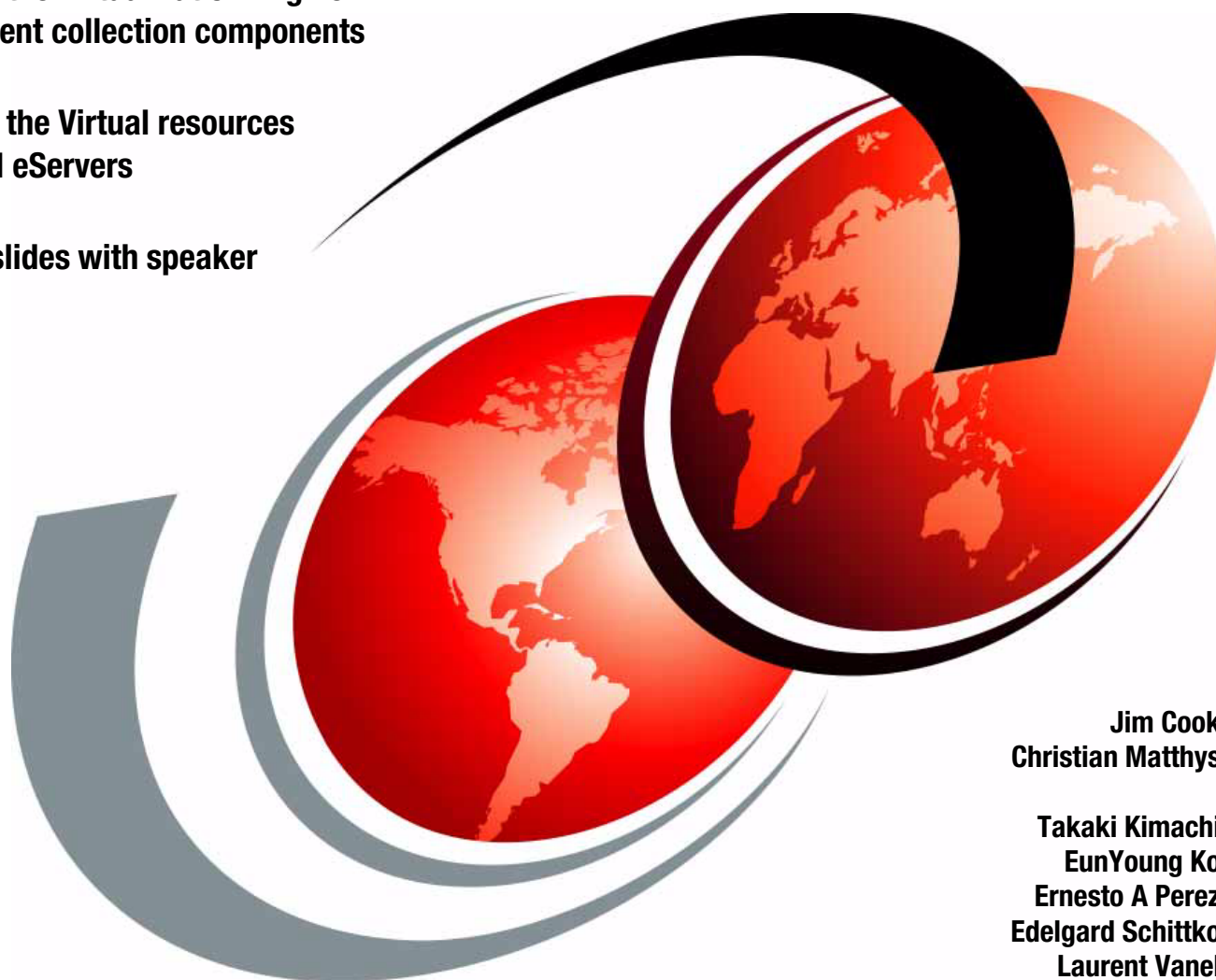


IBM Virtualization Engine Platform Version 2 Technical Presentation Guide

Describes the Virtualization Engine
management collection components

Describes the Virtual resources
for all IBM eServers

Provides slides with speaker
notes



Jim Cook
Christian Matthys

Takaki Kimachi
EunYoung Ko
Ernesto A Perez
Edelgard Schittko
Laurent Vanel

Redbooks



International Technical Support Organization

IBM Virtualization Engine Platform Version 2

December 2005

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

First Edition (December 2005)

This edition applies to the IBM Virtualization Engine Version 2.1

This document created or updated on December 19, 2005.

© Copyright International Business Machines Corporation 2005. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

- Notices** vii
- Trademarks viii

- Preface** ix
- The team that wrote this redbook. x
- Become a published author xii
- Comments welcome. xiii

- Chapter 1. Introduction to the Virtualization Engine platform** 1
- 1.1 The IBM Systems Agenda 2
- 1.2 From Web services to the On Demand Operating Environment. 5
- 1.3 The On Demand Operating Environment architecture details. 10
- 1.4 Standards: from OGSi to WSDM 13
- 1.5 The Virtualization Engine platform 20
- 1.6 Virtual Resources 24
- 1.7 The Virtualization Engine management collection 26
- 1.8 What’s new with the Virtualization Engine Version 2 28
- 1.9 The Virtualization Engine Version 2 products offerings 30
- 1.10 Virtualization: topology and terminology 33
- 1.11 Management servers: the supported operating systems 35
- 1.12 Managed servers: the supported operating systems 36

- Chapter 2. IBM Director 5.10** 37
- 2.1 IBM Director 5.10 overview 38
- 2.2 IBM Director 5.10 hardware components 42
- 2.3 IBM Director 5.10 software components 44
- 2.4 Overview of agent level functions 47
 - 2.4.1 IBM Director console and managed objects 49
 - 2.4.2 IBM Director console navigation views 51
- 2.5 What’s new in IBM Director 5.10, summary 53
- 2.6 Comprehensive platform management with IBM Director. 55
- 2.7 Topology of IBM Director component and Virtualization Engine console 59
- 2.8 IBM Director 5.10 cross platform functions 62
- 2.9 IBM Director extensions 65
- 2.10 Discovery cross platform functions 69
 - 2.10.1 Discovery: automatic preferences. 71
 - 2.10.2 Discovery preferences, naming templates 72
- 2.11 Inventory cross platform functions 73
 - 2.11.1 Inventory subtasks 74
 - 2.11.2 Inventory: custom collection, inventory monitors, and custom query 76
 - 2.11.3 Hardware Inventory query - i5/OS example 77
- 2.12 Monitors and Hardware Status cross platform functions. 79
- 2.13 Inventory monitors. 85
- 2.14 Hardware status 86
- 2.15 Event Action Plans cross platform functions. 88

- Chapter 3. Resource Dependency Service** 97
- 3.1 Modeling the I/T Infrastructure with virtual resources 98
- 3.2 A manageable resource 101

3.2.1	Anatomy of a manageable resource	101
3.2.2	Exploiting standards	102
3.3	Resource Dependency Service overview	107
3.4	Resource Dependency Service architecture	108
3.5	Intelligent Device Discovery function	111
3.6	Data Collectors function	114
3.7	Resource Dependency Service repositories	118
3.8	Resource Modeling framework	120
3.9	Positioning Resource Dependency Service and the Common Configuration Management Database	124
3.10	Analytics and correlation engines	126
3.11	Topology services	128
3.12	Runtime, packaging, installation and configuration	130
	Chapter 4. The Virtualization Engine console	133
4.1	Positioning the Virtualization Engine console	134
4.2	Virtualization Engine console functions	135
4.3	Virtualization Engine console architecture	139
4.4	The welcome page	141
4.5	The health center function	147
4.6	Interacting with the environment	157
4.7	The topology view	162
4.8	The Virtualization Engine console under the hood	169
4.9	Summary	171
	Chapter 5. Enterprise Workload Manager	173
5.1	Enterprise Workload Manager overview	174
5.2	Enterprise Workload Manager terminology	176
5.3	ARM instrumentation	178
5.3.1	ARM 4.0 API execution	180
5.3.2	Correlator and hop	181
5.3.3	ARM instrumented IBM products	183
5.3.4	ARM enablement	184
5.3.5	Non-ARM instrumented servers support	184
5.4	Enterprise Workload Manager domain policy	186
5.4.1	Domain policy and service policy	187
5.4.2	Service class	188
5.4.3	Domain policy, service policy, service class example	190
5.4.4	Workload classification	191
5.5	Performance Index	201
5.6	Enterprise Workload Manager load balancing	203
5.7	Enterprise Workload Manager LPAR CPU management	205
5.7.1	eServer p5/i5 LPAR concept	206
5.7.2	Partition Workload group	208
5.7.3	LPAR CPU management mechanism	210
5.8	Enterprise Workload Manager and TIO integration	212
5.9	EWLM V2.1 enhancements	214
	Chapter 6. Integrated Virtualization Manager for pSeries and Virtual Partition Manager for iSeries	215
6.1	Benefits from IVM and VPM	216
6.2	HMC, APV feature, and VIOS	218
6.3	Integrated Virtualization Manager	222
6.3.1	Comparing VIO Server with IVM to VIO Server with HMC	225

6.3.2	IVM design	228
6.3.3	IVM considerations	230
6.3.4	VIOS- IVM user interface	233
6.3.5	IVM and VIOS initial setup and partition preparation	234
6.3.6	Client LPAR Creation	236
6.4	Virtual Partition Manager.	240
6.4.1	Limitations and considerations with VPM	242
6.4.2	Preparing system for VPM	246
6.4.3	Defining new Linux partitions	250
Chapter 7. Security		253
7.1	The end to end security model	254
7.2	Need for single sign-on in the virtualization environment	256
7.3	Single Sign-On architecture in the Virtualization Engine Version 2.	259
7.4	Single sign-on set up.	262
7.5	Configuring the single sign-on in the Virtualization Engine console	264
7.6	IBM Director security.	268
Chapter 8. Getting Started		271
8.1	The Information Center	272
8.1.1	Overview	272
8.1.2	Access	274
8.1.3	General topics.	275
8.1.4	Interactive Resources	277
8.1.5	Printing	278
8.2	Planning advisor	280
8.2.1	Role of the planning advisor	280
8.2.2	Overview	282
8.2.3	Planning advisor flow	284
8.2.4	How to select your Virtualization Engine products.	286
8.2.5	How to select your management servers	287
8.2.6	How to define your managed servers	288
8.2.7	Your customized installation plan	289
8.3	The common runtime components	291
8.3.1	Virtualization Engine software components	291
8.3.2	Virtualization Engine common products	292
8.3.3	Repositories	295
8.3.4	User IDs, passwords, and keys.	296
8.4	Installation tasks	297
8.4.1	Deployment planning	297
8.4.2	Virtualization Engine installation task list	298
8.4.3	Additional installation options and tasks	304
8.4.4	Specific consideration for i5/OS systems edition	306
8.4.5	Uninstallation	307
8.5	Products packaging.	308
8.6	Architecture, performance, and sizing.	310
8.6.1	General architectural guidance	311
8.6.2	Hardware sizing considerations	315
8.6.3	Performance benchmarks.	319
Chapter 9. Virtual Resources: Server Specific virtualization functions		325
9.1	For xSeries and BladeCenter: Capacity Manager and System Availability.	327
9.2	For xSeries and BladeCenter: IBM Virtual Machine Manager.	329
9.3	For iSeries, pSeries and zSeries: Dynamic Logical Partitioning	331

9.4 For pSeries: Partition Load Manager	333
9.5 For iSeries and pSeries: Micro Partitioning	336
9.6 For iSeries and pSeries: Virtual I/O, Virtual LAN, Shared Ethernet	338
9.7 For iSeries, pSeries and zSeries: Capacity on Demand	340
9.8 For iSeries and pSeries: Simultaneous Multithreading	344
9.9 For iSeries: Multiple OS support	346
9.10 For zSeries: Multiple operating systems support with z/VM	349
9.11 For zSeries: HiperSockets	351
9.12 xSeries and BladeCenter summary	353
9.13 System z9 and zSeries summary	354
9.14 System p5 and pSeries summary	356
9.15 iSeries summary	358

Appendix A. Integrated Virtualization Manager versus Hardware Management Console

361

Abbreviations and acronyms	365
Glossary	369
Related publications	373
IBM Redbooks	373
Other publications	373
Online resources	374
How to get IBM Redbooks	375
Help from IBM	375
Index	377

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:

@server®	Intelligent Device Discovery®	PR/SM™
@server®	IntelliStation®	pSeries®
Redbooks (logo)  ™	iSeries™	Redbooks™
Asset ID™	IBM®	RS/6000®
AIX 5L™	IMS™	S/390®
AIX®	Lotus Notes®	ServeRAID™
AS/400®	Lotus®	System z9™
BladeCenter®	Micro-Partitioning™	Tivoli Enterprise™
Candle Management Server®	Multiprise®	Tivoli Enterprise Console®
Candle®	MVS™	Tivoli Management Environment®
CICS®	NetView®	Tivoli®
CICSplex®	Notes®	TotalStorage®
DB2 Connect™	OpenPower™	Virtualization Engine™
DB2 Universal Database™	OS/400®	VSE/ESA™
DB2®	Parallel Sysplex®	WebSphere®
Electronic Service Agent™	Passport Advantage®	xSeries®
Enterprise Storage Server®	PowerPC®	z/Architecture™
eServer™	Predictive Failure Analysis®	z/OS®
HiperSockets™	POWER™	zSeries®
HACMP™	POWER3™	z/VM®
i5/OS™	POWER4™	
Infoprint®	POWER5™	

The following terms are trademarks of other companies:

EJB, Java, JDBC, JDK, JMX, JVM, J2EE, J2SE, Power Management, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

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

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

Preface

Fundamental to the IT infrastructure simplification are the concepts of automation and virtualization.

The IBM Virtualization Engine is the delivery vehicle of the virtualization concepts; it includes innovations in resource virtualization as well as management, performance, modelling and visualization services for the infrastructure components in an heterogeneous environment.

A new version of the IBM Virtualization Engine has been announced and detailed in November 2005. The IBM Virtualization Engine platform provides the foundation to build an infrastructure more simple to manage; the IBM Virtualization Engine is based on open interfaces and industry standards. It provides a set of blocks for progressively building business oriented solutions, allowing customers to start where they want and to evolve at their own pace.

This new version, the IBM Virtualization Engine Version 2 is made of Virtual Resources which are integrated into the IBM @server systems and Virtual Management and Access components which are a multi-platform offering of system tools for a variety of server operating system environments. Some of its components are available since December 2005, other components will be made available in 1Q2006.

This redbook provides foils and technical information (that can be viewed as speakers notes) which describe all the existing virtualization features and products included in the Virtualization Engine Version 2:

- ▶ Chapter 1, "Introduction to the Virtualization Engine platform" on page 1 introduces the architecture environment and the concepts that guide the Virtualization Engine principles.
- ▶ Chapter 2, "IBM Director 5.10" on page 37 describes the IBM Director 5.10.
- ▶ Chapter 3, "Resource Dependency Service" on page 97 describes the Resource Dependency Services, a very new component of the Virtualization Engine.
- ▶ Chapter 4, "The Virtualization Engine console" on page 133 describes the new version of the Virtualization Engine console.
- ▶ Chapter 5, "Enterprise Workload Manager" on page 173 describes the new version of the Enterprise Workload Manager.
- ▶ Chapter 6, "Integrated Virtualization Manager for pSeries and Virtual Partition Manager for iSeries" on page 215 and Chapter 9, "Virtual Resources: Server Specific virtualization functions" on page 325 describe the Virtual Resources, specific for each IBM @server.
- ▶ Chapter 7, "Security" on page 253 describes the new security environment provided by the Virtualization Engine Version 2.
- ▶ Chapter 8, "Getting Started" on page 271 presents all the facilities offered in this Virtualization Engine Version 2 to start implement one or several of its components.

This redbook is suitable for IT Architects and for IT Specialists who want to understand how the virtualization components fit into the Service Oriented Architecture, to understand the products offerings in detail or to plan a project that includes the use of these features and products.

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, Poughkeepsie Center.

Jim Cook is a Senior Software Engineer at the ITSO, Rochester Center. He leads teams that produce iSeries™ Announcement presentation sets that are maintained on the iSeries technical support Web sites and presents at ITSO iSeries Forums internationally. Jim also produces IBM Redbooks™ about various iSeries and IBM eServer™ i5 topics.

Christian Matthys spent more than 25 years at IBM as a System Engineer, working with large mainframe-oriented customers. He spent three years as an ITSO project leader on assignment in Poughkeepsie, NY. In 2000 he joined the EMEA Design Center for On Demand Business, working with customers' projects to make use of the leading edge technologies, in particular the Autonomic Computing technologies. He was part of the launching team of the Start Your Engines workshops to promote the IBM Virtualization Engine. He works as an ITSO project leader based in the Product and Solutions Support Center in Montpellier (PSSC), France. He is a Consulting IBM IT Specialist certified.

Takaki Kimachi is an IBM Certified IT Specialist at the iSeries Techline, since 2000 in the areas of Linux® on iSeries, iSeries LPAR and hardware, covering Japan and Asia Pacific. He joined IBM Japan in 1984 as a Systems Engineer providing technical support for customers and business partners. His career has spanned 21 years in IBM midrange server S/36, AS/400 , and iSeries.

EunYoung Ko is an Advisory IT Specialist at FTSS in IBM Korea. She has five years experience of Dynix/ptx and five years of AIX . She currently works for the Field Technical Sales Support Team for pSeries®. Her mission includes various benchmark tests, performance tuning, troubleshooting, and solution implementation.

Ernesto A. Perez is an IBM Senior I/T Architect with over 20 years of experience in the I/T industry. Mr. Perez is a member the IBM team of specialized Client I/T Architects for FSS Americas, where he helps design business solutions for the financial services industry. His area of professional focus is the IBM On Demand Operating Environment (ODOE) with particular emphasis on its Infrastructure Services areas. Mr. Perez's domain of expertise also extends to leading-edge I/T virtualization technologies including Grid Computing solutions.

Edelgard Schittko is an iSeries Advisory I/T Specialist working for iSeries Technical Sales in IBM Germany. She holds a degree in mathematics from Humboldt University in Berlin, Germany. She has 14 years of experience in working with the AS/400, iSeries and Power5 server i5 in midrange customer and large customer environments. Her areas of experience include iSeries and Storage Area Network, iSeries Backup and Recovery and iSeries together with Virtualization Engine. Edelgard coauthored multiple redbooks.

Laurent Vanel is a Systems Architect in IBM France. He started as an IT specialist in the RS/6000 brand, now pSeries, 15 years ago. His IBM experience includes a 3-year assignment at the Austin ITSO. In his current position, he is responsible for the infrastructure proposed to large customers in the Industrial sector. He is also the technical focal point for the Virtualization Engine solutions for IBM Systems and Technology group in France.



Figure 0-1 From left to right: Laurent, Jim, Edelgard, Young, Ernie, Christian and Takaki

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

Paola Bari
IBM USA, Poughkeepsie, NY

Jeremy Bauer
IBM USA, Rochester, MN

Jim Beck
IBM USA, Rochester, MN

Pat Botz
IBM USA, Rochester, MN

Mark Cathcart
IBM USA, Somers, NY

Ramamohan Chennamsetty
IBM USA, Poughkeepsie, NY

G Michael Connolly
IBM USA, Poughkeepsie, NY

G Michael Connolly
IBM USA, Poughkeepsie, NY

Susan F Crowell
IBM USA, Rochester, MN

Jeff Van Heuklon
IBM USA, Rochester, MN

Norbert Masannek
IBM Germany, Munich

Annette Miller
IBM US

Tom Monza
IBM USA, Poughkeepsie, NY

Paula Muth
IBM USA, Rochester, MN

Madeline Nick
IBM Systems Group Technical Support, Poughkeepsie, NY

Padmini Prasad
IBM USA, Poughkeepsie, NY

Gregory Pruett
IBM USA, Raleigh, NC

Devon Snyder
IBM USA, Rochester, MN

Scott Urness
IBM USA, Rochester, MN

Jeff Thelen
IBM USA, Rochester, MN

Bill Vanca
IBM USA, Raleigh, NC

David Watts
IBM USA, Raleigh, NC

Romney White
IBM USA, Endicott, NY

Gudrun M Wolfgram
IBM USA, Rochester, MN

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

ibm.com/redbooks

- ▶ Send your comments in an email to:

redbook@us.ibm.com

- ▶ Mail your comments to:

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



Introduction to the Virtualization Engine platform

On November 8, 2005, IBM announced new IBM Virtualization Engine offerings under the generic name of the *Virtualization Engine platform* which is the IBM Virtualization Engine Version 2.

