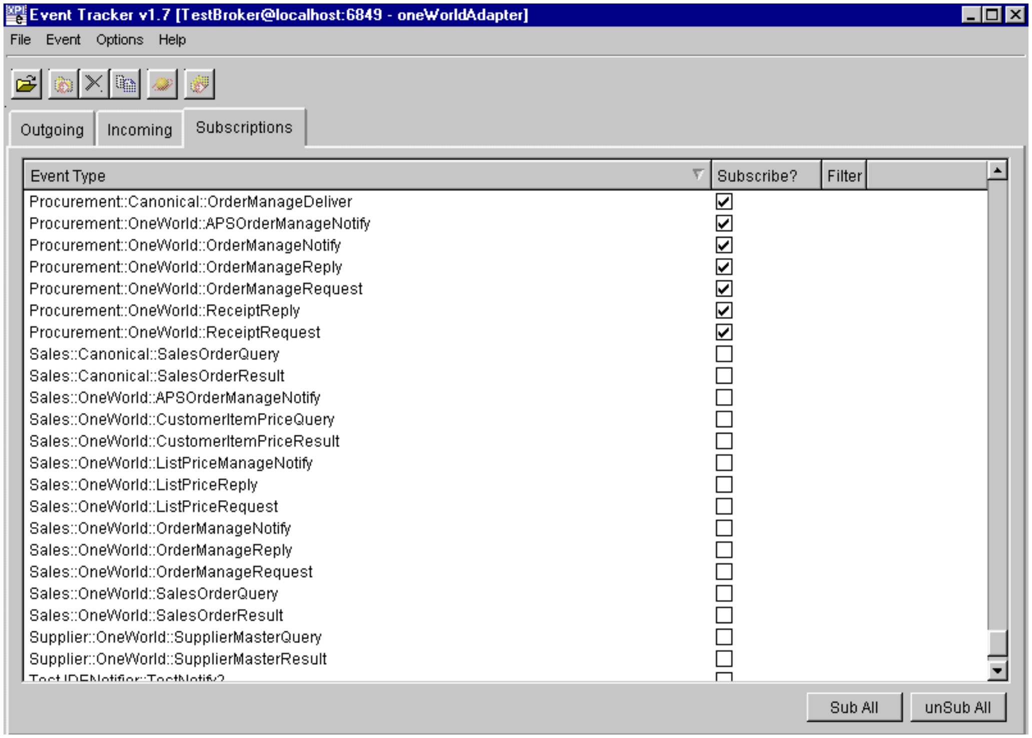


Figure 6-11 Event Tracker subscriptions

6.1.2 XBP

As we stated earlier, XBPs are fully executable business processes that tie together applications within a company or applications between companies. XPI is the middleware that provides the ability to connect applications and companies. XBPs, on the other hand, provide the contents of those connections. Just as J.D. Edwards provides a wide range of business functions that define the processes of conducting business, they now provide XBPs so those same business applications can be extended within and outside of a company's walls.

Types of XBPs (internal and external)

As we previously discussed, XPIx provides the mechanisms for the external workflow, security, and standards-based elements required for inter-company collaboration. It also provides the infrastructure for external XBPs.

In our example, we discussed how a purchase order flows into the OneWorld sales order processing. However, purchase orders vary by industry. The information in an electronics purchase order is different than what is contained in a chemicals purchase order. As a result, different standards evolve. In spite of the various standards and the differences between industries, purchase orders contain many common items, and the external XBPs target the commonalities.

While many standards evolve for inter-company transactions, internally every company maintains its own practices; there are no general standards. In this case, the “standard” is OneWorld and the transactions that flow into and out of it. As we mentioned earlier, these standard transactions are referred to as *canonical events*.

Available XBPs

J.D. Edwards has delivered seven sets of XBPs to date. The majority of the XBP sets are those that flow into either World or OneWorld. However, two XBP sets have been developed for specific third-party applications as well.

- **OneWorld XBP Set for e-Procurement**: XBPs to exchange financial, purchasing, pricing, and shipping information between OneWorld and XPI.
- **OneWorld XBP Set for CRM Sales**: XBPs to exchange customer, credit, and billing information between OneWorld and XPI.
- **OneWorld XBP Set for Storefront**: XBPs to exchange customer, item, and order information between OneWorld and XPI.
- **World XBP Set for e-Procurement**: Similar capability as the first set listed above, but for WorldSoftware rather than OneWorld.
- **World XBP Set for CRM Sales**: Similar capability as the second set listed above, but for WorldSoftware rather than OneWorld.
- **Ariba XBP Set for e-Procurement**: XBPs that connect Ariba to XPI. Once connected, Ariba can exchange information with either World or OneWorld using the first set or fourth set listed above.
- **Siebel BP Set for CRM Sales**: XBPs that connect Siebel to XPI. Once connected, Siebel can exchange information with either World or OneWorld using the second set or fifth set listed above.

6.1.3 XPIx

As the name suggests, XPIx is an extension of the XPI technology beyond the walls of a company. To extend outside of the company, you must consider many additional infrastructure items. For example, security, standards, and service availability must all be accounted. XPIx is composed of a server, tools, and a set of external XBPs. Various add-ons are also available from J.D. Edwards.

XPIx Server

The XPIx Server is the engine that runs XPIx. It is composed of a set of subsystems that each perform a necessary function to integrate XBPs with external trading partners.

The XPIx Server runs within WebSphere as a set of Java servlets. Each of the servlets is a subsystem that performs specific functions for XPIx:

- **General Subsystems**: Contains the general purpose processing functions such as the XML/XSL Parser and Generator for dealing with XML documents; managers for trading partners, users/groups, and audit trails.
- **Resource Library Manager**: Handles the storage and access of the definitions used by the subsystems. It contains document definitions, workflow definitions, etc.
- **Document Manager**: Maintains a mailbox paradigm for incoming and outgoing documents. It uses the Resource Library Manager to determine what subsystem incoming documents should flow to.
- **Workflow Engine**: Coordinates all of the internal business processing for a document. It reads information from the Resource Library Manager and invokes each of the other subsystems in the order specified in the work flow.
- **RosettaNet Engine**: Performs all of the basic services required to implement the rosettaNet protocols.
- **XPIx Transformer and Converter**: Maps between document formats. The transformer maps non-XML documents into an XML format and the Convert maps documents from one XML format to another.
- **XPIx Adapter**: Integrates XPIx with external software subsystems. The adapter works with a specific transformer to perform the necessary document conversions.
- **XPIx Security**: Determines the security techniques that will be used to convert outgoing and incoming documents. The security technique that is used depends on the sender or receiver of the document and the agreed security.

XPIx XBPs

The XPIx Server's purpose is to run XBPs that are extended beyond the back office. You can find a more complete explanation of XBPs in 6.1.2, "XBP" on page 212. The XBPs that are currently available for company to company work are based on RosettaNet, a standard for the electronics industry. The initial set of XBPs allow companies to order products and handle the details of the order. The list of XBPs and the RosettaNet identifier include:

- Request Purchase Order (PIP 3A4)
- Distribute Order Status (PIP 3A6)
- Query Order Status (PIP 3A5)
- Notify of Advance Shipment (PIP 3B2)
- Notify of Invoice (PIP 3C3)
- Notify of Remittance Advice (PIP 3C6)
- Query Price and Availability (PIP 3A2)

This list is being expanded to include other industry standards.

6.2 Installing XPI on the iSeries server

Installing XPI on the iSeries server requires preparation, installing and configuring the XPI middleware, and making changes to the OneWorld configuration to run with it. XPI comes with a set of installation documentation.

When installing on the iSeries Integrated xSeries Adapter, follow the Windows NT or Windows 2000 installation documentation. The following sections are intended to provide some information and suggestions to supplement the J.D. Edwards installation and configuration guides.

6.2.1 Minimum technical requirements

You can find the minimum technical requirements for running XPI in the Knowledge Garden. In the Products section, search for “XPI Minimum Technical Requirements” and select the **MTR** document. The requirements for running XPI on the Integrated xSeries Adapter are listed with the Windows 2000 or Windows NT requirements. Select the document according to how you install your iSeries adapter.

6.2.2 Installing the XPI Foundation code

Because XPI does not run on OS/400, installing XPI on iSeries requires the use of an Integrated xSeries Server.

Installing the Integrated xSeries Server

You can install the Integrated xSeries Server card with either Windows NT 4.0 or Windows 2000. We installed our server card with Windows NT 4.0, setting aside a 1 GB D: drive for system code and an 8 GB C: drive for XPI and its required software. After identifying the resource name for the 2890 adapter, follow the normal installation procedure. Here is the command that we used to install Windows NT 4.0 on our Integrated xSeries Server adapter:

```
INSWNTSVR NWSD(AS2NWS1) INSTYPE(*FULL) RSRCNAME(LIN05) DMNROLE(*SERVER)
WNTVER(*NT40) OPTION(*INSTALL) EVTLOG(*NONE) SVRSTGSIZE(1000 8000)
CVTNTFS(*YES) FULNAM('XPI Server') ORG(IBM)
```

If you installed Windows NT 4.0 on the Integrated xSeries Server, you need to change the display setting to 256 Color support. For more information, go to the iSeries InfoCenter at:
http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html

Search for the article "Installing the Savage 4 Video Driver for Windows NT 4.0 on the 2890 Integrated xSeries Server" and follow the instructions. This allows you to change the display settings to 256 color mode. This is required to complete the installation for XPIe Foundation.

If you have installed Windows 2000, you do not need to change the video driver.

Installing XPIe Foundation

After Integrated xSeries Server is installed and the display has been set to 256 colors, you can follow the installation instructions in the *XPIe Foundation Installation and Configuration Guide*. The instructions in the guide are complete, and we found that no additions or corrections are needed to complete the installation.

6.2.3 Configuring XPIe

As discussed earlier, XPI contains numerous adapters, including OneWorld, World Software, Ariba, CRM, etc., each of which can be configured for use. For example, we discuss the configuration of the OneWorld adapter.

Each adapter has a corresponding installation and configuration guide. The OneWorld adapter manual is the *OneWorld XPIe Adapter Installation and Configuration Guide*. We found the installation guide to be complete; you can follow it for the installation of the OneWorld adapter. We point out a few of the important settings of the configuration here.

OneWorld adapter properties

The OneWorld adapter has a set of base properties. Figure 6-12 shows the adapter configuration tool open to the OneWorld adapter base properties. The broker that is being used is setup during the XPIe Base installation process and it is referenced by the adapter. This is the broker that will pass the events related to the OneWorld adapter. In our example, we created a broker named *TestBroker* and it is located in the Integrated xSeries Server *as2nws1*.

A client group also needs to be created. We left ours with the default name of *oneWorldAdapter*.

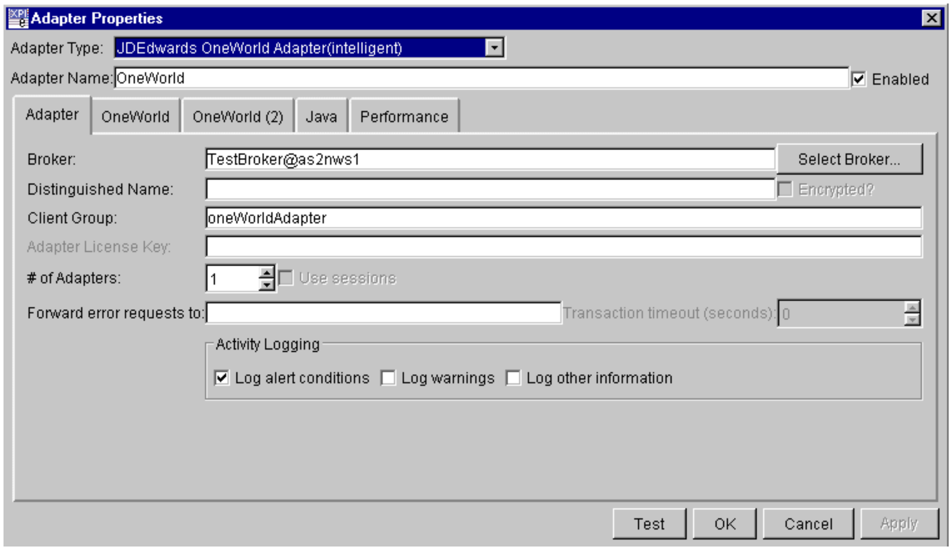


Figure 6-12 OneWorld adapter base properties

When you click the OneWorld tab, you see the OneWorld specific properties as shown in Figure 6-13. All connections from XPI to the OneWorld enterprise server use a single user ID. You need to enter the user ID and password that will be used for these connections. You also need to enter the environment to which you want XPI to connect.

In our example we used the user ID `JDE`, our proxy user ID. We used the environment `PD7333`, which is the production environment. You could initially use the test or development environment and then change the adapter configuration when you are ready to go into production.

This is the tab where you enter any real-time events that you need XPI to monitor. These events are defined by the XBP set that you are using. For example, `jdeab` is the OneWorld address book real-time event.

Event style sheets for XML conversion are also entered on this tab. We use the default outbound event style sheet, oubound.xsl, that J.D. Edwards provides.

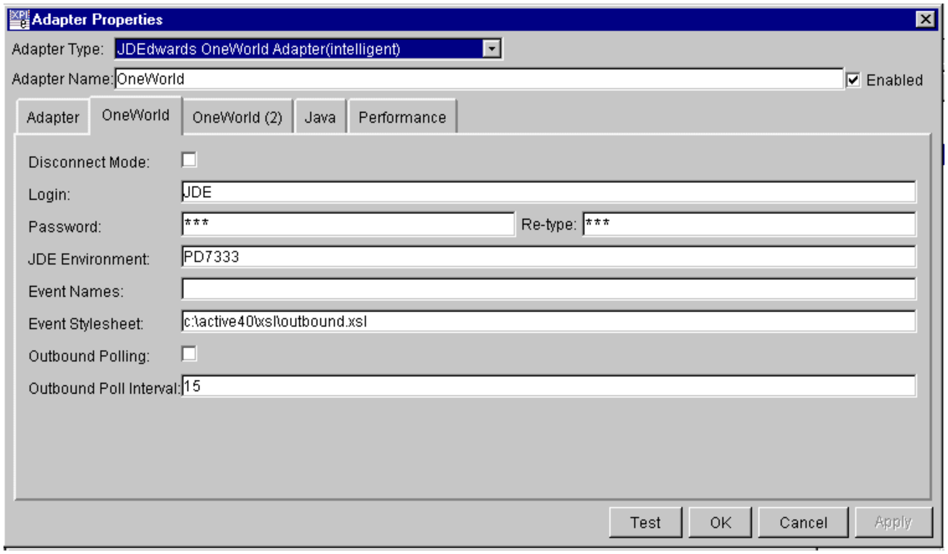


Figure 6-13 OneWorld adapter settings for connecting to the enterprise server

The OneWorld(2) tab contains the OneWorld logging settings that are used for the adapter.

The Java tab as shown in Figure 6-14 contains all of the necessary information for the adapter to run. Because the XPI adapters are written in Java, this page contains the information about the Java code (.jar files) that are needed for the OneWorld Adapter.

The Java code includes the pre-requisites that are needed by the OneWorld adapter, plus any specific Java code that is needed for your specific integration. In our example, we see two files, xalan.jar and xerces.jar, that will be used for XML document access. We also see a specific xpi_cdistInterop.jar file, which contains the necessary integrations for our purchase order example.

This tab also contains the information on the location of the INI file for the OneWorld adapter, the jdeinterop.ini file.

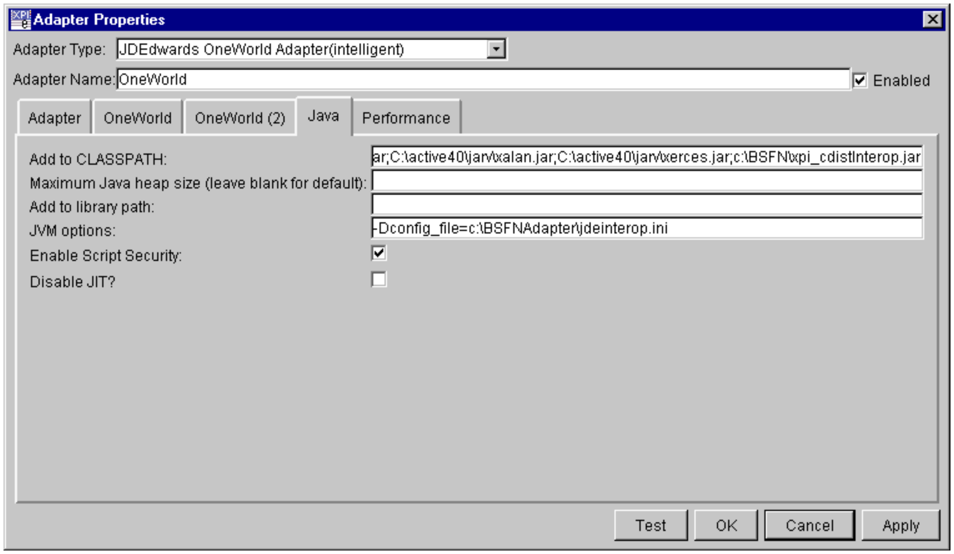


Figure 6-14 OneWorld adapter Java settings

The jdeinterop.ini file contains the information on where the OneWorld adapter will find the J.D. Edwards OneWorld enterprise server. This information is contained in the [INTEROP] stanza of the INI file. Example 6-1 shows our settings. This OneWorld adapter communicates with the enterprise server on DENAS2 at port 6090.

Example 6-1 The jdeintrerop.ini file

```
[INTEROP]
enterpriseServer=DENAS2
port=6090
```

6.2.4 Validating the configuration

After you complete the configuration, validate that it is set up correctly. Similar to running PORTTEST, the XPI adapters also contain a self-test. Each time you start the adapter, it performs a self-test. As you can see in Figure 6-15, the adapter test function validates that it is working correctly and can communicate with the enterprise server.

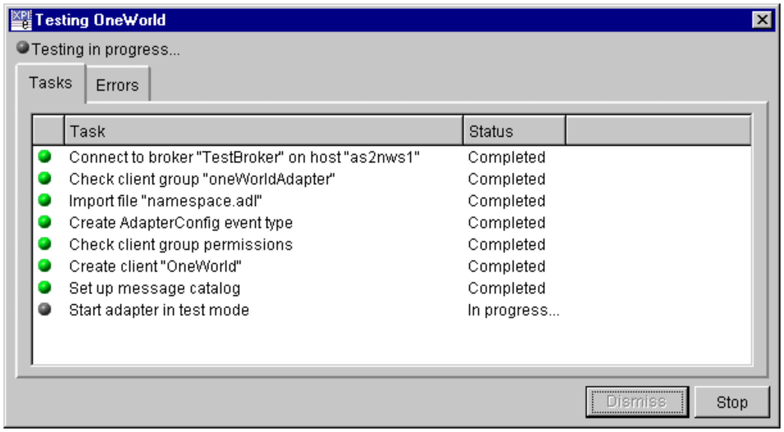


Figure 6-15 OneWorld adapter test

If there are any problems, you can choose to run the adapter in debug mode. To start the adapter in test mode, select the **Adapter** menu and select **Run in Debug Mode**. Figure 6-16 shows the messages that appear when running the adapter in debug mode.

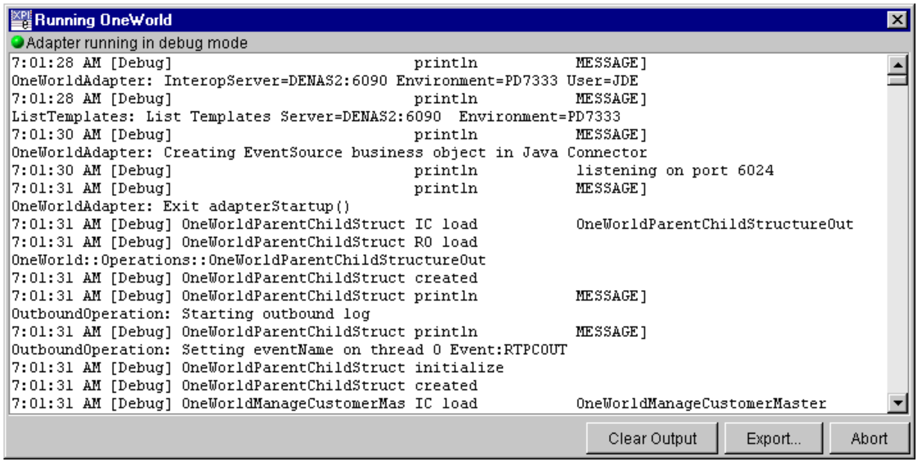


Figure 6-16 OneWorld adapter running in debug mode

6.2.5 Setting up OneWorld for XPI

As shown in Figure 6-4 on page 206, the XPI adapter communicates with OneWorld. To facilitate this communication, additional kernel jobs have been created to perform the various tasks required by XPI.

Interoperability Events Observer (IEO) kernel

The IEO kernel job is responsible for formatting the XML documents that are generated by OneWorld. The IEO kernel is called by both Call Object kernel jobs and UBE jobs when a message needs to be sent to XPI. The messages that are sent vary in size according to the document that is being processed. For example, a multi-line sales order that needs to be sent to XPI generates a large XML document.

Event Notification (EVN) kernel

The EVN kernel is responsible for managing subscriptions and forwarding the XML documents to subscribers.

XMLList kernel

The XMLList kernel job processes list template requests from the XPIe OneWorld adapter. This kernel is used with XBPs that use list templates.

Changes to the INI file

You must make several changes to your server INI file to use the new kernel jobs. The INI file is contained in your system directory (for example, B7333SYS). Edit the JDE member of the INI file, using the System Edit Utility (SEU), for example.

Because of the additional kernel jobs that we previously described, you must increase the number of kernel types supported. In the JDENET stanza of your configuration, increase the kernel range to 20, if yours is set lower. Example 6-2 shows an updated JDENET stanza.

Example 6-2 JDENET stanza changes

```
[JDENET]
maxKernelRanges=20
```

After you increase the number of kernel types, add a new stanza for each of the kernel types. Example 6-3 shows all of the new kernel types. Configure only the kernel types that you need. This example shows the maximum number of processes for each kernel type as 1. Set your maximums according to the workload that you expect.

Example 6-3 New kernel type stanzas

```
[JDENET_KERNEL_DEF16]
krnlName=XMLList Kernel
dispatchDLLName=XMLLIST
```

```
dispatchDLLFunction=XMLListDispatch
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1

[JDENET_KERNEL_DEF19]
KRNLNAME=EVN KERNEL
DISPATCHDLLNAME=JDEIE
DISPATCHDLLFUNCTION=JDEK_DISPATCHITMESSAGE
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1

[JDENET_KERNEL_DEF20]
KRNLNAME=IEO KERNEL
DISPATCHDLLNAME=JDEIEO
DISPATCHDLLFUNCTION=JDEK_DISPATCHIEOMESSAGE
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1
```

Finally, an interoperability stanza, as well as separate stanzas for each of the registered events, need to be added to your server INI file. Example 6-4 shows the setup for configuring the interoperability of purchase order events.

Example 6-4 Interoperability stanzas

```
[INTEROPERABILITY]
FilteredEvents=*ALL
RegisteredEvents=RTPOOUT,RTPOHDR,RTPODTL,RTPOAPS
[RTPOOUT]
DS1=D4302150B
DS2=D4302150C
DS3=D34A1050E
[RTPOHDR]
DS1=D4302150B
[RTPODTL]
DS1=D4302150C
[RTPOAPS]
DS1=D34A1050E
```

Performance considerations

You can find performance considerations for running XPI in the J.D. Edwards Knowledge Garden. Here, you can also find a best practices guide by searching for “XPI Tuning” alerts on the XPI Performance Alerts page.

7

SAN technology

This chapter discusses Storage Area Network (SAN) technology and how it relates to iSeries servers that have implemented J.D. Edwards software solutions. SAN technology implementations are becoming more common in organizations. The promised SAN benefits of simplified storage management, disk virtualization, backup, disaster recovery, and performance, to name a few, have been available on the iSeries server, often driven by the storage consolidation of OS/400, Windows NT, and UNIX platforms. Recently, IBM, in particular the iSeries servers, have extended system support of SAN technology.

This chapter offers a brief overview of the SAN technology on iSeries machines, discusses its implications with J.D. Edwards software, and provides configuration recommendations for incorporating SAN technology in a OneWorld environment.

7.1 Overview of iSeries storage

From the perspective of a pure iSeries server shop, there are few reasons to implement SAN technology. The capabilities of the iSeries server internal drives and the single-level storage architecture currently provide many of the benefits of SAN technology including:

- Storage consolidation
- Storage management
- Disk reliability
- Disk compression
- Virtualization

Disk capacity on the iSeries server either meets or exceeds comparable capacities from SAN vendors. Price performance on the iSeries server has been more cost effective compared to the SAN technology. The value of iSeries server and SAN technology interoperability is to provide customers the option of using SAN technology based on their *business* needs.

Implementations of SAN technology on an iSeries server as the primary platform have in many cases been for the purpose of *storage consolidation*. Storage consolidation is the ability to connect many systems, not all of the same platform, to a centralized disk storage. The connections may be direct or may include a hub or switch. The ability to connect UNIX, Windows NT, and iSeries servers to the same storage has been the major factor in all of the J.D. Edwards SAN implementations.

The iSeries server treats the SAN environment as a set of physical disk arms or spindles that it has control over. The SAN internally provides the iSeries server with a set of "virtual disks", called *logical unit number* (LUNs), which for all practical purposes, is a single disk to the iSeries server. The SAN manages the assignment of this LUN and its configuration apart from the iSeries server; it is transparent to the iSeries server. The LUN itself, configured by the SAN, may contain one or more disks and may be configured in a variety of RAID configurations. Optimization algorithms on the iSeries server treat this LUN as a single disk arm, although it may be spread across multiple disk devices.

For more information on SAN technology on the iSeries, see:
`http://www.ibm.com/san`

7.2 OneWorld implementation

This section discusses the ERP OneWorld software implementation on the iSeries server with SAN technology.

There are no known issues with the implementation of the OneWorld software application. There are however a few tricks that can be implemented to minimize the potential for problems related to interoperability between the iSeries server and the SAN. The SAN vendor should be contacted for information concerning recent problems, updates, and bug fixes with the iSeries server. SAN vendors maintain a list of supported Host Bus Adapter (HBA) cards and problems arising concerning connectivity should be addressed to the SAN vendor.

The following sections offers a brief discussion of the major concerns with SAN implementations on the iSeries server.