This chapter:

- ▶ Positions the role of the IBM Virtualization Engine platform in the IBM Systems Agenda, in Section 1.1, “The IBM Systems Agenda” on page 2.
- ▶ Positions the IBM Virtualization Engine into the On Demand Operating Environment architecture in Section 1.2, “From Web services to the On Demand Operating Environment” on page 5 through Section 1.3, “The On Demand Operating Environment architecture details” on page 10.
- ▶ Explains the importance of the Open Standards in Section 1.4, “Standards: from OGSI to WSDM” on page 13.
- ▶ Introduces the components of the new version of the IBM Virtualization Engine in Section 1.5, “The Virtualization Engine platform” on page 20 through Section 1.10, “Virtualization: topology and terminology” on page 33.

1.1 The IBM Systems Agenda






ibm.com/redbooks International Technical Support Organization 	
<h2>The IBM Agenda</h2>	
 <p>Virtualization</p>	<p>The IBM Virtualization Engine platform is a comprehensive virtualization portfolio enabling customers to virtualize resources</p>
 <p>Openness</p>	<p>Commit to openness to redefine how to acquire systems, avoid being “locked in” and achieve needed integration.</p>
 <p>Collaborative Innovation</p>	<p>Collaborate to innovate with the capability to share information, processes and functions across the enterprise and beyond, as well as participation in industry innovation networks</p>
 IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.	

Figure 1-1 The IBM Systems agenda

In October 2003, IBM announced the *On Demand Business* strategy. On Demand Business puts the emphasis on business responsiveness and flexibility based on Information Technology (IT) to make that happen.

The *On Demand Operating Environment* is the computing environment that is required for building and managing an On Demand Business. It is a flexible, integrated, and reliable IT infrastructure that uniquely serves the business. The On Demand Operating Environment allows business objectives to drive the IT processes, rather than adjusting the business logic around the IT infrastructure. To become an On Demand Business a company needs:

- ▶ Componentization of business processes, to create the flexibility and innovation at the business level.
- ▶ Applications modularity to support the business processes.
- ▶ IT infrastructure simplification to be able to support the changes the business needs.

Fundamental to the IT infrastructure simplification are the concepts of automation and virtualization:

- ▶ *Automation* enables an IT infrastructure to manage many day-to-day tasks itself. With a self-managing infrastructure, efficiency is increased and resource allocation simplified. A fully automated IT infrastructure can sense changing conditions, such as surges in demand or isolated application errors, and can spot trends that could lead to costly system downtime. The infrastructure automatically responds by taking corrective actions that ensure IT resources remain aligned with business goals.

- ▶ *Virtualization* is the ability to separate the direct dependency of an application to a physical resource. Through virtualization, an enterprise:
 - Has a single, consolidated view of, and easy access to, all available resources in the network, regardless of location.
 - Efficiently accesses and manages those resources to reduce operations and systems management costs while maintaining needed capacity.
 - Responds dynamically to the application needs of its users.
 - Gathers and accesses information across the organization quickly to gain competitive advantage.

The IBM Virtualization Engine Version 1 has been announced and made available in August 2004 and addresses the virtualization aspects of the service oriented infrastructure.

The IBM Systems Agenda extends the value of virtualization. The IBM Systems agenda is a new contribution of the IBM hardware capability to the On Demand Business strategy. Several years ago, IBM announced the IBM @server, then IBM combined the server, the storage, and the technology business together. IBM is now committed to break down barriers between components and enable them to be integrated. The three core design principles for the IBM Systems agenda are:

- ▶ Virtualize everything
- ▶ Commit to openness
- ▶ Collaborate to innovate

What does “Virtualize everything” means?

Virtualize everything means:

- ▶ Continued enhancements to the virtualization facilities have to further simplify the infrastructure environment by integrating many of the features into the systems themselves; for small or midsize business, with limited IT resources, virtualization functions need to simplify the platform management.
- ▶ Systems need to better monitor and manage themselves: this will increase continuous availability, improve security, improve resource utilization, and improve quality of service levels. When virtualization capabilities are implemented, the automation concepts become more effective. The highest level of automation is achieved through autonomic computing technologies, another IBM initiative¹.
- ▶ Each component will be part of the virtualization capabilities: storage, processors, networks, applications, data, and users.

What does “Commit to Openness” means?

Commit to openness means:

- ▶ IBM is committed to contribute to make open standards and openness part of its solutions: openness reduces risk through freedom of choice.
- ▶ IBM wants to offer systems that leverage technology innovation and fit within any IT environment, from small businesses, that have limited resources, to very large multinational corporations which have to deal with very complex non homogeneous environments.
- ▶ IBM wants to offer systems that combine servers, operating systems, storage, middleware, networking and applications that enable rapid deployment of a business solution, based on industry standards.

¹ To know more about the IBM Autonomic Initiative, visit the following Web site: <http://www-03.ibm.com/autonomic/>

- ▶ For clients who want the flexibility to customize systems in their environment, IBM will deliver components that can be easily integrated because based on standards.

What does “Collaborate to Innovate” means?

Collaborate to innovate means:

- ▶ Simplification across the enterprise infrastructure will provide more data security, more business flexibility, more business opportunity.
- ▶ Collaboration with Business Partners, Systems Integrators and Independent Software Vendors will offer complete and quicker business solutions.
- ▶ To accelerate innovation, collaboration with the industry is fundamental.

The IBM Virtualization Engine Version 2, announced on November 2005, is the latest advance in virtualization technology IBM offering. It offers unprecedented interoperability across systems, storage and networking. Open interfaces and implementation of standards allow companies such as Cisco, VMware, Network Appliance and others to participate in this infrastructure. The IBM Virtualization Engine is one of the key components of the On Demand Operating Environment architecture.

1.2 From Web services to the On Demand Operating Environment

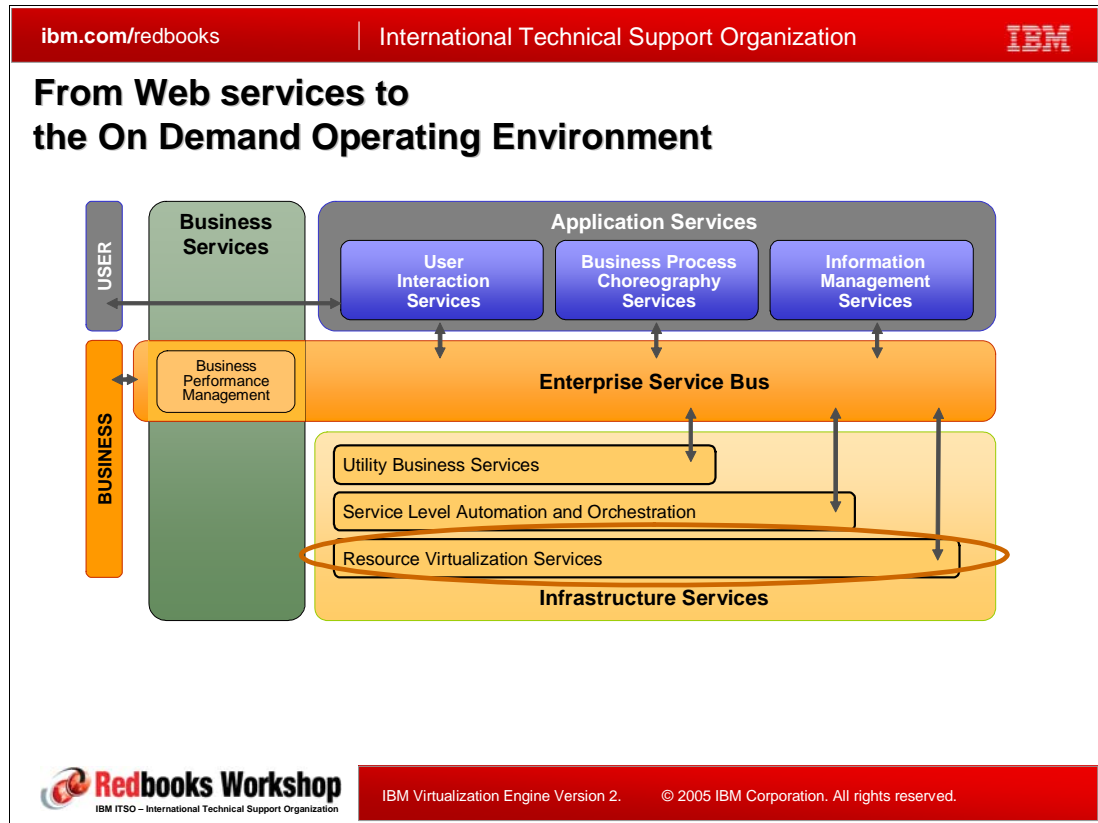


Figure 1-2 From Web services to the On Demand Operating Environment

To understand the role of the Virtualization Engine and its importance, you need to understand first how the On Demand Operating Environment is designed.

The On Demand Operating Environment (ODOE) is based upon the concept of a *Service Oriented Architecture* (SOA) which itself expands the vision of *Web services*. The following sections explain the relationship between these three major entities:

- ▶ The Web services
- ▶ The Service Oriented Architecture
- ▶ The On Demand Operating Environment

Web services

A Web service is a software component designed to support interoperable machine-to-machine interaction over a network. Web services are basically Web-based applications designed to support application to application communication. Unlike typical applications, Web services literally describe themselves to the outside world telling what functions they perform, how they can be accessed, and what kinds of data they require.

They can be developed using any language, and they can be deployed on any platform: from the tiniest device to the largest super computer. This is the fundamental evolution: the service consumer doesn't have to be concerned with the implementation or realization of the service, as long as it supports the required functionality and quality of service.

From a practical point of view, a Web service is a program component deployed within some runtime environment which is responsible for executing the code and for dispatching messages. A Web service is *stateless*, meaning that it is a service whose implementation maintains no dynamic data, but which acts upon resources based on messages it sends and receives. The resources themselves may be *stateful*. A Web service will not maintain a dynamic state: that means it will not maintain a state for which the service is responsible between message exchanges with its requestors. The advantage is that a stateless Web service can be restarted following failure without concern for its history or prior interactions, and more copies can be created or destroyed in response to the changing load. Operations on stateless Web services imply that the necessary parameters are passed on each Web service request or invocation.


Web services standards

Web services technologies are based on XML components specified by independent standards bodies. The current and more mature core Web services standards are SOAP, WSDL and UDDI.

- ▶ XML (eXtensible Markup Language) provides a platform-neutral mechanism to represent data and is also the foundation for all other Web services protocol. One important ramification of the use of XML is that any application, written in any language, running on any platform, can interpret Web services messages, descriptions and discovery mechanisms. No specific middleware is needed and any application can interpret a SOAP message using standard XML protocols.
- ▶ SOAP (Single Object Access Protocol) defines the data communication protocol for Web services. The mechanism by which the data are encoded in XML is totally extensible, and in fact can be specified within each SOAP message. SOAP defines an abstract binding framework that allows SOAP messages to be transferred using a variety of underlying transfer protocols. Additional information can be included with a SOAP message to provide directive or control information to the service, such as security information, transaction context, message correlation information, session indicators, or management information
- ▶ WSDL (Web Services Description Language) is used to describe the interface (the service contract) offered by each individual and indivisible web service and is used to bind the client to the service. A WSDL document describes what functionality a Web service offers (the service type), how it communicates (binding of the abstract interface to a concrete set of protocols), and where to find it (the service implementation).
- ▶ UDDI (Universal Description, Discovery and Integration) plays, as a registry, the role of a service broker, a means to advertise and discover Web services. UDDI is particularly useful if the service consumer doesn't know which type of Web service it wants to use, which service providers provide the service, or where to go to find the services. If a service consumer already knows this information, then it can use a simpler, more direct form of discovery, such as the Web Services Inspection Language (WS-Inspection), an XML format that can be used to inspect a Web site for available Web services. UDDI itself is a Web Service and users communicate with it using SOAP messages. UDDI manages information about service types (a pointer to the WSDL document that describes the service bindings - the what part) and service providers (a pointer to the WSDL document that describes the service binding- the how part), and it provides mechanisms to categorize, find, and bind to services

These standards are not the only ones used to implement Web services; a more complete list of open standards used by the Web services is shown in Figure 1-3.

ibm.com/redbooks		International Technical Support Organization		IBM
Web Services standards				
Business Domain Specific Extensions	...	Business		
Distributed Management	WSDM,WS-Manageability	Management		
Provisioning	WS-Provisioning			
Security	WS-Security	Security		
Security Policy	WS-SecurityPolicy			
Secure Conversation	WS-SecureConversation			
Trusted Message	WS-Trust			
Federated Identity	WS-Federation			
Portal & Presentation	WSRP	Portal & Presentation		
Asynchronous Services	ASAP	Transaction and Business Process		
Transaction	WS-Transactions,WS-Coordination, WS-CAF			
Orchestration	BPEL4WS,WS-CDL			
Events and Notification	WS-Eventing,WS-Notification	Messaging		
Multiple Messages Sessions	WS-Enumeration,WS-Transfer			
Routing/Addressing	WS-Addressing,WS-MessageDelivery			
Reliable Messaging	WS-ReliableMessaging,WS-Reliability			
Message Packaging	SOAP, MTOM			
Publication and Discovery	UDDI,WSIL	Metadata		
Policy	WS-Policy, WS-PolicyAssertions			
Base Service and Message Description	WSDL			
Metadata Retrieval	WS-MetadataExchange			



IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 1-3 Web services standards

A service oriented architecture

A service-oriented architecture (SOA) is an application framework that takes business applications and breaks them down into individual business functions and processes, called services. An SOA lets you build, deploy and integrate these services independent of applications and the computing platforms on which they run.

When Web services specifications define the details needed to implement services and interact with them, the service oriented architecture is an approach to build distributed systems that deliver application functionality as services to end-user applications. In using SOA to design distributed applications, you can expand the use of Web services from simple models to systems of arbitrary complexity.

SOA aims to develop applications that rely on functions, components and objects. SOA is neither a product nor a platform. SOA defines a set of patterns and guidelines for creating loosely coupled, business-aligned services. Because of the separation of concerns between description, implementation, and binding, this architecture provides unprecedented flexibility in responsiveness to new business opportunities.

SOA views every application or resource as a service implementing a specific, identifiable set of (business) functions. Each element of the architecture is a service that together implements the operating environment capabilities. This is accomplished through integration, infrastructure management, and business driven development capabilities, all interconnected using a service oriented architecture allowing the management of the IT environment in business terms. A service in SOA is an application function packaged as a reusable component for use in a business process; services can be invoked through defined

communication protocols. SOA can be based on Web services, but it may use other technologies instead.

SOA defines an interaction model between three primary parties:

- ▶ The *service provider*, who publishes a service description and provides the implementation for the service.
- ▶ A *service requester*, who can either use the Uniform Resource Identifier (URI) for the service description directly or can find the service description in a service registry then bind and invoke the service. The service requester uses the contract to bind the client to the service.
- ▶ The *service broker* who provides and maintains the service registry. It gives the consumer directions on where to find the service and its service contract.

The service requester has no knowledge of the technical details of the provider's implementation, such as the programming language, deployment platform, etc. The service requester typically invokes operations through messages, a request message and the response, rather than through the use of APIs or file formats. The service interaction must be well-defined and using standards is the key element to make work such an architecture. In particular WSDL defines an XML vocabulary for defining the contract between a service requester and a service provider in terms of the request and response messages.

A fundamental advantage of SOA is the decoupling of the service provider and the service consumer through open standards and protocols. Individual software assets become the building blocks to develop other applications. The complexity of systems is reduced by using a common style of interaction that works with both new and legacy code.

The conceptual model of SOA applies to the virtualization of both business functions and physical infrastructure. It spans the construction of applications as well as their deployment and management. Clients only see a collection of business services and are interested in their quality of service; the On Demand operating environment shields them from the details of application assembly and service delivery.

The On Demand Operating Environment

The On Demand Operating Environment consists of three main elements as shown in Figure 1-2 on page 5:

▶ Integration services

The On Demand Business services programming model is based on application development through component (service) assembly. On demand application builders use the services in the integration category to create new business services. The services in this category facilitate integration as well as provide business functions to be integrated. Integration services are hosted by *application services* that provide container facilities to simplify their participation in interactions with other integration services and with On Demand operating environment infrastructure services. On demand integration service developer's focus on realizing the business logic they care about, assembling integration services that provide required business function, and declaring expected quality of service.

▶ Infrastructure services

The services in the infrastructure category provide and manage the infrastructure into which business services and their constituents are deployed. Besides the fact that they implement very different capabilities supporting a variety of On Demand operating environment patterns, the main difference between two categories of services is which user roles build and use them. Middleware providers and ISVs build *infrastructure*

elements, while service oriented infrastructure and application builders build *integration elements*.

Note: The *resource virtualization services*, the core topic of this redbook, are part of these infrastructure services.

► The enterprise service bus

SOA applications are built from services. The enterprise service bus (ESB) is the component that provides access to the services and so enables the building of SOA applications.

Services interact through the Enterprise Service Bus, which facilitates mediated interactions between service end points. The ESB supports event-based interactions as well as message exchange for service request handling. The ESB provides a common model for messages and events. All messages can become events if the deploying service binds the message to a topic in the event space.

The ESB is more than just a transport layer: it must provide a mediation support to facilitate service interactions, it must support a variety of ways to get on and off the bus such as adapter support for existing applications or business connections. To support these different ways to get on and off the bus, it must support service interaction with a wide variety of service endpoints. It is likely that each endpoint will have its own integration techniques, protocols, security models and so on. This level of complexity should be hidden from service consumers.

Without an ESB infrastructure, any service consumer that needs to invoke a service needs to connect directly to a service provider using the protocol, transport, and interaction pattern that is used by the provider.

In an ESB, there is no direct connection between the requestor and provider. Requestors access the ESB to invoke services, and the ESB acts as an intermediary by passing the request to the provider using the appropriate protocol, transport, and interaction pattern for the provider. With an ESB, the infrastructure shields the consumer from the details of how to connect to the provider.

One of the most important insights of the On Demand Operating Environment is that a *common pattern* supports both application services and infrastructure services.

The three facets of the On Demand Operating Environment (integration services, infrastructure services and Enterprise Service Bus) provide the capabilities required to enable business performance management, or, management of business services to meet business goals; in particular:

- Integration services are instrumented to produce business events.
- Infrastructure services produce IT-level events reporting the status of resources business services use that can be correlated to business events produced by those services.

Both kinds of events (infrastructure and business related) are encoded using a common event description format and therefore can be propagated through the infrastructure and correlated, or aggregated, independently of their source.

1.3 The On Demand Operating Environment architecture details

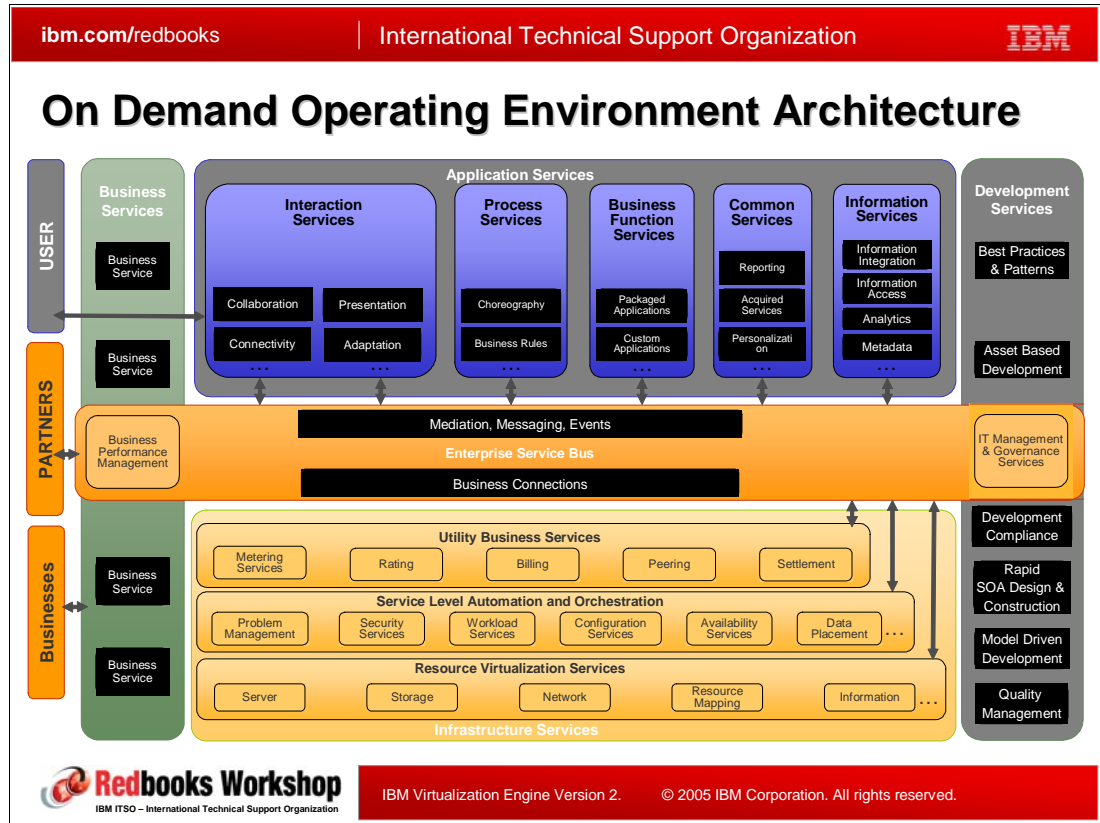


Figure 1-4 The On Demand Operating Environment detailed architecture

This section provides more details about the ODOE architecture and the infrastructure services where the *virtualization services* are implemented. You can find more detailed information about the application services, the development services and the business, user and partners services in the following books:

- ▶ *Patterns: Implementing an SOA using an Enterprise Service Bus*, SG24-6346.
- ▶ *Patterns: Integrating Enterprise Service Buses in a Service-Oriented Architecture*, SG24-6773
- ▶ *Patterns: Implementing Self-Service in an SOA Environment*, SG24-6680
- ▶ *Patterns: SOA with an Enterprise Service Bus in WebSphere Application Server V6*, SG24-6494

The services in the infrastructure category provide and manage the infrastructure represented by a pool of resources into which business services are deployed. Infrastructure services realize the facilities the service oriented infrastructure provides. Infrastructure services live in application containers like integration services. The infrastructure services are constructed using the same principles as business services: process choreography, adapters, business rules, etc. These services are categorized into three different layers, as shown in Figure 1-4.

Utility business services

The *utility business services* define what use of which resource is to be metered, defined and allowed for rating packages to be applied. They support functions like billing, metering, rating, peering, settlement.

- ▶ The rating/billing component provides the ability to translate technical measurements into monetary units and then bill the subscriber appropriately.
- ▶ Metering provides the ability to enable the collection of information related to the subscriber's service and resource utilization.
- ▶ Peering/settlement perform accounting information for resources and services the business service uses for itself.

Service-level automation and orchestration services

The *service-level automation and orchestration services* enable the system resource to self-configure, self-heal, self-optimize and self-protect themselves. these functions are enabled by a set of services provided by autonomic managers and resource managers. The following list describes the services associated with service-level automation and orchestration.

- ▶ Availability services let you manage the availability of the various infrastructure components based on service-level agreements. It includes solutions such as HACMP™, backup and restore, data mirroring, etc. The overall objective is to provide a policy-declared level of resilience.
- ▶ Configuration services provide the ability to adapt to changes in the infrastructure based on goals and policies.
- ▶ Problem management services provide appropriate logging and tracing facilities to be able to find out non apparent reasons.
- ▶ Security services provide the ability to achieve self-protection. Intrusions must be detected and the systems have to react dynamically and prevent damage.
- ▶ Workload services provide the ability to balance performance and workload when the load exceeds the installed capacity based on service level agreements and business policies.
- ▶ Data placement services handle creation, maintenance, and destruction of caches and replicas of data, based on placement policies.

Resource virtualization services

Resource virtualization services enable a level of indirection between applications use of physical resources and the actual allocation of physical resources underneath. They are supported by standards. they are used by the service-level automation and orchestration layer.

Resource virtualization services enable a level of indirection between applications' use of physical resources and the actual allocation of physical resources underneath. They are supported by standards like Web Services Distributed Management (WSDM) and a set of XML vocabularies that manage Web services and distributed resources. These services are used by the service-level automation and orchestration layer.

Examples of resource virtualization services are:

- ▶ Server virtualization services enable the virtualization of the physical servers across the infrastructure, using capabilities such as clustering, partitioning, and virtual machines.
- ▶ Storage virtualization services enable the virtualization of the persistent information across the distributed environment, using capabilities such as data grids.
- ▶ Network virtualization services enable the virtualization of the physical network, using capabilities like VPN, VLAN, and so on.
- ▶ Resource mapping services are responsible for extraction of customer legacy resource repositories and the transformation to canonical resource types for resource composition

in support of business service deployment and also autonomic management interaction with on demand resources during runtime.

- ▶ Information services provide the means to manage and exploit information represented by documents, e-mail, images, music, and other unstructured data. These services coordinate and enforce policies describing the life cycle and management of the information including access, security, versioning, archiving, and retention.

1.4 Standards: from OGSi to WSDM

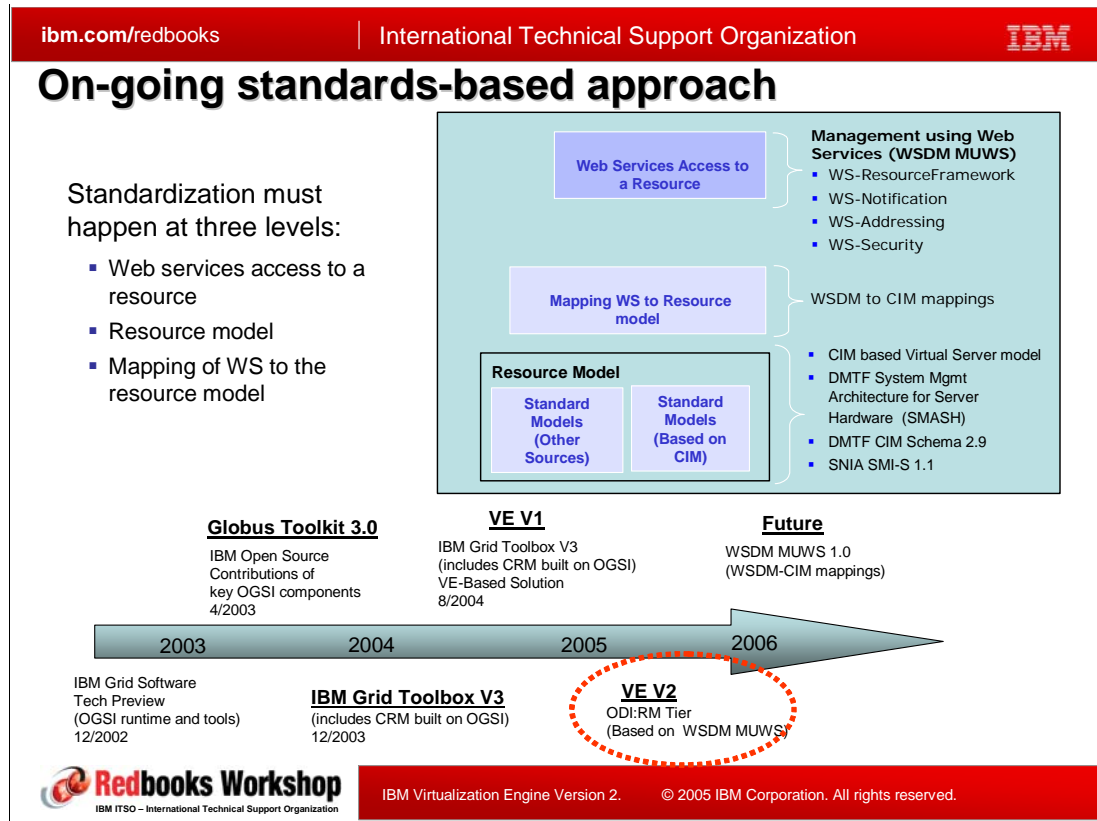


Figure 1-5 Open standards evolution

As already said, standards are critical to enabling the IBM On Demand Business strategy and the virtualization services because they provide the rules that ensure interoperability between IBM products, the products of other vendors and systems, processes, services and so on. This is important to understand why we want to use these standards, how they evolved and where they are going. This section describes the evolution of the standards and which ones are the most important and why.

First, do not confuse Open Standards with Open Source² even most Open Source projects are related to one or more Open Standards. In most instances, Open Source projects are the first to provide a reference implementation to a specification or standard proposed by an Open Standards body. Open standards promise interoperability and portability; they provide customers with choice, flexibility, speed, and skills; they are chaperoned by independent bodies like IETF, W3C, IEEE, OASIS, JCP, OMG, ECMA etc.

The following sections describe how the standards evolved and why they are all important for the On Demand Operating Environment architecture.

Open Grid Services Infrastructure

Grid technology was among the first to promote a virtual computing environment through the use of a set of services which represented the abstraction of heterogeneous resources. This is why the Global Grid Forum³ (GGF) has developed the Open Grid Services Architecture (OGSA); OGSA defines a standard architecture and a set of services for use in grid

² Open Source Initiative (OSI) is a non-profit corporation dedicated to managing and promoting the Open Source Definition for the good of the community (more on the following Web site: <http://www.opensource.org/>).

applications; OGSA uses Web services as its main technology to enable grid services. Grid services provide for the controlled management of distributed and often long-live state objects.

The Open Grid Services Infrastructure (OGSI), also developed by the GGF, is the base infrastructure on which OGSA is built. OGSI defines the conventions and specifications of the grid services, their protocols and interface' behavior, properties and attributes, based on Web services specifications plus many extensions.

OGSI is concerned primarily with creating, addressing, inspecting, and managing the lifetime of stateful Grid services. OGSI defines a component model by using extended WSDL and XML schema definition improving new services in order to fit grid needs. OGSI introduces standard factory and registration interfaces for creating and discovering Grid services.

One extension proposed by OGSI to the Web services community was the mechanism of Service Data whose role is to expose the instance of a service to requestors, providing a *stateful* system. Stateful services allow service consumers to query and to manipulate service data that persist across service interactions

For more information about Grid services, you can refer to *Grid Services Programming and Application Enablement*, SG24-6100.

WS-Addressing

SOAP did not provide a standard way to specify where a message is going, how to return a response, or where to report an error. The Web Service Definition Language (WSDL) 2.0, defined a new specification, the *WS-Addressing specification*. The WS-Addressing specification defines a standard for incorporating message addressing information into web services messages. WS-Addressing also defines a standard for including service-specific attributes within an address for use in routing the message to a service or for use by the destination service itself.

Specifically, this specification defines XML elements to identify Web service *endpoints* and to secure end-to-end endpoint identification in messages. This specification enables messaging systems to support message transmission through networks that include processing nodes such as endpoint managers, firewalls, and gateways in a transport-neutral manner.

WS-Addressing defines two interesting interoperable constructs: endpoint references and message information headers (or message addressing properties).

- ▶ A Web services *endpoint* is a referenceable entity or resource where Web services messages can be targeted. Endpoint references convey the information needed to identify/reference a Web services endpoint, and may be used in several different ways: for example they can be used to provide addresses for individual messages sent to and from Web services. The only required element of an endpoint reference is the address. The other elements: reference properties, reference parameters, port type, service name, and policy elements (defined by the WS-Policy specification). are all optional.
- ▶ To deal with this last usage case, this specification defines a family of *message information headers* that allows uniform addressing of messages independent of underlying transport. These message information headers convey end-to-end message characteristics including addressing for source and destination endpoints as well as message identity. In particular, the WS-Addressing specification defines a reply-to address as an extension of the SOAP envelope header element. The service requestor enqueues a request to the port specified in the WSDL document but doesn't wait; it can process other work until the response arrives. When the service provider is ready to send the response, it uses the

³ The Global Grid Forum (GGF) is the community of users, developers, and vendors leading the global standardization effort for grid computing. The GGF Web site is: <http://www.gridforum.org/>

reply-to address from the SOAP:Header of the request message. The message information headers define the full set of addressing information that can be attached to a SOAP message. Most of the fields are optional; the only required fields are the To and Action fields, each of which specifies a URI; when a reply is expected, a MessageId element must also be present.

A postal mail address provides some analogy: the destination address is written on the face of the envelope and must contain a name, a street, a city, etc. in a specific well known order to be delivered at the right place:

- ▶ The *format* of the address is analogous to the endpoint reference.
- ▶ The *layout* of the address on the envelope is analogous to WS-Addressing's message addressing properties: if you write the city first, then the name, then the street, the post office will not be able to deliver your mail!

WS-RF

If Web services are supposed to be stateless, or considered as a stateless message processor, message exchanges are, in many cases, supposed to enable access/update to state maintained by other system components; those servers, file systems, databases or other entities, can be considered to be stateful resources. The link between one or more stateful resource and a Web service is the *Implied Resource Pattern* to formalize the relationship between Web services and the stateful resources through a conventional use of WS-Addressing.

The WS-Resource Framework (WS-RF) is a refactoring of some OGIS interfaces to produce a framework of independently useful Web services standards. WS-RF is a set of six Web Services specifications that define terms such as the *WS-Resource* approach to modeling and managing state in a Web services context. The WS-RF definition relies upon the WS-Addressing specification. The specifications of these interfaces retain all of the essential capabilities present in OGIS, while changing some of the syntax to exploit WS-Addressing and adopting a different terminology.

The WS-Resource construct is a means of expressing the relationship between stateful resources and Web services. The WS-Resource framework is a set of existing Web services specifications that define a rendering of the WS-Resource approach in terms of specific message exchanges and related XML definitions. These specifications allow the programmer to declare and implement the association between a Web service and one or more stateful resources. They describe the means by which a view of the state of the resource is defined and associated with a Web services description, forming the overall type definition of a WS-Resource. They also describe how the state of a WS-Resource is made accessible through a Web service interface, and define related mechanisms concerned with WS-Resource grouping and addressing.

A WS-Resource can be defined as the composition of a Web service and a stateful resource that is expressed as an association of an XML document with a defined type with a Web services portType, and addressed and accessed according to the implied resource pattern.

WS-RF is concerned primarily with stateful resources; the framework provides the means to express states as stateful resources. The implied resource pattern is a set of conventions on Web services technologies based on the well known specifications: XML, WSDL, and WS-Addressing. A stateful resource that participates in the implied resource pattern is called a WS-Resource. Implicitly it means that the requestor does not provide the identity if the resource has an explicit parameter in the body of the request message. The context used to designate the implied stateful resource is encapsulated in the WS-Addressing endpoint reference used to address the target Web service at its endpoint. WS-RF uses different

constructs to model a stateful resource and a Web service, while OGSF used the same construct for both.

In summary we can say that the WS-RF goal is to allow stateful resources manipulation via Web services operations. WS-RF adopts the endpoint reference construct defined in the WS-Addressing specification as an XML syntax for identifying Web service endpoints. The creation of a WS-Resource in WS-RF terms is really the creation of a stateful Web service in OGSF terms.

Web Services Distributed Management

Web Services Distributed Management (WSDM) allows a resource and all of its services to be manageable in a standard and interoperable manner. The WSDM specifications define:

- ▶ How to use Web services to expose manageable resources, this is the Management Using Web Services (MUWS) specification.
- ▶ How to expose manageable Web service implementations, this is the Management of Web Services (MOWS) specification.

Using both specifications, it is possible to build reliable and accountable management systems. Figure 1-6 describes the global concept of Web services manageability.

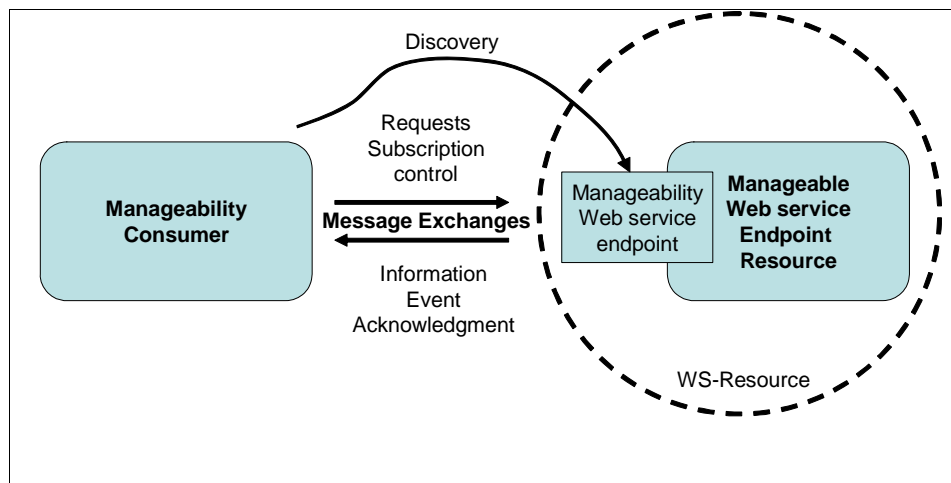


Figure 1-6 Management of Web services concepts

Web Services Distributed Management is a recently approved OASIS⁴ standard (March 2005) to use Web services architecture and technology to manage distributed resource Web services as well as non-Web services infrastructures (in this latter case, a replacement for SNMP).

To manage a Web services environment, the components (the Web services endpoints), have to be accessible and managed. A Web service endpoint provides access to a manageable resource. In order to discover the Web service endpoint providing access to a particular manageable resource, the requestor needs to obtain an *Endpoint Reference* (EPR), as defined by the WS-Addressing specification, and then obtains any other required descriptions, like a WSDL document, an XML Schema, or a policy document. A Web service endpoint providing access to some manageable resource is called a *manageability endpoint*. To exchange messages with a manageability endpoint, a manageability consumer needs to understand all of the required descriptions for the endpoint. The manageability consumer

⁴ The Organization for the Advancement of Structured Information Standards (OASIS) is a not-for-profit, global consortium that drives the development, convergence, and adoption of e-business standards (for more information please refer to the following Web site: <http://www.oasis-open.org>).

sends messages targeted to the manageable resource by using information contained in the EPR, for example, an address and some reference properties.