Managing disk storage on the iSeries and the SAN

The introduction of an external data storage requires additional system resources to manage this storage apart from the iSeries server. This management is automatically performed by the iSeries server operating system only if the storage is internal. However, since the external storage is not completely under the control of the iSeries server, additional expertise and management are required to maintain this resource.

For example, the iSeries server does automatic performance tuning of a disk that has been made available from a SAN. In an attempt to increase performance, the SAN administrator makes changes to its configuration such as increasing the size of the LUN. The iSeries server automatically responds to these changes and, in turn, makes system performance changes of its own.

This adjusting continues until a steady state of the system is achieved. Since the result may not be the one as desired originally, considerable time and resources may be encountered in managing disk storage on the iSeries server. To avoid these problems, keep to a minimum the changes to the disk configurations. You should also monitor changes and carefully manage their effects.

Performance issues between the iSeries and SAN

Performance issues can arise between the SAN and the iSeries server such as optimization conflicts between the disk cache algorithms on the iSeries server and SAN. Also, application program usage on the iSeries server can cause automatic changes in the disk cache allocation that is driven by the iSeries server disk cache algorithm. Alone, this is a very desirable effect because it enhances the overall server performance.

However, in the case of the SAN, another variable is introduced, namely the *SAN disk cache*. The SAN disk cache also attempts to adjust its allocation according to a different set of rules. The net effect is disk cache contention causing a less than optimal performance for both the SAN and iSeries server. Each system continues to automatically adjust its cache until a steady state is achieved. Performance issues with iSeries server expert cache are discussed later in this chapter.

Interoperability concerns

Interoperability concerns between the SAN and iSeries server include:

- Host Bus Adapter (HBA) cards

  HBA cards and their corresponding drivers are often overlooked as a possible candidate for problems. Consult an HBA card vendor Web site for any known issues with the iSeries server device drivers or connections to SAN devices. Most of the issues are with interoperability, and device driver patches for the iSeries server are readily available. SAN vendors also have an affinity to specific HBA card vendors.

  We recommend that you use a single HBA card vendor in all systems in a given SAN architecture.

- SAN switches and hubs

  SAN switches and hubs and the software that controls them not only creates another level of management but introduces another point of failure and area of complexity for the OneWorld software implementation. The general recommendation is to limit or entirely eliminate the number of switches and hubs if possible.

- Other iSeries server and non-iSeries server interaction

  Depending on the SAN configuration, there may be OneWorld performance issues that are introduced with the sharing of a centralized disk storage by iSeries server and other non-iSeries servers. iSeries servers use a different blocking size for disks than other systems, and therefore, iSeries servers' SAN LUN disks must be dedicated as part of the initial configuration of the SAN.

These concerns and a few tricks are discussed in more detail later in this chapter.

7.3 iSeries and SAN technology considerations

There are many factors to take into consideration when implementing SAN technology on an iSeries server. The purpose of this section is to introduce some of the most common concerns.

7.3.1 Internal versus external storage

The iSeries server is designed and optimized on a distributed I/O processor (IOP) topology. This topology load balances the disk I/O and system processes equally in a parallel fashion. Internal and external storage is treated equally on disk I/O channels. Therefore, you can expect performance on the internal storage, by the nature of the shorter data path, to better scale and have higher throughput than the external SAN storage. The SAN data path connection, given the latency that is introduced into the data path by the distance and speed of the connection, creates overhead that contributes to a decreased performance.

Performance on an iSeries server and SAN also depends on the CPU, main memory, IOP processor, and SAN configuration. You must carefully examine sources of response times and throughput to determine the primary bottlenecks. Increasing the speed and capability of one resource may bring only a short term or small benefit because another resource limitation is encountered. Use the iSeries server capacity planning tools (see Chapter 2, “Pre-installation planning, sizing, capacity planning” on page 15) to determine the current resource limitations at different conditions of use to decide whether there is a true issue of SAN performance.

At low throughput, both internal and external disk subsystems perform at about the same level. The performance characterizations of iSeries server internal and external SAN disk storage can change dramatically at higher levels. Internal disk, in general, performs better, but this depends on the configuration, setup, and application environment variables that contribute to overall performance.

7.3.2 iSeries server expert cache

Expert cache is a feature that buffers data from the disk storage into memory in anticipation of its usage. This feature is key for increasing performance on applications that are read intensive. The expert cache algorithm is based both on the logical and physical characteristics of disk utilization. The expert cache mode can be selected for each memory pool using the CHGSHRPOOL command with the paging options parameter of *FIXED, *CALC, or USRDFN as follows:

- When expert cache is OFF (*FIXED), there is no buffering of data.
- When expert cache is ON (*CALC), buffering of data occurs with the default strategy.

- When expert cache is ON (USRDFN), buffering of data occurs with the user defined strategy.

Expert cache improves the performance of both the internal and external SAN storage because it can perform much of the costly operations asynchronously to the application.

On at least one OneWorld implementation with the IBM Enterprise Storage Server (ESS), both system throughput and response time performance increases were observed when expert cache was turned on. This indicates that the read/write caching algorithm of the IBM ESS SAN, in general, works well with expert cache on the iSeries server to improve performance.

7.3.3 SAN storage cache

The read/write caching algorithm of an external SAN cache samples the data and adjusts the cache based on the current local sampling information without the benefit of viewing both the logical and physical resources.

SAN storage cache must be adequate enough to provide an extra read/write cache to mask resource conflicts that can occur as the iSeries server storage manager arm utilization parameters, adjusted with the SAN arm utilization parameters. Limited SAN storage cache can have unexpected results depending on the application and usage.

7.3.4 SAN configuration: Dedicated versus shared disk

Shared disk configuration refers to the layout of the storage on a SAN. In SAN implementations with iSeries servers, SAN disks are typically not shared with non-iSeries servers. The scope of this discussion revolves around the sharing of a set of SAN disk assignments between one or more iSeries servers connecting to the same SAN.

A dedicated SAN configuration reserves an entire set of disks (called a *disk array*, *disk group*, or *rank*) to be used exclusively by a single iSeries server. In this configuration, the SAN partitions its internal disk, and in some cases its cache, between a single iSeries server and other systems. The advantage of this configuration is that there is no contention for these resources with other systems. The single iSeries server operating system does not struggle to optimize the performance of the LUNs that the SAN presents to the system as “virtual drives”.

For the shared environment, there are two main configurations:

- In a two-system iSeries server configuration, two separate disk arrays are configured. Each array is assigned to a specific iSeries server.

- A single disk array is created to serve both iSeries servers. The LUNs created from this single disk array are split between these two systems.

The performance on a dedicated subsystem, in general, outperforms a shared configuration. The reason is that both the iSeries server and SAN resources depend on resources such as cache, disk arms, processors, memory, and buses. When more than one request for these resources is needed, as in the case of a shared configuration, the system must split those resources among the requesting agents. In the case of a dedicated system configuration, there is a lower conflicting overhead and, therefore, an anticipated increase in performance.

Disk operations on a SAN do not provide a mechanism for request prioritization based on application or type of request. Therefore, a system with a large disk storage volume using a SAN can be expected to perform at a lesser level than that of internal storage.

7.3.5 Fibre versus SCSI connection

There are two different types of connections, fibre and SCSI. Both types of connections can be installed in either a system unit or expansion tower. SCSI connections are only available for earlier AS/400 models and attach via an SPD expansion slot.

Fibre connection is available in newer iSeries servers, such as 2xx and 8xx models. Fibre connection to the iSeries is through the PCI I/O tower via a high-speed fibre cable providing 100 MB/sec throughput in a single connection.

The differentiator of having fibre or SCSI channel support on the iSeries server is the connectivity and distance advantages. SCSI distance limitation with ultra-wide SCSI-2 technology is 25 meters. Fibre distance limitations depend on the capability of the customer to implement short- and long-wave technology. Short-wave technology allows connections to be up to 175 and 500 meters depending on the size of the fibre cable. Long-wave technology can extend this distance up to 10 kilometers with long-wave to short-wave hub converters.

7.3.6 Multi-pathing from iSeries to SAN

You must consider the number of connections to a single iSeries server. Load balancing the performance between multiple connections to the SAN is a common practice. This technique increases the effective bandwidth of the data exchange between the disk storage on the SAN and iSeries server.

7.3.7 Connectivity of iSeries to SAN

When configuring LUN sizes, you must be mindful to anticipate growth and the limit on the number of SCSI targets (if you have a server that supports SCSI attachment to SAN). AS/400e server SCSI target assignments are limited to 16. A LUN size that is too small does not allow disk capacity scaling, and LUN sizes that are too large may waste space. Dual connections on the iSeries server decrease the number of available SCSI targets by half. Fibre connectivity does not have this limitation.

On new SAN implementations with iSeries servers, we recommend that you make a fibre connection directly to the SAN. Since the iSeries server treats internal and external disks the same, a direct connection to the SAN from the iSeries would eliminate any initial configuration issues dealing with the complexity of switch and hub architecture and the software needed to manage them. Any additional hops, like those encountered through a hub or switch, increase the overall latency and application response times.

You should also note the factors of latency of each of these devices. Any time data is transferred over a switch or hub, a small (in the range of a few nanoseconds) latency is introduced into the transmission of the data. This latency is overhead that the device in the network introduces in the architecture. Usually it cannot be changed because it is inherent to the physical hardware of the device. Therefore, you must use care to limit the introduction of base performance inhibitors such as SAN switches, hubs, and extended distance connections.

7.4 Reference materials

The following redbooks contain more information on SAN and storage:

- *AS/400 Disk Storage Topic and Tools*, SG24-5693, uncovers the basics of iSeries disk storage. This redbook is designed to help AS/400 and iSeries system administrators and operators gain a broad understanding of AS/400 and iSeries disk storage, architecture, and management.
- *Storage Consolidation in SAN Environments*, SG24-5987, walks you through the process of planning and implementing storage consolidation in a storage area network. The redbook presents several real storage consolidation scenarios for improving system management, scalability, flexibility, and availability.
- *Implementing an Open IBM SAN*, SG24-6116, looks at the entire IBM SAN product portfolio and explains each product's place and function in a storage area network.

- *The IBM SAN Survival Guide*, SG24-6143, describes all IBM Storage Area Network (SAN) products and explains how to design and plan a SAN.
- *Introduction to SAN Distance Solutions*, SG24-6408, provides more information on how to best configure distance solutions in a storage area network.
- *Introduction to Storage Area Network (SAN)*, SG24-5470, offers simplified storage management, scalability, flexibility, availability, and improved data access, movement, and backup. This redbook is written for those who market, plan for, or implement Storage Area Networks, such as IBM system engineers, IBM Business Partners, system administrators, system programmers, storage administrators, and other technical support and operations managers and staff.

7.5 Conclusion

SAN technology brings great benefit in an e-commerce application such as J.D. Edwards OneWorld. The distributed nature of the different component systems that make up an ERP implementation can be extensive. SAN technology can be used to simplify the data storage and maintenance of at least some of these components. Although the iSeries server already has many of the SAN benefits, other systems such as UNIX and Windows NT do not. It is this consolidation that brings the largest benefit to using SAN in an iSeries server environment.

8

Clustering

Globalization, competition, and e-commerce are driving the business need for enterprise-wide applications, such as OneWorld, to be continuously available to its end-user community. This translates into eliminating expected downtime for production environment backups and taking measures to prevent outages due to system failure or catastrophic incidents.

Although the iSeries server has one of the best records in the industry for uptime, often this need goes beyond the capabilities of a single system. When an application is unavailable, a business may suffer losses in productivity and revenue. The primary benefit of implementing a clustering solution is the minimization of downtime costs associated with unplanned outages such as system failures and planned downtime for maintenance and backups.

Clustering multiple iSeries servers or partitions provides a way to increase an application's availability, enabling an enterprise to retain its customer base, increase its market share, and improve quality of service. In September 2000, OneWorld became one of the first client server applications to earn usage of the ClusterProven trademark by meeting specifications established by IBM.

Clustering for OneWorld Xe on the iSeries and AS/400e servers is supported on OS/400 V4R4 and higher and OneWorld Version B73.3.3 or higher, along with the middleware offered by one of the cluster middleware partners.

This chapter outlines the deployment of an iSeries high availability solution for OneWorld Xe using clustering products offered by three independent IBM cluster middleware business partners – DataMirror, Lakeview Technology, and Vision Solutions.

8.1 Clustering architecture

This section provides a high level overview of the iSeries clustering design. Then, it compares and contrasts iSeries clustering with other platform implementations and with its predecessor – the high availability products that are still offered by the iSeries cluster middleware vendors.

8.1.1 iSeries clustering design

The benefits of deploying an iSeries-based cluster for OneWorld Xe applications can be found in its design. Clustering services are provided by a collaboration between OS/400 APIs, cluster middleware, and application environmental definitions from the independent software vendor (in this case J.D. Edwards). The middleware provides data resiliency and cluster management functions. The APIs provide OS/400 cluster resource services, and J.D. Edwards provides the information for application resiliency. Clustering can be deployed over standard communication protocols such as TCP/IP, ATM, SNA, or OptiConnect.

The iSeries clustering solution is architected as a leaderless or peer cluster of two to 128 partitioned or independent iSeries servers. An iSeries cluster implementation groups the participating servers (nodes) into one or more cluster resources called the *recovery domain*. A recovery domain can be a subset of the nodes in a cluster, and each cluster node may participate in multiple recovery domains.

Resources that are grouped together for the purposes of business resumption are known as *Cluster Resources Groups (CRGs)*. The CRG defines the recovery/accessibility characteristics or behavior for that group of resources. One characteristic involves the “near real time” replication of the dynamic components of an application’s environment on the primary iSeries node to one or more of the secondary iSeries nodes. Which objects to replicate are determined by an application resiliency component called an *object specifier file*. The “health” of the primary node is monitored via heartbeat services on the secondary nodes. An automated or manual switchover process involves a clustering exit program, and an IP takeover address is used to redirect end-user demand to the secondary node designated to assume the role of a primary node.

8.1.2 iSeries clustering and other platforms

Typical UNIX and Windows/Intel implementations of clustering have their roots in the hardware components of the computer or storage manufacturers. This is a local solution for recovery from the hardware failure of a specific server and load balancing between application servers. SCSI cabling is used to cross-connect partitioned storage that contains the full disk image of the operating environment. This includes swap and temporary files, which makes it prohibitively expensive from a communications standpoint to duplicate and maintain on a near real-time basis larger application environments across a wide area network. Often only the application executables are duplicated across single-purpose servers sharing the same database on a single shared-disk subsystem. This local redundancy approach is more of a high availability solution, not a disaster recovery one.

iSeries clustering's "shared nothing" implementation is a software-oriented solution. The shared disk subsystems that can represent a single point of failure in UNIX and Intel-based clustering solutions are avoided since complete systems represent cluster nodes. The design has its basis is in remote data replication for business operations, disaster recovery, and application distribution. No special cabling is required other than a LAN or WAN connection, so there are no distance limitations that are sometimes encountered with other platforms. Application information is selected for duplication at the object level. Therefore, only committed permanent changes to essential business information are replicated via standard communication protocols (SNA/TCP IP) to the backup nodes. This keeps communications overhead at a minimum.

8.1.3 iSeries clustering and high availability

High availability on the iSeries was the predecessor to clustering. Both solutions are offered by the three iSeries cluster middleware vendors – Vision Solutions, DataMirror, and Lakeview Technology. Both offerings are similar in purpose and function. The term "high availability" is used to describe one of the advantages of clustering. This section compares and contrasts the qualities of clustering and high availability.

Both solutions are capable of providing continuous availability of mission critical applications to the end-user community. Both solutions perform near real-time object and data replication using OS/400 journaling, have latency and heartbeat monitoring facilities, and provide automated or manual switchover functionality.

The high availability (or HA) solutions were developed entirely by each of the three cluster middleware vendors. They generally work above the operating system level to provide continuous availability and do not take advantage of OS/400 clustering services. The application data and objects to be replicated are identified through a discovery process in the project planning phase and defined to the HA software through a manual interface. A customized program is developed by the implementation team to perform the switchover.

iSeries clustering solutions are generally developed by the application software vendor, who creates clustering object components according to IBM specifications. These components contain information on object and data replication. They also contain startup and shutdown procedures that are used by cluster middleware and IBM APIs to automatically create and activate the clustering environment. Many of the clustering functions work within and below the operating system level to provide a greater level of application and data resiliency than the high availability offerings.

8.1.4 Other iSeries availability solutions

Other redundant availability solutions exist for the iSeries platform and its applications but are neither as resilient nor as complete as clustering. Some of these are switched disk, storage area networks, data replication, and remote journaling.

Switched disks

Switched disks were made available on the iSeries platform with the 8xx models at V5R1. With the current implementation, two iSeries servers or partitions can connect via High Speed Link (HSL) to switch disk units with business data on them from one system to the other. Currently it is limited to disks assigned to an independent ASP (IASP) that contains IFS objects only. From a recovery standpoint, this solution only protects against non-disk subsystem hardware failures and is an incomplete solution for OneWorld Xe.

Storage Area Network (SAN)

Storage Area Networks have been used as an alternative or supplement to internal disks by iSeries customers for quite some time, but they are relatively new in the high availability arena. The configuration involves a redundant iSeries server attached to redundant SANs with the recovery iSeries server turned off, but its SAN turned on. The production iSeries SAN keeps the recovery SAN current during business operations. When a failover occurs, the recovery iSeries system is IPLed and “becomes” the production machine.

One problem with this approach is that the iSeries servers that are involved have to be exact duplicates of each other, raising the cost of implementation if system resource requirements are lower in a recovery environment. There may also be configuration issues when the system is IPLed because OS/400 stores its system and hardware configuration on disk. In addition the failover time may be lengthy because the recovery system must undergo an abnormal IPL.

A second configuration may be a single iSeries server attached to two SANs with one being a backup the other. This approach still requires an abnormal IPL and introduces a single point of failure by only offering data, not application resiliency.

Data replication

Data replication involves the transmission of changes to physical files (SQL tables) on a near real-time basis from one system to another, enabling the duplication of an application database. This product is offered by the cluster middleware vendors and other third parties.

One issue with this approach is that the failover process and activation or deactivation of the environment are the responsibility of the customer. Another issue on the OneWorld environment is that there are some IFS objects that J.D. Edwards recommends for replication that cannot be addressed by these products.

Remote journaling

Remote journaling involves the transmission of journal receiver contents from one iSeries server to a second recovery system. In the event of a system failure, the journal changes can be applied to a copy of the database on the second machine. The customer is required to develop and test both the apply process and the failover procedures.

In a high transaction environment, the failover time may be very lengthy if the apply process is built for net change and file updates are not performed on a near real-time basis. This method is best used in conjunction with a clustering implementation where there is a high volume of transactions. Refer to 8.4, “Performance tips” on page 249, for a further discussion on deploying remote journaling.

8.2 Typical execution

This section describes the application of iSeries clustering to OneWorld Xe. It provides an example of its execution under one of the more common server architectures implemented by OneWorld Xe iSeries customers.

8.2.1 Application to OneWorld Xe

Today, iSeries clustering has been proven on two OneWorld Xe server configurations:

- Physical two tier (fat client)
- Virtual three tier (thin client)

The iSeries-specific clustering processes are modifiable and implemented using industry standard TCP/IP redirection and cluster naming conventions to make the actual iSeries server perform client services transparent to the client. This makes it possible to deploy clustering within other iSeries/OneWorld Xe server configurations such as physical three tier (client, application server, and database server) and environments supporting TSE or browser-based clients.

The implementation involves the use of a “cluster name” associated with a “takeover” or “floating” TCP/IP address by the client, participating cluster nodes, and the managing network. The cluster name and its address are defined to the network; the client is configured to request OneWorld services from the enterprise server via a cluster name; each iSeries node in the cluster gets the cluster name specified in its host table and the OneWorld initialization file. Cluster services are configured on each iSeries node to create a TCP/IP interface associated with the cluster name’s IP address.

The resilient application and data replication for the Cluster Resource Group (CRG) are created by the cluster middleware product on all nodes using the object specifier file, clustering data areas, and an application CRG exit program provided by J.D. Edwards. The deployment server retains its role in distributing OneWorld application changes.

Clustering OneWorld Xe provides greater application resiliency in the event of a hardware failure. Although the software uses commitment control, it is done to ensure the completion of business transactions from an application execution standpoint. It is not deployed across the entire suite or all components of a particular application.

8.2.2 Virtual three-tier, two node cluster

This section explains how to do a virtual three-tier, two-node cluster setup, operation, failover, and return to normal operations.

Setup

The major components for a basic two-node virtual three-tier configuration (where a single iSeries server acts as both the application and database server) are illustrated in Figure 8-1.

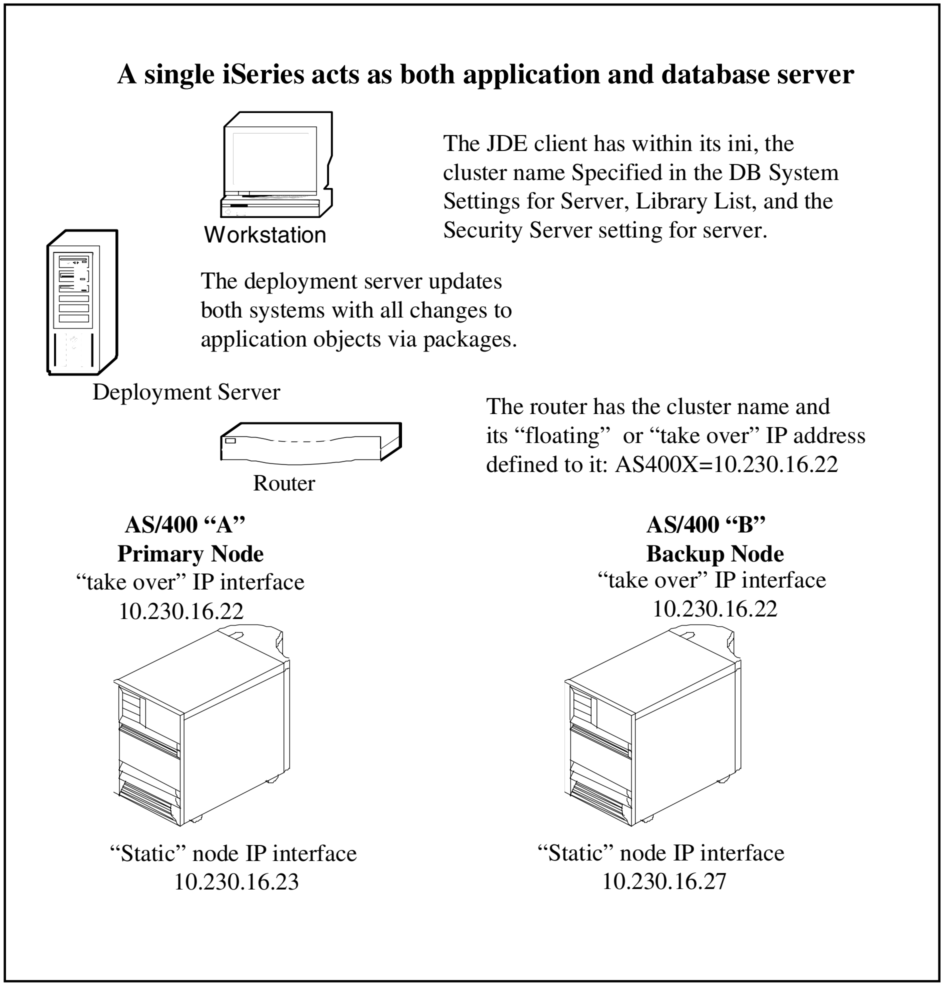


Figure 8-1 Virtual three-tier, two node cluster setup

Each iSeries node (either an independent machine or a logical partition) has at least:

- A host table entry with the cluster name associated with the “takeover” IP address.
- A TCP/IP interface defined to it by clustering services with the same “takeover” TCP/IP address. Only one server (the primary node) can have the interface supporting this address active at any point in time.
- A second static TCP/IP interface defined with a unique TCP/IP address for object and data replication.

- Equal versions of OS/400 (V4R4 or later) at equal PTF levels configured the same or as close as possible. Any PTFs required by the cluster middleware vendor to make their product operational must also be installed.
- Equal versions of OneWorld Xe at equal service pack levels configured the same or as close as possible.
- Equal (the same content) cluster exit programs, object specifier files, and clustering data areas for OneWorld Xe. In addition, the object specifier file version stated in the input data area must match the format of the object specifier file.
- Cluster middleware installed and configured.
- The cluster name specified under the CLUSTER stanza in the JDE INI file.
- Synchronized copies of the business data objects listed in the clustering object specifier file.

The client has the cluster name specified within its INI file. The cluster name is also used to set up the Object Configuration Mapping for the clustered environment. For details on this, see the "iSeries Clustering" chapter in the *OneWorld Xe Workstation and Server Administration Guide*.

Operation

A client request for OneWorld services is routed via the cluster name to the iSeries node with the activated "takeover" interface. The active node is called the *primary node*. Client-oriented changes to the OneWorld database and Integrated File System (IFS) objects resulting from business activity are replicated to the designated secondary nodes on a near real-time basis by the cluster middleware. Any application changes such as upgrades or modifications are distributed by the deployment server. This is illustrated in Figure 8-2.