The WSDM specification defines how the manageability of Web service endpoints and resources exposed as Web services can be accessed via Web services and uses the two following specifications:

- ▶ Management Using Web Services (MUWS) – is a specification that describes how to define, advertise and exercise manageability capabilities through Web services mechanisms, independently of the type of resources managed and the model used to describe these resources

MUWS defines *how to represent* and access the manageability interfaces of resources as web services, and defines a core set of capabilities, such as resource identity, metrics, configuration, relationships, etc. . MUWS uses the same mechanisms, for obtaining EPRs and their associated descriptions, as used by regular Web service implementations, and their applications. MUWS is based on a number of Web services existing specifications.

- ▶ Management Of Web Services (MOWS) – defines the manageability capabilities of Web services. MOWS defines *how to manage* web services as resources, and how to describe and access that manageability using MUWS. This specification focuses on defining manageability capabilities of the Web service endpoints. MOWS also delivers the mechanisms and methodologies that enable manageable web services applications to interoperate across enterprise and organizational boundaries. WOWS can be described as the runtime monitoring and management of SOAP messages.

MOWS is an application of management using MUWS to the resources which are elements of the Web Services Architecture. WSDM specification defines how the manageability of Web service endpoints and resources exposed as Web services can be accessed via Web services.

Figure 1-7 positions the MOWS and MUWS roles specifications from a systems management view.

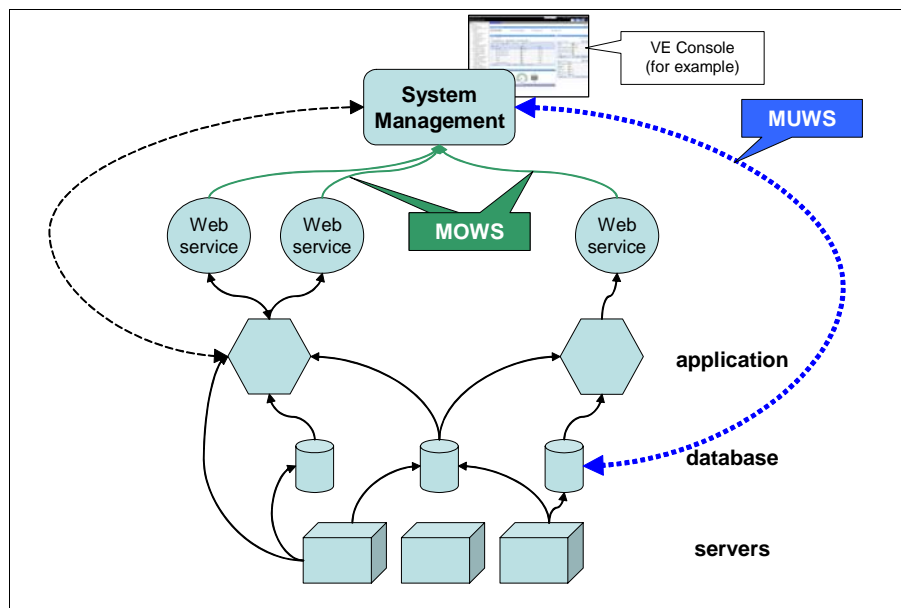


Figure 1-7 The roles of MOWS and MUWS

These standards are particularly important to understand the resource modeling available from the Virtualization Engine and more specific standards are developed in Section 3.2, “A manageable resource” on page 101.

Future perspective

IBM is also involved in the Open Services Gateway Initiative (OSGi⁵) through its Service Management Framework (SMF): SMF is an implementation of the OSGi Service Platform specification and provides a production-ready software management framework for network-delivered applications.

Figure 1-8 describes how to position these standards efforts:

- ▶ Stateful Web services (manageable resources) are hosted in a WebSphere® runtime, using the full WS-RF specifications.
- ▶ OSGi/SMF targets the environment where the size of the endpoint implementation needs to be limited and where a stateless Web service is enough to provide the functionality.

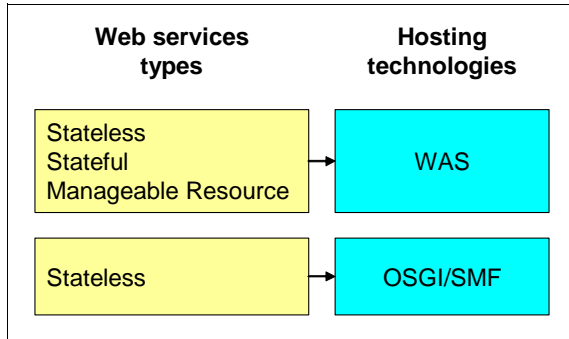


Figure 1-8 OSGI positioning

Another direction is to expose Common Interface Model (CIM) resources as Web services, as described in Section 3.2.2, “Exploiting standards” on page 102.

Note

Figure 1-5 on page 13 mentions other important organization or standards:

- SNIA SMI-S:** The Storage Networking Industry Association (SNIA) is an association of producers and consumers of storage networking products whose goal is to further storage networking technology and applications. SMI-S is a standard storage management interface developed by SNIA.
- DMTF SMASH:** The DMTF's Systems Management Architecture for Server Hardware (SMASH) initiative is a suite of specifications that deliver architectural semantics, industry standard protocols and profiles to unify the management of the data center. (<http://www.dmtf.org/standards/smash/>)
- DMTF CIM:** The DMTF Common Interface Model (CIM) provides a common definition of management information for systems, networks, applications and services, and allows for vendor extensions. CIM's common definitions enable vendors to exchange semantically rich management information between systems throughout the network

⁵ The Open Services Gateway Initiative (OSGi) was founded in March 1999. Its mission is to create open specifications for the network delivery of managed services to local networks and devices. The OSGi organization is the leading standard for Internet services to homes, cars, small offices, and other environments. To know more you can refer to the following Web site: <http://www.osgi.org>

CIMOM: CIM Object managers (CIMOM) are implementations but not part of the CIM standard; it is used to describe server implementations providing access to instances of CIM models.

1.5 The Virtualization Engine platform

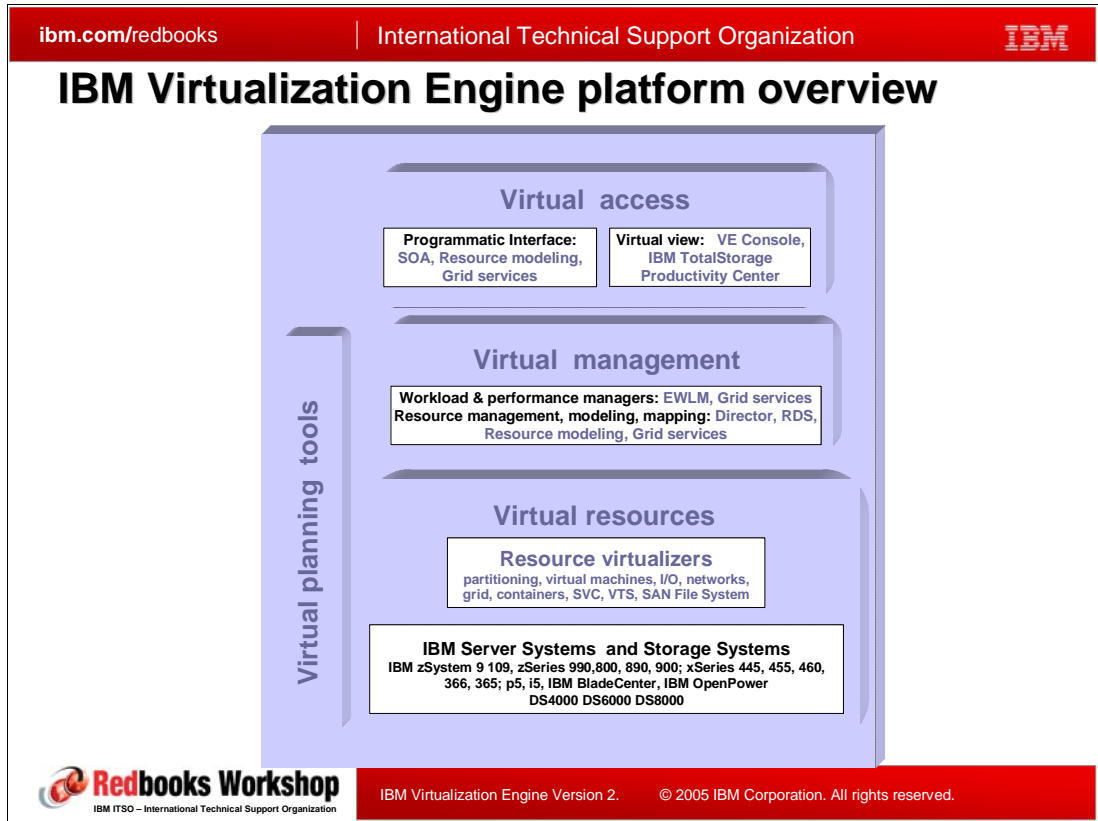


Figure 1-9 The Virtualization Engine platform

The Vision

In the on demand environment, the real benefits of virtualization will lie beyond the individual resources, servers and devices. Customers need the ability to manage non homogeneous infrastructures as single logical units with better automation that span multiple types and layers of hardware and operating systems. More virtualization features can and will be imbedded in the physical layer (hardware and hypervisor): alone they will not be able to provide all the benefits of the virtualization capabilities.

On July 2005, when the IBM Systems Agenda was unveiled, as described in Section 1.1, "The IBM Systems Agenda" on page 2, IBM expanded the traditional view of virtualization.

Virtualization is an heterogeneous IT Infrastructure dynamically provisioned, managed by business rules, where work is scheduled and monitored as it flows through all layers of the system, from the network, through firewalls, web servers, application servers, database servers and back.

Add to this dynamic infrastructure an event driven, autonomic, management system based on a SOA, which can discover resources, monitor and visualize their status, respond to changes in hardware configuration and that can exploit both IBM and non-IBM systems. Combine this with access and management driven by industry standard web services interfaces, and you have the "Virtualize Everything" IBM idea.

The traditional view tends to focus on partitioning; this view broadens and expands virtualization to:

- ▶ Allow multiple systems to come together and perform together in a larger context.
- ▶ Give a common way to address component-level devices .

A large system can be made to look like many smaller systems, to include capabilities to make many smaller components (systems and devices such as storage and networking) look and perform as one. This expanded definition of virtualization is delivered through the consistent use of open standards—the foundation of the IBM Virtualization Engine platform.

Until now virtualization has been used primarily to increase hardware utilization; virtualization will play a more and more major role in simplifying IT management; in particular it will be extended from single servers to scale-out aggregations of servers, storage, and network components: making many small systems look like one from a management perspective.

IBM is using this new view to take the next steps in virtualization by transforming the Virtualization Engine into an integrated platform of technologies that broadens and enhances the current capabilities. The IBM Virtualization Engine platform includes a collection of management and performance services and consistent, comprehensive access and views. This platform approach supports a business policy-based infrastructure that allows for:

- ▶ Flexible starting points based on current and future business and IT requirements.
- ▶ Virtualizing systems resources to ease systems management.
- ▶ Many options that recognize one size does not fit all.
- ▶ Virtual management applied to systems that make many small resources look like one system.
- ▶ Consistent expansion—add what is needed when it is needed.
- ▶ Open interfaces and standards to allow partners to participate in the infrastructure.

Virtual resources

The Virtualization Engine platform is based on virtual resources that are included in all IBM server and storage systems, as well as select server and storage systems from other vendors such as HP, Sun™ and EMC. Just a few of the technologies available with most systems include:

- ▶ Hypervisor: hypervisors support partitioning and dynamic resource movement across multiple operating system environments.
- ▶ VLAN: Virtual Local Area Network, VLAN, helps provide network virtualization capabilities that allow you to prioritize traffic on shared networks.
- ▶ Virtual I/O: Virtual I/O helps provide the ability to dedicate I/O adaptors and devices to a virtual server, allowing the on demand allocation and management of I/O devices.

These embedded technologies allow resource virtualizers to facilitate the virtualization of resources within a server or storage device. They prepare the virtual resources for the management and performance capabilities provided by the Virtualization Engine management collection.

Resource virtualizers include:

- ▶ Virtual machines.
- ▶ IBM Director.
- ▶ Grid technologies.

- ▶ SAN Volume Controller and SAN File System.
- ▶ Third-party virtualizers such as VMware and Cisco.

Resource virtualizers enable:

- ▶ Systems management ease of use capabilities.
- ▶ More robust network capabilities.
- ▶ Greater flexibility in storage virtualization deployment and management.

Chapter 9, “Virtual Resources: Server Specific virtualization functions” on page 325 describes the most important Virtual Resources components.

Virtual management

The Virtualization Engine management focus is to provide workload and performance managers, and resource management tools in relation with the resource virtualizers.

It includes three *products*:

- ▶ A new version of the IBM Director, described in Chapter 2, “IBM Director 5.10” on page 37.
- ▶ A new product, Resource Dependency Services (RDS) described in Chapter 3, “Resource Dependency Service” on page 97.
- ▶ A new version of the IBM Enterprise Workload Manager (EWLM), described in Chapter 5, “Enterprise Workload Manager” on page 173.

They all focus on simplifying and optimizing the operation of virtual systems.

Virtual access

The IBM Virtualization Engine's virtual access provides a virtual view and a programmatic interface.

- ▶ The virtual view includes a new version of the Virtualization Engine console which is a single Web-based interface for administering all virtual resources such as resource health, monitoring, and integration access to planning tools. The new version of the Virtualization Engine is described in Chapter 4, “The Virtualization Engine console” on page 133.
- ▶ The programmatic interface offers the capability to access system resources as services. This allows access to the system infrastructure through a Service Oriented Architecture programming model. The key programmatic interfaces used in the Virtualization Engine platform today provide standards at the following levels:
 - Web services access to a resource.
 - Mapping of Web services to the resource model.
 - The resource model.

Virtual planning tools

These tools help you plan and build your virtualized environment.

Workload Estimator

The IBM eServer Workload Estimator provides sizing recommendations for a number of workloads for the iSeries, pSeries and xSeries . The tool provides projections for simple to complex topologies including single server or partitioned deployments. Projections can be made for new or existing workloads. The Estimator will recommend the model, memory and DASD requirements that are necessary to meet reasonable performance expectations. The IBM eServer Workload Estimator is available in 2 forms.

- ▶ The on-line version can be accessed at:

<http://www.ibm.com/eserver/series/support/estimator>.

This is the recommended access method.

- ▶ There is also a download form available from:

<http://www.ibm.com/eserver/series/support/supporthome.nsf/Document/16533356>.

Planning advisor

The Virtualization Engine planning advisor is an interactive tool that produces a customized plan based on your input. The planning advisor will allow you:

- ▶ To identify what systems services you want to install.
- ▶ To guide you for the design of your Virtualization Engine topology.
- ▶ To get the information to order the Virtualization Engine.
- ▶ To understand the prerequisites you need before installing the components.
- ▶ To generate a customized installation flow charts.
- ▶ To print and record your customized plan before beginning the installation process.

Based on your input, the planning advisor generates recommendations to help you:

- ▶ Order: the planning advisor interview helps you decide which systems services to include in your Virtualization Engine management collection.
- ▶ Deploy: once you determine which systems services meet your business needs, the planning advisor recommends how to deploy the components of the Virtualization Engine across your enterprise topology.
- ▶ Install: finally, the planning advisor generates customized documentation to help you prepare your environment.

Section 8.2, "Planning advisor" on page 280 describes the planning advisor of the IBM Virtualization Engine Version 2.

1.6 Virtual Resources

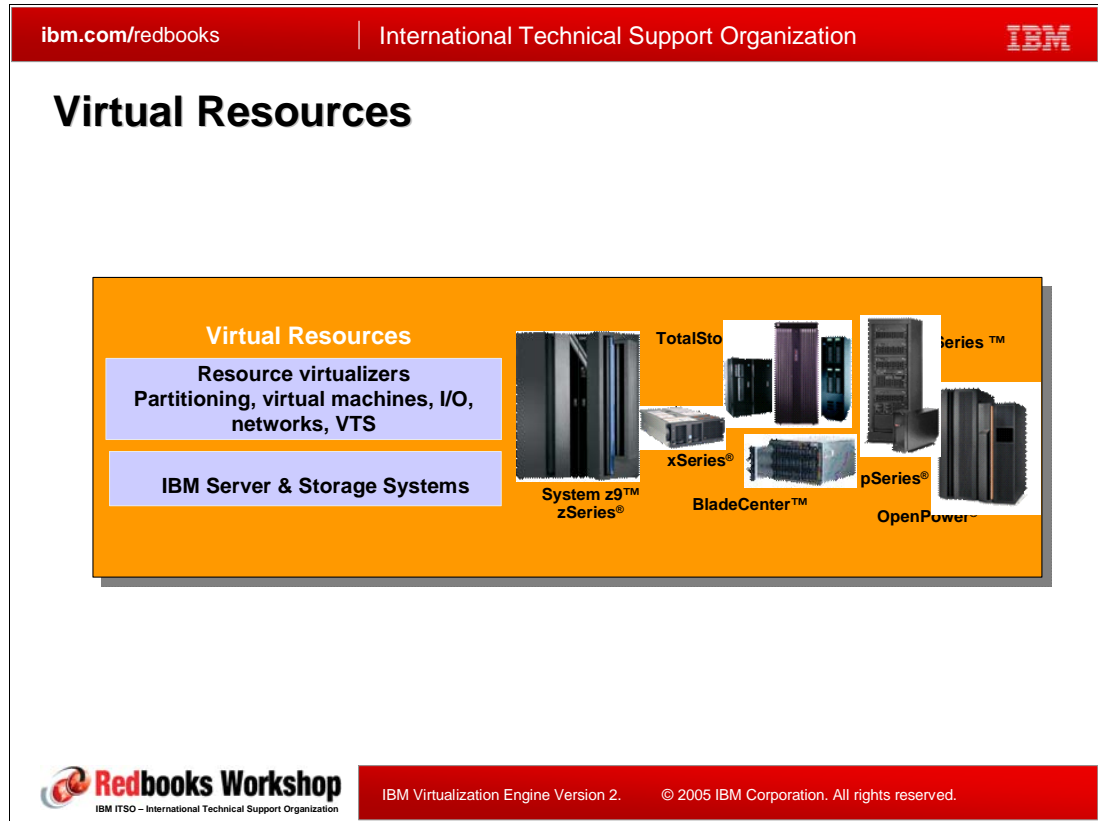


Figure 1-10 IBM Virtualization Engine Platform: Virtual Resources

Virtual resources are virtualization technologies integrated within all IBM Systems hardware (IBM servers, storage systems and network) that help simplify the infrastructure. They are the primary technologies to maximize the infrastructure availability and to make a better use of the resource utilization. These technologies, usually, are the starting point to consolidate the components of the infrastructure. They cover physical resource virtualization technologies as well as operating system virtualization technologies.

Physical resource virtualization

Physical resource virtualization technologies can be found at different infrastructure layers: at the server, the storage, and the network layers.

- ▶ *Server virtualization* allows a single server to be used by multiple applications, middleware and operating systems concurrently. Some of the technologies have been available for years though they continue to evolve. They include virtual memory, physical partitioning, logical partitioning, I/O virtualization, etc. The Integrated Virtualization Manager (IVM), the Virtual I/O Server (VIOS), and the Hosting Partition Infrastructure are examples of the evolution of these technologies.
- ▶ *Storage virtualization* enables customers to make changes in the storage infrastructure with no or little impact to applications; this is of a particular importance in regards to the increase in storage needs. The storage virtualization technologies cover all the storage infrastructure levels: they include technologies at the disk controller level, at the block level, at the file level and at the tape level.

- ▶ *Network virtualization* allows pooling and sharing of network components to make communication across IT infrastructures more efficient and more secure. The network virtualization technologies include Open Systems Adapter Express cards(OSA), Virtual Ethernet, and Virtual IP Addressing (VIPA) facilities.