The process shown in Figure 8-2 is explained here:

1. The J.D. Edwards client accesses the primary node server via the INI cluster specifications at session initiation.
2. The router directs client traffic to the "active" or primary node via its cluster name and the "takeover" or "floating" IP address.
3. The "primary node" iSeries server (AS400A):
   - Provides client services via the active TCP/IP interface with the "takeover" or floating IP address
   - Has OneWorld Xe services started via the cluster exit program
   - Forwards changes to OneWorld Xe business data to the designated secondary nodes

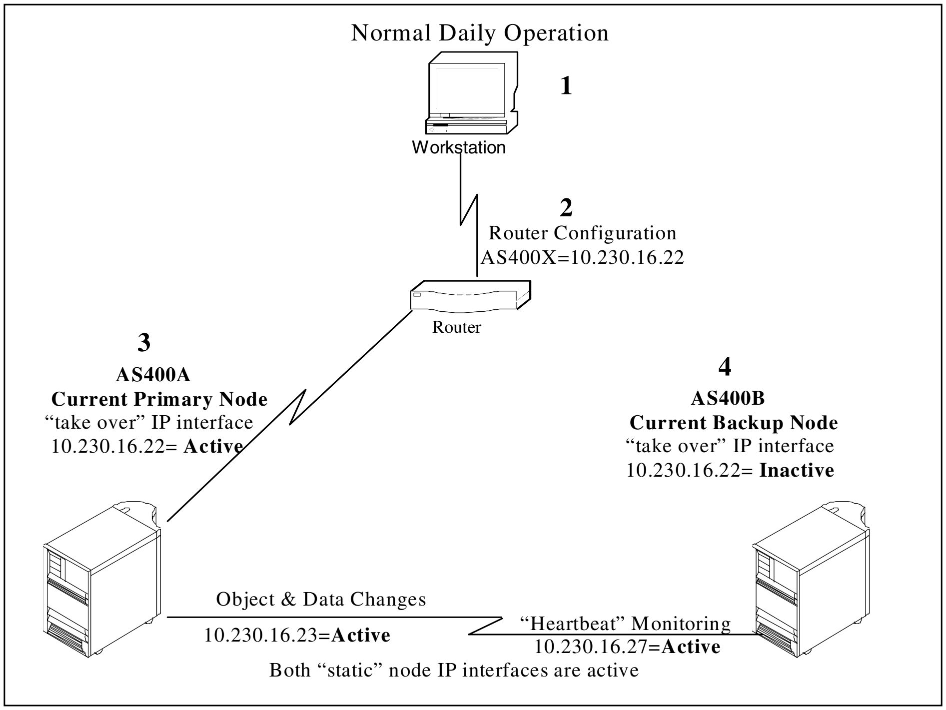


Figure 8-2 Virtual three-tier, two node cluster operation, normal daily operation

4. The “Backup” or “Designated Secondary” nodes iSeries server AS400B:
   - Monitors the “heartbeat” of iSeries server AS400A via the static TCP/IP interface
   - Receives and applies business data changes to the OneWorld Xe objects

Failover

The cluster middleware uses OS/400 APIs and resilient application components supplied by J.D. Edwards to switch over (also known as a “role swap” or “failover”) client services from the primary node to a designated secondary node participating in the cluster. This can be done manually or configured to execute automatically. The execution time of the switchover can depend on the processing power of the new primary node, the quantity of queued database or iSeries object changes to be applied, and the speed of the network.

When the client makes a request for enterprise server resources during or following a switchover, it posts a warning to the end user and waits a length of time specified in its initialization file before attempting a reconnect. On the reconnect attempt, the client request is routed to the new primary node. This event is illustrated in Figure 8-3.

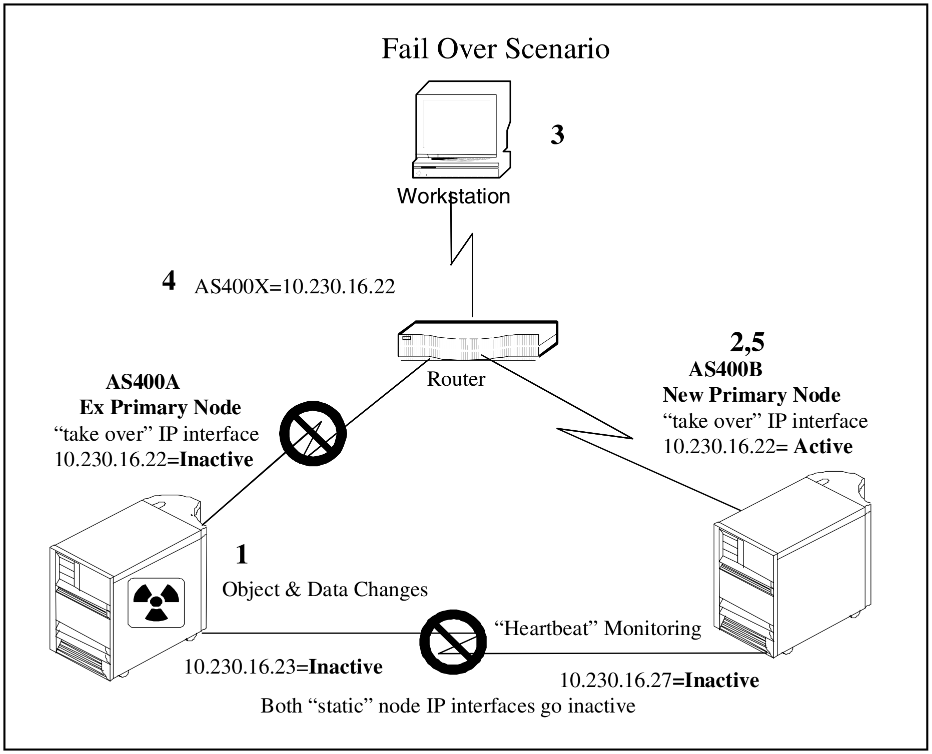


Figure 8-3 Virtual three-tier two tier cluster operation, failover scenario

The process shown in Figure 8-3 is explained here:

1. Client communication via the TCP/IP interface to the primary node is ended due to an administrator request or an actual outage detected by the "heartbeat" monitor.
2. On the designated secondary node, the IBM clustering APIs react to the event and invoke OneWorld's clustering exit program. This ends the application if it is active, applies any queued database or object updates, restarts OneWorld, and activates the takeover interface to begin hosts services.
3. At the next request for server resources, the client waits for a response and then attempts a reconnect.

4. The router directs client traffic to the “new” primary node (AS400B) via the IP takeover address associated with the cluster name.
5. On the “new” primary node, changes to business data and objects are queued to be forwarded and applied to the original primary node when it is made available to the network. Optionally changes can be forwarded and applied to other available secondary nodes within the cluster.

Operations

To return to normal operations, a scheduled, manual failover process is initiated as shown in Figure 8-4.

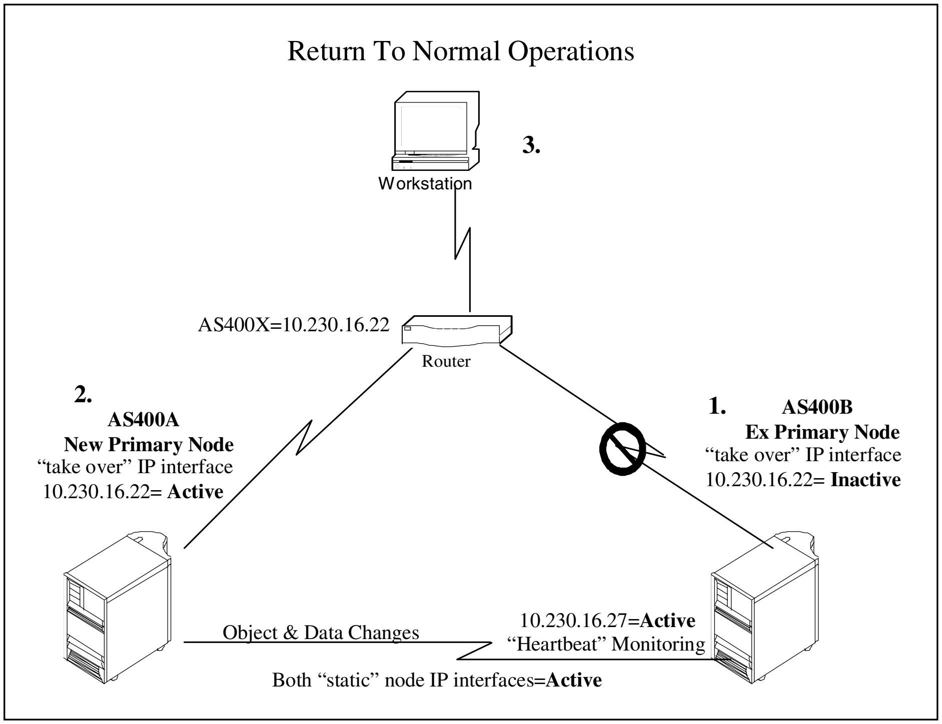


Figure 8-4 Virtual three-tier, two node virtual operation: Return to normal

The process shown in Figure 8-4 is explained here:

1. On the iSeries server AS400B (the “ex” primary node), the static TCP/IP interface is started. A switchover command is issued that ends the failover interface and initiates the clustering exit program on iSeries server AS400A so that it can assume the role of the new primary node.
2. On iSeries server AS400A (the “new” primary node), the clustering exit program brings any specified objects current by applying the changes

transmitted from iSeries server AS400B, activates the failover TCP/IP interface, ends OneWorld (if active), and restarts it. Processes supporting the replication of the specified object changes to the backup (“ex” primary node) are started.

3. On the client, at the next request for server resources, the client waits for a response on the length of time specified in its INI file. When this value is exceeded, OneWorld notifies the end user of a lost server connection and advises them to take appropriate action, and then attempts to reconnect.
4. The router directs client traffic to the new primary node AS400A via the cluster name and its takeover IP address.

8.3 Installation process

This section describes the installation process, discusses planning and installation issues, and provides a shell project plan to implement iSeries clustering with OneWorld Xe.

Installing a clustering solution on an iSeries server for OneWorld Xe is an exercise in business recovery planning, testing, and implementation. The project involves the participation of application development, operations, systems administration, and network staff. All three cluster middleware vendors offer a methodology and consulting services to ensure a successful installation. Upon activation, change control procedures should be established that are comprehensive enough to identify any modifications to the clustered environment that impact end-user application availability. In addition, the failover process should be tested periodically to ensure operational readiness.

8.3.1 Planning considerations

The typical data center running OneWorld is a heterogeneous environment that includes multiple hardware platform, each running more than one application. An application could depend on the exchange of data with another. Use care in identifying the role each system and application plays in supporting mission critical functions to ensure complete availability during a planned or unplanned outage. The requirements definition phase may reveal that additional clustering solutions may need to be implemented for other platforms, as well as other applications running on the iSeries server that may need to be added to the clustering solution.

If additional applications or operational procedures (user account management, print management, etc.) are to be included in the cluster, it is necessary to catalog all the objects that make them up and determine if they need to be replicated. Not all OS/400 object types are supported by all cluster middleware products. The replication of some that are supported can have a significant impact on CPU and disk resources on both the primary and secondary nodes. Consult you cluster middleware vendor for information concerning object types supported and their impact on server performance.

There may be additional hardware or reconfiguration requirements for the iSeries nodes participated in the cluster to provide optimal performance in its role the primary or secondary node. In a high transaction environment, the replication process occurring between the nodes may impact end-user service levels if they run over the same network segment. Therefore, it may be prudent to acquire separate communication IOPs to set up a private connection between the nodes for object replication. In addition, the journal capture and apply processes for object replication can impact CPU and disk resources on both the primary and secondary nodes. It is advisable to acquire additional DASD arms for the journaling activity. See 8.4, “Performance tips” on page 249, for more information on this topic.

J.D. Edwards posts updates to iSeries clustering on the Knowledge Garden. For planning and implementation, you can obtain the following items from there:

- *OneWorld Xe Minimum Technical Requirements for AS/400 Cluster Services*. These tables list minimum hardware and software technical requirements for deploying OneWorld Xe cluster services on the iSeries server.
- Latest version of *The Server Workstation and Administration Guide for Xe Addendum for AS/400 Clustering*. This document has step-by-step information on configuring OneWorld Xe for iSeries clustering.
- The clustering components (data areas, specifier file, exit program, and specifier file update program) can be downloaded in the form of an executable and installed on each iSeries server participating in the cluster.

Justification for hardware to support secondary nodes may also be an issue. Even after the costs of down time are understood, there is often resistance to obtain funding for a system that is assumed to be idle until a failover occurs. Some customers have successfully overcome this by combining it with a request to upgrade or acquire a development, test, or reporting system.

Additional network hardware or software may also need to be acquired to support virtual host name and IP address switching. At a minimum, some configuration work by network staff is required, so be sure to include project time in planning for those individuals.

From a planning standpoint, the extent of modifications to iSeries OneWorld Xe objects need to be determined to be sure all of the objects required for replication have been identified. J.D. Edwards can only supply a list of the objects recommended for replication and their locations library or the directory path they reside after an out-of-the -box installation. If customization efforts have changed the location or name of any of these objects, the specifier data has to be changed accordingly. Therefore, you need to include time for researching these modifications in the project plan.

8.3.2 Installation considerations

For installation, iSeries and OneWorld Xe Configurable Network Computing (CNC) expertise is required. Within the J.D. Edwards Global Advanced Technologies Services division, the Continuous Operations group has consultants with this background available for assistance in installing iSeries clustering for OneWorld Xe.

J.D. Edwards can supply a list of the objects and their location (the library or the directory path they reside in. Contact the Continuous Operations Group within Global Advanced Technology Services (GATS) for the latest recommended object list for your release and service pack level. At the time this redbook was written, the out-of-the-box installation object specifier list for an environment is:

- xxxxDTA-all objects
- xxxxCTL-all objects
- xx7333DNT-all objects
- DD7333-all objects
- OL7333-all objects
- SVM7333-the object F986110 only

In addition, within the IFS directory for an environment, the specfile directory objects with .xdb and .ddb extensions should also be included in the specifier file:

- dstmpl
- gbrlink
- gbrspec
- rdaspec
- rdatext

Prior to configuring the Ethernet IP interfaces, the failover IP address used with the cluster middleware vendor's product should not exist in the host table on the iSeries server or be defined as an interface. If it is, this may cause the process that creates the resilient application to fail.

The process itself creates the interface. In the process, an OS400 API assigns the Ethernet card supporting the first listed interface the failover interface. As a result, once the cluster middleware is installed and configured, review the TCP/IP interface to make sure it was assigned to the correct Ethernet card.

When you configure the clustering software, be sure that none of the clustering middleware jobs monopolize or conflict with other jobs in the QBATCH subsystem. QBATCH is sometimes configured to be the default job QUEUE for UBEs.

The format of the object specifier file changed between releases V4R4 and V4R5 of OS/400. The release parameter in the QCSTHAAPPI data area (the first six bytes) needs to match the format of the file. Please consult you cluster middleware vendor for the proper parameter values.

If the data needs to be re synchronized after the cluster middleware is installed and configured consider doing this by saving(with access paths) and restoring the objects listed in the specifier file between the primary and secondary nodes as opposed to letting the cluster middleware automatically do it. The save/restore method is significantly faster.

8.3.3 Sample installation plan

This section provides a sample installation plan. In the *planning* phase, make sure you perform the following steps:

1. Determine the business requirements.
2. Develop failover and fail back requirements.
3. Assess application requirements.
4. Assess server requirements.
5. Assess network requirements.
6. Develop a capacity plan.
7. Develop an architecture design.
8. Determine physical requirements.
9. Develop an implementation plan.

For the *implementation*, perform the following steps:

1. Select and acquire the necessary hardware.
2. Select and acquire the necessary software. This includes the cluster middleware and any service packs, “one of” PTFs or patches (including any required by the cluster middleware vendor) for each participating node.
3. Install and configure the hardware.
4. Install and configure the operating system with the goal to have matching configurations on each node.
5. Apply the operating system service packs with the goal to bring each node to equal patch or PTF levels.
6. Apply any required “one of” operating system PTFs to bring each node to equal patch or PTF levels.

7. Install and configure OneWorld with the goal to have matching configurations one each node.
8. Apply OneWorld service packs to bring each node to equal service pack levels.
9. Apply any required “one of” OneWorld patches with the goal to bring each node to equal patch levels.
10. Test OneWorld connectivity between the client and server.
11. Define the cluster name to the LAN/WAN environment:
    a. Configure iSeries failover Ethernet card, IP interface, and host table entry (for each iSeries node).
    b. Configure routers to use the cluster name and failover IP address.
    c. Test the connectivity.
12. Define a cluster environment to OneWorld:
    a. Configure client INI entries.
    b. Configure iSeries INI entries (for each iSeries node).
    c. Configure OCM.
    d. Synchronize business data and objects between primary and secondary nodes.
    e. Test with read only procedures (this can be done without the cluster middleware by activating/deactivating the failover interface).
13. Install and configure the cluster middleware:
    a. Install the clustering middleware.
    b. Install the J.D. Edwards-supplied clustering components.
    c. Identify additional non-OneWorld objects to replicate (this could be other applications or administrative objects such as user profiles, job queues, etc.)
    d. Determine the J.D. Edwards objects to replicate.
    e. Specify the objects to replicate (add object list to the J.D. Edwards-supplied object specifier file).
    f. Update the J.D. Edwards-supplied clustering input data area.
    g. Remove failover TCP/IP interfaces from all nodes (allow the cluster middleware to recreate them).
    h. Re-synchronize the OneWorld XE environments if necessary.
    i. Configure the cluster middleware.

j. Test the failover with read-only application functions.

k. Test the failover with update application functions.

8.4 Performance tips

This section offers some additional information on getting the best performance from your iSeries server or servers in a clustered configuration.

Clustering has some impact on system resource utilization in the areas of disk I/O and the CPU cycles to support its replication processes. Although there are three different middleware implementations for iSeries clustering, they all use OS/400 journaling (audit, IFS, and data journals) to capture changes to the objects named in the specifier file on the primary node and apply them to the application environment on the secondary node.

Understanding this and implementing the techniques associated with configuring and tuning an iSeries server for this type of I/O activity can be beneficial. It can help you get the most throughput for the least amount of system resources.

8.4.1 Primary node performance

Clustering processes typically consume no more than 5% of the CPU on the primary node. However, this can be greater in environments with very high transaction activity. The primary node CPU resources are also impacted by the types of objects selected for replication. The configuration and tuning strategy for this node should include:

- Following the journaling performance configuration guidelines for user ASP isolation discussed in the subsequent section. It is more important to do this on the primary node than the secondary node if there are budget limitations.
- Isolating the cluster jobs to use a separate memory pool. This can yield a greater degree of visibility to the resources consumed, making it easier to tune and control the processes.
- If there are additional objects to replicate outside of the list of OneWorld objects mentioned in 8.3, “Installation process” on page 244, they should be evaluated for inclusion on a “need” basis only in a business recovery scenario.

8.4.2 Secondary node performance

Clustering processes on the secondary node can be more resource-intensive than the primary node. This is due to the apply process, where the change information for the objects selected for replication is used to update the resilient application. The latency between updates to objects on the primary and secondary nodes can depend on their quantity, the communication bandwidth, and the CPU and disk resources allocated to secondary node for clustering support. The degree of latency can affect the fail of switchover time as well. Therefore, it is important to provide enough system resources to keep latency at a minimum. The configuration strategy for configuring and tuning the secondary nodes for optimum performance should include:

- Following the journaling performance configuration guidelines for user ASP isolation discussed in the next section.
- If the resilient application is transaction-intensive (more than 300,000 updates per hour), consider deploying remote journaling as described in the next section.
- If the secondary node is used for other purposes, consider isolating the cluster solution and the resilient application to a dedicated logical partition. This helps prevent other functions that the system is used for from encroaching system resources needed for clustering. On the newer 2xx and 8xx servers, partition resources can be dynamically re-allocated to the resilient applications in the event a failover occurs.
- If logical partitioning is not implemented, isolate the cluster jobs to utilizing a separate memory pool. This can yield a greater degree of visibility to the resources consumed, making it easier to tune and control the processes.
- For OneWorld Xe, the cluster application needs exclusive access to this environment. Most of the OneWorld processes that appear to be read only actually update files, so any use of the OneWorld instance being replicated on a secondary node should be thoroughly evaluated to prevent contention.

8.4.3 Configuration for journaling performance

For optimum journaling performance a non-library user ASP should be created with additional disk arms (not arms supporting the existing disk configuration; this could degrade production performance). The ASP should contain a minimum of two disk arms. It should be based on the total size of the receivers being kept there that occupy less than 70% of the available space. Only the journal receiver supporting clustering should be placed in that ASP. This can keep the journal writes from competing with the application I/O activity. The seek distance for the disk arms supporting receiver is kept at a minimum because the writes are sequential, ensuring optimum performance. Use the smallest capacity drives available with the highest rotational speed and support it with the fastest disk controller available with the most write cache. Spread the disk arms across multiple controllers if possible. For additional performance in a high transaction environment, consider using mirrored protection instead of RAID-5 for the journal ASP. In a server environment that uses both a SAN and internal disk drives, creating the user ASP with internal disks usually provides the best performance.

Implementing remote journaling with the batch journal caching RPQ 5799-JBC can improve replication throughput and shorten failover time in high transaction environments. It is supported by all of the cluster middleware vendors.

Compared to local journaling, it operates below the MI to perform microcode-to-microcode transport and memory-to-memory transfer of database changes between the iSeries servers. This results in less disk writes on the primary node for greater I/O efficiency. Much of the journaling work is shifted to the secondary node, freeing up CPU cycles on the primary node. If operated in synchronous mode (with receipt confirmation), database images are forwarded on a real-time basis. The journal images reach the secondary node before reaching the disk on the primary, ensuring retention of all database changes.

To tune or configure remote journaling for optimum performance, you need to:

1. Configure the ASPs supporting the journal receiver with the same amount of disk arms on both the primary and secondary nodes.
2. Use the recommendations in the first paragraph to set up journaling, including using mirrored protection for the journal receiver ASP.
3. Do not include internal(access path) entries in the journal.
4. Make sure the base memory pool has enough additional space in it for memory to memory transfer for catch up and reconnect. 100 MB is a good amount for this.

8.5 Operational and maintenance considerations

Once the clustering environment is implemented, there are tasks that need to be performed on a periodic basis to ensure operability when the need arises. This section describes common operational and maintenance issues concerning the installation and maintenance of iSeries clustering for OneWorld Xe.

8.5.1 Operational considerations

Due to its high degree of automation, clustering requires little monitoring. Each cluster middleware vendor provides Web-based GUI monitoring and management tools for remote cluster support. On implementation, the major items to monitor are:

- **Journal ASP utilization**

  ASP utilization should stay below 70%. The QSYSOPR message queue should be monitored daily for ASP overflow conditions.

- **Cluster message queue utilization**

  The message queues supporting clustering should be monitored periodically for errors in sending, receiving, and applying database changes. If failovers are performed, the messages should be reviewed after the event to become familiar with what constitutes a complete and successful failover. The cluster middleware vendor should be contacted to discuss anything that looks out of order.

- **Latency**

  Because the backlog of transactions to be applied to the secondary nodes can affect the length of a failover, check latency daily and track it to make sure there are enough system resources available to keep it as low as possible.

8.5.2 Maintenance considerations

As the business environment evolves, existing applications are enhanced or replaced and new ones are implemented. Their impact to the clustering environment needs to be examined to asses their relation to the resilient applications and impact on server resources. Upon implementation, the failover should be retested.

Journal management should be left up to the recommendations of the cluster vendor.

Finally, the cluster should be tested periodically to ensure failover is successful and complete when needed.

8.6 Sources of information

This section provides additional sources of information on iSeries clustering and OneWorld Xe.

8.6.1 Clustering information

- Cluster middleware vendor Web sites:
  - Lakeview Technology: http://www.lakeviewtech.com
  - DataMirror Corporation: http://www.datamirror.com
  - Vision Solutions: http://www.visionsolutions.com

- IBM iSeries clustering Web site:
  http://www-1.ibm.com/servers/eserver/iseries/ha/
- Redbook Web site:
  http://publib-b.boulder.ibm.com/Redbooks.nsf/Portals
  – *High Availability on the AS/400 System: A System Managers Guide,* REDP0111
  – *Clustering and IASPs for Higher Availability on the IBM @server iSeries Server*, SG24-5194

8.6.2 Installation information

"The AS/400 Clustering Addendum to the Server and Workstation Administration Guide for Xe" is located on J.D. Edwards Knowledge Garden Web site. You can access this site by clicking the KG Login link at: http://www.jdedwards.com/

9

Performance management and tuning

This chapter introduces the process of performance management and tuning, as it relates to OneWorld Xe on iSeries in a WebSphere Application Server environment. It begins with a brief review of the performance process, and then moves into the specific tools we have chosen to use for application analysis. For each of the selected tools, we provide a concise description of how the tool is used and the specific advantages that each brings to the measurement process.

For more information on the WebSphere Application Server and Java performance on the iSeries server, see *Java and WebSphere Performance on IBM @server iSeries Server*s, SG24-6256.

9.1 iSeries performance overview

A OneWorld Xe production environment can be complex, depending on the number of physical tiers, each of which becomes a performance management sub-component. Each sub-component and the inter-connecting communications layers must be managed appropriately to ensure optimal performance of the total system.

This section looks at the components of an iSeries-based system and how each component can affect performance. Then it examines these components in more detail and shows you how to use basic iSeries systems managements tools to locate potential problem areas. It gives you guidelines for what constitutes good and bad performance. Finally, it offers suggestions on how to improve existing performance.

This chapter is not a substitute for reading the appropriate documentation for each tool, but contains sufficient detail that a person familiar with the tool will see how to apply it to the problem at hand.

9.1.1 Components of performance

OneWorld Xe can be configured to run in a number of environments. Initially we consider the virtual three-tier environment where the application server, master business functions, and database server run on one iSeries server. Then we look at adding advanced functionality (and more complexity) by changing our environment to use a WebSphere-based server to handle browser-based clients.

In Figure 9-1, a OneWorld PC-based client is attached to a virtual 3-tier iSeries. Most of the transaction based processing is performed by the iSeries and the graphical rendering is performed by the PC client. From a performance viewpoint, the iSeries server can be treated as a black box, and the client sends its work in the form of transaction-based information and the iSeries responds. This is similar to the way we processed traditional interactive work. However under the covers, if we examine the way work is processed on the iSeries, we find a number of different types of servers, all inter-communicating to handle different parts of the transaction requests.

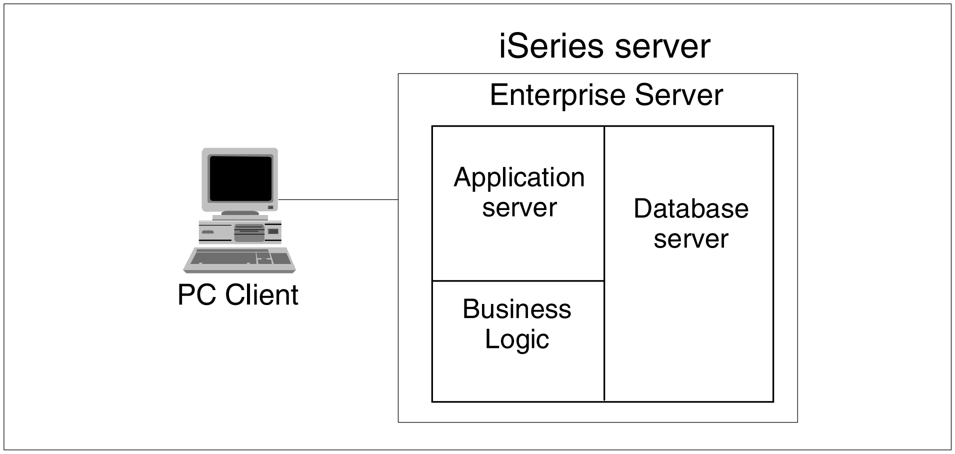


Figure 9-1 PC client connected to a virtual 3-tier OneWorld XE iSeries configuration

Each server function (for example, the database server) consists of many server jobs that wait to process information. The number of jobs has been predetermined as part of the installation process, depending on variables like the number of concurrent users.