Operating System virtualization

Two virtualization technologies are mainly used, depending on the platform:

- ▶ The z/VM system, available on the IBM @server zSeries for more than 30 years is very dynamic. It allows real and virtual resources to be shared and switched dynamically without outages between virtual machines; several hundreds of virtual machines can be hosted on a single.
- ▶ VMware⁶ provides a set of tools, very similar to the z/VM facilities, for partitioning, consolidating and managing IBM @server xSeries and IBM BladeCenters. Though the idea is the same as z/VM, given the difference in the size of the target box, the facilities are not totally equivalent.

Some of these technologies are detailed in Chapter 9, “Virtual Resources: Server Specific virtualization functions” on page 325.

⁶ More information can be found at the following Web site: <http://www.vmware.com/>

1.7 The Virtualization Engine management collection

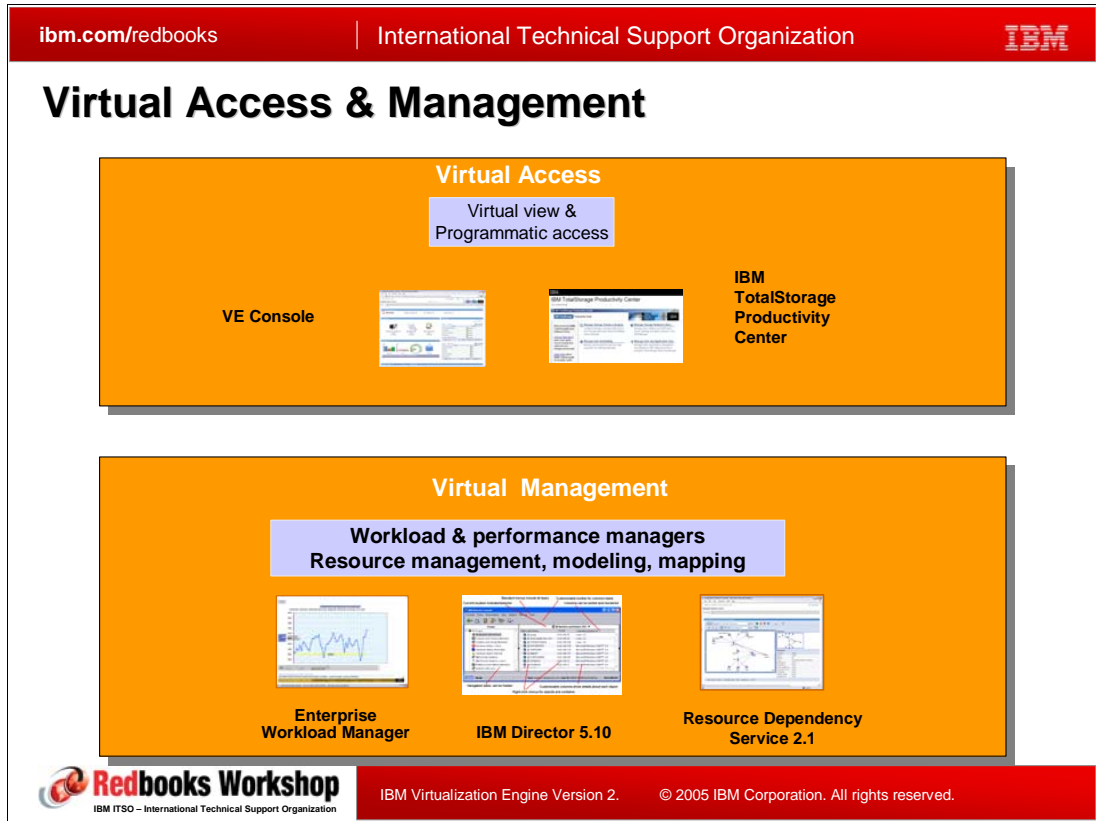


Figure 1-11 IBM Virtualization Engine Virtual Access and Virtual Management components

The Virtualization Engine management collection includes all the components of the virtual management and virtualization access aspects described previously. The goal is to provide common tools across the heterogeneous infrastructure.

The tools include the following products:

- ▶ The IBM Director provides a common, consistent solution for managing hardware and virtual partitions not only for xSeries platform support; the hardware configuration of systems can be remotely discovered, monitored and automated. It can be bridged with the Virtualization Engine console, offering one single console focal point for administrators. The IBM Director is described in Chapter 2, "IBM Director 5.10" on page 37.
- ▶ The Resource Dependency Service (RDS) provides a logical representation of hardware and software resources using a model based on open standards using Web services. Resource Dependency Service is described in Chapter 3, "Resource Dependency Service" on page 97.
- ▶ The Virtualization Engine console provides a single portal for administering the virtual resources; it allows a consolidated view of the performance, health, and topology across multiple heterogeneous operating systems. The Virtualization Engine console is described in Chapter 4, "The Virtualization Engine console" on page 133.

- ▶ Enterprise Workload Manager (EWLM) provides a consolidated end-to-end view of transactions performance and supports dynamic adjustments to CPU allocations across logical partitions (LPARs) for Power5 servers.

EWLM is described in Chapter 5, “Enterprise Workload Manager” on page 173.

The intent is to deliver all the functions on all the eServer platforms; today the following functions are available:

- ▶ iSeries
 - EWLM domain manager and managed nodes; EWLM is able to manage LPARs based on business goals
 - IBM Director
 - Resource Dependency Service for Linux
 - The Virtualization Engine console
- ▶ pSeries
 - EWLM domain manager and managed nodes; EWLM is able to manage LPARs based on business goals
 - IBM Director
 - Resource Dependency Service for Linux
 - The Integrated Virtualization Manager (IVM) on select systems
 - The Virtualization Engine console
- ▶ xSeries and BladeCenter
 - EWLM
 - IBM Director
 - Resource Dependency Service for Linux
 - Virtual Machine Manager
- ▶ On the zSeries:
 - EWLM domain manager on z/OS and Linux; EWLM managed nodes for z/OS and Linux
 - IBM Director with the Virtualization Engine console for Linux
 - Resource Dependency Service for Linux
- ▶ Storage

Restriction: Though the IBM TotalStorage components provide integrated functions with the Virtualization Engine, this redbook does not provide many details about the storage features.

1.8 What's new with the Virtualization Engine Version 2

ibm.com/redbooks
International Technical Support Organization

What's new in 2H?

- **Integrated Virtualization Manager**
 - Simplified partition creation & management
- **IBM Director**
 - New capabilities & platforms
- **Enterprise Workload Manager**
 - Partition management based on business goals
 - Additional Platform support (z/OS, Linux, and HP/UX)
- **Resource Dependency Services**
 - Map resources to workloads
- **Virtual Tape for Open (LTO)**
- **NetApp SVC Interoperability**

IBM ITSO – International Technical Support Organization
IBM Virtualization Engine Version 2.
© 2005 IBM Corporation. All rights reserved.

Figure 1-12 What's new in the IBM Virtualization Engine Version 2, compared to Version 1

The IBM Virtualization Engine Version 2 is the follow-on to the IBM Virtualization Engine Version 1 available since August 2004.

Since August 2004, IBM had two offerings:

- ▶ The *IBM Virtualization Engine, for Systems*, which focused on utilizing the technologies integrated into the IBM eServers, such as Hypervisor, VLAN, Virtual I/O, etc.
- ▶ The *IBM Virtualization Engine, for Enterprise* which provided a collection of multiplatform system tools to help simplify the management and utilization of IT resources. such as Enterprise Workload Manager, Tivoli Provisioning Manager, Director Multiplatform, Virtualization Engine console and the IBM Grid Toolbox V3 for Multiplatforms.

In November 2005, IBM announced the IBM Virtualization Engine Version 2 that is an evolution of the two previous offerings (sometimes they were called *suites*). The following products are now delivered:

- ▶ Enterprise WorkLoad Manager (EWLM) Version 2.1, a new version of EWLM, delivers new operating systems support and workload management on pSeries systems.
- ▶ IBM Director Version 5.10, a new version of IBM Director, integrates functions of IBM Director and IBM Director Multiplatform.
- ▶ Resource Dependency Services (RDS) is a new component of the IBM Virtualization Engine.
- ▶ The Virtualization Engine console 2.1 extends the functions already available in Version 1 and offers new facilities, integrating more products.

28 IBM Virtualization Engine Platform Version 2

Not delivered within the IBM Virtualization Engine platform Version 2, the new announced following components however deliver new virtualization facilities:

- ▶ Integrated Virtualization Manager (IVM) provides the LPAR support on IBM @server pSeries without a physical Hardware Management Console (HMC). With the Integrated Virtualization Manager, IBM is simplifying p5 systems virtualization deployment. Using a new wizard-driven, browser-based user interface, clients can create a MicroPartition ready for installation of either AIX 5L or Linux operating system. IVM comes packaged at no additional cost with the optional purchase of Advanced POWER Virtualization on IBM's p5 servers. IVM is designed to reduce the complexity and time involved to setup and configure multiple partition environments, using IBM's Micro-Partitioning capabilities, in small environments.

IVM is detailed in Chapter 6, "Integrated Virtualization Manager for pSeries and Virtual Partition Manager for iSeries" on page 215.

- ▶ The TS7510 Virtualization Engine is a high-performance, high-capacity open systems virtual tape product designed to augment the tape backup and restore process in large tape environments. The TS7510 is designed to aid in the tape backup and restore process by helping to provide customers significant operational and throughput efficiencies via tape virtualization, namely: The Virtualization Engine TS7510 is a new member of the family of IBM Virtualization Engine for storage offerings designed to help enhance Information Lifecycle Management by improving the utilization of tape resources. The TS7510 Virtualization Engine combines hardware and software into an integrated solution enabling tape virtualization for open systems servers connecting over Fibre Channel physical connections.

The Virtualization Engine TS7510 Software V1.1 (5639-CC7) provides the software for the Virtualization Engine TS7510 and for a management console. The TS7510 utilizes IBM technology and software to emulate IBM tape libraries, drives, and media. This emulation is designed to help customers achieve the following operational and throughput efficiencies:

- Backup window reduction
- Restore time reduction
- Data sharing and resource virtualization facilitation
- Operational efficiencies
- Low total cost of ownership (TCO)
- Improved sharing of tape libraries across applications and servers

- ▶ The IBM TotalStorage® SAN Volume Controller (SVC) is a key component in the IBM virtualization storage technologies. At the heart of SAN Volume Controller's many benefits is its ability to consolidate devices from various suppliers, in different locations, into a single view of storage resource.

In particular, in 1Q2005, IBM announced a partnership with Network Appliance (NetApp⁷). This partnership wants to promote the integration of Network Appliance's software applications with IBM's Tivoli Storage Manager product line, a backup and recovery package designed specifically for use with partner's products.

Tivoli Provisioning Manager (TPM) and the IBM Grid Toolbox V3 for Multiplatforms are not delivered with the Virtualization Engine platform Version 2:

- ▶ TPM, a key part of the ODOE architecture, will interact with the virtualization components; it is distributed through the IBM Software Tivoli brand.
- ▶ The IBM Grid Tool box can be downloaded from the following Web site:

<https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=grid>

⁷ More information at the following Web site: <http://www.netapp.com/>

1.9 The Virtualization Engine Version 2 products offerings

ibm.com/redbooks
International Technical Support Organization

IBM Virtualization Engine Products

- **IBM Virtualization Engine Enterprise Workload Manager V2.1**
 - 5733-EWA IBM Virtualization Engine Enterprise Workload Manager for i5/OS V2.1
 - 5733-EWM IBM Virtualization Engine Enterprise Workload Manager V2.1
 - 5655-EWM IBM Virtualization Engine Enterprise Workload Manager for z/OS V2.1
 - 5648-F08 IBM Virtualization Engine and Infrastructure Services for Linux on System z9 and z/Series, V2.1
- **IBM Director 5.10 with Virtualization Engine Console**
 - 5733-DR1 IBM Director with Virtualization Engine Console for i5/OS, V5.10
 - 5733-DIR IBM Director with Virtualization Engine Console, V5.10
 - 5648-F08 IBM Virtualization Engine and Infrastructure services for Linux on System z9 and z/Series, V2.1
- **IBM Virtualization Engine Resource Dependency Service V2.1**
 - 5733-RDS IBM Virtualization Engine Resource Dependency Service for Linux on Power, V2.1
 - 5648-F08 IBM Virtualization Engine and Infrastructure Services for Linux on System z9 and z/Series, V2.1

IBM ITSO - International Technical Support Organization
IBM Virtualization Engine Version 2.
© 2005 IBM Corporation. All rights reserved.

Figure 1-13 IBM Virtualization Engine Version 2 products

The following products are now available.

To get EWLM

- ▶ 5733-EWM IBM Virtualization Engine Enterprise Workload Manager, V2.1.0:
 - It provides the EWLM domain manager and the common runtime components for AIX or Linux on Power, or Windows or Linux on xSeries hosting server.
 - It includes the EWLM managed agents for AIX, i5/OS , Linux, and HP-UX.
 - A specific feature includes the EWLM managed agents for Windows and Solaris .
 - This product replaces the 5724-I71 product, which was available with the Virtualization Engine Release 1.

Note: The 5660-EWM, 5661-EWM, 5662-EWM, 5664-EWM, are maintenance-product specific names which include the Enterprise Workload Manager with different maintenance durations.

- ▶ 5733-EWA, IBM Virtualization Engine Enterprise Workload Manager for i5/OS, V2.1:
 - It includes the EWLM domain manager, the common runtime components and the EWLM managed agents for AIX, i5/OS and Linux inside the same iSeries box (no agents on external servers).
 - It is included in iSeries Enterprise Edition.
 - This product replaces the Virtualization Engine for iSeries option 5733-Virtualization Engine1.

- It is a \$0-priced product (the support is fee-based).

Note: The 5660-EWA, 5661-EWA, 5662-EWA, 5664-EWA, are maintenance product specifics names which include the Enterprise Workload Manager for i5/OS with different maintenance durations.

- ▶ 5655-EWM, IBM Virtualization Engine Enterprise Workload manager for z/OS, V2.1:
 - It includes the EWLM domain manager, the common runtime components and the EWLM managed agents for z/OS.
 - To obtain the EWLM agents for managing AIX, i5/OS, Linux, HP-UX, Solaris, or Windows, you need to order the specific features of the 5648-F08 product, described below.
 - It is a \$0-priced product (the support is fee-based).
- ▶ 5648-F08, IBM Virtualization Engine and infrastructure services for Linux on System z9™ z/Series, V2.1; under this product name the Virtualization Engine components have separate and optional feature codes:
 - It includes the IBM Director V5,10 (the IBM Director console), the Virtualization Engine console, the IBM Director Agents, the management access point interface to z/VM and the common runtime components. Altogether these components are named the *base components*.
 - A specific priced feature includes the EWLM managed agents for AIX, i5/OS, Linux, and HP-UX.
 - A specific priced feature includes the EWLM managed agents for Windows and Solaris.
 - Other priced features can include Resource Dependency Service and the IBM Director extensions.
 - The base component is a \$0-price based product (fee-based for Service and Support).

To get IBM Director

- ▶ 5733-DIR, IBM Director with Virtualization Engine console, V5.10:
 - It provides the IBM Director server for AIX, or Linux on Power, or Windows or Linux on xSeries (depending on the feature you select).
 - It includes the Virtualization Engine console, the common runtime components and all IBM Director agents.
 - This product replaces the 5724-I71 product which was available with the IBM Virtualization Engine Release 1.
- ▶ 5733-DR1, IBM Director with Virtualization Engine console for i5/OS, V5.10, shipped with every i5/OS copy:
 - It includes the IBM Director server, the Director console and the agents for i5/OS.
 - It includes the Virtualization Engine console and the common runtime components.
 - The product is a \$0-price based product (fee-based for Service and Support).
- ▶ 5648-F08, IBM Virtualization Engine and infrastructure services for Linux on System z9 and z/Series, V2.1; under this product name all the Virtualization Engine components have separate and optional feature codes:
 - It includes the Virtualization Engine console, the IBM Director (with the IBM Director console), the IBM Director Agents and the common runtime components. Altogether they are named the *base components*.
 - A specific priced feature includes the IBM Director Extensions.
 - Other priced features can include Resource Dependency Service.
 - The *base components* is a \$0-price based product (fee-based for Service and Support).

Note: IBM Director may be shipped without the Virtualization Engine components using the following product names:

- ▶ 5765-DR1, IBM Director for pSeries, V5.10:
 - It includes the IBM Director server and agents for i/5OS.
 - It includes the Virtualization Engine console and the common runtime components.
 - It is available to IBM eServer pSeries customers at no additional charge
- ▶ 5697-NFD, IBM Director, V5.10, shipped with every xSeries server:
 - It includes the IBM Director server and agents for i/5OS.
 - It includes the Virtualization Engine console and the common runtime components.

Note: The 5660-DIR, 5661-DIR, 5662-DIR, 5664-DIR, are maintenance product specifics names which include the Director with the Virtualization Engine console for POWER with different maintenance durations.

To get Resource Dependency Service

- ▶ 5733-RDS, IBM Virtualization Engine Resource Dependency Service, V2.1:
 - It provides Resource Dependency Service for Linux on POWER or Linux on xSeries (depending on the feature you select).
 - It includes the Resource Dependency Service server, the resource modeling, the Virtualization Engine console and the common runtime components.

Note: The 5660-RDS, 5661-RDS, 5662-RDS, 5664-RDS, are maintenance product specifics names which include the Resource Dependency Service with different maintenance durations.

- ▶ 5648-F08, IBM Virtualization Engine and infrastructure services for Linux on z/Series, V2.1; under this product name all the Virtualization Engine components have separate and optional feature codes:
 - It includes the Virtualization Engine console, the IBM Director (with the IBM Director console), the IBM Director Agents and the common runtime components. Altogether they are named the *base components*.
 - A specific priced feature includes the Resource Dependency Service management server and the resource modeling.
 - Other priced features can include the IBM Director Extensions and the EWLM components.
 - The *base components* is a \$0-price based product (fee-based for Service and Support).

Note: General considerations for the 5648-F08 product:

- ▶ The *base components* can be ordered alone.
- ▶ There is no need to order all features altogether, at the same time.
- ▶ There is no need to install the Virtualization Engine console if you have components which do not use it; however the *base components* include the Virtualization Engine console and you will receive it.

How these products are packaged is detailed in 8.5, “Products packaging” on page 308.

1.10 Virtualization: topology and terminology

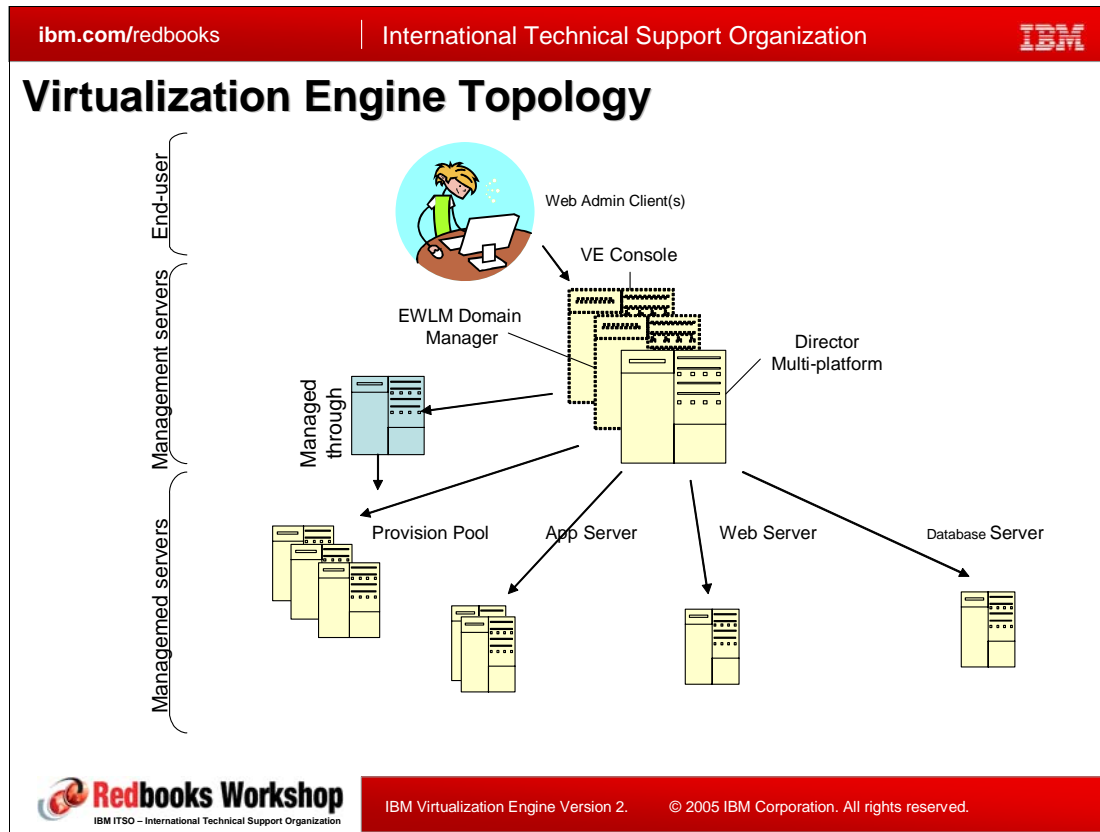


Figure 1-14 Topology and naming

The following terms are used when we present this type of environment:

Management server

Defines the platform with intelligence functions to issue instructions to enforce policies. A management server collects and may monitor data. Figure 1-14 depicts three management servers: one for the Virtualization Engine console, one for EWLM, one for IBM Director. All the management servers can be hosted in one box (refer to 8.6, "Architecture, performance, and sizing" on page 310 for more information about configuration choices and dependencies).

Managed servers (or managed node)

Defines the platform being managed or provisioned. They may have some intelligence to perform functions on a single server basis.

Managed from

Defines the system with the Virtualization Engine common runtime components installed.

Managed to

Defines the end-point that is being managed.

Managed through

Defines the management server that is being used as the resource instrumentation for Virtualization Engine services. Managed through servers must be used when

the management source does not have the function capability. Otherwise a managed through server is not part of the topology.

Common runtime


Defines the technologies that provide functions leveraged by the features for basic fabric capabilities.

Manageable resource (or managed resource)

Defines a resource that has a management interface with operations that control the state and the function of the resource.

1.11 Management servers: the supported operating systems

ibm.com/redbooks		International Technical Support Organization				IBM	
Operating Systems for <u>Management Servers</u>							
	Power Linux	xLinux	zLinux	I5/OS	AIX	Windows	z/OS
EWLM 2.1	RHEL 4.0 SLES 9 Note 1	RHEL 4.0 SLES 9 (32 bit)	RHEL 4.0 SLES 9 (64-bit)	I5/OS V5R3	AIX 5.3	W2003 (SE and EE) (32 bit) Note 2	z/OS 1.6 (and later) (or z/OS.e V1.6+)
RDS	RHEL 4.0 SLES 9 Note 1	RHEL 4.0 SLES 9	RHEL 4.0 SLES 9 (31-bit & 64 bit)				
IBM Director 5.10	RHEL 4.0 SLES 9 Note 1	RHEL 4.0 SLES 9 (32 bit)	RHEL 4.0 SLES 9 (31 bit & 64 bit)	I5/OS V5R3	AIX 5.3	W2003 (SE and EE) (32 bit) Note 2	
VE Console 2.1	RHEL 4.0 SLES 9	RHEL 4.0 SLES 9 (32 bit)	RHEL 4.0 SLES 9 (31-bit & 64 bit) Note 3	I5/OS V5R3	AIX 5.3	W2003 (32 bit)	

 IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 1-15 Operating Systems supported for management servers

The table provided in Figure 1-15 describes which level of operating systems support the *management servers*.


Explanation of the *Notes* on the foil:

- ▶ Note 1: on iSeries, xLinux requires the IXA hardware.
- ▶ Note 2: on iSeries Windows the IXS/IXA hardware.
- ▶ Note 3: The IBM Tivoli Directory Server and DB2® UDB common components must be installed in a 31-bit Linux partition.

Note: Before planning, pay specific attention for the Linux environment, the 31-bit or 64-bit support. For example, at the time of writing, ITDS, one of the common runtime components runs 31-bit only and so must be installed on a different machine if you plan to install the management server in a 64-bit Linux environment.

1.12 Managed servers: the supported operating systems

ibm.com/redbooks		International Technical Support Organization						IBM	
Operating Systems for <u>Managed</u> Servers									
	Power Linux	xLinux	zLinux	I5/OS	AIX	Windows	z/OS (z/OS.e)	Sun	HP
EWLM	SLES 9 Note 1 Note 3	SLES 9 Note 1	SLES 9 Note 5	I5/OS 5.3	AIX 5.2 or 5.3 Note 2	W2000 W2003 (SE & EE) Note 4	V1R6 and later	Sun Solaris 9	HP-UX 11iV1
IBM Director Agents	RHEL 4 SLES 9 Note 3	RHEL 4 SLES 9	RHEL 4 SLES 9	I5/OS 5.3	AIX 5.3	W2003 (SE & EE)			

 IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 1-16 Operating systems supported as managed servers

The table provided in Figure 1-16 describes which operating systems can support the *managed* servers.

Explanation of the *Notes* on the foil:

- ▶ Note 1: availability expected in 1Q2006
- ▶ Note 2: the use of new functionality on EWLM V2.1 requires AIX 5.3
- ▶ Note 3: on iSeries Linux on POWER™ requires the IXA hardware.
- ▶ Note 4: on iSeries Windows 2000 and Windows 2003 require the IXS/IXA hardware.
- ▶ Note 5: 31-bit and 64-bit support



IBM Director 5.10

One of the objectives of the Virtualization Engine platform is to provide cross hardware, cross operating system, and cross application management functions. The IBM Director product and its components have a time tested suite of base functions and extended components that address multiple hardware platforms and associated operating systems.

This chapter provides an overview of IBM Director functions through 5.10, with selected screen captures and examples of selected IBM Director cross-platform functions. Primary topics covered include:

- ▶ Summary of IBM Director multiple hardware platform and operating system functions:
 - Base functions across multiple operating systems and hardware platforms
 - Extended functions available on specific hardware platforms and operating systems
- ▶ New in IBM Director 5.10 console terminology, screen navigation changes, and agent function levels (level 0, level 1, and level 2)
- ▶ IBM Director base managed object discovery and functions for inventory, resource and process monitoring, and job scheduling
- ▶ Selected IBM Director extended functions such as software package distribution.

2.1 IBM Director 5.10 overview

ibm.com/redbooks | International Technical Support Organization

IBM Director Cross Platform Functions

- ❑ **Functions**
 - > Discovery, Inventory
 - > Monitor, Hardware Status
 - > Event Action Plans
 - > Process Manager
 - > Group Management
 - > Remote Session/Remote Control
 - > File Transfer
 - > Task Scheduler
 - > Software Distribution/Update Assistant
- ❑ **Operating Systems**
 - > Microsoft Windows
 - > Intel Linux
 - > Power Linux
 - > AIX
 - > i5/OS
 - > zSeries Linux environments
- ❑ **Hardware platforms**
 - > pSeries
 - > iSeries
 - > xSeries
 - > zSeries
 - > BladeCenter
 - > OpenPower

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 2-1 IBM Director overview - cross platform functions

Figure 2-1 shows an impressive, but significantly summarized list of IBM Director's capabilities over a set of supported hardware platforms and operating systems.

IBM Director is a set of components that run on several different hardware platforms and operating systems, which offer a single point of management and automation for a wide range of "system management" functions.

The IBM Director product has been available for several years offering a wide set of management facilities, primarily on xSeries-based hardware platforms running a Windows operating system. IBM Director functions can complement and be integrated with higher level management tools, such as those available with the Tivoli suite of tools.

During 2004 IBM Director Multiplatform 4.2 became available which expanded most of its IBM Director 4.1 functions to a wider set of hardware platforms and operating systems.

IBM Director 5.10 became available during November 2005, and, though the word "multiplatform" has been removed from its product name, 5.10 extends the range of functions available on multiple operating systems. IBM Director 5.10 platform coverage includes Microsoft Windows, Intel Linux, Power Linux, AIX, i5/OS and zSeries Linux environments across pSeries, iSeries, xSeries, zSeries, BladeCenter and OpenPower™ systems.

Note: In this chapter:

- ▶ The term *pSeries* represents pSeries hardware platforms, IBM eServer p5 hardware platforms and IBM System p5 hardware platforms.
- ▶ The term *iSeries* represents iSeries hardware platforms, IBM eServer i5 hardware platforms, and POWER5-based i5 hardware platforms.

The following is a brief list of the major IBM Director functions:

- ▶ Collect and display hardware and software information on managed systems (managed nodes)
- ▶ Monitor hardware status
- ▶ Monitor important operating system and application processes. This includes metrics such as processor utilization, jobs/processes started and stopped, and more. These generate events that can be simply recorded or defined to take some specific action, such as running an operating system command or calling a pager device.
- ▶ Create and manage dynamic and static groups of systems
- ▶ Establish remote workstation sessions to a managed system
- ▶ Transfer files and manage directories on systems and groups
- ▶ Schedule and monitor non-interactive tasks, such as collecting inventory and transferring data
- ▶ Depending on operating system capabilities, deploy firmware, device drivers, and software packages to systems or groups
- ▶ Provide core corrective actions, distributed commands, and hardware control for both servers and storage.

Note: In this chapter we do not cover IBM Director TotalStorage management capabilities and touch lightly on IBM Director running z/OS and Linux partitions. A new version of *IBM Director 5.10*, SG24-6188 is planned for early 2006.

Rather than immediately delving into the detailed functional descriptions we first set the environment in which IBM Director functions excel and discuss IBM Director terminology. We defer until later in this chapter, (2.8, "IBM Director 5.10 cross platform functions" on page 62) a more comprehensive description of the major functions.

We do, here however, remind you as we stated previously, IBM Director 5.10 covers most of today's well-know operating systems - Microsoft Windows, Intel Linux, Power Linux, AIX, i5/OS and zSeries Linux environments across pSeries, iSeries, xSeries, zSeries, BladeCenter and OpenPower. Several new extensions in IBM Director 5.10 are focused on delivering enhanced management capabilities for iSeries, pSeries and zSeries servers, and ease of use enhancements, including:

- ▶ z/VM Center is a new extension to IBM Director for Linux on System z9 and zSeries that saves time and simplifies the creation and grouping of virtual servers, and the deployment of Linux systems under z/VM. Through a drag and drop interface, IBM Director allows for fast, easy, and repeatable deployments of new virtual servers.
- ▶ Hardware Management Console (HMC) integration into IBM Director 5.10 allows IBM Director to work through the HMC to perform discovery, inventory, health monitoring and event management of central electronics complexes and manage logical partitions running on the system being managed by the specific HMC.

Note that this HMC support is planned to be available the first half of 2006. Remember any statements made about IBM plans are subject to change without notice.

- ▶ A streamlined IBM Director console user interface that allows more customization in what sets of information are displayed and the order in which that information is displayed.
- ▶ 3 levels of IBM Director Agent (discussed later in this chapter) functions which offer more flexibility in trade-off for functions versus hardware resource requirements.
- ▶ Storage Managed Objects which provide enhanced storage device discovery, inventory, health monitoring and event management. IBM Director 5.10 understands the Storage Management Initiative Specification and through it can manage storage devices, including IBM TotalStorage DS300 and DS400 storage subsystems.

Note that this presentation guide does not include further information on IBM Director and TotalStorage capabilities.

IBM Director's functions are grouped into an IBM Director Server, IBM Director Agent (3 levels with 5.10), and an IBM Director console.

IBM Director is also compatible with operating systems running all other Single Byte Character String (SBCS) languages for which it is not translated for. In such cases strings will be shown in English.

Unicode is supported; limitations may apply, please refer to your local support.

Please note that IBM Director components have such a wealth of functions, we cannot cover all of them in this presentation guide. We list three excellent sources of additional information:

- ▶ IBM Systems Software Information Center IBM Director content:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp>

Select IBM Systems Software Information Center. Expand Systems Management → Solutions → . IBM Director

When starting to consider using IBM Director, we recommend specifically reviewing the IBM Director Installation and Configuration Guide (V5.10). Important sections of this document include planning information and a comprehensive listing of which operating systems support IBM Director Server, IBM Director Agent, and IBM Director console and the operating systems supporting a specific IBM Director task. To quickly access this guide, get to the IBM Director information center web page as described above. Then:

Select Documentation and resources → IBM Director Installation and Configuration Guide PDF

- ▶ the IBM Director website

www.ibm.com/servers/eserver/xseries/systems_management/ibm_director

- ▶ IBM International Technical Support Center (ITSO) redbooks

www.ibm.com/redbooks

For example, search *IBM Director 5.10*, SG24-6188

Many of the descriptions included in this chapter are based upon the IBM Director content at these websites and from the IBM Director console component's on-line help information.

In the following topics we take a graphical view of the IBM Director server, agent, and console components in a network to provide a perspective of how these components work together to provide the functions. We also introduce the 5.10 term *managed objects*. This term is new to IBM Director starting with 5.10 and is for the purpose of collectively addressing "all the things" managed by IBM Director.

Following these topics we then launch into more details on these functions with screen examples of selected functions.

2.2 IBM Director 5.10 hardware components



Figure 2-2 IBM Director 5.10 Hardware components

IBM Director 5.10 has four major hardware components:

- ▶ **Management Server**

This server is “the manager” from the IBM Director point of view for all IBM Director information collected from managed systems (nodes). Most functions are requested by the IBM Director console to the IBM Director server. The specific function could be performed by the server for a specified managed node. The server contains all collected inventory information. The IBM Director console connects to the server to define actions and report inventory and events that have been defined.

- ▶ **Managed systems (also referred to as managed nodes)**

This includes all servers, desktops, laptops, workstations, storage subsystems managed by the IBM Director server. This can include the server system itself, the managed systems, and SNMP (Simple Network Management Protocol) devices.

Since the Information Center Planning Advisor for Virtualization Center console, which can include the IBM Director in its planning steps, uses the term *managed node*, we use the term managed system and managed node interchangeably in this chapter. A managed node (or system) is a unique hardware platform and installed operating system combination. The term managed node/system is meant to distinguish possible IBM Director functions capable on these “systems” versus the more basic management functions that can be performed with a managed SNMP device.

- ▶ **SNMP devices**

Printers, computers, switches and other hardware with Simple Network Management Protocol (SNMP) enabled and these are managed by IBM Director Management Server. That is they have no IBM Director software installed.

► Network

A network consists of one or more systems, devices, and the communication hardware and enabling them to communicate. From an IBM Director viewpoint systems and devices are “discovered” to be managed by an IBM Director server. In the IBM Director environment, this network is normally using Ethernet (wire or wireless) with TCP/IP as network protocol.

There is a command line interface that can be used to do many of the functions that are available from the console.

2.3 IBM Director 5.10 software components

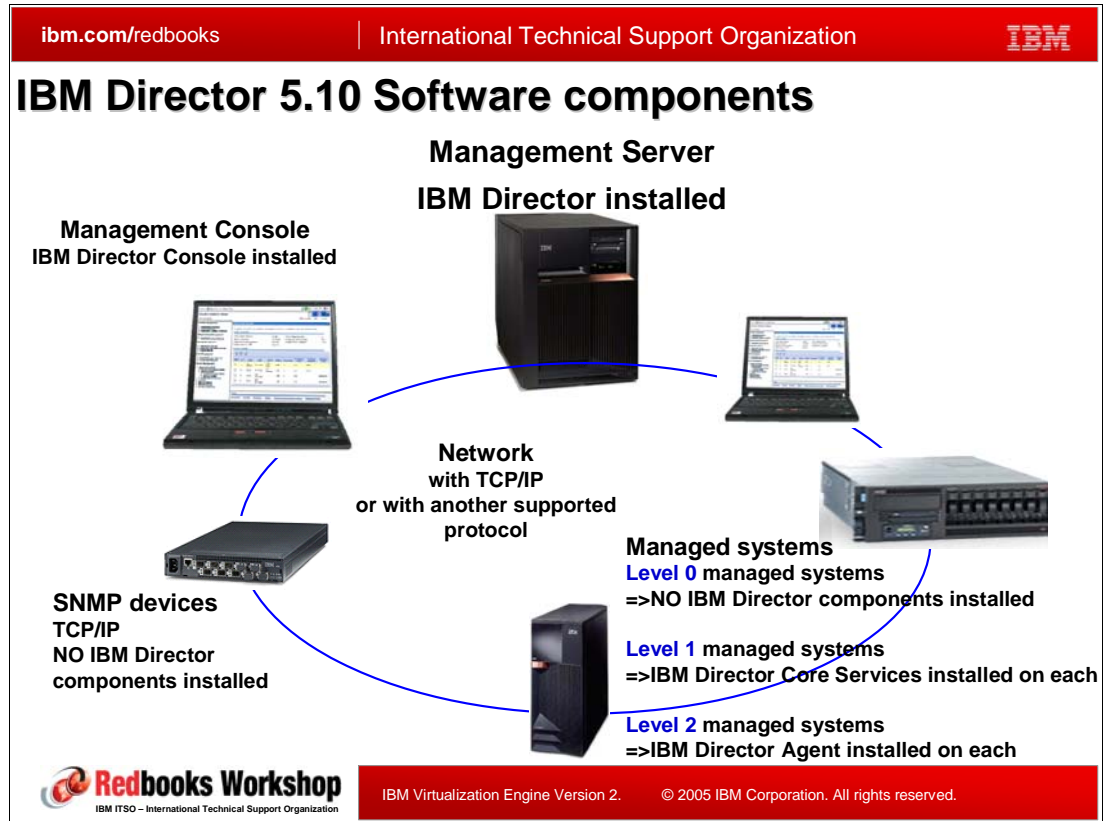


Figure 2-3 IBM Director 5.10 Software components

IBM Director 5.10 software components are placed on top of the hardware network and include the following:

- ▶ IBM Director Server installed on a management server
 - A database product/component must be installed on your “server system” prior to installing IBM Director Server. IBM Director Server requires this database to store and query its own “director database unique” information.
 - The database could be local or remote, in most environments. Typically the database is on the same system/partition as the IBM Director server.
 - IBM Director Server uses the database to store inventory information in a central repository. This inventory enables managing assets from a central point.

Note: Before installing IBM Director Server, you need to decide which database product/component you want to use. The Apache Derby Database is embedded in IBM Director Server itself and is the default database on all platforms except for i5/OS.

The default database for i5/OS is IBM DB2, which is embedded within with i5/OS itself. When using IBM Director server on a hardware platform other than an iSeries system, you certainly can install DB2 UDB rather than use the default Apache Derby Database.