Figure 9-2 shows how system resources are used when processing these requests. This is simplified for clarity. Other "lesser" components are covered later in this chapter.

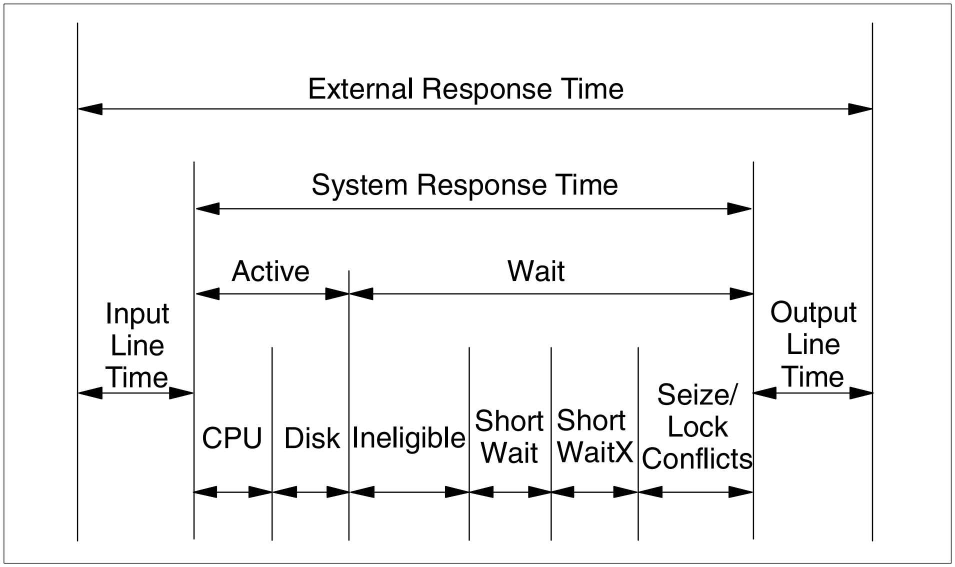


Figure 9-2 Components of server response time

Each transaction uses some amount of communications line capacity, CPU time, and main storage; does a number of disk accesses; and has to be scheduled for the CPU using a priority classification. The server response time experienced by a user is the total of many components:

- There is a transmission time delay for the transaction to reach the CPU. (This can be significant in some cases such as token-ring or remote workstation.)
- Once the transaction reaches the system, the system's response time measurement begins:
  - The server job may have to wait for an activity level at the system.
  - Once the activity level is entered, resource utilization begins, which includes:
    - CPU processing time (including CPU queuing)
    - Disk I/O time (including disk queuing)
  - There are also periods of inactivity during which the server job waits in any one of a number of states, including:
    - Ineligible time (not holding an activity level)
    - Short waits (still holding an activity level)
    - Short waits – extended (not holding an activity level)
    - Object or record seize wait conflict (still holding an activity level)
    - Object or record lock wait conflict (not holding an activity level)
- Finally, there is a transmission delay in the response returned to the user.

When increasing capacity or providing support for browser-based clients, it is quite common to add a separate Web server. This uses the Apache HTTP Server (powered by Apache) in V5R1, WebSphere Application Server middleware and J.D. Edwards provided Java servlets. A typical configuration is shown in Figure 9-3.

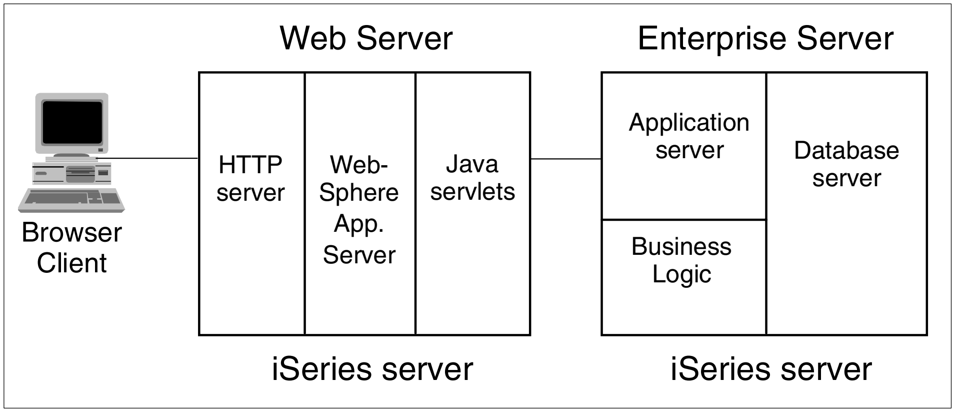


Figure 9-3 Typical configuration to support OneWorld Xe browser-based clients

With this configuration, each system would be considered a distinct entity. From a performance perspective, each must be managed separately in the first instance and then treated as a component of the total solution.

As a general rule, the workload profile for the Web server is very different from the enterprise server. The Web server is characterized by high CPU activity, multi-threaded jobs and minimal disk activity, supporting extensive communications, and datastream parsing and conversion. The enterprise server is characterized by significant CPU used by the application server and business rule validation, supported by lower CPU and significant disk activity by the database server. Consequently, each system is sized and tuned differently. But nevertheless, all the systems collectively play integral roles in providing optimal performance for the total system solution.

9.2 Queuing theory and utilization guidelines

This section reviews how queuing theory relates to performance and how you can use this knowledge to understand performance management. Then it examines the guidelines for resource utilizations, based on our recently acquired queuing theory knowledge.

9.2.1 Queuing theory

Customer expectations for a single job or specific transaction must be balanced against realistic expectations when many jobs are active during the same period of time. The work of a single job or transaction within that job is comprised of several tasks or services. The work given to a task is called a *request* or a *unit-of-work*. The task is also called a *server*, and the time taken to complete processing the request is called the *service time*. A single server can service only one request at a time. Multiple requests must wait for service by the server.

Using Figure 9-2 as a reference, the servers of the components of response time include CPU time and disk I/O time. As the figure shows, there are wait times associated with these servers, including waiting for CPU and waiting for disk I/O. These wait times are associated with queuing for the server. The higher the server utilization is, the greater the wait or queuing time is.

Queuing is a concept that applies to computer resources just as it applies to people waiting in line at the supermarket or waiting to access an Automated Teller Machine (ATM). In general, the time it takes to service a request or unit of work, whether it be a request to complete the purchase at the supermarket counter, complete a cash withdrawal at the ATM, perform a disk I/O operation, or use the CPU, depends on three primary parameters:

- The number of “waiters” in the line ahead of a new request
- The number of servers responding to requests
- The service time to complete a request once given to the server, which is a function of the speed of the server and the amount of work to do

There are mathematical equations to determine the effect of queuing. The formula for computing the Queuing Multiplier (QM) assumes:

- Work arrives in a normal (or Poisson) distribution pattern.
- Requests for the resources are not all for the same amount.

As the utilization of a server increases (resulting in the server handling more work), queuing accounts for part of the longer work (or request) completion time. For an interactive transaction, this can often be a significant component of long response times.

The Queuing Multiplier is a measure of the effect of queuing. Figure 9-4 shows the approximate QM for a single server queue. Using a simple example, assume the CPU is 67% utilized. The mathematical equation says the QM for a single CPU is 3. A QM of 3 means that on average, there are a total of three requests in the queue (you and two requests for work ahead of you). Therefore, if it takes an average of 0.2 seconds of CPU to service a request, it takes a minimum of 0.6 seconds to use the CPU (server response time = QM x stand-alone service time).

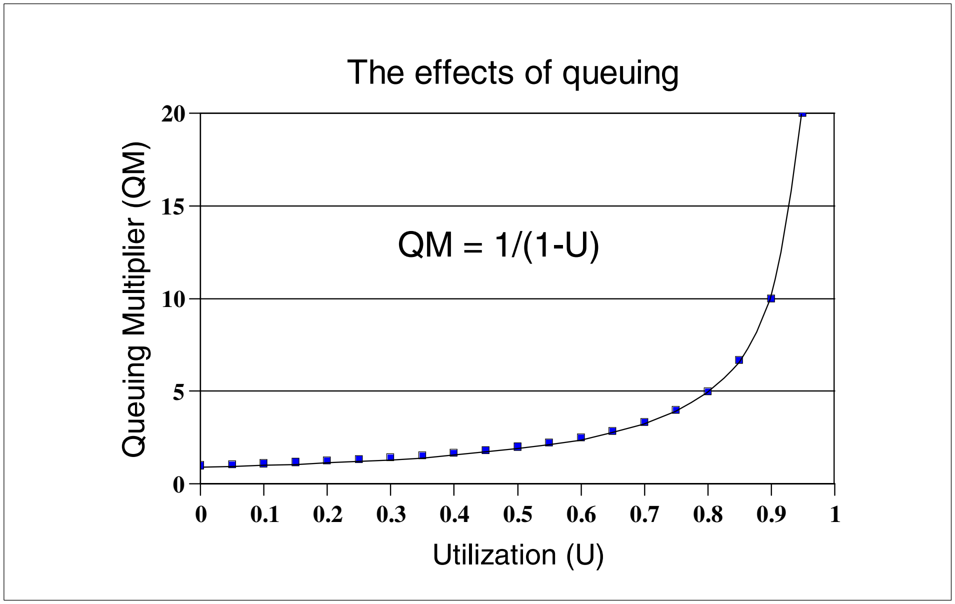


Figure 9-4 Effects of utilization on the Queuing Multiplier for a single server queue

The components of response time show that CPU is only one of the resources (servers) involved in response time. Disk service time, which is a function of the disk utilization and the disk QM, must also be factored into response time expectations. In real environments, additional wait times, such as exceptional wait times, must be factored in as well. These exceptional wait times (waiting for record or object locks, waiting for communication line data transmission, and so on) can play an important part in actual performance results and must be included when analyzing performance problems.

The simple queuing theory equation that was previously discussed assumes a single queue of requestors and a single server. In the high-end of the iSeries product range, some models have multiple processors (N-way) that have more than one CPU, even though there is only a single queue of requestors (Task Dispatch Queue). In this situation, the increased number of servers reduces the Queuing Multiplier and the average queue length.

As Figure 9-5 shows, for a given Queuing Multiplier, the QM value is lower as the number of servers increase. Conversely, this means that for a given Queuing Multiplier, multiple server systems can sustain higher CPU utilization than a single server and can maintain more stable response times, when utilization varies.

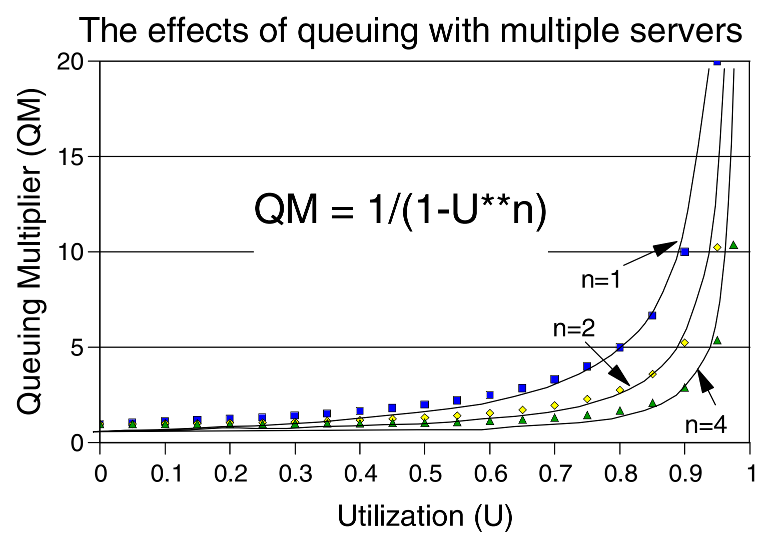


Figure 9-5 Effects of utilization on the Queuing Multiplier for a multiple server queue

The curve in Figure 9-5 shows the utilization at various rates and the significance of the “knee”. The knee of the curve is the point where a change in utilization produces a correspondingly higher change in the Queuing Multiplier. That is, the change along the Y-axis (Queuing Multiplier) is significantly greater than the change along the X-axis (utilization).

The knee of this curve is the maximum utilization point that a certain resource should be driven up to. After this “knee”, service time becomes less stable and may increase dramatically for small utilization increases. The objective of setting and operating at or below the maximum resource utilization threshold guidelines is to minimize queuing and provide a stable environment with consistent response time.

9.2.2 Utilization guidelines

Not all resources (servers) have the same effect on performance at the same utilization values. There are different recommended maximum values for the different resources, such as CPU, disk, memory, adapters, remote lines, IOPs, and so on.

Table 9-1 contains guidelines that you can use to determine if your system is healthy and operating with minimal resource constraints. This is a type of macro health check in which you take the system's temperature and blood pressure. There are other areas where we may be able to improve on performance, but if any of these guidelines are exceeded, then it is best to focus on remedying these.

Table 9-1 Resource utilization guidelines

Resource description	Good	Acceptable	Poor
CPU 1 Processor[a]	< 70%	70-80%	> 80%
CPU 2 Processors	< 76%	76-83%	> 83%[b]
CPU 4 Processors	< 81%	81-86%	> 86%
Disk Arm	< 40%	40-50%	> 50%
Disk IOA/IOP	< 60%	70%	> 80%
IOA/IOP Local	< 25%	35%	> 40%
IOP Multifunction	< 35%	45%	> 50%
IOA/IOP Communications	< 35%	45%	> 50%
IOA/IOP LAN	< 35%	40%	> 50%

a. This refers to CPU utilization of jobs whose priorities are equal to or higher than the interactive job priorities.
b. In a multiple processor environment, the guidelines are more sensitive to utilization above the values specified in the "Good" column.

You can find many of these parameters in the System Report and Component Report, which is part of IBM Performance Tools for iSeries (5722-PT1).

9.3 Setting performance objectives

You need to establish performance objectives. Having done this, you can take action to meet these objectives. Performance comes in many types. Different customers have different objectives, and like a car, the system may be tuned differently to meet specific requirements.

For one customer, the primary objective may be average transaction response time; for another customer, it may be transaction throughput; for yet another customer, it could be running short batch jobs. For another customer, it may be a combination of the three requirements, depending on time of day. So look at your business goals or objectives and write down your performance objectives. Talk to different department managers and users. An important objective that is often overlooked is consistency of response time.

Once you establish your performance objectives, translate them into utilization objectives for components of your system. For example, you may have an objective that says, “I want sub-second response time for user transactions”. This could translate into:

- An average CPU utilization of 40% for interactive users
- Batch CPU must not exceed an average of 35%
- Average total CPU is less than 75%
- Memory paging must average less than 200 pages per second
- Average disk arm utilization must be less than 10%

These values are typical of those run by many J.D. Edwards OneWorld Xe customers and are consistent with the values found in Table 9-1. They may vary for each customer, but represent the utilization guidelines that a performance consultant may recommend.

9.4 Basic iSeries server tuning

This section covers some of the basic operating system parameters that influence the way a Java-based application performs on an iSeries server. Both system values and subsystem settings may have significant affects on the application performance.

9.4.1 Manual tuning versus automatic tuning

You must first decide whether to tune your system manually or automatically. You may decide to use the automatic tuning to provide you with a starting point for tuning the system. Do this by using the automatic adjustment during the medium to heavy workload on your system and then turn it off. After you turn off the automatic adjuster, you may decide to adjust the pool sizes and activity level settings manually with 10% changes as explained in “Tuning the main storage with the WRKSYSSTS command” on page 277.

In both cases, you must check the settings of system values manually because the automatic tuner only changes the pool sizes and activity levels. This is determined with the setting of the system value QPFRADJ. The setting of this system value gives you the following levels of IBM-provided automatic tuning:

- 0 = No adjustment
- 1 = Adjustment at IPL
- 2 = Adjustment at IPL and automatic adjustment
- 3 = Automatic adjustment

The difference between settings 2 and 3 is that when using the value 3, the tuner remembers the values that were used before the IPL and starts the IPL with these values. When using the value of 2, the system resets the memory pool sizes and activity levels and starts tuning with the default values. We recommend that you use value 3 if you decide to use the automated tuning.

When you use manual tuning, you decide how much of the system resources are allocated to each subsystem and how many jobs are allowed to use the CPU at any given time.

If left at its default values, sometimes the automatic tuner dynamically changes the pools and does not leave them with enough main storage. Then the tuner is unable to respond fast enough to instantaneous work demands in that pool. To prevent this from happening, you can add a setting to your pools that sets a minimum and maximum of memory for that pool. Adjustments are then made within these predefined limits. Use the Change Shared Pool (CHGSHRPOOL) CL command and the parameters MINPCT and MAXPCT to set upper and lower limits.

9.4.2 Verifying the settings of system values

System values are to the jobs running on an iSeries server what libraries are to Java. They define the basic characteristics of the operating system and all the jobs being supported by the operating system.

There are a number of system values that you should consider changing with either the Change System Value (CHGSYSVAL) or Work with a System Value (WRKSYSVAL) commands. The following sections describe each of these system values and the recommended settings. The numbers that are recommended have been found to work during hundreds of real-life cases.

Allocation system values

Every job on the iSeris server has a Work Control Block Table Entry (WCBT) that the operating system uses to keep track of the jobs in the system. A WCBT entry can be seen as a pre-fabricated frame for a job that is filled with the jobs' run attributes by the time the job becomes active within a subsystem. A job holds the WCBT entry as long as the job is active and as long as there exists one spooled output file for the job on the system.

The allocation system values are:

- **QACTJOB**: Initial number of active jobs
- **QADLACTJ**: Additional number of active jobs
- **QADLSPLA**: Spooling control block additional storage
- **QADLTOTJ**: Additional number of total jobs
- **QJOBMSGQFL**: Job message queue full action
- **QJOBMSGQMX**: Maximum size of job message queue
- **QJOBMSGQSZ**: Job message queue initial size
- **QJOBMSGQTL**: Job message queue maximum initial size
- **QJOBSPLA**: Spooling control block initial size
- **QMAXJOB**: Maximum number of jobs
- **QMAXSPLF**: Maximum spooled files
- **QRCLSPLSTG**: Reclaim spool storage
- **QTOTJOB**: Initial total number of jobs

You should first verify the settings of the allocation system values on your system. In some cases, this may be the only action needed to remove the performance problem. We discuss the most important system values and their settings in the following sections.

QTOTJOB (Initial total number of jobs)

This specifies the initial number of Work Control Block Table entries created during IPL. You have two options to find out the required setting of this system value:

- WRKSYSSTS command
- DSPJOBTBL command

Enter the `WRKSYSSTS` command, press F21 to set the assistance level to Advanced. The Work with System Status display appears similar to the example in Figure 9-6. Pay attention to the value shown in the Jobs in system field. Add 20% to the number and set it to be the new system value.

We recommend that you use the advanced assistance level because it provides us all the required information at a glance.

Make sure that there is not an excessive number of dormant spooled files. Each spooled file corresponds to a job, and the job slot is not released until the spooled file is printed or deleted. It is a matter of good system housekeeping to monitor for spooled files that are no longer needed and remove them periodically from the system. This should be part of nightly operations.

```
                           Work with System Status                   AS05
                                                           10/12/01  10:12:35
% CPU used . . . . . . . :        97.8    System ASP . . . . . . . :    87.74 G
% DB capability  . . . . :        12.5    % system ASP used  . . . :    28.9425
Elapsed time . . . . . . :    00:00:03    Total aux stg  . . . . . :    144.7 G
Jobs in system . . . . . :         443    Current unprotect used . :     3135 M
% perm addresses . . . . :        .009    Maximum unprotect  . . . :     3756 M
% temp addresses . . . . :        .015

Sys      Pool   Reserved    Max  ----DB-----  --Non-DB---  Act-    Wait-   Act-
Pool    Size M   Size M     Act  Fault Pages  Fault Pages  Wait    Inel    Inel
  1     493.68   200.95   +++++     .0    .0     .0    .0     .0      .0      .0
  2    2356.60     3.35      20     .0    .0     .0    .0   2990      .0      .0
  3      40.95      .00       5     .0    .0     .0    .0     .0      .0      .0
  4     504.75      .00       9     .0    .0    1.9   1.9   39.0      .0      .0
  5     200.00      .00     150     .0    .9     .0    .0     .0      .0      .0
  6     500.00      .99     200     .0    .0     .9  25.0  13153      .0      .0



                                                                          Bottom
===>
```

Figure 9-6 WRKSYSSTS with advanced assistance level setting

Enter the `DSPJOBTBL` command and pay attention to the number shown in the Entries in use column. Add 20% to that value and set that to be your system value.

Figure 9-7 shows you what to expect when you enter the DSPJOBTBL command.

```
                          Display Job Tables                   AS05
                                                    10/12/01  10:07:24
Permanent job structures:               Temporary job structures:
  Initial  . . . . :   30                  Initial  . . . . :   20
  Additional . . . :   30                  Additional . . . :   10
  Available  . . . :   441                 Available  . . . :   26
  Total  . . . . . :   864
  Maximum  . . . . :   163520


                        ---------------------Entries---------------------
     Table         Size         Total    Available        In-use       Other
         1       885504           864          441           443           0


                                                                      Bottom
Press Enter to continue.

F3=Exit   F5=Refresh   F11=In-use entries   F12=Cancel
```

Figure 9-7 DSPJOBTBL showing the total number of WCBT entries

In these examples, the value 30 represents the setting of the system value QTOTJOB, and the value 20 represents the setting of the system value QACTJOB. The number of In-use Entries shows how many jobs are active and how many jobs are not active but have output files left on the system. Compare Figure 9-7 with Figure 9-8.

```
                              Display Job Tables                    AS05
                                                           10/12/01  10:07:24
Permanent job structures:                 Temporary job structures:
  Initial  . . . . :   30                   Initial  . . . . :   20
  Additional . . . :   30                   Additional . . . :   10
  Available  . . . :   441                  Available  . . . :   26
  Total  . . . . . :   864
  Maximum  . . . . :   163520

             -----------In-use Entries------------
                                       Job       Output
     Table         Active             Queue       Queue
         1            271                 0         172



                                                                       Bottom
Press Enter to continue.

F3=Exit   F5=Refresh   F11=Total entries   F12=Cancel
```

Figure 9-8 DSPJOBTBL showing the WCBT entries currently in use

The setting of this system value should be high enough not to be exceeded between IPLs. You should suspect an incorrect setting of this system value if your system is suffering from mysterious slowdowns that happen periodically. If the number of jobs in the system exceeds the number specified with this system value, all the jobs in the system are paged out from the main storage, and the amount of WCBT entries specified with system value QADLTOTJ is created.

After this is completed all the jobs are paged back in to the main storage, and the normal processing continues until the amount of the additional WCBT entries created is used up and the jobs are paged out again. Based on our experience, we prefer setting QTOTJOB too large rather than setting it too small. The default value shipped with the operating system is 30, which usually is too small. A change to this system value takes effect at the next IPL.

QADLTOTJ (Additional number of total jobs)

This value specifies how many WCBT entries are created if the amount of jobs in system exceeds the number specified with system value QTOTJOB. The default value of 10 should be enough provided that the setting of QTOTJOB is sufficient. The larger this value is, the longer it takes to create the additional WCBT entries. The normal workload on the system is not being processed while the additional entries are created. We suggest that you use the default setting to minimize the performance impact. A change to this system value takes effect immediately.

QACTJOB (Initial number of active jobs)

This value specifies the initial number of active jobs for which storage is allocated during initial program load (IPL). An active job is a job that has started running but has not ended. The amount of storage allocated for each active job is approximately 110K. This storage is in addition to the storage allocated using the system value QTOTJOB. A change to this system value takes effect at the next IPL. The shipped value is 20. A reasonable value to assign to QACTJOB is your estimate of the number of active jobs on a typically heavy-use day. This can be done by viewing the active jobs field on the active jobs display (WRKACTJOB command).

```
                             Work with Active Jobs                   AS05
                                                          10/12/01  10:13:42
CPU %:   99.1     Elapsed time:   00:05:43     Active jobs:   271

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...

Opt  Subsystem/Job  User        Type  CPU %  Function        Status
     QBATCH         QSYS        SBS      .0                   DEQW
     QCMN           QSYS        SBS      .0                   DEQW
     QCTL           QSYS        SBS      .0                   DEQW
        QSYSSCD     QPGMR       BCH      .0  PGM-QEZSCNEP     EVTW
     QEJBSBS        QSYS        SBS      .0                   DEQW
       QEJBADMIN    QEJB        BCI      .0  PGM-QEJBADMIN    JVAW
       QEJBMNTR     QEJB        ASJ      .0  PGM-QEJBMNTR     EVTW
       TEAM16       QEJB        BCI    81.4  PGM-QEJBSVR      JVAW
       TEAM16ADMN   QEJB        BCI      .1  PGM-QEJBADMIN    JVAW

                                                                   More...
Parameters or command
===>
F3=Exit   F5=Refresh       F7=Find      F10=Restart statistics
F11=Display elapsed data   F12=Cancel   F23=More options   F24=More keys
```

Figure 9-9 Using WRKACTJOB to determine the QACTJOB system value setting

Set the QACTJOB system value high enough so that storage does not need to be allocated for additional active jobs using the system value QADLACTJ.

QADLACTJ (Additional number of active jobs)

This specifies the additional number of active jobs that have storage allocated when the initial number of active jobs (the system value QACTJOB) is reached. An active job is a job that has started running but has not ended. Auxiliary storage is allocated whenever the number of active jobs exceeds the storage which has already been allocated.

The amount of storage allocated for each job is approximately 110K. Setting the number close to 1 can cause frequent interruptions when many additional jobs are needed. The number should not be set too high because the time required to add additional storage should be minimized. A change to this system value takes effect immediately.

Storage system values

The storage system values are a set of system values that affect to how the system storage is being used. The ones that you should set correctly are:

- QBASACTLVL
- QBASPOOL
- QMAXACTLVL
- QMCHPOOL

QBASACTLVL (Base storage pool activity level)

Increasing this value reduces or eliminates thread transitions into the ineligible state. Initial choice of value should be (arbitrarily) high, and then as implementation proceeds, monitor and decrease the value if necessary.

This value determines the maximum number of threads (not jobs) running in the *BASE pool that can use the processor concurrently. If the activity level is too low, the threads may transition to the ineligible condition. If the activity level is too high, excessive page faulting may occur.

It is important to increase this value for systems that are executing a large number of threads. Having a setting of too low may slowdown or even hang the system.

Threads running on the system can be in any of the following states:

- Active
- Wait
- Ineligible

An active thread exists in main storage and processes work requested by the application. A thread in the wait state needs a resource that is not available. An ineligible thread has work to do, but the system is unable to accept more work at that time because an activity level is not available.

Use the WRKSYSSTS command to adjust this setting. Increasing the value highlighted in the example shown in Figure 9-10 reduces or eliminates the thread transitions into the ineligible state.

The setting of QBASACTLVL only affects the activity level setting for the *BASE pool, use the WRKSHRPOOL or WRKSYSSTS commands to change the activity level settings for other pools.

```
                            Work with System Status                    AS05
                                                             10/12/01  10:12:35
% CPU used . . . . . . . :       97.8    System ASP . . . . . . . :    87.74 G
% DB capability  . . . . :       12.5    % system ASP used  . . . :    28.9425
Elapsed time . . . . . . :   00:00:03    Total aux stg  . . . . . :    144.7 G
Jobs in system . . . . . :        443    Current unprotect used . :     3135 M
% perm addresses . . . . :       .009    Maximum unprotect  . . . :     3756 M
% temp addresses . . . . :       .015

Sys      Pool   Reserved    Max  ----DB-----  --Non-DB---  Act-    Wait-   Act-
Pool    Size M   Size M     Act  Fault Pages  Fault Pages  Wait    Inel    Inel
  1     493.68   200.95   +++++     .0    .0     .0    .0     .0      .0      .0
  2    2356.60     3.35      20     .0    .0     .0    .0   2990    1329      70
  3      40.95      .00       5     .0    .0     .0    .0     .0      .0      .0
  4     504.75      .00       9     .0    .0    1.9   1.9   39.0      .0      .0
  5     200.00      .00     150     .0    .9     .0    .0     .0      .0      .0
  6     800.00      .99     200     .0    .0     .9  25.0  13153      .0      .0



                                                                         Bottom
===>
F21=Select assistance level
```

Figure 9-10 Using WRKSYSSTS to change the QBASACTLVL

> **Note:** By default, everything except interactive jobs and printer jobs run in the *BASE pool. This includes the WebSphere Application Server jobs.

If you are using a system with a mixed load of traditional high-level language (HLL) code and Java code, consider creating a separate pool for the Java servlets.

QBASPOOL (Base storage pool minimum size)
This specifies the minimum value of main storage that is left to the *BASE pool after all the subsystems are started. If this value is set too high, there will be not enough storage left for the additional memory pools. A significantly low amount of memory in the *BASE pool may have a negative effect on your subsystem monitors, communications jobs, and so on.

QMAXACTLVL (Maximum activity level of system)
This specifies the number of threads that can compete at the same time for main storage and processor resources. For all active subsystems, the sum of all threads running in all memory pools cannot exceed QMAXACTLVL. If a thread cannot be processed because the activity level has been reached, the thread is held until another thread reaches a time slice end or along wait. In Java or WebSphere environments, this usually generates a transaction rollback. This value should be larger than the sum of the activity levels for all your memory pools. If QMAXACTLVL is smaller, activity levels in the memory pools may not be used. We recommend setting it to *NOMAX.

A change to this system value takes effect immediately.

QMCHPOOL (Machine storage pool size)
This is perhaps the most important system value because this value specifies the amount of main storage that is reserved for both the operating system programs and Licensed Internal Code. Enter the WRKSYSSTS command as shown in Figure 9-6 on page 267. Then at look at the Reserved Size column for memory pool 1 (*MACHINE). Multiply that number by two and set that number to this system value.

Your system performance may be severely affected if this system value is too small. Your application performance may be unsatisfactory if this value is too large. There may be enough of main storage available in the system to support your workload, but this is not where it is needed.

> **Note:** The system values QTOTJOB and QACTJOB require an IPL for the changes to become effective.

9.4.3 Setting the memory pool sizes and activity levels manually

The manual adjustment can be done by using both the WRKSHRPOOL and WRKSYSSTS commands. This section shows how to use the WRKSYSSTS command for tuning the system. All the displays were created with using the advanced assistance level.

Determining the memory pool size

If data is already in main storage, it can be referred to independently of the memory pool it is in. However, if needed data does not exist in any memory pool, it is brought into the same memory pool for the job that referred to it (this is known as a page fault). As data is transferred into a memory pool, other data is displaced and, if changed, it is automatically recorded on disk (this is called paging). The memory pool size should be large enough to keep data transfers (into and out of main storage) at a reasonable level because the rate affects performance.

1. Determine in which pool the QEJBSBS jobs are running:
   a. Enter the `WRKSBS` command. Then you see a display like the example in Figure 9-11.

```
                               Work with Subsystems
                                                             System:   AS05
Type options, press Enter.
  4=End subsystem   5=Display subsystem description
  8=Work with subsystem jobs

                        Total     -----------Subsystem Pools------------
Opt  Subsystem     Storage (M)    1   2   3   4   5   6   7   8   9  10
     QBATCH               .00     2
     QCMN                 .00     2
     QCTL                 .00     2
     QEJBSBS              .00     2   6
     QHTTPSVR             .00     2
     QINTER               .00     2   4
     QSERVER              .00     2
     QSPL                 .00     2   3
     QSYSWRK              .00     2   5
     QUSRWRK              .00     2

                                                                      Bottom
Parameters or command
===>
F3=Exit   F5=Refresh   F11=Display system data   F12=Cancel
F14=Work with system status
```

Figure 9-11 Using the WRKSBS command to relate memory pools with WRKSYSSTS

   b. Press F14 or enter the `WRKSYSSTS` command to see the Work with System Status display (Figure 9-12). Note that the pool 2 for subsystem QEJBSBS is *not* the pool 2 on the WRKSYSSTS display. The memory pools are allocated in the order in which the subsystems are started. In this case, pool 6 on the WRKSYSSTS display is pool 2 for the QEJBSBS subsystem.

```
                            Work with System Status                    AS05
                                                             10/12/01  10:12:35
% CPU used . . . . . . . . :       97.8    System ASP . . . . . . . :    87.74 G
% DB capability  . . . . :       12.5    % system ASP used  . . . :    28.9425
Elapsed time . . . . . . . :   00:00:03    Total aux stg  . . . . . :    144.7 G
Jobs in system . . . . . . :        443    Current unprotect used . :     3135 M
% perm addresses . . . . . :       .009    Maximum unprotect  . . . :     3756 M
% temp addresses . . . . . :       .015

Sys      Pool   Reserved    Max  ----DB-----  --Non-DB---  Act-    Wait-   Act-
Pool    Size M   Size M     Act  Fault Pages  Fault Pages  Wait    Inel    Inel
  1     493.68   200.95   +++++     .0     .0     .0    .0     .0      .0      .0
  2    2356.60     3.35      20     .0     .0     .0    .0   2990      .0      .0
  3      40.95      .00       5     .0     .0     .0    .0     .0      .0      .0
  4     504.75      .00       9     .0     .0    1.9   1.9   39.0      .0      .0
  5     200.00      .00     150     .0     .9   19.0  40.0     .0      .0      .0
  6     800.00      .99     200     .0     .0   10.9 125.0  13153      .0      .0



                                                                          Bottom
===>
F21=Select assistance level
```

Figure 9-12 Using WRKSYSSTS to determine the size of a memory pool

2. Display the faulting data that represents the rate that data is moved between main storage and disk:
   - This is shown by the two columns in the middle of the display; the columns are labeled DB Faults and Pages and NonDB Faults and Pages.
   - Every row represents a separate memory pool.
   - In our example, we concentrate on pool 6.
3. Pay attention to the page fault rate (DB versus NDB)
   - DB faults and pages indicate how the data is moved.
   - NonDB faults and Pages represent the program code, job work areas, variables, etc.
   - Increase the pool size to reduce faulting or reduce the amount of activity levels (described next).

Determining the activity level setting for a pool

To determine the activity level setting for a pool, follow these steps:

1. Display the transition data that is high-lighted in Figure 9-13.
2. Pay attention to the three rightmost columns.
3. Every row represents a separate memory pool.
4. It is normal for threads to go from active status to wait status.
5. The rate of change is important here. Increase activity level if the rate of transitions from wait to ineligible is approaching the rate of active to wait.
6. Increase or decrease the activity levels with ten percent increments.
7. Remember that Java and WebSphere threads need more activity levels per pool than the legacy applications used to need.

```
                            Work with System Status                   AS05
                                                            10/12/01  10:12:35
% CPU used . . . . . . . :       97.8    System ASP . . . . . . . :    87.74 G
% DB capability  . . . . :       12.5    % system ASP used  . . . :    28.9425
Elapsed time . . . . . . :   00:00:03    Total aux stg  . . . . . :    144.7 G
Jobs in system . . . . . :        443    Current unprotect used . :     3135 M
% perm addresses . . . . :       .009    Maximum unprotect  . . . :     3756 M
% temp addresses . . . . :       .015

Sys      Pool   Reserved    Max  ----DB-----  --Non-DB---  Act-   Wait-  Act-
Pool    Size M   Size M     Act  Fault Pages  Fault Pages  Wait   Inel   Inel
  1     493.68   200.95   +++++     .0    .0     .0    .0    .0      .0     .0
  2    2356.60     3.35      20     .0    .0     .0    .0  2990      .0     .0
  3      40.95      .00       5     .0    .0     .0    .0    .0      .0     .0
  4     504.75      .00       9     .0    .0    1.9   1.9  39.0      .0     .0
  5     200.00      .00     150     .0    .9   19.0  40.0    .0      .0     .0
  6     800.00      .99     200     .0    .0   10.9 125.0 13153      .0     .0

                                                                        Bottom
===>
F21=Select assistance level
```

Figure 9-13 Using WRKSYSSTS to monitor the transition data

See *OS/400 Work Management (V4R5)*, SC41-5306, for a thorough description of system values and other work management-related functions.

Tuning the main storage with the WRKSYSSTS command

The Work with System Status (WRKSYSSTS) command provides a list of page faulting and wait-to-ineligible transitions for each main memory pool. Set the assistance level to intermediate or advanced. Use the following steps to determine acceptable levels of faulting and W->I transitions:

1. The machine pool (pool 1) should have fewer than 10 faults per second (sum of Db and Non DB faults). Increase the pool by 10% until the faults per second are satisfactory.
2. If your machine pool is experiencing very low page fault rates (<0.4), you should decrease the size of the machine pool. If the page fault rate is this low, you may be affecting work in some other pools.
3. If only system jobs and subsystem programs are running *BASE, the fault rate for that pool should be less than 30 faults per second. You need to decrease another pool to increase *BASE.
4. For interactive pools, the W->I should be small (less than 10% of A->W). If you see any W->I, increase the MAXACT by 5-10 until the W->I is 0. After you increase the MAXACT value, press F10 to reset the statistics; do not use F5 to refresh. You should wait at least 1 minute between refreshes.
5. For user pools, the fault rate alone is not necessarily a measure of good or poor performance. Response time and throughput are the actual measures of performance. Therefore, you should tune your pools by moving storage from pools with better performance to pools with poor performance. Continue to move the storage to the pool with poor performance until performance improves. Do not decrease a pool by more than 10% at one time.

9.4.4 Common WebSphere/Java jobs

This section lists some of the common WebSphere/Java related jobs and their corresponding subsystems. You can use this information to help determine where your workload issues may be.

Java-related jobs and subsystems

Table 9-2 helps you to combine the Java and WebSphere-related jobs and subsystems with each other.

Table 9-2 Jobs and subsystems related to Java and WebSphere

Subsystem	Job Name	Job type	Description
The QEJBSBS subsystem holds the WebSphere Application Server related jobs. Generally all WebSphere instances run in this subsystem.	<instance name>	BCI	Application server running job. All WebSphere Java code, Servlets, JSPs, and EJBs execute in this job.
	<instance name>ADMN	BCI	Administration server for the instance.
	<instance name>MNTR	BCH	Monitor (nanny) job.
The QHTTPSVR subsystem holds the HTTP Server-related jobs. Generally all HTTP Server instances run in this subsystem.	<instance name>	BCH	Primary instance manager for the HTTP Server.
	<instance name>	BCI	Usually several of these jobs, each capable of managing incoming requests.
The QINTER subsystem holds the interactive jobs. Since a traditional interactive job is unable to execute threads, running a Java program from an interactive prompt results in a batch immediate job being spawned.	QJVACMDSRV	BCI	Java invoked by an interactive job.
The QSYSWRK subsystem holds system-related jobs. Jobs in this subsystem are generally pre-start jobs.	QSQSRVR	PJ	Usually a large number of these jobs that provide the database pooling for SQL access to DB2.
	QRWTSRVR	PJ	Used for DDM access to Database files.
The QSERVER subsystem holds jobs related to communication.	QZRCSRVS	PJ	Used for RPC (also known as DPC).
The QSERVER subsystem holds jobs related to the TCP/IP servers.	QZDASOINIT	PJ	Provides database pooling for the Toolbox JDBC driver.

9.4.5 Working with the prestart jobs

A prestart job is a batch job that starts running before a program on a remote system sends a program start request. Prestart jobs are different from all the other jobs because they use prestart job entries to determine which program, class, and memory pool to use when they are started. Within a prestart job entry, you specify attributes that the subsystem uses to create and manage a pool of prestart jobs.

The prestart jobs QSQSRVR and QZDASOINIT actually perform the access to database that your OneWorld applications request. The native JDBC functions are carried out by the QSQSRVR jobs running on QSYSWRK, whereas the IBM Toolbox uses the QZDASOINT jobs that are running on QSERVER subsystem. Make sure that these server jobs have the resources they need when they need them. Otherwise, the users will experience bad response times or even time outs.

The easiest way to provide these server jobs with the memory and activity level is to direct these jobs to a memory pool that no other jobs are using. We use the QSQSRVR jobs here as an example, but the same approach may be used with the QZDASOINIT jobs.

You can improve the server job performance by answering the following questions:

- How many prestart jobs are running on the system?
- How many prestart jobs are started with the subsystem?
- How many prestart jobs do you need to process the workload?
- How do you provide resources for the prestart jobs?

Determining the number of active prestart jobs

Before you change the number of prestart server jobs, you should know how many of these jobs you need. By default, these server jobs run in the subsystem QSYSWRK. The easiest way to decide this is to enter the Display Active Prestart Jobs (DSPACTPJ) command as shown in Figure 9-14.

```
                          Display Active Prestart Jobs
                                                          10/12/01  09:04:14
 Subsystem  . . . . . :   QSYSWRK        Reset date . . . . . :   10/10/01
 Program  . . . . . . :   QSQSRVR        Reset time . . . . . :   11:34:03
   Library  . . . . . :     QSYS          Elapsed time . . . . :
0045:30:11

 Prestart jobs:
   Current number . . . . . . . . . . . . . . . . :   38
   Average number . . . . . . . . . . . . . . . . :   38.1
   Peak number  . . . . . . . . . . . . . . . . . :   52

 Prestart jobs in use:
   Current number . . . . . . . . . . . . . . . . :   34
   Average number . . . . . . . . . . . . . . . . :   34.0
   Peak number  . . . . . . . . . . . . . . . . . :   49
 Program start requests:
   Current number waiting . . . . . . . . . . . . :   0
   Average number waiting . . . . . . . . . . . . :   .0
   Peak number waiting  . . . . . . . . . . . . . :   0
   Average wait time  . . . . . . . . . . . . . . :   00:00:00.0
   Number accepted  . . . . . . . . . . . . . . . :   4619
   Number rejected  . . . . . . . . . . . . . . . :   0
```

Figure 9-14 Display Active Prestart Jobs

When comparing these values with the values shown in Figure 9-17 on page 283, we decide that the default values are not sufficient enough.

Determining the number prestart jobs you start

This is shown in the susbsystem descriptions prestart job entries part. Enter the Display Subsystem Description command:

```
DSPSBSD SBSD(QSYSWRK)
```

Then you see the Display Subsystem Description display (Figure 9-15).

```
                         Display Subsystem Description
                                                           System:   AS05
Subsystem description:   QSYSWRK       Library:   QSYS
Status:   ACTIVE

Select one of the following:

     1. Operational attributes
     2. Pool definitions
     3. Autostart job entries
     4. Work station name entries
     5. Work station type entries
     6. Job queue entries
     7. Routing entries
     8. Communications entries
     9. Remote location name entries
    10. Prestart job entries


                                                                 More...
Selection or command
===>10

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
```

Figure 9-15 Display Subsystem Description

Choose option 10 for all Prestart job entries as shown in Figure 9-16.

```
                         Display Prestart Job Entries
                                                            System:   AS05
Subsystem description:   QSYSWRK       Status:   ACTIVE

Type options, press Enter.
  5=Display details

Opt     Program        Library        User Profile
        QANEAGNT       QSYS            QUSER
        QIWVPPJT       QIWS            QUSER
        QRWTSRVR       QSYS            QUSER
 5      QSQSRVR        QSYS            QUSER
        QSRRATBL       QSYS            QUSER
        QTMMSRVR       QTCP            QTCP
        QTMSCLCP       QTCP            QTCP
        QTMSSRCP       QTCP            QTCP
        Q5BWHSRV       QSYS            QUSER

                                                                      Bottom
F3=Exit   F9=Display all detailed descriptions   F12=Cancel
```

Figure 9-16   Display Prestart Job Entries

Choose option 5 to see the details for the prestart job entry QSQSRVR as shown on Figure 9-17.

```
                     Display Prestart Job Entry Detail
                                                            System:   AS05
Subsystem description:   QSYSWRK       Status:   ACTIVE

Program  . . . . . . . . . . . . . . . . . . . . . . :   QSQSRVR
  Library  . . . . . . . . . . . . . . . . . . . . . :     QSYS
User profile . . . . . . . . . . . . . . . . . . . . :   QUSER
Job  . . . . . . . . . . . . . . . . . . . . . . . . :   QSQSRVR
Job description  . . . . . . . . . . . . . . . . . . :   *USRPRF
  Library  . . . . . . . . . . . . . . . . . . . . . :
Start jobs . . . . . . . . . . . . . . . . . . . . . :   *NO
Initial number of jobs . . . . . . . . . . . . . . . :   5
Threshold  . . . . . . . . . . . . . . . . . . . . . :   2
Additional number of jobs  . . . . . . . . . . . . . :   2
Maximum number of jobs . . . . . . . . . . . . . . . :   *NOMAX
Maximum number of uses . . . . . . . . . . . . . . . :   200
Wait for job . . . . . . . . . . . . . . . . . . . . :   *YES
Pool identifier  . . . . . . . . . . . . . . . . . . :     2
Class  . . . . . . . . . . . . . . . . . . . . . . . :     QSYSCLS20
 Library  . . . . . . . . . . . . . . . . . . . . . . :      QSYS
 Number of jobs to use class   . . . . . . . . . . . :      *CALC
Class  . . . . . . . . . . . . . . . . . . . . . . . :     *NONE
 Library  . . . . . . . . . . . . . . . . . . . . . . :
 Number of jobs to use class  . . . . . . . . . . :    *CALC


Press Enter to continue.

F3=Exit   F12=Cancel   F14=Display previous entry
```

Figure 9-17 Display Prestart Job Entry Detail

On this screen, you see that the operating system starts five prestart jobs when subsystem QSYSWRK is started. When you compare this display to the one in Figure 9-14 on page 280, you see that there were 38 additional prestart jobs started during the time that the subsystem has been active.

We recommend that you start the required amount of prestart jobs during the time of starting the subsystem (usually during IPL) because the overhead by that time is less noticeable. Usually it is a good practice to start more prestart jobs than what you actually need. This is because the overhead of having too many prestart jobs is much less than the overhead of dynamically creating the prestart jobs.

Determining the amount of prestart jobs you need

The amount of prestart jobs needed is based on the application needs. If you have an application that uses a large number of database connects and disconnects, you need a lot of prestart jobs. If your application uses a relatively small number of database connects and disconnects, you will suffice with a relatively small amount of these prestart jobs on your system.

The same applies to the number of concurrent users. If there is only a handful of concurrent users, you need fewer prestart jobs than you need when you have hundreds of users on the application at the same time. Usually, the default value shipped with the operating system is not sufficient.

Enter the DSPACTPJ command and observe the output as shown in Figure 9-14 on page 280. You should be able to determine the number of prestart jobs you have currently running and make sure that you start at least that amount of jobs when your subsystem is being started.

Specifying the number of prestart jobs started

To do this, use the Change Prestart Job Entry (CHGPJE) command as shown in Figure 9-18.

```
                      Change Prestart Job Entry (CHGPJE)

Type choices, press Enter.

Subsystem description . . . . . SBSD          > QSYSWRK
  Library . . . . . . . . . . .                   *LIBL
Program . . . . . . . . . . . . PGM           > QSQSRVR
  Library . . . . . . . . . . .                   *LIBL
User profile . . . . . . . . . . USER            *SAME
Start jobs . . . . . . . . . . . STRJOBS         *SAME
Initial number of jobs . . . . . INLJOBS       > XXXXX
Threshold  . . . . . . . . . . . THRESHOLD     > XXXXX
Additional number of jobs  . . . ADLJOBS         YYYYY
Maximum number of jobs . . . . . MAXJOBS         *SAME
Job name . . . . . . . . . . . . JOB             *SAME
Job description  . . . . . . . . JOBD            *SAME
  Library  . . . . . . . . . . .

Maximum number of uses . . . . . MAXUSE          *SAME
Wait for job . . . . . . . . . . WAIT            *SAME
Pool identifier  . . . . . . . . POOLID        > 2
Class:                           CLS
  Class  . . . . . . . . . . . .               > YOURCLASS
    Library  . . . . . . . . . .               >   YOURLIB
  Number of jobs to use class  .                 *SAME
  Class  . . . . . . . . . . . .                 *SAME
    Library  . . . . . . . . . .
  Number of jobs to use class  .                 *SAME

F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 9-18   Change Prestart Job Entry command

Note that you can specify the prestart jobs to use a class different from the one provided with the operating system. The default is to use the class QSYSCLS20.

9.5 Performance tools: A top-down approach

Determine whether your system is running optimally. If not, you need to improve its efficiency by choosing from the many performance tools that are available.

9.5.1 Which performance tool to use

The most productive way to monitor performance is to collect data on a regular basis and track specific key parameters against a standard. A performance problem may occur for a number of reasons, including increased workload, application software change, operating system upgrade, applying program temporary fixes (PTFs), hardware malfunctions, communications degradation, etc. Using this process, you can analyze the performance impact before and after significant changes or events occur.

Many customers constantly collect sample performance data 24 hours a day, seven days per week. This is saved off the system regularly, after key performance parameters are extracted. This can be used as a basis for a meaningful management report, reflecting increasing workload (for example, increased business transactions), and used for capacity planning purposes.

Depending on the type of problem and the depth of measurement, you can use a number of different tools. Table 9-3 lists the tools we chose for the process. Of course, other tools are available, but we found that the tools shown here cover most issues that you are likely to encounter.

Table 9-3 Performance analysis tools

Phase	Data collected by	Data analyzed by
System Analysis	Collection Services	Management Central
WebSphere Application Server Analysis	WebSphere Administration Instance	WebSphere Application Server Resource Analyzer
Java/Application Analysis	DMPJVM	Manually
	Performance Explorer (PEX) in Stats mode	Performance Explorer Stats mode
	Performance Explorer in Trace Mode	PTDV
	Performance Explorer in Trace Mode	Performance Explorer in Profile Mode
	DB Monitor	Manually
	SQL Visual Explain	Manually

9.6 System level tools

System level tools refer to tools capable of measuring such things as CPU usage, memory pool page faulting, disk utilization and so on. There are a great number of these tools on the market, including one supplied by IBM – Management Central.

9.6.1 Management Central

Use Management Central's Collection Services to collect performance data for future analysis by the Performance Tools for iSeries licensed program (5769-PT1) or other performance report applications. If you prefer to view real-time performance data, Management Central provides an easy-to-use graphical interface for monitoring system performance.

Figure 9-19 shows an example of a Management Central CPU monitor.

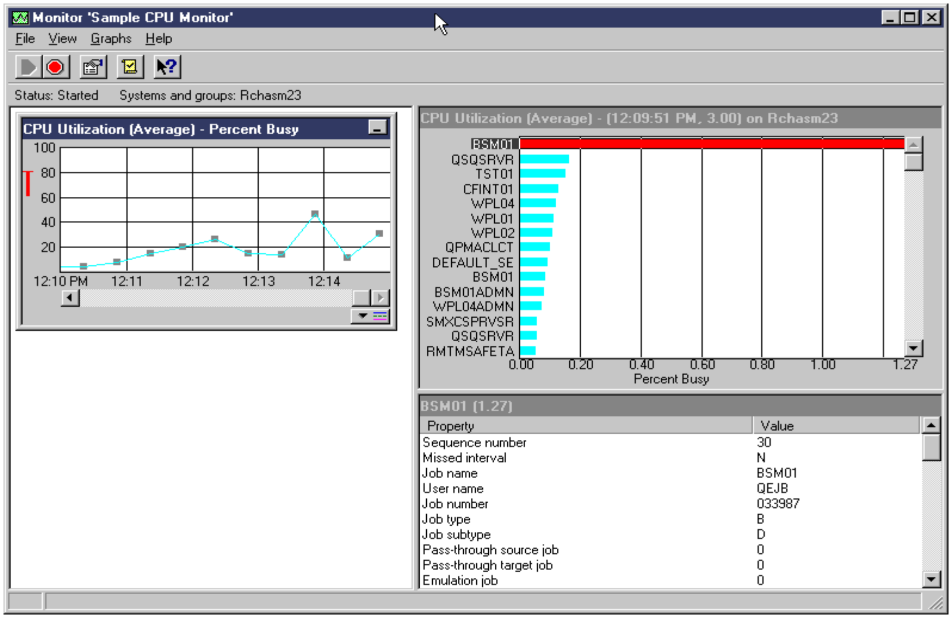


Figure 9-19 Management Central sample view

To collect and store performance data for future analysis, you can perform any of the following tasks:

- Start Collection Services on a single system
- Start Collection Services on system groups
- Start Collection Services automatically

Collection Services collects data that identifies the relative amount of system resource used by different areas of your system. When you collect and analyze this information on a regular basis, you help balance your resources better, which in turn gives you the best performance from your system. You can customize your data collections so you collect only the data you want.

You can use database files generated from collection services with the Performance Tools licensed program (5769-PT1) or other applications to produce performance reports.

9.7 WebSphere Application Server level tools

WebSphere level tools are capable of measuring the internal response time and resources used by the WebSphere Application Server. At present, the best tool available is the WebSphere Resource Analyzer.

9.7.1 WebSphere Resource Analyzer

WebSphere Resource Analyzer is a performance monitor used for WebSphere Application Server *Advanced Edition*.

In WebSphere Application Server Version 3.5, WebSphere Resource Analyzer is a free-of-charge, Technology Preview feature, available for download at:
http://www.ibm.com/software/webservers/appserv/download_ra.html

In WebSphere Application Server version 4.0, WebSphere Resource Analyzer is part of WebSphere Administrative Console, available as one of the options on the menu bar.

The Analyzer retrieves performance data by periodically polling the administrative server. Data is collected continuously and retrieved as needed from within the Analyzer. The level of data to collect is specified by using the WebSphere Administrative Console. Then, the Analyzer's graphical interface is used to retrieve and view the data in a table or chart, or to store the data in a log file.

The WebSphere Resource Analyzer provides a wide range of performance data for two kinds of resources:

- WebSphere resources (for example, enterprise beans and servlets)
- Run-time resources (for example, Java Virtual Machine (JVM) memory, application server thread pools, and database connection pools)

Performance data includes statistical data (such as the response time for each method invocation of an enterprise bean) and load data (such as the average size of a database connection pool during a specified time interval). This data is reported for individual resources and aggregated for multiple resources.

Figure 9-20 shows sample output from the WebSphere Resource Analyzer. Depending on which aspects of performance are being measured, the Analyzer can help you:

- View data in real time, or during specified time intervals. For example, data can be displayed in intervals showing performance during the last minute, the last 5 minutes, the last 10 minutes, and the last 20 minutes.
- View data in chart form, allowing comparisons of one or more statistical values for a given resource on the same chart.
- Record current performance data in a log and replay performance data from previous sessions.
- Compare data for a single resource to an aggregate (group) of resources on a single node.

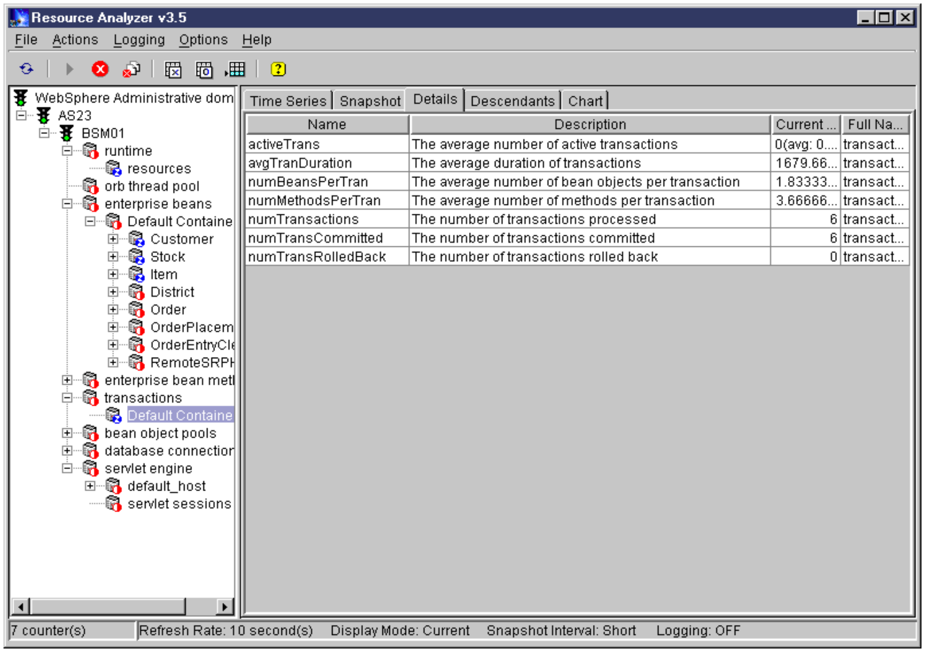


Figure 9-20 WebSphere Resource Analyzer

9.8 Java/application level tools

Application level tools are capable of looking "under the covers" of the running application. A number of tools are available, each with various levels of complexity. For a quick peek at what is going on in your JVM, use the Dump Java Virtual Machine (DMPJVM) command. For more in-depth analysis, use Performance Trace Data Visualizer (PTDV). And for database and SQL optimization, use DB Monitor and Visual Explain.

9.8.1 DMPJVM

Dump Java Virtual Machine (DMPJVM) is a standard OS/400 command. It dumps information about the Java Virtual Machine (JVM) for a specified job. The information is dumped using printer file QSYSPRT. The user data for the QSYSPRT file is DMPJVM. The dump includes formatted information about the classpath, garbage collection, and threads associated with the JVM as follows:

- The classpath
- Garbage collection
- Threads that are associated with the Java Virtual Machine

Figure 9-21 shows the *classpath* information output from the DMPJVM command. The classpath information includes information about the JDK version that was used.