- The database component/product supported by IBM Director server is dependent on the base operating system on the server being used for server inventory information. For additional details about supported databases, go to the IBM Director information center and search for "director database"; then select the database link.
- IBM Director Server is based on Java
 - You must have a supported Java Developer Toolkit on the system. For example, IBM Director installation on i5/OS system, requires product 5722-JV1 with option *BASE (IBM Developer Kit for Java) and option 6 (Java Developer Kit 1.4).
- The Management Server must be connected via a network with all Managed Systems. Normally this is a TCP/IP network protocol.
- ▶ IBM Director Console must be installed on at least one workstation functioning as a Management Console somewhere in the network. You can have multiple workstations with the IBM Director console installed, but this requires planned coordinated and secured use by more than one IBM Director console.
 - IBM Director Console is a Java based implementation
 - Management Console is connected via network (normally using TCP/IP protocol) with the Management Server and all Managed Systems
- ▶ IBM Director Core Services or IBM Director Agent Level 2 is installed on Managed Systems
 - IBM Director Agent is a Java implementation on all platforms except Windows
- ▶ A supported network protocol like TCP/IP should be installed as prerequisite

Please note, starting with V 5.10, IBM Director provides three different levels of managed systems. The agents levels ("0", "1", "2") have different software requirements and provide a set of functions that correspond to the "level number":

- ▶ Level 0 is base function which really involves no IBM Director software to be installed
- ▶ Level 1, which requires a base level of IBM Director agent software to be installed
- ▶ Level 2 installs the full IBM Director agent software, which enables the highest IBM Director agent functions.

You need to understand the functions available at each agent level and the associated software required to do proper planning and management of your network. You may choose to have different "agent levels" on groups of systems or devices in your network, depending on your management requirements.

Because of the importance of understanding the functions available at each "agent level," the following topic provides a short description of these new with 5.10 levels for IBM Director managed systems. Then we expand on the IBM Director Management Server functions. There are several new or changed IBM Director agent and server functions with V5.10. We note some of the more significant V 5.10 functions or interfaces, but not all of them in the following sections.

Though not discussed in detail in this chapter each operating systems supporting IBM Director server, console, and agent components has security facilities that interact with IBM Director to control which users can do which IBM Director function.

For example, on i5/OS either a command interface or an iSeries Access for Windows iSeries Navigator interface must be used to "permit" usage of IBM Director server and agent functions. Figure 2-4 is an example of enabling IBM Director using the iSeries Navigator Application Administration interface.

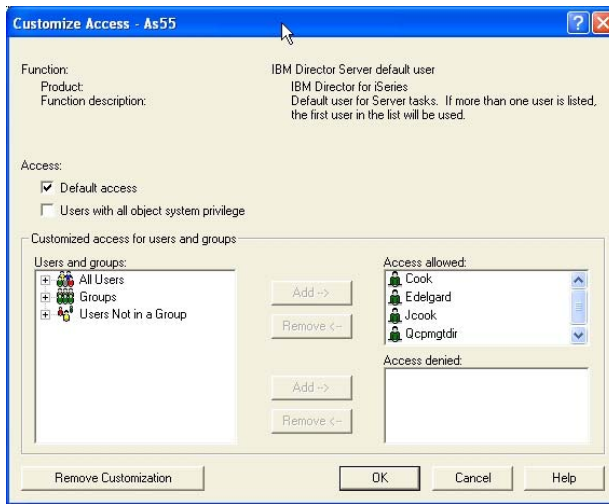


Figure 2-4 i5/OS example of enabling user access to the IBM Director server

2.4 Overview of agent level functions

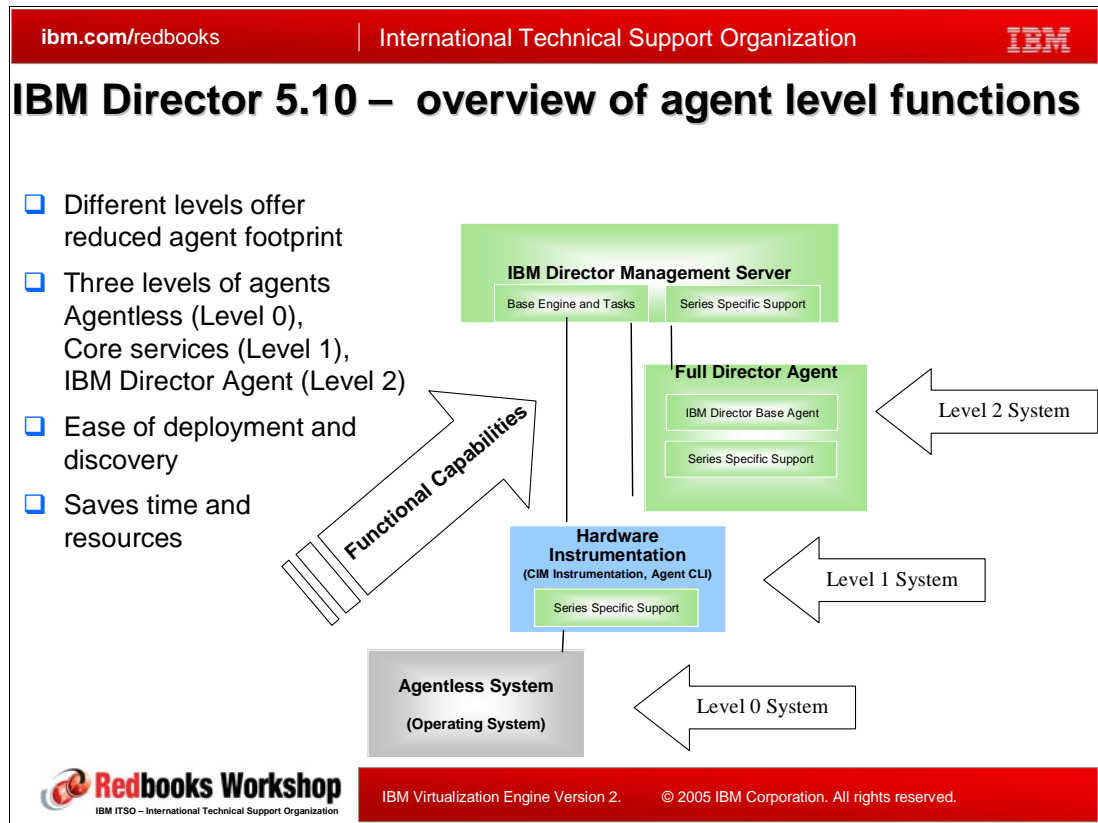


Figure 2-5 IBM Director 5.10 - Support for Levels of Director Systems

With IBM Director 5.10 its agent functions are separated into three levels - level 0, level 1, and level 2. The following summarizes the functions now available for each agent level.

- ▶ **Level-0: Agentless Systems:** These systems do not have any IBM Director software or Common Information Model (CIM) instrumentation installed on them. Level-0 managed systems can be IBM or non-IBM servers, desktop computers, workstations, and mobile computers.

A Level-0 managed system does have the minimum set of protocols that are required for that system to be managed by IBM Director. Such a system must have, at a minimum, software supporting the Secure Shell (SSH) or Distributed Component Object Model (DCOM) protocol.

Such an agent is typically running one of the following IBM Director supported operating systems:

- Linux
- Windows

IBM Director discovers Level-0 managed systems by verifying the IP addresses on your network and scanning the ports of those addresses using the SSH or DCOM protocols. The range of IP addresses that are verified is governed by the IBM Director discovery preferences that you configure in IBM Director Console. By default, IBM Director uses the range of addresses that are in the IP domain of the management server.

When a Level-0 managed system is discovered, it is locked by default. You can unlock the system by requesting access to it through IBM Director Console. If the object that is

discovered supports SSH but is not a computer system (for example, a Remote Supervisor Adapter (RSA)), the object will be displayed in IBM Director Console but will not support any tasks.

Systems and other network devices that support the SNMP protocol will display as SNMP Device Managed Objects in IBM Director Console.

The attributes that are returned for a Level-0 managed system include:

- Locked
 - System Name
 - TCP/IP Addresses
 - System State
 - MAC Address
 - System Presence Check Setting
- Unlocked
 - Computer Name
 - Architecture
 - OS Major Version
 - OS Minor Version
 - Access Denied
 - Operating System
 - Unique System ID
 - System UUID (xSeries)
 - Machine Type
 - Model Number
 - Serial Number

After you discover and unlock a Level-0 managed system, you can perform the following tasks on that system:

- Collect inventory that is available from the operating system
- Install Level 1: Core Services or Level 2: IBM Director Agent by using software distribution
- Restart the operating system (Linux only)
- Run command-line programs (only if SSH is present)
- ▶ Level-1: IBM Director Core Services provides managed systems with a subset of the IBM Director Agent Level 2 functionality that is used to communicate with and administer that system. Specifically, level 1 provides hardware alerts and status information that can flow to the IBM Director server or some other management server, such as a Tivoli product.

A Level-1 managed system is any system that has the IBM Director Level 1 Agent Core Services installed but does not have the IBM Director Level 2 Agent installed. Level-1 managed systems can be IBM servers, desktop computers, workstations, and mobile computers.

You can perform the following tasks on a Level-1 managed system:

- Collect inventory
- Install Level 2: IBM Director Agent using software distribution
- Manage events using event action plans, event subscription, and the event log
- Monitor hardware status
- Restart or shutdown the managed system
- Run command-line programs

IBM Director Level 1 Core Services is supported on systems that are running the following operating systems:

- Linux (xSeries, System p5 and pSeries®, and System z9 and zSeries)
- Windows

- ▶ Level-2: IBM Director Agent Level 2 provides managed systems with the full complement of IBM Director Agent functionality that is used to communicate with and administer that system. The functionality of IBM Director Agent Level 2 on a managed system varies, depending on the operating system and platform.

A Level-2 managed system is any system that has IBM Director Agent Level 2 installed. Level-2 managed systems can be IBM or non-IBM servers, desktop computers, workstations, and mobile computers.

IBM Director Agent Level 2 is supported on systems that are running the following operating systems.

- AIX
- i5/OS
- Linux (xSeries, System p5 and pSeries, and System z9 and zSeries)
- NetWare
- Windows

Focusing on IBM Director 5.10, we selected to show the “managed objects” in Figure 2-6 arranged according to “object type.” We take this opportunity to explain several of things shown in Figure 2-6.

2.4.1 IBM Director console and managed objects

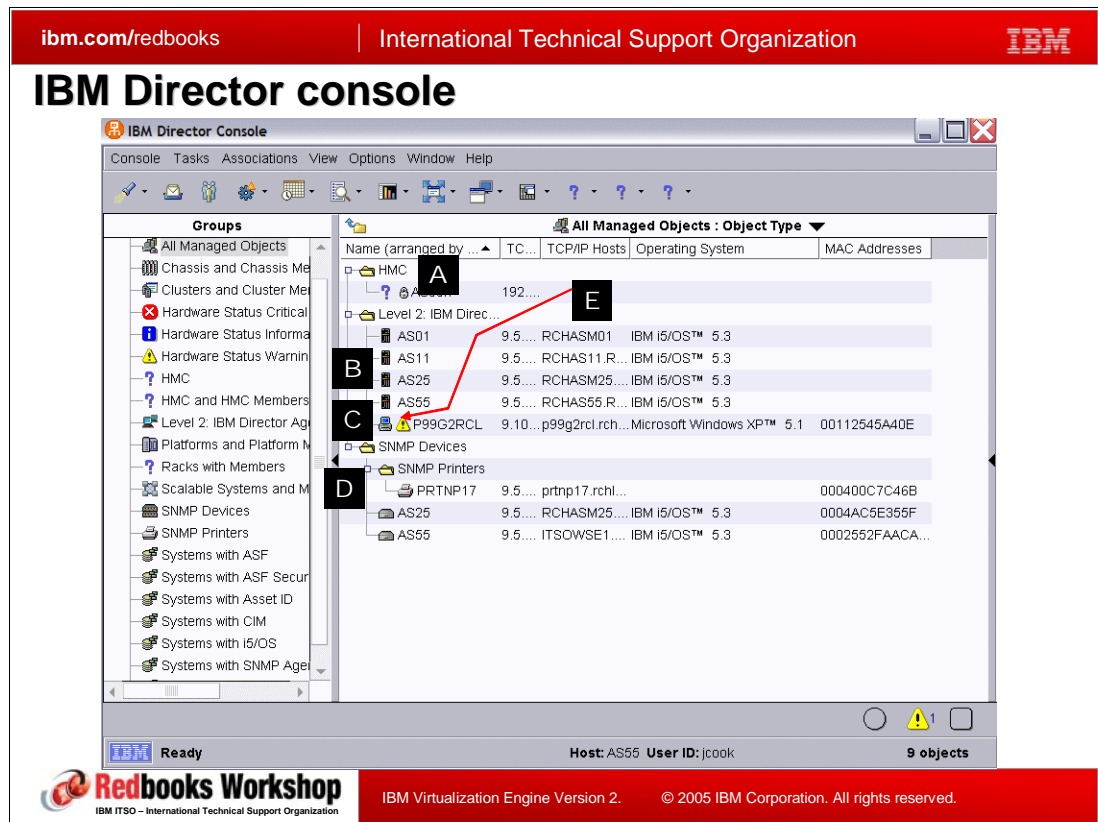


Figure 2-6 IBM Director 5.10 console - All Managed Objects

IBM Director 5.10 introduces the term managed objects to represent the managed systems and devices (such as an SNMP printer). In previous releases the term used was systems and devices.

In our example network shown in Figure 2-6, we selected All Managed Objects. Before capturing the window shown, we used the column heading Name (arranged by.) to sort the managed objects in the order shown. This window shows

- ▶ A Hardware Management Console (HMC) device that is used to manage system AS55 (A)
- ▶ Four iSeries and IBM eServer i5 systems (B).
- ▶ A Windows IBM Director console system, P99G2RC, all IBM Director Agent level 2 agents (C).
- ▶ SNMP classified devices, such as LAN printer PRTNP17 (D).

While this chapter on IBM Director 5.10 is not as detailed as the redbook *IBM Director 5.10*, SG24-6188, our example shown in Figure 2-6 demonstrates IBM Director 5.10 console screen customization we did before capturing the window shown:

- ▶ Selected to view according to IBM Director 5.10 object types
- ▶ Renamed the original names “discovered” by IBM Director for all four our iSeries and IBM eServer i5 systems shown in the Arranged by ... column
- ▶ “Squeezed” the TCP IP Address and formal host name columns so you cannot see the full text in these columns
- ▶ “Switched off” the optionally shown right pane for available Tasks that you can perform on the managed object.

Two important notes:

1. Note the yellow triangle with the ! icon (warning) (E) for the Windows workstation running both IBM Director 5.10 Console and Agent level 2. This workstation has two LAN adapters. Though not shown in this chapter, if you select P99G2RCL and perform the Hardware status task you would see the warning is for a second LAN adapter that is disabled. A disabled adapter may or may not be a problem, depending on your environment.
2. Two of the iSeries systems are “discovered” as both IBM Director 5.10 Agent level 2 objects and also as SNMP devices. This is because agent is installed on these systems (actually partitions on iSeries LPARed systems) and the i5/OS partition has its SNMP code activated.

Optionally, with IBM Director 5.10 Windows Console, you can delete these systems from being viewed under the SNMP object type. This is your choice.

2.4.2 IBM Director console navigation views

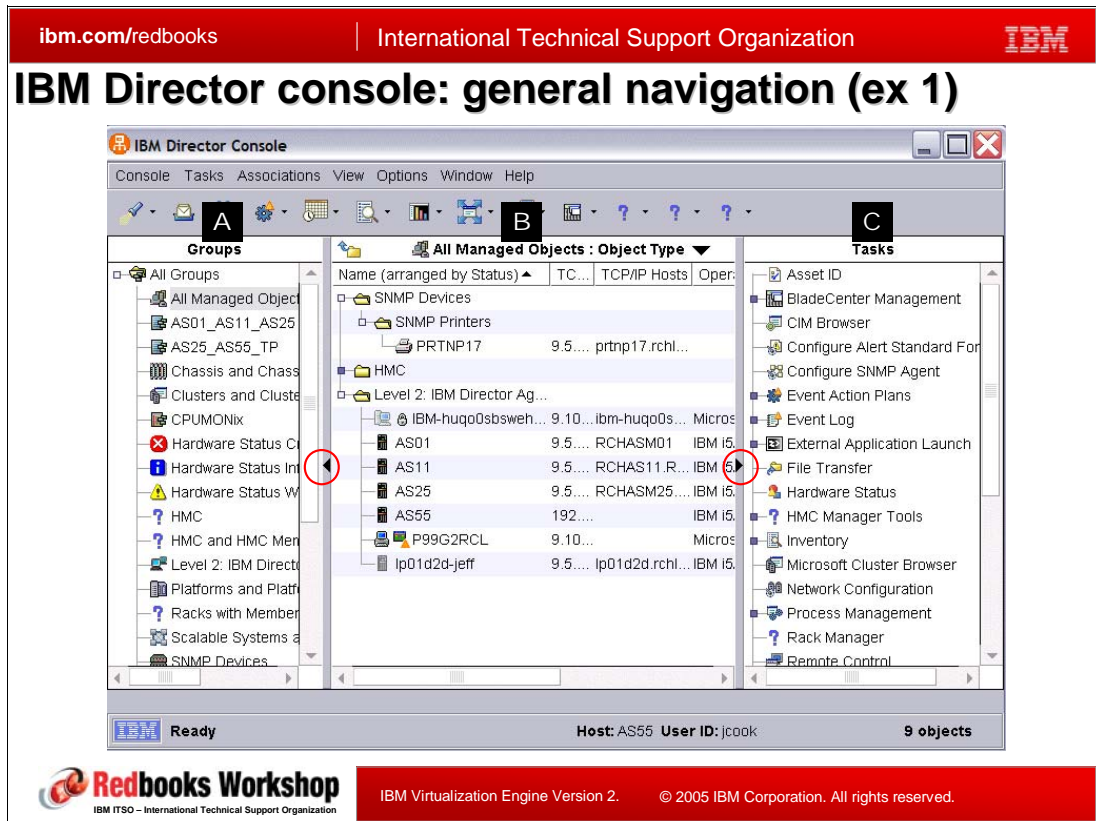


Figure 2-7 IBM Director console - general navigation window example 1

Figure 2-7 and Figure 2-8 show two examples of the console window.

- ▶ In Figure 2-7 you see an example of the window typically used. It essentially is divided into 3 separate panes - left (Groups, **A**), center (All Managed Objects, **B**), and right (Tasks, **C**). Using your mouse you can widen or make narrower the width of each of the 3 panes.