```
Java Virtual Machine Information  034866/QEJB/WPL01
 ....................................................................
 . Classpath                                                        .
 ....................................................................
 java.version=1.2
 /QIBM/ProdData/Java400/jdk12/lib/jdkptf12.zip:/QIBM/ProdData/Java400/jdk12/
 lib/rt.jar:/QIBM/ProdData/Java400/jdk12/lib/i18n.jar:/QIBM/ProdData/Java400
 /ext/IBMmisc.jar:/QIBM/ProdData/Java400/ext/jssl.jar:/QIBM/ProdData/Java400
 /ext/ibmjssl.jar:/QIBM/ProdData/Java400/:/qibm/proddata/http/public/jt400/l
 ib/jt400.zip:/qibm/userdata/webasadv/WPL01:/qibm/userdata/webasadv/WPL01/Or
 derEntryBeans.jar:/qibm/userdata/webasadv/WPL01/nonejb.jar:/QIBM/ProdData/J
 ava400/ext/db2_classes.jar:/QIBM/ProdData/java400/ext/db2_classes.jar:/QIBM
 /ProdData/WebASAdv/lib/wsa400.jar:/QIBM/UserData/WebASAdv/WPL01/properties:
 /QIBM/ProdData/WebASAdv/lib/server/ibmwebas.jar:/QIBM/ProdData/WebASAdv/lib
 /servlet.jar:/QIBM/ProdData/WebASAdv/lib/webtlsrn.jar:/QIBM/ProdData/WebASA
 dv/lib/server/ns.jar:/QIBM/ProdData/WebASAdv/lib/ejs.jar:/QIBM/ProdData/Web
 ASAdv/lib/ujc.jar:/QIBM/ProdData/WebASAdv/lib/server/repository.jar:/QIBM/P
 rodData/WebASAdv/lib/server/admin.jar:/QIBM/ProdData/WebASAdv/lib/was20cm.j
 ar:/QIBM/ProdData/WebASAdv/lib/server/tasks.jar:/QIBM/ProdData/WebASAdv/lib
```

Figure 9-21 DMPJVM output: Classpath detail

Figure 9-22 shows the *garbage collection* information produced by DMPJVM. Using the DMPJVM command a few times over the life of your application can give you information about the frequency of your garbage collections and can help you determine if you have memory leaks.

```
.....................................................................
. Garbage Collection                                                .
.....................................................................
Garbage collector parameters
   Initial size: 32768 K
   Max size: *NOMAX
Current values
   Heap size: 89376 K
   Garbage collections: 30
.....................................................................
```

Figure 9-22 DMPJVM output: Garbage collection detail

Figure 9-23 shows the *thread information* output of the DMPJVM command. You can see the call stack of any threads running as well their current status and any locking situations they may be in.

```
 *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+...
 . Thread information                                               .
 ....................................................................
 Information for 49 thread(s) of 49 thread(s) processed
 Thread:  00000013 Thread-0
   TDE:  B000400006B9D000
   Thread priority: 5
   Thread status: Waiting
   Wait object: com/ibm/ejs/sm/server/ManagedServer
   Thread group: main
   Runnable: java/lang/Thread
  Stack:
    com/ibm/ejs/sm/server/ManagedServer.awaitShutdown()V+13 (ManagedServer.java:1046)
     com/ibm/ejs/sm/server/ManagedServer.main([Ljava/lang/String;)V+109
(ManagedServer.java:141)
  Locks:
     None
  ....................................................................
 Thread:  00000043 Alarm Thread 1
.
.
```

Figure 9-23 DMPJVM output: Information for first thread

The DMPJVM is an excellent tool for a quick overview of your application. If you are required to go deeper, PTDV is a good choice to use.

9.8.2 Performance Explorer (PEX)

Performance Explorer is a trace and profiling tool that ships with OS/400. It has the ability to analyze virtually every aspect of your running code and provide profiling, statistical, and trace information.

PEX is a set of CL commands that come with OS/400. It is used for detailed performance data collection that gives you the ability to analyze virtually every aspect of your running code and acquire profiling, statistical, and trace information for your needs. PEX allows you to collect performance data in three different modes:

- **Statistical**: Identifies applications and IBM programs or modules that consume excessive CPU use or that perform a high number of disk I/O operations. Typically, you use the statistical type to identify programs that should be investigated further as potential performance bottlenecks.
- **Profile**: Identifies high-level (HLL) programs that consume excessive CPU utilization based on source program statement numbers.
- **Trace**: Gathers a historical trace of performance activity generated by one or more jobs on the system. The trace type gathers very specific information about when and in what order events occurred. The trace type collects detailed program, Licensed Internal Code task, OS/400 job, and object reference information. The amount of data gathered by trace is enormous, that's why you would typically use trace after you narrow down your performance problem to a specific program or a job.

To define the mode as well as the details of your data collection, you create a PEX definition. The PEX definition is stored as a database member QAPEXDFN in library QUSRSYS. Then you start PEX using the Start PEX (STRPEX) command. After you finish, you end the PEX collection with the End PEX (ENDPEX) command. PEX performance data is stored in files QAYPExxx in the library QPEXDATA. The ENDPEX command places the collected data into members in each of the files in QPEXDATA.

To check your PEX definitions, use the following command:

```
WRKMBRPDM FILE(QUSRSYS/QAPEXDFN)
```

To check your data collections, use the following command:

```
WRKMBRPDM FILE(QPEXDATA/QAYPExxx)
```

While PEX collects the performance data, it does not include any data analysis tool. You can analyze the data collected with PEX by using any of the following options:

- Using the Print PEX Report (PRTPEXRPT) command, which is a part of Performance Tools for iSeries. PRTPEXRPT takes all the information from the files in QPEXDATA, creates a readable report, and places it into a spooled file.
- Creating your own queries against the files QAYPExxx in library QPEXDATA.
- Using the Performance Trace Data Visualizer (PTDV) tool.

PEX helps to find the causes of performance problems that cannot be identified by using other tools that are used for more general, system-wide performance monitoring. It is designed for application developers who are interested in understanding or improving the performance of their programs. It is also useful for users who are knowledgeable in performance management to help identify and isolate complex performance problems.

You should use PEX after you have tried the other tools. It gathers vast amounts of data that can be analyzed to isolate the factors involved in a performance problem.

For more information on PEX, refer to *Performance Tools for iSeries* on the iSeries Information Center at:
http://as400bks.rochester.ibm.com/pubs/html/as400/v5r1/ic2924/index.htm

9.8.3 Performance Trace Data Visualizer (PTDV)

Performance Trace Data Visualizer is a graphical analysis tool that visualizes Java information from a PEX trace. PTDV is available as a download from the IBM AlphaWorks site at: http://www.alphaworks.ibm.com

PTDV is a Java application that can be used for performance analysis of applications running on iSeries. PTDV works with the performance data collected by PEX trace mode. It allows you to view program flows and obtain such details as:

- CPU time
- Wall clock time
- Number of CPU cycles
- Number of instructions that may be summarized by:
  - Trace
  - Job
  - Thread
  - Procedures

- Number of objects created
- Information about Java locking behavior

With PTDV, you can sort or hide columns, export data, and perform many levels of data summarization.

PTDV ships with an excellent tutorial that describes all the features of the product and takes you through an example trace. We do not repeat the contents of this tutorial here, but we believe a brief overview of some of the features of the products are useful. For detailed information on performing some of the tasks described, refer to the PTDV manual.

Preparing to use PTDV

PTDV relies on event data that is generated by the Java programs being traced. Before you can collect data for PTDV, normally you must make sure that your Java programs are enabled for performance collection. Specify the following parameter in the command line for the JVMs in the WebSphere Administration Console (failure to do this results in no data being shown by PTDV):

```
-Dos400.enbpfrcol=1
```

Remember to turn off performance enablement after you finish using PTDV, because there is extra overhead in collecting performance data.

Collecting data for PTDV

Once your JVMs are prepared for performance collection and you have restarted your WebSphere Servers, you can start the trace. The trace is started using the STRPEX command and ended using the ENDPEX command. Keep the tracing period short because PEX can collect large amounts of data.

Visualizing performance data with PTDV

Here is an example of using PTDV to analyze the result of a trace. When PTDV starts, the trace is downloaded and the main panel appears as shown in Figure 9-24.

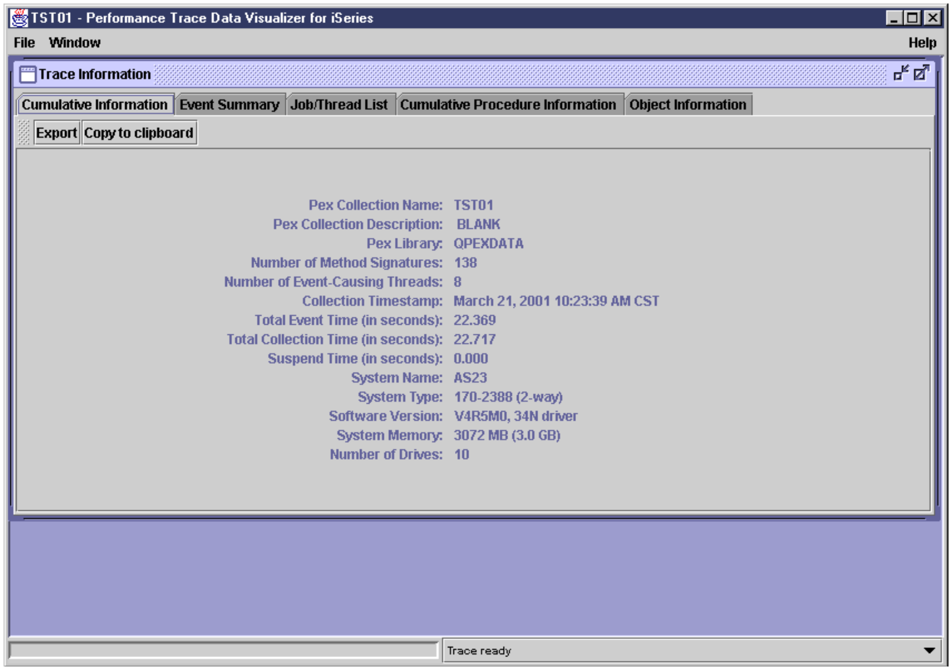


Figure 9-24 Cumulative Information tab: Initial summary

PTDV shows many frames that contain different information. The frames can be identified by the title "Trace Information" in this case. PTDV allows you to move, resize, maximize, and minimize the frames that it shows.

Like most PTDV frames, the Trace Information frame has several tabs across the top. These tabs divide the information into manageable pieces. By clicking the various tabs, different panes appears with information related to the name of the tab you clicked.

For the Trace Information frame, the tabs (panes) are:

- Cumulative Information
- Event Summary
- Job/Thread List
- Cumulative Procedure Information
- Object Information

Cumulative Information

The Cumulative Information tab is shown in Figure 9-24. It shows you basic information about the PEX collection and the system it was collected on.

Event Summary

The Event Summary pane shows the number of events of each type contained in the trace. The events are subdivided according to their PEX types ("Java Events" and "Machine Interface Program Bracketing Events") and subtypes ("Java Object Create/Delete", "Java Entry", and "Java Exit"). The top line contains the total number of events.

Notice that the table has two columns of event counts: "Events in Trace" and "Events Processed". Most of the time, these two columns are identical. However, if you click the Stop button while PTDV is processing events, the numbers in the "Events Processed" column will be lower than the "Events in Trace", because not all events were processed.

Job/Thread List

From the Job/Thread List pane, you can see all of the jobs that were active during your PEX collection, and the threads in those jobs. Figure 9-25 shows an example. The jobs are divided into two categories: "Event-causing jobs" and "Non-event-causing jobs".

The event-causing jobs are the ones you are most likely interested in; they are the ones that caused PEX events. However, it is sometimes useful to look at the non-event-causing jobs to make sure that nothing unexpected was going on with the rest of the system while your program was running. In our sample, however, there are no non-event-causing jobs, because we performed the trace on the application server instance.

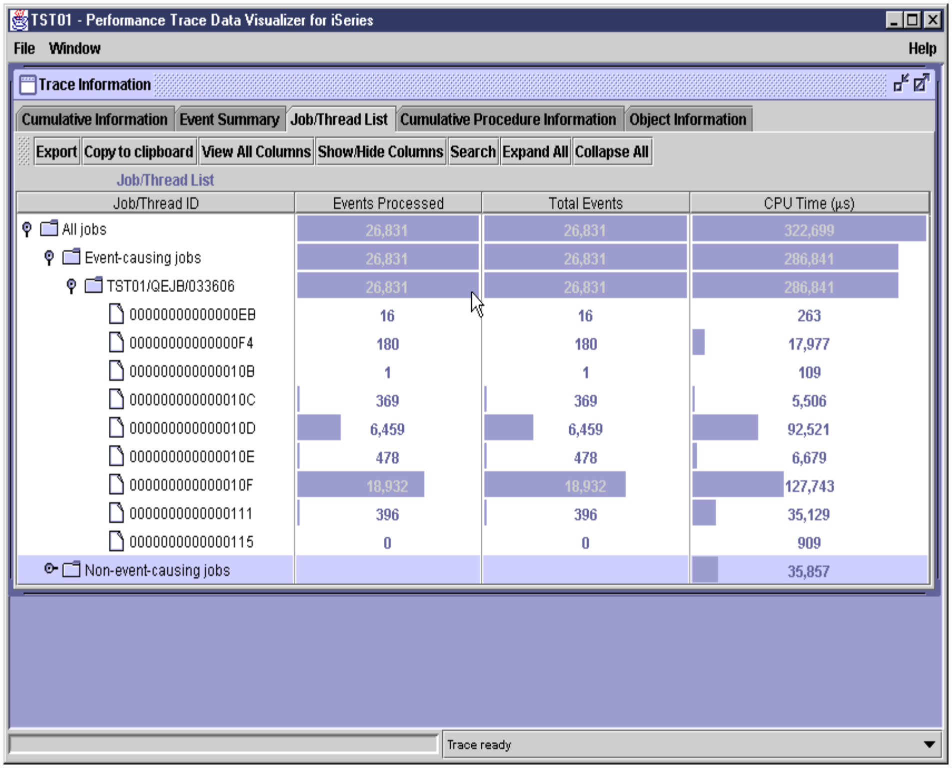


Figure 9-25 Job/Thread List

To the right of the job and thread names, you can see various information about the job or thread. This is typical of most PTDV panes. While the information shown in the columns by default is usually the most useful, other information is available as well. You can click the View All Columns button to quickly display all of the available columns.

You can also right-click a column in the table header, and a menu includes the option to "Hide Column".

If you click Show/Hide Columns at the top of the pane, you see a list of the available columns as shown in Figure 9-26. You can select which of those columns you want to view. This is often the easiest method to show only the information you are interested in.

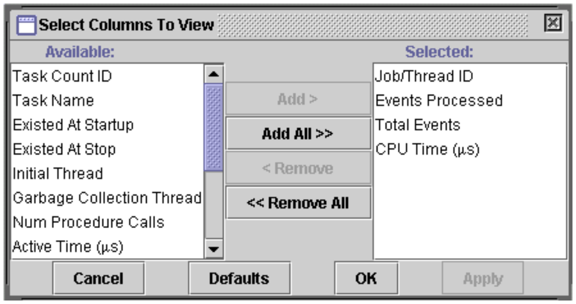


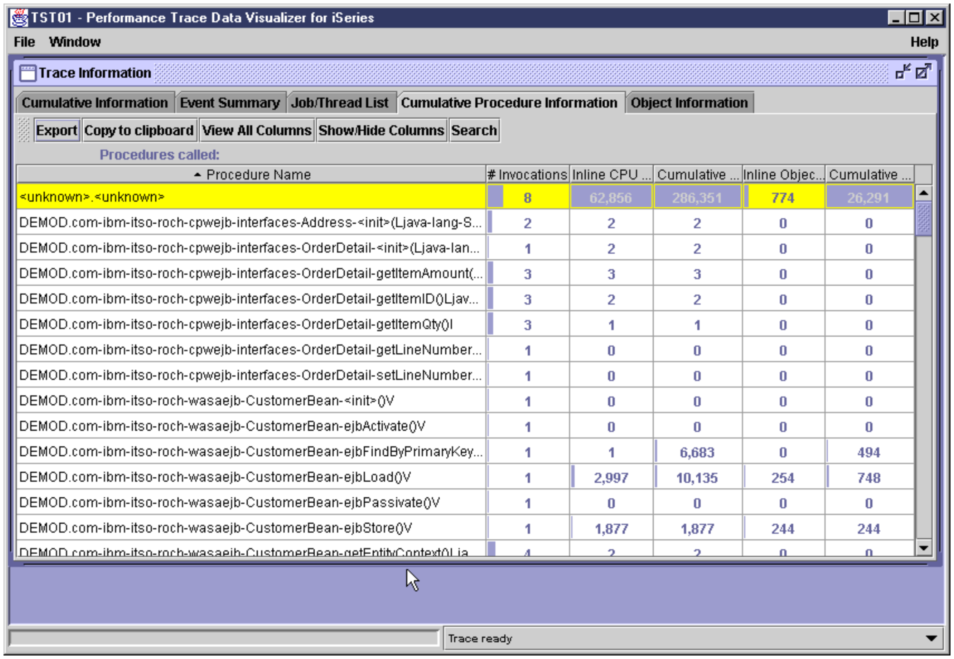
Figure 9-26 Selecting columns

Object Information

The Object Information pane shows you information about the objects created during the trace. This includes the number of objects of each type that were created and the size of those objects.

Cumulative Procedure Information

The Cumulative Procedure Information pane contains summary information about all of the procedures/methods called during the trace, as shown in Figure 9-27.



Procedure Name	# Invocations	Inline CPU ...	Cumulative ...	Inline Objec...	Cumulative ...
<unknown>.<unknown>	8	62,856	286,351	774	26,291
DEMOD.com-ibm-itso-roch-cpwejb-interfaces-Address-<init>(Ljava-lang-S...	2	2	2	0	0
DEMOD.com-ibm-itso-roch-cpwejb-interfaces-OrderDetail-<init>(Ljava-lan...	1	2	2	0	0
DEMOD.com-ibm-itso-roch-cpwejb-interfaces-OrderDetail-getItemAmount(...	3	3	3	0	0
DEMOD.com-ibm-itso-roch-cpwejb-interfaces-OrderDetail-getItemID()Ljav...	3	2	2	0	0
DEMOD.com-ibm-itso-roch-cpwejb-interfaces-OrderDetail-getItemQty()I	3	1	1	0	0
DEMOD.com-ibm-itso-roch-cpwejb-interfaces-OrderDetail-getLineNumber...	1	0	0	0	0
DEMOD.com-ibm-itso-roch-cpwejb-interfaces-OrderDetail-setLineNumber...	1	0	0	0	0
DEMOD.com-ibm-itso-roch-wasaejb-CustomerBean-<init>()V	1	0	0	0	0
DEMOD.com-ibm-itso-roch-wasaejb-CustomerBean-ejbActivate()V	1	0	0	0	0
DEMOD.com-ibm-itso-roch-wasaejb-CustomerBean-ejbFindByPrimaryKey...	1	1	6,683	0	494
DEMOD.com-ibm-itso-roch-wasaejb-CustomerBean-ejbLoad()V	1	2,997	10,135	254	748
DEMOD.com-ibm-itso-roch-wasaejb-CustomerBean-ejbPassivate()V	1	0	0	0	0
DEMOD.com-ibm-itso-roch-wasaejb-CustomerBean-ejbStore()V	1	1,877	1,877	244	244
DEMOD.com-ibm-itso-roch-wasaejb-CustomerBean-getEntityContext()Lja...	4	2	2	0	0

Figure 9-27 Cumulative Procedure Information

The Procedure Name column contains the names of all the methods called. The first part of the name (DEMOD) tells you how the iSeries executed the method. DEMOD means Direct Execution Mode (running a optimized Java program), and JITC means the Just In Time Compiler. Calls to JNI functions may include other modules.

Notice that in several cases, there are "inline" and "cumulative" versions of a column. For example, there is "Inline CPU Time (µs)" and "Cumulative CPU time (µs)". In these cases, *inline* refers to the time spent on this method only, while *cumulative* refers to the time spent on this method and all of the methods called by this method.

You can click any of the column headings to sort the output in that column. This provides a good entry point to determine which methods are the most like performance bottlenecks.

In addition, PTDV provides drill-down capabilities for most columns. Refer to the PTDV tutorial for an extensive description.

9.8.4 DB Monitor

Both Database Monitor for iSeries (DB Monitor) and SQL Visual Explain (see 9.8.5, “SQL Visual Explain” on page 301) are relevant to more than Java programming.

DB Monitor is a tool that can be used to analyze database performance problems after SQL requests have completed running. The DB Monitor tool has been part of OS/400 since V3R6.

DB Monitor is used to gather database and performance data generated when SQL queries are executed. Then using customized SQL programs, the analyst can view, analyze, and determine the most appropriate actions to take in order to generate more efficient SQL queries.

DB Monitor data is most useful if the user has a basic knowledge of iSeries query optimization techniques.

Collecting DB Monitor data

Use the STRDBMON command to start the DB Monitor. DB Monitor collects information on previously started jobs or new jobs started after the monitor collection has begun. Because of the volume of data collected, try to gather data for a specific job only. This makes analysis easier and keeps the DB Monitor file smaller. If this is not possible, collect data on all of the jobs and use queries to select the specific jobs of interest.

It should also be noted that when the DB Monitor is gathering data, a significant amount of CPU utilization (20 to 30 percent) and disk usage may be temporarily required.

Start Database Monitor (STRDBMON) parameters

The following list highlights the STRDBMON parameters and explains how they are used:

- **OUTFILE**: The file name for the results file is required, but the library name is optional. The file is created if it does not exist; it is reused if it exists.
- **OUTMBR**: This parameter defaults to the first member in the file. Specify the "ADD" or "REPLACE" option (the default is "REPLACE"). The "ADD" option causes the new results to append to the end of the file.
- **JOB**: This parameter defaults to the job when you issue the STRDBMON command. You can specify one job or *ALL jobs – no subsetting allowed. Two DB Monitors can collect data on the same job.
- **TYPE**: With this parameter, you can specify the type of data to be collected – *SUMMARY, which is the default option, or *DETAIL. For most cases, *SUMMARY provides all of the necessary analysis data.
- **FRCRCD**: With this parameter, you can specify how often to force monitor records to the results file. For most cases, the default of *CALC is acceptable. You can specify a larger number to reduce the overhead of the DB Monitor; a smaller number increases the overhead.
- **COMMENT**: Using this parameter, you can provide a description of the collection. If the DB Monitor is started using the Start Performance Monitor (STRPFRMON) command, JOB(*ALL) is used for the JOB option and data is placed in the QAPMDBMON file in the QPFRDATA library using the same member name as specified for the STRPFRMON command. Note that you need to use the End Database Monitor (ENDDBMON *ALL) command to end the DB Monitor for all jobs.

End Database Monitor (ENDDBMON) parameters

The ENDDBMON command parameters and their functions are highlighted in the following list:

- **JOB**: You can specify a particular job name or end all jobs (*ALL). If a particular job name is used, DB Monitor only ends the monitor that was started with that same job name. It is possible to end one monitor on a job and still have another monitor collecting on that same job.
- **COMMENT**: This parameter allows the user to provide a description of the data collection.

Identifying and tuning problem areas

There are many different methods to identify problems and tune the troublesome database statements. One of the most common methods is to identify the most dominating, time-consuming queries and work on each of them individually. Another method is to leverage global information and to use this information to look for indexes that definitely need to be created.

Initially, concentrate on repetitious non-reusable ODPs, table scans, and long index builds. Also, look for repetitious short-running queries that are not optimized well. Joins and sorts can be more difficult to analyze. If joins and sorts are accounting for a significant portion of run time, they need to be addressed as well.

Fine tuning smaller problems should be done after large problems have been addressed. Often, adding indexes can be used to address many performance problem areas.

9.8.5 SQL Visual Explain

SQL Visual Explain is a tool used to visualize the optimization method selected by the DB2 engine when a query is performed. Visual Explain provides a graphical representation and makes it easier for the analyst to determine the actions to take (such as creating a new index).

Visual Explain can be used in the following ways:

- On SQL statements stored in database performance monitor data
- Detailed monitor data (STRDBMON or Detailed SQL Performance Monitor)
- Monitor data has to be collected on V4R5 or later to guarantee that all of the information is available for drawing the picture (can be another system)
- Via the "Recent Performance Monitor" task in the SQL Script Center
- On SQL statements in the SQL Script Center

Running Visual Explain

Running Visual Explain from Run SQL Scripts is sometimes called running Visual Explain *proactively*. That's because you can run the query, make changes, and then run it again to see how the changes affected the implementation. However, if your query is long running, you may want to collect the data using a detailed SQL performance monitor and analyze it later using Visual Explain. You can explain a statement without running it, but the data gathered is based on estimates made by the Query Optimizer.

Running Visual Explain reactively

Running Visual Explain using detailed performance monitor data is sometimes referred to as running Visual Explain *reactively*, because the query that is explained has already run. However, if your query is long running, this may be your best option. In addition, the data collected by your detailed performance monitor can be more accurate than explaining a query from Run SQL scripts without actually running it.

1. In the Operations Navigator window, expand **your server-> Database-> SQL Performance Monitors**.
2. Right-click the detailed SQL performance monitor you want to use and select **List Explainable Statements**.
3. In the SQL statements monitored list, select the SQL statement you want to visually explain.
4. Click **Visual Explain** and the SQL statement is displayed, as a graph, in the Visual Explain window.

Figure 9-28 shows an example of the output from Visual Explain.

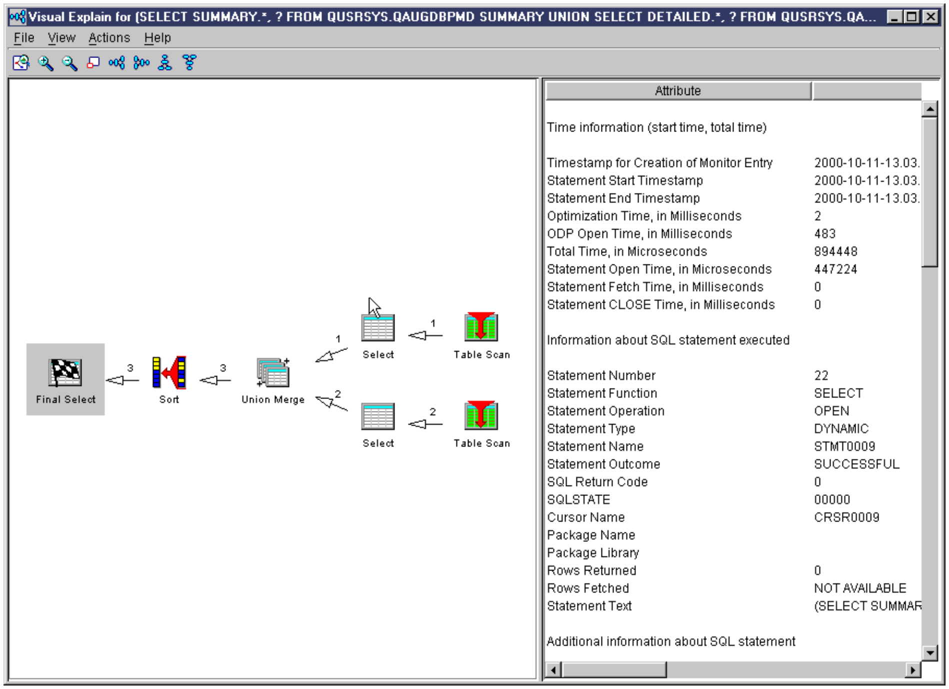


Figure 9-28 Example of output from Visual Explain

Creating an SQL performance monitor

Creating a new SQL performance monitor produces a new instance of a monitor on your system. You can have multiple instances of monitors running on you system at one time. However, there can only be one monitor instance monitoring all jobs. When collecting information for all jobs, the monitor collects on previously started jobs or new jobs started after the monitor is created. However, when you end a performance monitor, the instance of the monitor is terminated and cannot continue, where a paused monitor can be restarted. To create an SQL performance monitor, follow these steps:

1. In the Operations Navigator window, expand **your server-> Database**.
2. Right-click **SQL Performance Monitor** and select **New**.
3. Select **Summary** or **Detailed**.
4. Specify the name you want to give the monitor in the Name field.
5. Specify the library in which you want to save the information that the monitor gathers in the Library field.
6. If you want to specify the maximum amount of system memory that the monitor is allowed to consume, specify the size in MB in the Storage (MB) field.
7. Click the **Monitored Jobs** tab.
8. If you want to monitor all of the available jobs and another monitor is not monitoring all jobs, select **All**. Only one monitor can monitor all jobs at a time.
9. If you want to monitor only certain jobs, select a job that you want to monitor in the Available jobs list and click **Select**. Repeat this step for each job that you want to monitor. Individual jobs can be monitored by only one active monitor at a time.
10. If you selected a job that you do not want to monitor or if a job you selected is already being monitored, select that job in the Selected jobs list and click **Remove**.
11. If you selected Summary monitor, click the **Data to Collect** tab.
12. On the Data to Collect tab, select the types of data that you want to collect. If you want to collect all types of data, click **Select All**.
13. Click **OK**. The performance monitor starts and runs until it is ended or paused.

Running Visual Explain from Run SQL Scripts

To run Visual Explain from Run SQL Scripts, you need to follow these steps:

1. In the Operations Navigator window, expand **your server-> Database**.
2. Right-click **Database** and select **Run SQL Scripts**.

3. If you have an existing SQL statement that you want to use, copy that statement and paste it into the Run SQL Scripts window.
4. If you do not have an existing SQL statement, type a statement in the Run SQL Scripts window.
5. Highlight the text of your statement and select **Explain** (or **Run and Explain**) from the Visual Explain menu. Or you can use the corresponding toolbar buttons.

If you choose *Explain* from the Visual Explain menu, the system only displays your query diagram without actually running it. If you choose *Run and Explain* from the Visual Explain menu, your query is run by the system before it is displayed. The results of the query (if any) are displayed in the Run SQL Scripts window. Running and displaying your query could take a significant amount of time, but the information displayed is more complete and accurate.

9.9 Communication performance considerations

Efficient communication performance is an essential component of total system performance, since in a multi-tier environment, the components are all interconnected. Communications can be broadly categorized into three groups:

- **Wide area networks (WANs)**

  WANs usually rely on relatively slow connections (9600 to 56,000 bits per second) and may be either dial-up or point-to-point. Many access the Internet using this type of connection, and adequate performance is available using the OneWorld Xe Web browser interface.

- **Local area networks (LANs)**

  LANs are often used in locations where the client system is located in close proximity to the computer system or systems that require relatively high speed access. LANs make it possible to share resources such as printers and file servers. Typical LANs operate from as low as 1.5 Mb per second (such as cable modems) to 1 Gb per second. Most common today is 100 Mb per second.

  LAN hubs often employ high-speed switches or routers to maximize throughput. These switches or routers can usually support attachment connections of different speed and often have firewall software running for greater security.

- **High speed links**

  High speed links (> 1 Gb per second) are often used for system interconnection and connection of system to shared resources such as large disk enclosures. These often use fibre optics.

Although there are many protocols and standards in use today, we focus on the two most common standards, Ethernet and TCP/IP.

9.9.1 High-speed interconnection between systems

Figure 9-29 shows a typical customer environment where the Web server is located on a different system to the enterprise server. In this situation, it is best to have a high speed communications link between the two systems. On iSeries, this can be accomplished through the use a 1 Gb per second adapters. Some older iSeries server do not support the 1 Gb per second adapters, in which case, it may be possible to use multiple 100 Mb per second adapters.

In large OneWorld Xe environments, an additional communications link can be added to service JDBC database requests from the JAS servlets to the DB2 UDB for iSeries database. Only about 5 to 10% of total requests are JDBC requests, so this option should only be considered if the link between the two systems becomes saturated (for example, greater than 30 to 40% line utilization).

The Performance Tools System Report and the Performance Tools component Report show line utilization and adapter utilization.

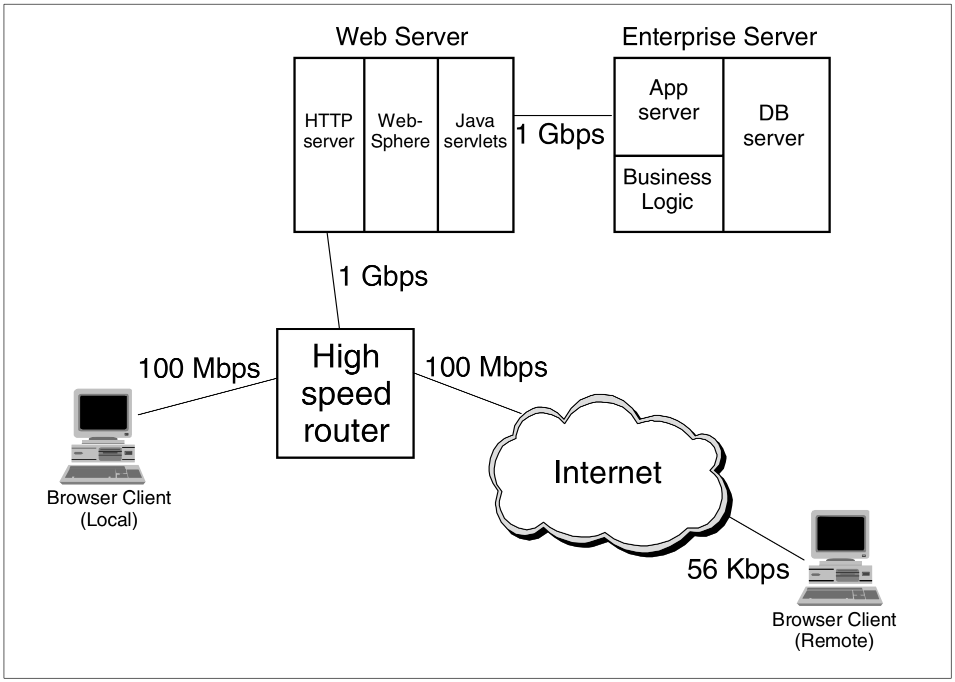


Figure 9-29 A typical customer communications network with high-speed connections

Table 9-4 shows the values that should be used for optimal performance when setting up TCP/IP over Ethernet communications lines on iSeries for 1 Gb per second lines and 100 Mb per second lines.

Table 9-4 Tuning TCP/IP over Ethernet lines on iSeries

Line speed	**Max Transmission Unit**	**TCP/IP buffer size**
100 Mb per second	1492	65536
1 Gb per second	8996	1024000

The Maximum Transmission Unit (MTU) size specifies the maximum size (in bytes) of the IP datagram that can be transmitted through a particular route. This value can be changed using the OS/400 Change TCP/IP Route (CHGTCPRTE) command.

The initial TCP/IP Buffer size can be changed using the OS/400 Change TCP/IP Attribute (CHGTCPA) command.

PC clients should be connected through a different network to the network connecting system to system. Where possible use a point-to-point connection between systems. A back-to-back Ethernet cable can be used for this purpose. iSeries communication adapters support this type of connection, which allows full duplex communications, and no delay due to routers or hubs.

PC clients can be connected via an Ethernet hub or router, but this network should be monitored to make sure that its does not become overloaded. As a guide, the network should not be more than 30% utilized, to avoid excessive delays due to queueing. Use additional iSeries adapters and hubs if this guideline is regularly exceeded.

9.10 PC performance implications

The OneWorld Xe browser interface should generally run very efficiently on most modern PCs. However, some newer browsers such as Netscape 6.01 are more resource intensive and require faster CPUs greater than 500 Mhz. Also, Microsoft has introduced later versions of their operation systems, such as Windows 2000 and Windows XP, which use more resources.

Prior to implementation, test various PC hardware and software configurations, with recommendations being made for the minimum hardware requirements to support the proposed software combinations.

During implementation and system testing, PC resources can be monitored using the standard tools provided by Microsoft. For example, Windows NT ships with a monitor called *Windows NT Task Manager*. One option is performance monitoring, showing CPU and memory usage. Use this data to make sure that your proposed PC configuration provides acceptable performance. Make sure that you take into account other tasks that may be concurrently run on the PC and include this in your PC sizing.

9.11 Optimizing the Java garbage collector

Java garbage collection can consume considerable CPU resource. Therefore, it is important to understand how, by changing various parameters and system resources, this CPU can be minimized, without causing any degradation in performance.

When a Java program is run, the Java Virtual Machine (JVM) uses a storage mechanism called a *heap* to store objects. When objects are instantiated, they are allocated storage in the heap. As a program runs it uses more and more heap storage as more objects are created. The rate of object creation is application dependent.

When an object is no longer required, it is no longer referenced. The space for this object is still allocated until the garbage collector runs and compresses the heap, removing all objects that are no longer referenced. If the garbage collector wakes up too frequently and has to work hard, it can use excessive CPU resource. By balancing the initial heap allocation and total available memory, it is possible to significantly reduce garbage collector overhead.

9.11.1 Tuning methodology

Probably the most important factor in tuning the garbage collector is the *initial heap size parameter*. In OneWorld Xe, Java servlets run in each WebSphere Application Server. Depending on the number of concurrent users, there may be multiple applications servers. For OS/400 V5R1 and WebSphere 3.5.4, each WebSphere Application Server can handle at least 500 users.

The initial heap size controls how often the garbage collector wakes up and compresses heap storage. If the initial heap size is set too low, the garbage collector wakes frequently and has to work very hard to keep the heap very small. This constant processing often consumes excessive CPU.

If the initial heap size is set too large, memory may be wasted, and in a worst case situation, this may lead to excessive disk paging. Therefore, the goal is to set the initial heap size to a value that is close to its steady state value where the creation of objects and their disposal are in balance.

There is a trade-off between memory and CPU. It may be preferable to buy more memory to reduce CPU, since memory usually costs less than a CPU upgrade.

The next section shows you how to obtain the current heap size and use this value to set the initial heap size.

9.11.2 Tuning the garbage collector

The easiest way to monitor heap storage is by using the DMPJVM command as explained in 9.8.1, “DMPJVM” on page 290. Figure 9-22 on page 291 shows an example of the initial heap size at 32 MB and the current heap size at approximately 89 MB. This JVM is not well-tuned.

The initial heap size should be similar in size to the current heap size, so a suggested initial heap size, in this case, would be 80 MB. Run the Performance Monitor before and after these changes and see if the CPU reduces considerably. Before you make this change, make sure you have enough main storage on the system and in the pool where the JVMs run, since you are effectively increasing memory requirements by about 50 to 60 MB. If you have insufficient memory, you may experience an increase in paging that would have a detrimental effect on the system.

The actual CPU used by the garbage collector can be obtained using PEX Trace collecting trace points using the TPROF definition and using the Print PEX Report (PTRPEXRPT) with *PROFILE option. The *PROFILE option normally allows sampling down to 100 mS while the PEX Trace collection point allows sampling down to 1 mS.

First you must create a PEX Trace definition to capture the TPROF data:

```
ADDPEXDFN DFN(TPROF) TYPE(*TRACE) JOB(*ALL) MAXSTG(4000000) INTERVAL(1)
TRCTYPE(*SLTEVT) SLTEVT(*YES) BASEVT((*PMCO))
```

Then run the trace using the STRPEX and ENDPEX commands.

Print the report with the PRTPEXRPT command using the following options (where TRACE01 is the name of the member that holds the trace data):

```
PRTPEXRPT MBR(TRACE01) TYPE(*PROFILE) PROFILEOPT(*SAMPLECOUNT *PROCEDURE)
```

Look for procedures with names that begin with JAVAGC.

Figure 9-30 shows a sample page from a Performance Explorer report. It highlights the garbage collector procedures. The Hit % column approximates the CPU percentage used (in fact, it shows the task that was running in the CPU at the time the 1mS counter expired). By increasing the initial heap size, it may be possible to reduce the CPU consumed by the garbage collector.

```
                              Performance Explorer Report                                 11/05/00  8:06:18
                               Profile Information                                             Page   48
Library . . : JDEWSTUNE
Member. . . : TPROFRC001
Description : Sample TPROF report to for Java Garbage Cololector
Histogram Hit  Hit  Cum   Start         Map   Stmt    Name
          Cnt  %    %     Addr          Flag  Nbr

      **  288 7.7  7.7 FFFFFFFFC6043948 ++   000000  JITGLU2/#jitglu2
      **  217 5.8 13.5 FFFFFFFFB3D95C94 ++   000135  JAVAGC/setObjectToMarkColor__6JavaGCFP10JavaObjecti
       *  194 5.2 18.7 FFFFFFFFB3D99A5C ++   0010A7  JAVAGC/sweepReuseSegment__27JavaThreadSystemGCCollector
       *  109 2.9 21.6 FFFFFFFFB3D95F94 ++   0001F5  JAVAGC/markGrayCollector__6JavaGCFP10JavaObject
           82 2.2 23.8 21B139F65A7EA7D0 ==        0  JAVA400_JD DEMOD/java-io-BufferedInputStream-read()I
           79 2.1 25.9 FFFFFFFFFE9B1AC8 ++   000000  JVALOCK/#jvalock
           49 1.3 27.2 FFFFFFFFFEA560B0 ++   000088  JAVADEEP/createjavaxarraybla
           42 1.1 28.3 28D2ECD1AA7A8EE0 ==        0  JAVA400_JD DEMOD/java-util-BitSet-get(I)Z
           39 1.0 29.4 FFFFFFFFB3DA2798 ++   00005E  JVANWOBJ/NewObjectSlow__13JavaNewObjectFP9JavaClassUlT2i
           36 1.0 31.3 FFFFFFFFFEA55FE0 ++   000054  JAVADEEP/javaxresolveinterfacebla
           34 0.9 32.2 D84F6C30E30C34B4 ==   000000  JITC/com-jdedwards-ifc-util-Hashtable-tableIndex
           34 0.9 33.2 E76B35D5790ED760 ==   000000  JITC/com-ibm-as400-access-DBStoragePool-getUnusedStorage
           33 0.9 34.0 FFFFFFFFFF0050D0 ++   000034  LLGLUE/_llglue
           33 0.9 34.9 28D2ECD1AA3C3780 ==        0  JAVA400_JD GLUEMOD/_UPCALL(_2)
           33 0.9 35.8 FFFFFFFFFEA56694 ++   000201  JAVADEEP/javaxloadexception
           31 0.8 36.6 FFFFFFFFB3DA2630 ++   000004  JVANWOBJ/NewObject128__13JavaNewObjectFP9JavaClass
           27 0.7 37.3 E0DF2BB9BE0E8434 ==   000000  JITC/com-jdedwards-ifc-util-CompactInputStream
           27 0.7 38.1 FFFFFFFFFF8019B0 ++   00006C  CFPLBLA/bla_md1_Pulsar
           27 0.7 38.8 FFFFFFFFC47B8180 ++   00046C  JAVANEW/getBucketOfBlocksOutlineProb__7JavaNewFUlT1P18
           26 0.7 39.5 FFFFFFFFFEA55E90 ++   000000  JAVADEEP/checkjavaxcastbla
```

*Figure 9-30 Sample PEX *Profile report showing resources consumed by the Java garbage collector*

The initial heap size is set using the WebSphere administrative console. Use the general command line. For example, use the format `-Xms512m` to set the heap size to 512 MB.

9.12 Scaling the OneWorld Xe environment

Scaling is the process of increasing the amount of work done per unit time. Usually it is accomplished by adding more users or more batch jobs. When this is done, more hardware resources are usually required. If sufficient resources are not available on one system, more systems may have to be added. This is known as *horizontal scaling*. In a OneWorld Xe environment, you can easily achieve this through the exploitation of the n-tier architecture.

In some situations, it may be necessary to increase the number of WebSphere Application Servers on a single system to overcome contention within an application. This is known as *vertical scaling*. This type of scaling is most often seen on medium to large installations where the number of concurrent users exceeds 1000.

Sometimes, there is contention on specific resources such as database indexes or database records. These must be resolved. This is a software architectural limitation. Generally adding more hardware of faster hardware only temporarily fixes this problem. Usually it requires an application change.

9.12.1 Record locks and index seizes

It is quite common in applications to encounter record locks and index seizes when running with large numbers of concurrent users, or if a particular batch job is locking records for update for extended periods of time.

In actual fact, OneWorld is well engineered and seldom suffers from this problem. However, sometimes when OneWorld is customized by third-party programmers, if care is not taken, these problems can occur. Sometimes it is years after the modifications were made, since the symptoms may not manifest themselves until high numbers of transactions are processed.

Seizes usually occur when a column in a row is updated, and this column is the key in another view. By definition, DB2 maintains that all indexes must be current (assuming immediate maintenance of indexes). Therefore, when a row is added or changed if any indexes need to be changed, the index is seized and all users are momentarily unable to access the index. The updating process occurs in microcode and executes at very high priority, so most seizes occur for very short periods of time, and do not adversely affect operations. But seizes also depend on the number of indexes built over a table and the frequency of changes to key columns.

Locks, by comparison, happen at the table level or the row level. They occur when an applications locks a row for update or when an application needs exclusive control of an object, for example saving a table.

Seizes and locks result in excessive wait time, which normally is close to zero, but can rapidly increase to become a major factor in response time.

If you suspect that either seizes or locks are causing problems on your system, it is possible to run a trace that can identify those objects and jobs causing the problem, so that remedial action can be taken. This usually involves changing the application code or scheduling jobs to run at a different time.

Prior to V5R1, there was a trace option in the STRPFRMON command and an option to print the lock report, using the GO PERFORM menu and then option 3, Print Performance Reports.

In V5R1, the concept of *collection services* in Management Central was introduced. This collected sample data only. The trace information and the Seize/Lock Report are performed using CL commands.

Use the Start Performance Trace (STRPFRTRC), End Performance Trace (ENDPFRTRC), and Print Lock Report (PRTLCKRPT) commands to show you which objects are being locked (held) and which jobs/tasks are locking (holding) the objects.

Once you identify the objects that are being locked or seized and the jobs or program which are holding the objects, an action plan can be developed to remedy the situation.

9.12.2 Additional WebSphere Application Servers

One way to increase throughput of the Web server is to add instances of the WebSphere Application Server. For each WebSphere Application Server, an additional HTTP Server is added, as well as an additional Java application server, to run the servlets. This gives more than one path for an application to flow through the Web server.

Each HTTP Server would have its own IP address or port number. Load balancing is required to ensure that the workloads are evenly spread across HTTP Servers. This can be done by configuring the servlet that handles redirection.

This works in the following manner. Each browser initially connects with a particular IP address and say port 1025. Other HTTP Servers listen on other ports, for example 1026 to 1030 if there is a total of six HTTP Servers. When the request is made, the redirector servlet changes the URL to use a different port, and resubmits the HTTP request. For the duration of the connection, this browser would use the updated port number (contained in all subsequent URL requests).

You can find more details in the *OneWorld Installation Guide* on how to configure multiple WebSphere Application Servers and the redirector servlet.

This approach can enhance performance. We found through laboratory testing that there is an upper limit to the number of concurrent users that can connect through one WebSphere Application Server instance when running the OneWorld servlets. This is platform specific and depends on the transaction type mix. On the iSeries server, this is approximately 500 users, using WebSphere 3.5.4 and V5R1.

9.12.3 Additional enterprise application servers

For very large numbers of concurrent users, typically greater than 3000, it is possible to create multiple instances of the OneWorld application server. This increases parallelism in the application server and removes some minor contention points. This is similar to creating another application server for another company or for a test version. But in this case, they are running the same company in parallel.

There is a requirement that if you run multiple instances of the application server then you must run at least the same number (or more) of instances of the WebSphere Application Server.

You can find more details on installing multiple enterprise application servers in the *OneWorld Installation Guide*.

A

Collaborative solutions

This appendix provides overview descriptions of four J.D. Edwards solution areas that, along with OneWorld Xe, provide key functionality in the J.D. Edwards collaborative product line. Through its own engineering and development as well as strategic acquisition, J.D. Edwards has assembled an integrated software product line that offers its customers a full range of agile collaborative solutions for the connected economy. The solutions featured in this appendix are:

- Advanced Planning
- Collaborative CRM
- Advance Order Configurator
- Enterprise Content Manager

The information presented in this appendix was gathered from J.D. Edwards press releases and Web pages accessed through J.D. Edwards public Web site.

For information about other solutions and specific applications, visit the J.D. Edwards Web site at: `http://www.jdedwards.com`

Advanced Planning solutions

With J.D. Edwards Advanced Planning, you can link with your partners, suppliers, and customers to coordinate shared demand, production, and distribution plans. Advanced Planning produces plans that simultaneously satisfy multiple constraints. Any resource can be a constraint – materials, machines, tools, skilled labor, and time. With constraint-based planning, you gain feasible plans that realistically balance the load across constrained resources.

Advanced Planning also enables you to optimize performance based on the goals you define. It allows for weighted trade-offs among conflicting business objectives, such as customer service and inventory investment, or due date performance and throughput maximization.

The J.D. Edwards Advanced Planning product group is comprised of five key components:

- Demand Planning
- Order Promising
- Production and Distribution Planning
- Production Scheduling
- Strategic Network Optimization

Collaborative CRM

J.D. Edwards collaborative commerce Customer Relationship Management (CRM) solution is a customer-oriented, B2B solution for optimizing planning, marketing, sales, fulfillment, delivery, and service. It recognizes that billing and delivery are every bit as important as sales force automation in terms of the overall customer experience.

With the acquisition of Charlotte, North Carolina-based firm YOUcentric in the Fall of 2001 J.D. Edwards rounded out its CRM solution, which now includes:

- Sales Force Automation
- Marketing Automation
- Partner Relationship Management
- Customer Self-Service
- Customer Service and Warranty Management
- sophisticated Sales Configuration
- Contact Center

Adding to the Customer Self Service, Service Warranty Management, and storefront integration already in place, the integrated CRM solution that results from this acquisition enhances J.D. Edward's integrated ERP and APS solutions. This provides customers with a comprehensive, robust, and powerful collaborative-commerce solution.

Advanced Order Configurator

Built to accommodate a to-order/postponement manufacturing environment, this solution allows you to visually configure a product to the customer's specifications in real time. When a customer requests a variation to a product, you can enter the information and generate a visual representation of the newly custom-configured product as well as the supporting technical documents the manufacturing department needs.

If the customer's request is an invalid configuration, you are notified immediately, avoiding bottlenecks and production delays. In addition, you can:

- Deploy product knowledge in the field, on the fly via a laptop
- Reduce quote and sales cycle times
- View results instantaneously
- Eliminate costly configuration errors
- Accelerate customer communication

The Advanced Order Configurator application is also available in a fully Web-deployed version referred to as Advanced Web Order Configurator (AWOC). This suite of applications also includes:

- The Advanced Order Configurator Quote for in-house sales order entry
- The Advanced Order Configurator Pro for AutoCAD integration

All of these applications can deploy the same product catalogs.

Enterprise Content Manager

Enterprise Content Manager lets customers create a corporate knowledge repository from unstructured mission critical data, such and policies, ISO procedures, product data sheets, memos, presentation, and spreadsheets, to encourage collaboration across geographic, department, and source boundaries. Repurposing the content in the data repository enables people to work more efficiently. Users can write the content once and then use it an unlimited number of times. This “single source” concept – embedded in Enterprise Content

Manager lets customers develop content faster by building on the information they already have. Combined with Enterprise Content Manager's faster to language methodology, customers can significantly save time and money when it comes to translating content into different languages.

Combined with J.D. Edwards Customizable Content for OneWorld Xe, Enterprise Content Manager gives customers access to over 70 user guides together with end-user training material for reuse, enabling users to produce customized training materials and procedural documentation for specific organizational needs.

Additional documentation

You can find additional documentation, including technical requirements, deployment options, information on integration, and end-user guides, in the J.D. Edwards Knowledge Garden under Product Publications.

For information regarding Knowledge Garden access, visit the J.D. Edwards online support site at:
`http://www.jdedwards.com/public/0,1921,0%257E533%257E,00.html`

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 319.

- *High Availability on the AS/400 System: A System Managers Guide*, REDP0111
- *J.D. Edwards OneWorld Implementation for AS/400*, SG24-5195
- *Introduction to Storage Area Network (SAN)*, SG24-5470
- *AS/400 Disk Storage Topic and Tools*, SG24-5693
- *Clustering and IASPs for Higher Availability on the IBM @server iSeries Server*, SG24-5194
- *Storage Consolidation in SAN Environments*, SG24-5987
- *Consolidating Windows 2000 Servers in iSeries: An Implementation Guide for the IBM Integrated xSeries Server for iSeries*, SG24-6056
- *Implementing an Open IBM SAN*, SG24-6116
- *The IBM SAN Survival Guide*, SG24-6143
- *Java and WebSphere Performance on IBM @server iSeries Server*s, SG24-6256
- *Introduction to SAN Distance Solutions*, SG24-6408

Other resources

These publications are also relevant as further information sources:

- *OS/400 Work Management (V4R5)*, SC41-5306
- *OneWorld Installation Guide (AS/400 Systems) Xe*
- *OneWorld Installation Reference Guide (All Systems)*
- *OneWorld Xe Workstation and Server Administration Guide*
- *OneWorld Service Pack Installation Guide (iSeries Server)*

Referenced Web sites

These Web sites are also relevant as further information sources:

- IBM Workload Estimator for iSeries:
http://www.as400service.ibm.com/servlet/EstimatorServlet
- iSeries Performance Capabilities Reference:
http://ca-web.rchland.ibm.com/perform/perfguideup/v5r11perfguide/v5r11perfguide.pdf
- J.D. Edwards Knowledge Garden: http://www.jdedwards.com
- J.D. Edwards online support site:
http://www.jdedwards.com/public/0,1921,0%257E533%257E,00.html
- Microsoft's Hardware Compatibility List (HCL) for Windows NT and Windows 2000:
http://support.microsoft.com/default.aspx?scid=kb;EN-US;q131303
- Informational Authorized Problem Analysis Reports (APARs):
http://www.ibm.com/servers/eserver/iseries/service/erp/jdesupport.htm
- PM/400: http://www.ibm.com/servers/eserver/iseries/pm400/
- Electronic Service Agent public site:
http://publib.boulder.ibm.com/as400_sd/sdsadoc.html
- Tech Studio article on Electronic Support Access:
http://www.iseries.ibm.com/tstudio/planning/esa/esa.htm
- Online Disk Arms Calculator:
http://www-1.ibm.com/servers/eserver/iseries/perfmgmt/odac.htm
- IBM/J.D. Edwards Alliance: http://www.ibm.com/erp/jdedwards
- ICC Alliance Web site: http://www.ibm.com/erp/jdedwards
- Sizing Web site: http://www.ibm.com/erp/sizing
- iSeries support for J.D. Edwards Web site:
http://www.ibm.com/servers/eserver/iseries/service/erp/jdesupport.htm
- Client Access Web site: http://www.as400.ibm.com/clientaccess
- iSeries InfoCenter:
http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html
- SAN technology on the iSeries: http://www.ibm.com/san
- Lakeview Technology: http://www.lakeviewtech.com
- DataMirror Corporation: http://www.datamirror.com

- Vision Solutions: http://www.visionsolutions.com
- IBM iSeries high availability and clustering Web site: http://www-1.ibm.com/servers/eserver/iseries/ha/
- IBM AlphaWorks site: http://www.alphaworks.ibm.com

How to get IBM Redbooks

Search for additional Redbooks or Redpieces, view, download, or order hardcopy from the Redbooks Web site:

ibm.com/redbooks

Also download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become Redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Acronyms

ACL Access control lists

AIO All in One

ASP Active Server Pages

ASP Auxilliary storage pool

AWOC Advanced Web Order Configurator

B2B Business-to-business

CNC Configurable Network Computing

CPU Central processing unit

CPW Commercial Processing Workload

CRG Cluster Resource Group

CRM Customer Relationship Management

CRP Conference Room Pilot

EIP Enterprise Integrated Portal

EJB Enterprise Java Beans

ERP Enterprise Resource Planning

ESS Enterprise Storage Server

EVN Event Notification

FTP File Transfer Protocol

GATS Global Advanced Technology Services

HA High availability

HBA Host Bus Adapter

HLL High-level language

HSL High Speed Link

IEO Interoperability Events Observer

IOP Input/output processor

JAS Java Application Server

JDBC Java Database Connection

JIT Just In Time

JSP JavaServer Pages

JVM Java Virtual Machine

KMBI Knowledge Management/Business Intelligence

LAN Local area networks

LIC Licensed Internal Code

LUN Logical Unit Number

LVD Low Voltage Differential

NER Named Event Rules

OCM Object Configuration Manager

PTDV Performance Trace Data Visualizer

PTF Program temporary fix

RPQ Request Per Quotation

SAN Storage Area Network

SAR Software Action Requests

SAW Server Administration Workbench

TSE Terminal Server Edition

UBE Universal Batch Engine

V3T Virtual 3-Tier

WAN Wide area network

WCBT Work Control Block Table

WCBTE Work Control Block Table Entry

WWAT Worldwide Advanced Technologies

XBP eXtended Business Process

XPI eXtended Process Integration

XPIe Enterprise XPI

XPIx Extended XPI

Index

Symbols
.jar files 3
"J" environment 9

Numerics
5722-JC1 8
5733-AS3 18
5733-WA3 18
5769-PT1 287

A
access control lists (ACL) 192
ACL (access control lists) 192
Active Server Pages (ASP) 5
active users 55
active-to-ineligible transition 26
activity level 273, 276
adapter 205
Adapter Configuration Tool 209
Add RDB Directory Entry 86
ADDNWSSTGL command 66
ADDRDBDIRE command 86
Advanced Order Configurator 315
Advanced Planning 314
Advanced Web Order Configurator (AWOC) 315
agent 204, 207
All in One (AIO) configuration 182
allocation system values 266
AlphaWorks 293
APAR (Informational Authorized Problem Analysis Report) 17
API (application program interface) 8
Application level Tools 290
application program interface (API) 8
application server 1–2
applications run 56
Ariba XBP Set for e-Procurement 213
AS/400 system values 107
ASP (Active Server Pages) 5
ASP utilization 252
assistance level 266
asynchronous business function processing 200
AT&T Global Network 21
authorization code 63
automatic tuning 264
availability solutions 236
AWOC (Advanced Web Order Configurator) 315

B
B2B 314
B7333 directory 123
B7333SYS library 128
B9861101 Business Function 129
base business function (BSFN) 12
batch sizing 58
benchmark process 56
  applications run 56
benchmark scripts 56
BEST/1 20, 25
binary large object (BLOB) 63
BLOB (binary large object) 63
broker server 204–205
browser 19
BSFN (base business function) 12
business function mappings 9
business function processing 200
business logic 9

C
C business functions 9
caching Java serialized object 199
canonical event 207, 213
capacity planning 15, 19
CCSID 102
Central Objects tablespace 13
CFGTCP command 68
Change Shared Pool (CHGSHRPOOL) command 265
CHGPJE 285
CHGSHRPOOL (Change Shared Pool) command 265
CHGSYSVAL 265
CHGTCPA 306
CHGTCPRTE 306
classpath 290

Client Access Express using the internal LAN 68
Client Access TCP/IP configuration tasks 68
CLRIPC 119, 139
cluster message queue utilization 252
cluster name 238
Cluster Resources Group (CRG) 234
clustering 235, 238
  architecture 234
  exit program 234
coexistent merges 87
Collection Services 286
COMMENT 300
Commercial Processing Workload (CPW) 24
communication performance 304
Competency Center 54
Conference Room Pilot (CRP) 62
configuring XPIe 216
connection pool setting 198
connectivity 230
Consider Growth recommendation 25
CPW (Commercial Processing Workload) 24
CRG (Cluster Resources Group) 234
CRM (Customer Relationship Management) 314
cross reference table 207
CRP (Conference Room Pilot) 62
CRTNWSSTG command 66
Custom Plan 87
Customer Relationship Management (CRM) 314

D
data collection 294
Data Dictionary 78
data replication 237
data source setup 87
Data Sources 90
DB Monitor 286, 299
dedicated disk 228
DEMOD (Direct Execution Mode) 299
denial of service (DOS) 192
deployment server 63
  FTP configuration verification 82
  host table entry 82
  paging file size 69
  setup 87
Developer Kit for Java 8
Direct Execution Mode (DEMOD) 299
disk array 228
disk group 228
disk sizing 57
disk storage 225
DMPJVM (Dump Java Virtual Machine) command 286, 290
Document Manager 214
DOS (denial of service) 192
DSPACTPJ 280
DSPJOBTBL 266–267
DSPSBSD 281
Dump Java (DMPJVM) command 286
Dump Java Virtual Machine (DMPJVM) command 290

E
EIP (Enterprise Integrated Portal) 5
EJB (Enterprise JavaBean) 3
Electronic Service Agent 20
End Performance Trace (ENDPFRTRC) command 311
End PEX (ENDPEX) command 292
ENDDBMON 300
ENDNET 119, 135, 139
ENDPEX (End PEX) command 292
ENDPFRTRC (End Performance Trace) command 311
Enterprise Content Manager 315
Enterprise Integrated Portal (EIP) 5
Enterprise JavaBean (EJB) 3
enterprise server 1
  host table entry 81
  setup 87
  TCP/IP protocol 80
Environment Checker 67
environment setup 87
event 205
Event Notification (EVN) kernel 221
Event Tracker 211
EVN (Event Notification) kernel 221
expert cache 227
eXtended Process Integration (XPI) 6, 203
external storage 227
external XBP 212

F
F00821 table 131
F0083 table 131
F91013 table 131
failover 241

FDA (Form Design Aid) 200
fibre connection 229
Form Design Aid (FDA) 200
form.js file 201
FRCRCD 300
FTP configuration verification 82

G
garbage collection 291
  setting 187
garbage collector 307–308
GATS (Global Advanced Technology Services) 16
General Subsystems 214
generation cycle 14
Global Advanced Technology Services (GATS) 16
GO PERFORM 311

H
Hardware Compatibility List (HCL) 17
HBA cards 226
heap 307
heap size memory setting 185
heartbeat service 234
high availability 235
high-speed interconnection 305
horizontal scaling 309
Host Bus Adapter (HBA) cards 226
host installation process 118
host table entry 81–82
HTML client 3, 18
htmlclient.ini 200
htmloverrides.ini file 200
HTTP Server 2, 190
  (original) 190
  (powered by Apache) 190
  persistence connection 192
HTTP threads 190
HTTP user profile swapping 193

I
IBM Developer Kit for Java 8
IBM HTTP Server for iSeries 2
  (original) 2
  (powered by Apache) 2
IBM OneWorld sizing questionnaire 55
IBM sizing process 58
IBM Toolbox for Java 8
IBM Workload Estimator 20
IBM/J.D Edwards International Competency Center 54
IEO (Interoperability Events Observer) kernel 221
index seize 310
information broker 204–205
Informational APAR 17
Informational Authorized Problem Analysis Report (APAR) 17
INI file changes 221
initial heap size parameter 307
Installation Planner 86–87
installation process 244
Installation Workbench 86, 100
installing OneWorld Xe 61
installing XPI 215
Installing XPIe Foundation 216
Integrated xSeries Server 64
intelligent adapters 204
Inter Process Communications (IPC) 137
interconnection 305
internal storage 227
internal XBP 212
Internet Explorer 19
interoperability 226
Interoperability Events Observer (IEO) kernel 221
IP takeover address 234
IPC (Inter Process Communications) 137
iSeries server
  availability solutions 236
  clustering 235
  clustering installation process 244
  connectivity to SAN 230
  disk storage 225
  expert cache 227
  eXtended Process Integration (XPI) 203
  high availability 235
  JDBC driver 201
  performance issues 225
  SAN technology considerations 227
  storage 224
  tuning 264

J
J.D. Edwards
  Knowledge Garden 16
  middleware 8
  sizing process 54

jar file 3
JAS server 1, 197
  architecture overview 2
  hardware requirements 17
  software requirements 18
JAS servlet connections 187
JAS software 3
jas.ini file 7, 198–199
Java and HTML Generator 13
Java Application Server 1
Java archive (jar) file 3
Java Database Connectivity (JDBC) 8
Java environment 9
Java garbage collector 307
Java job 277
Java model over OWProxy 11
Java Persistent Objects Cross Reference 13
Java Persistent Objects Table 13
Java program optimization 188
Java serialized object 3, 13, 199
Java Virtual Machine (JVM) 8, 184
JavaServer Page (JSP) 3
JDBC (Java Database Connectivity) 8
JDBC connection pool setting 198
JDBC driver 8
  on iSeries server 201
  options for iSeries 8
JDBODBC.DLL 90
JDE user profile 130
jde.ini file 111, 113, 140, 145, 171
JDEDST 116
JDEdwardsOneWorldB7333 77
JDEFIN 116
JDEHRM 116
JDEKRNL.DDB 83
JDEKRNL.XDB 83
JDEMFG 116
JDENet communication middleware 9
JDENET connection 199
JDEPLAN.MDB 63, 83
JDETECH 116
JITC (Just In Time Compiler) 188, 299
JOB 300
job queue 116
journal ASP utilization 252
journaling performance 250
JSP (JavaServer Page) 3
JTOPEN 202
Just In Time Compiler (JITC) 188, 299
JVM (Java Virtual Machine) 8, 184

K

KMBI (Knowledge Management/Business Intelligence) 5
Knowledge Garden 16
Knowledge Management/Business Intelligence (KMBI) 5, 321

L

LAN (local area network) 304
latency 252
LIBRARY file 113
library installation 78
LINKBSFN program 135
local area network (LAN) 304
location setup 87
lock 310
logic 9
logical unit number (LUN) 224
LUN (logical unit number) 224

M

Management Central 286–287
manual tuning 264
MAXACT 271
media object queues 129
memory pool
  determining the size 274
  size and activity level 273
Microsoft Access 84
Microsoft Hardware Compatibility List (HCL) 17
middleware 8
model
  CPU 43
  creating 27
  growth 37
multi-line edit 200
multiple foundation install 140

N

Named Event Rules (NER) 9
native calls 8
native driver 8
NER (Named Event Rules) 9
Netscape Navigator 19
notification event 206

O
object specifier file 234
OCM mapping 129
ODAC (Online Disk Arms Calculator) 57
OneWorld
  architecture differences 11
  business logic 9
  functionality differences 12
  implementation 224
  JAS software 3
  sizing 54
  sizing non-OneWorld workloads 56
OneWorld Adapter 216
OneWorld B73.3.2
  compared to OneWorld Xe 10
    architecture differences 11
    functionality differences 12
OneWorld Java Application Server 1
OneWorld Planner 63
OneWorld Portal 5
OneWorld sizing questionnaire 55
OneWorld XBP Set for CRM Sales 213
OneWorld XBP Set for e-Procurement 213
OneWorld XBP Set for Storefront 213
OneWorld Xe 238
  capacity planning 15
  clustering 238
  compared to OneWorld B73.3.2 10
    architecture differences 11
    functionality differences 12
  installation 61
  JAS server 1, 197
  pre-installation planning 15–16
  scaling the environment 309
  sizing 15
Online Disk Arms Calculator (ODAC) 57
operation 240
OUTFILE 300
OUTMBR 300
OWProxy 11

P
package build 14
package setup, builds 87
paging file size 69
PC performance 306
peak workload 55
performance data 294
Performance Explorer (PEX) 286, 292
performance issues 225
Performance Management/400 (PM/400) 20
performance monitor 303
performance objectives 263
performance tools 285
Performance Tools for iSeries 287
Performance Trace Data Visualizer (PTDV) 293
persistence 192
persistent connection 192
PEX (Performance Explorer) 286, 292
plan type 87
Plan Validation 99
Planner Path Code 63
platform-independent objects 75
PM/400 20
  with IBM Workload Estimator 24
PORTTEST 118, 137
pre-installation planning 15–16
prestart jobs 279
primary node 240
  performance 249
Print Lock Report (PRTLCKRPT) command 311
Print PEX Report (PRTPEXRPT) command 293
Processors on Demand 25
profile swapping 193
PRTLCKRPT (Print Lock Report) command 311
PRTPEXRPT (Print PEX Report) command 293
PTDV 286
  data collection 294
  visualizing performance data 294
PTDV (Performance Trace Data Visualizer) 293
PTRPEXRPT 308

Q
QACTJOB 266, 270
QADLACTJ 266, 271
QADLSPLA 266
QADLTOTJ 266, 269
QALWOBJRST 108, 135
QALWUSRDMN 135
QAPEXDFN 292
QBASACTLVL 271
QBASPOOL 273
QCMN subsystem 279
QEJBSBS subsystem 277
QJOBMSGQFL 107, 135–136, 266
QJOBMSGQMX 266

QJOBMSGQSZ 266
QJOBMSGQTL 266
QJOBSPLA 266
QJVACMDSRV 278
QMAXACTLVL 273
QMAXJOB 266
QMAXSPLF 266
QMCHPOOL 273
QPEXDATA 292
QPFRADJ 265
QQRYDEGREE 108
QRCLSPLSTG 266
QRWTSRVR 278
QSECURITY 107
QSERVER subsystem 279
QSQSRVR 8, 278–279
QSYSWRK subsystem 8
QTOTJOB 266
queuing theory 259–260
QUSRWRK susbsystem 8
QZDAINIT 112
QZDASOINIT 8, 278–279
QZRCSRVS 278
QZSCSRVS 112
QZSOSIGN 112

R

R4094A batch application 131
rank 228
record lock 310
recovery domain 234
Redbooks Web site 319
  Contact us xiv
remote journaling 237
request 260
Resource Analyzer 286
Resource Library Manager 214
resources 205
role swap 241
RosettaNet 214
RosettaNet Engine 214

S

SAN
  connectivity with iSeries 230
  considerations for iSeries 227
  disk cache 226
  disk storage 225
  performance issues 225
  storage cache 228
SAN (Storage Area Network) 223, 236
SAR (Software Action Request) 16
SAW 6
scaling the OneWorld Xe environment 309
scripts 56
SCSI connection 229
secondary node performance 250
Selected System recommendation 25
serialized object 13, 199
server 260
Server Administration Workbench (SAW) 6
Server Monitoring Configuration (SMC) 6
service time 260
setup of other servers 87
SETUP.INF file 118
shared disk 228
shared nothing implementation 235
Siebel BP Set for CRM Sales 213
silent post with multi-line edit 200
sizing 15, 54
  batch sizing 58
  disk sizing 57
  IBM sizing process 58
  International Competency Center 54
  methodology 54
sizing questionnaire 55
SMC (Server Monitoring Configuration) 6
Software Action Request (SAR) 16
SQL performance monitor 303
SQL Visual Explain 286, 301
Start Performance Trace (STRPFRTRC) command 311
Start PEX (STRPEX) command 292
STBEST command 43
steady-state 26
storage 224
Storage Area Network (SAN) 223, 236
storage cache 228
storage consolidation 224
storage system values 271
STRDBMON 300
STRPEX (Start PEX) command 292
STRPFRMON 311
STRPFRTRC (Start Performance Trace) command 311
subsystems 277
switched disk 236

System Analysis 286
system level tools 287
system sizing 59
system values 265

T
TCP/IP protocol, enterprise server 80
Terminal Server Edition 127
Terminal Services 63
thread information 291
thread state 271
Toolbox driver 8
transition data 276
TSE 127
tuning 264
TYPE 300
type 1 C business functions 9
type 1 NER 9
Typical Plan 87

U
UBE (Universal Batch Engine) 58
unit-of-work 260
Universal Batch Engine (UBE) 58
user profile swapping 193
utilization guideline 259, 263

V
verifying host installation 118
vertical scaling 310
Virtual 3-Tier (V3T) configuration 182
virtual drive 228
virtual private network (VPN) 21
virtual three-tier, two-node cluster 238
Visual Explain 301
  running proactively 301
  running reactively 302
Visual Integrator 210
VPN (virtual private network) 21

W
WAN (wide area network) 304
WCBT (Work Control Block Table Entry) 266
Web application server 1
Web server 2
WebSphere Administrative Console 184
WebSphere Application Server 3, 184
  Advanced Edition 18
  edition 18
  Java Virtual Machine 184
  number of instances 184
  Standard Edition 18
WebSphere Application Server 3.5 11
WebSphere job 277
WebSphere Resource Analyzer 288
wide area network (WAN) 304
Windows 2000 post installation 64
Windows NT Task Manager 307
Word Search tables 131
Work Control Block Table Entry (WCBT) 266
Workflow Engine 214
Workload Estimator 20, 49
  proposed configuration 53
  sensitivity to variations 53
  with PM/400 24
  workload complexity 52
  workload detail entry 51
  workload type selection 50
World XBP Set for CRM Sales 213
World XBP Set for e-Procurement 213
Worldwide Advanced Technologies (WWAT) 16
WRKSHRPOOL 272
WRKSYSSTS 266, 272–273, 277
WRKSYSVAL 265
WWAT (Worldwide Advanced Technologies) 16

X
XBP 212
XMLList kernel 221
XPI (eXtended Process Integration) 6, 203
XPI Foundation code 215
XPIe 216
  configuration 216
  tools 208
XPIe (enterprise) 204
XPIx 213
XPIx (extended) 204
XPIx Adapter 214
XPIx Security 214
XPIx Server 214
XPIx Transformer and Converter 214
XPIx XBPs 214

Y
YOUcentric 314

IBM

Redbooks

J.D. Edwards OneWorld Xe

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

IBM®

J.D. Edwards OneWorld Xe

Implementation on IBM @server iSeries Servers

Redbooks

Follow this complete implementation guide for OneWorld Xe on iSeries

Set up an HTML client and WebSphere Application Server

Learn valuable performance management and tuning tips

This IBM Redbook explains the specific tasks associated with the implementation of OneWorld Xe service pack 17 on the IBM @server iSeries server with OS/400 V5R1 and WebSphere Application Server Version 3.5. It is based on a collection of knowledge from J.D. Edwards consultants and IBM professionals who have experience with J.D. Edwards OneWorld Xe and the iSeries server.

This redbook is designed to assist technical people among J.D. Edwards OneWorld customers, OneWorld consultants, business partners, and IBM service representatives. It targets these professionals who are directly involved with the implementation of a total business solution. Such solutions consist of iSeries server hardware, OS/400, DB2 UDB for iSeries database, OneWorld solutions, and supplemental solution products.

You should use this redbook in conjunction with *J.D. Edwards OneWorld Implementation for AS/400*, SG24-5195. This publication explains detailed concepts and all implementation steps of earlier versions of OneWorld, most of which still apply to OneWorld Xe. You should also use this redbook together with the J.D. Edwards manuals that are provided with the OneWorld software.

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

SG24-6529-00 ISBN 0738424285