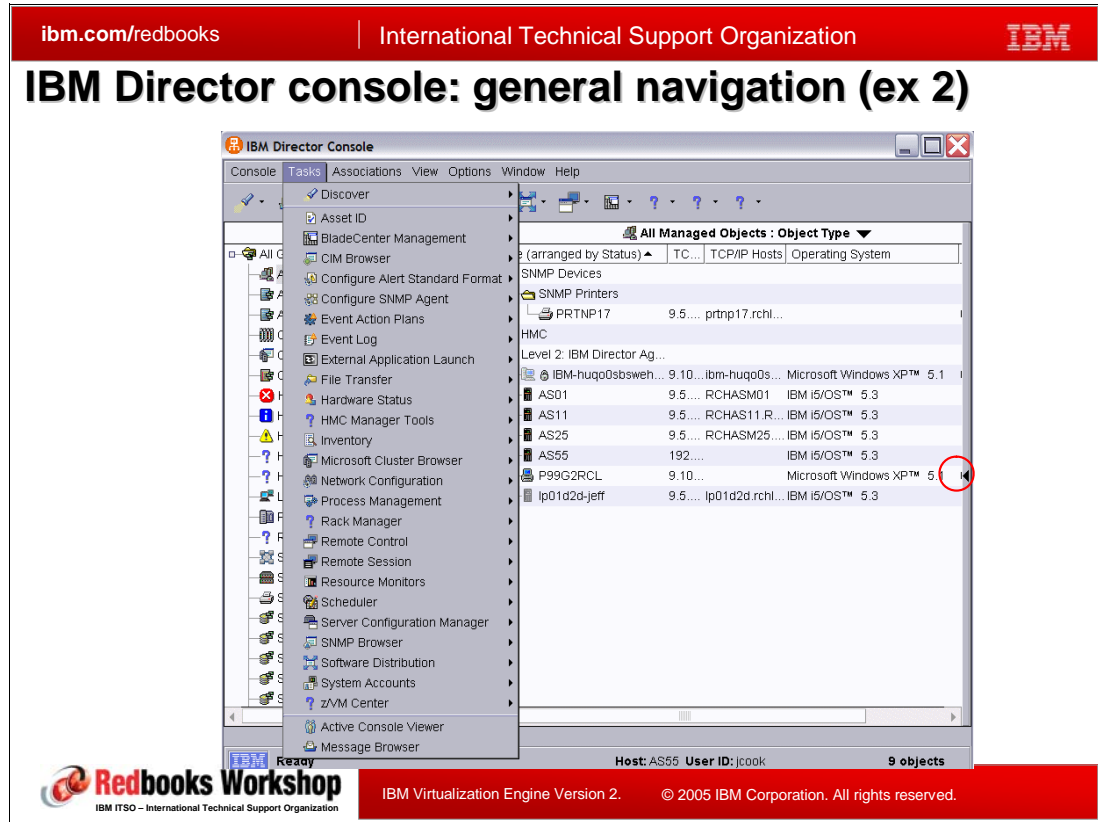


Figure 2-8 IBM Director console - general navigation window example 2

- ▶ In Figure 2-8 we have “dropped” showing the tasks pane. The tasks pane is essentially a graphical representation of the tasks menu list shown in the example.

You can use the circled “arrowheads” to cause a pane to appear or disappear. With 5.10 there is increased flexibility in controlling the columns of information being displayed and the order of columns. Using the View menu bar options there are several ways to order the information being shown in the Managed Objects pane.

Though not shown, there is a “ticker tape line” area available at the bottom of the console window. Text your IBM Director administrator has defined can be continuously shown. Typically such text would be defined as part of an action plan for a monitored event that occurred.

2.5 What's new in IBM Director 5.10, summary

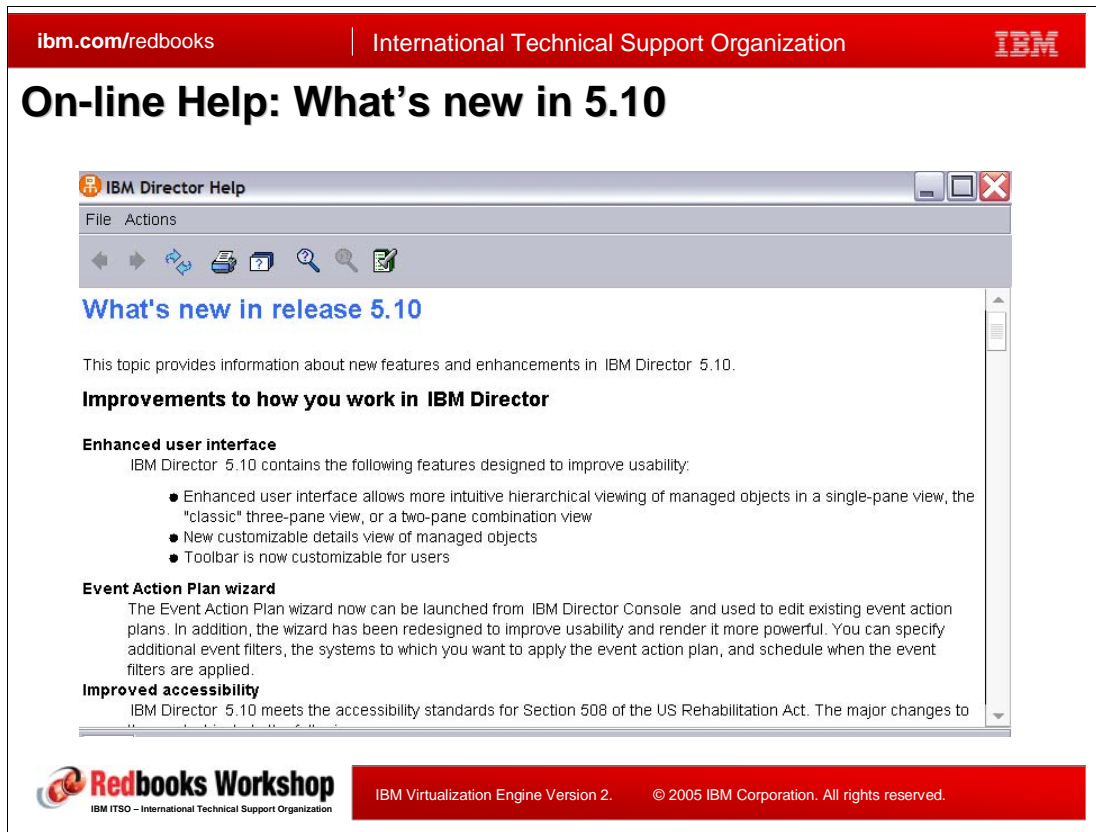


Figure 2-9 IBM Director - on-line help and what's new

IBM Director has so many functions through 5.10 that it is impossible to cover all of them in this presentation. We recommend using IBM Systems Software Information Center to read and investigate capabilities beyond what is covered in this chapter. There is also the *IBM Director 5.10*, SG24-6188-02, redbook, available in draft and planned for publication in 2006. However, we want you to know that when IBM Director console is installed on a workstation there is extensive on-line help that has a nice set of what is new in 5.10 information. The initial window is shown in Figure 2-9.

From a central console, a user or an administrator (depends on authorization level) can perform the following functions, as illustrated with Figure 2-10 on page 55:

- ▶ Monitor & Alert
- ▶ Optimize
- ▶ Analyze and Troubleshoot
- ▶ Maintain
- ▶ Deploy

You can manage a set of heterogeneous hardware platforms and operating systems, which could include many managed objects, depending upon your network structure.

Tip: You must plan carefully your IBM Director Agent licenses if you will have an environment with “thousands” of managed objects (SNMP devices and IBM Director agents).”

Currently the number of IBM Director level 2 Agents is limited to 5000 licenses. For the other levels 0 = Agentless Management and level and 1 = Core Services (sometimes named as Common Information Model (CIM) instrumented) are not subject to the 5000 limit.

While IBM Director 5.10 is interactively oriented, there are options for scheduling and unattended activities, as indicated in Figure 2-10 on page 55.

We begin a set of slides that focus on describing in more detail, the cross platform functions.

2.6 Comprehensive platform management with IBM Director

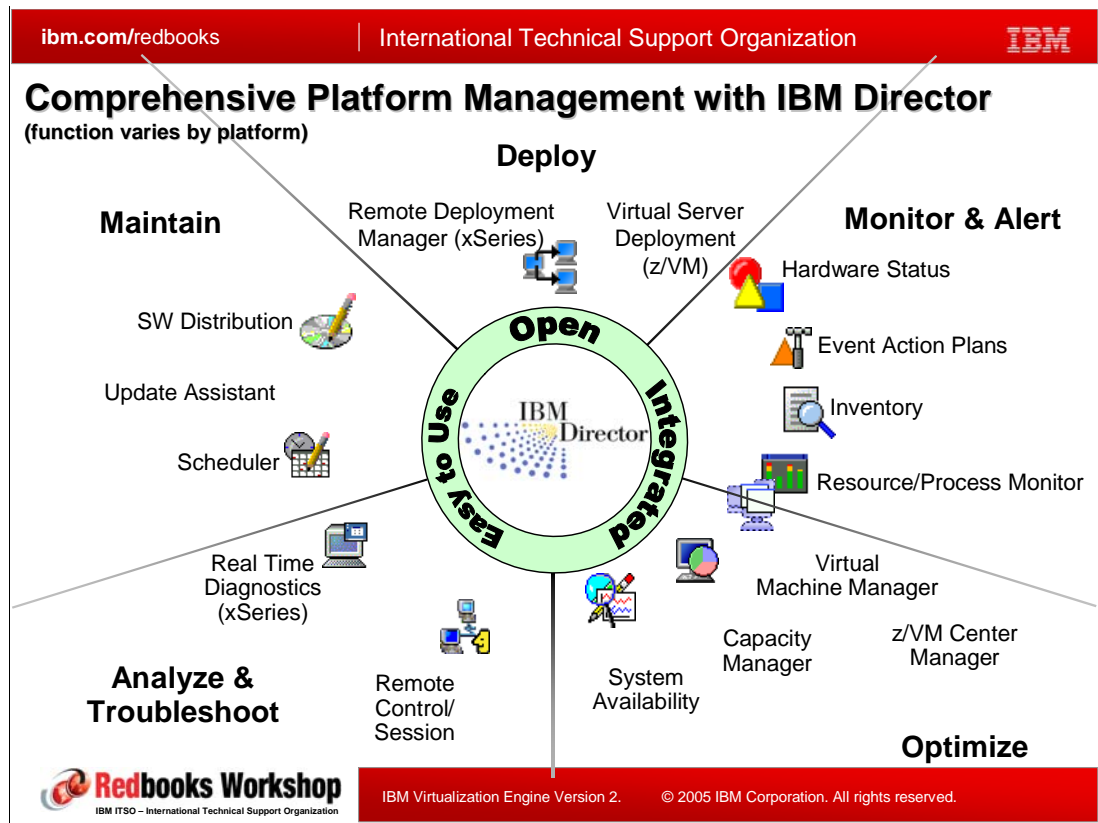


Figure 2-10 Comprehensive Platform Management with Director

We started this chapter with an overview of the IBM Director functions across operating systems and hardware platforms, depicted general network configurations, defined IBM Director Agent levels and some general IBM Director terminology and IBM Director console window navigation capabilities.

Before describing several cross platform functions, we provide the next few slides, repeating some function categorization and network topology information from slightly different viewpoints:

- ▶ IBM Director components from a management function category
- ▶ IBM Director topology from a user interface, management server, and managed agent viewpoint.

Deploy

- ▶ **Remote Deployment Manager (xSeries):** Remote Deployment Manager (RDM) facilitates remote deployment of both IBM and non-IBM systems. RDM allows for remote unattended installation of new and existing systems. RDM helps automate deployment tasks such as initial operating system installation, BIOS updates, and disposal of retired systems. All of these tasks can be done without visiting the remote system.

This means only one IBM Director console is required to both remotely manage and deploy supported systems. The combination of IBM Director, RDM, and Software Distribution Premium Edition provides a powerful set of tools for both management and deployment.

- ▶ Virtual Server Deployment (Z/VM): this is the Virtual Server task part of z/VM. See the Virtual Server Deployment description under Optimize - z/VM Center Manager on the next page.

Monitor and Alert

- ▶ Inventory: You can collect (manually or periodically via the Scheduler) hardware and software inventory information from the managed agent systems and devices onto the IBM Director server's database. This inventory information can be viewed (queried) in many different ways from the IBM Director console. You can have the latest information on hand to assist you with other tasks based upon an installed hardware feature of software produce release level.
- ▶ Hardware Status and Resource/Process monitors: You can monitor for hardware changes and resource utilization or changes in active processes (process/job started, ended, number active and so forth).
- ▶ Event Action Plans: You can define thresholds (such as 70% CPU utilization) for monitored hardware or resources to define an automated action plan, where appropriate. You associate a selected action with a threshold for a specific monitor within an Event Action Plan. You can specify filters, view the event logs and specify how long to keep records of events for later review across a time period. A filter can be used, for example to say if something happens, it must remain true for 5 minutes before signalling the event.

Optimize

- ▶ Virtual Machine Manager: IBM Virtual Machine Manager (VMM) is an extension to IBM Director that allows you to manage both physical and virtual machines from a single console. With VMM, you can manage both VMware ESX Server and Microsoft Virtual Server environments using IBM Director. VMM also integrates VMware VirtualCenter and IBM Director for advanced virtual machine management.
- ▶ System Availability: System Availability is an optional, no-charge IBM Director extension that helps you track and report the availability of your servers. Its automated capabilities and flexible graphical views save you time tracking the availability of your servers so that you can rapidly respond to problems and help maintain maximum system availability.
This capability is supported on Windows and Linux operating systems.
- ▶ Capacity Manager: Capacity Manager is an optional IBM Director extension that helps optimize your servers' performance and availability. It monitors the utilization of your servers' resources, helps identify existing or future bottlenecks and makes recommendations for improvement. Its predictive, proactive capabilities help reduce downtime costs. It has built-in, intelligent features to optimize server throughput, utilization and performance.
This capability is supported on Windows and Linux operating systems.
- ▶ z/VM Center Manager: This is the z/VM Center task and the z/VM environment that z/VM Center manages. Use z/VM Center to provision Linux systems on virtual hardware that is based on real IBM System z9 and zSeries hardware and the z/VM hypervisor. z/VM Center uses the IBM System z9 and zSeries virtualization technologies, in particular z/VM, to provision System z9 and zSeries resources in form of z/VM virtual servers. To z/VM, a z/VM virtual server is a guest virtual machine.

z/VM Center and IBM Director have two tasks:

- Virtual Server Deployment

With the Virtual Server Deployment task, you can define configurations of z/VM virtual servers and save them as virtual server templates.

Use the Virtual Server Deployment task to manage individual z/VM virtual servers and operating system instances and to set up templates and Linux guest systems.

A Linux guest system is a combination of a Linux instance and the z/VM virtual server on which the Linux instance is installed.

– Server Complexes

With the Server Complexes task you can manage configurations of Linux guest systems. A server complex is a configuration profile for Linux guest systems and includes both Linux and z/VM aspects. A server complex can define network settings, Linux configuration scripts, disk access, and VM Resource Manager (VMM) performance goals.

You can automatically configure a Linux guest system by assigning it to a server complex. You can also create a new Linux guest system within a server complex. When creating a new Linux instance in a server complex, you automatically create a z/VM virtual server with a Linux instance that is configured according to the server complex. For creating a Linux guest system, you require a virtual server template and an operating system template that have been created by the Virtual Server Deployment task.

You can make changes to a server complex and then apply the configuration changes to all Linux instances in the server complex

A z/VM guest virtual machine consumes a portion of the processor cycles, memory, and I/O bandwidth of the System z9 or zSeries hardware. All operating systems that can run natively on System z9 or zSeries can also run on a z/VM virtual server.

Analyze and Troubleshoot

- ▶ Remote Control/Session: You can use the base Remote Control task to manage a remote Level-2 system by displaying the screen image of the Level-2 managed system on a management console. You can specify whether remote control “takes over” keyboard and mouse interfaces or merely watches what the remote workstation user does.

This is supported on Level-2 managed systems running Windows® only. You cannot use Remote Control on SNMP devices

- ▶ Real Time diagnostics (xSeries): IBM Real Time Diagnostics is systems management software that runs on your IBM eServer® xSeries® system while the operating system is active. It can help prevent downtime; thereby increasing your system's availability.

Real Time Diagnostics supports operating in an IBM Director environment for Windows 2003, Windows 2000, Red Hat Linux EL 3.0, and Red Hat AS 2.1. Diagnostics for IBM eServer BladeCenter™ platforms can be done without taking the blade or unit off line.

This supports Windows and Linux operating systems.

Maintain

- ▶ Scheduler: You can use the base IBM Director Scheduler to run a job composed of single non interactive task or set of non interactive tasks at a later time. Only non interactive tasks, which are defined as tasks that do not require any user input or interaction, can be scheduled. You can specify an exact date and time you want the job to be started, or you can schedule a job to repeat automatically at a specified interval.
- ▶ Update Assistant and Software Distribution: The base Update Assistant function provides the ability to import IBM-provided software updates and distribute them with the IBM Director Software Distribution. IBM Updates can be imported from UpdateXpress media, Director CDs, and from files downloaded from an IBM Web site. The importer lets you import one or multiple updates in one instance. Each update consists of multiple files, including the actual update and an XML file that describes the update. Both files are

required to import the update into IBM Update Assistant. One example of this would be to download an IBM Director Agent for i5/OS from an IBM Director for i5/OS server.

The capability to define your own software package for distribution requires the additional charge IBM Premium Edition software distribution capabilities. The additional capabilities include

- Creating your own simple software package of, for example a set of files to be distributed periodically to managed agent systems
- Creating and distributing wizards for Microsoft Windows Installer packages, Linux RPM packages, AIX InstallP packages, i5/OS Restore Library, Licensed Program, and Restore Object packages.

More details about the wide spectrum of functions provided by IBM Director 5.10 is discussed in Section 2.8, “IBM Director 5.10 cross platform functions” on page 62

2.7 Topology of IBM Director component and Virtualization Engine console

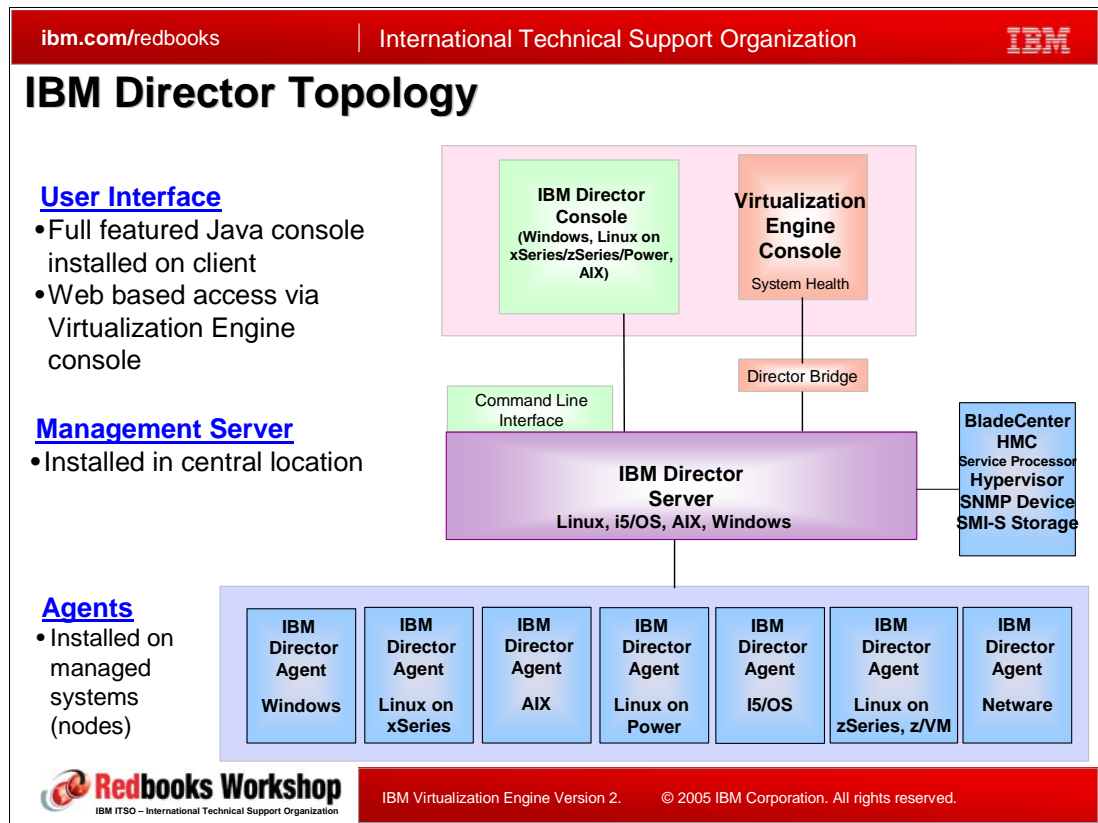


Figure 2-11 IBM Director Topology

The topology carries over from earlier releases to IBM Director 5.10. There are still three major building blocks. At the top of this figure you see The IBM Director console itself and the optional Virtualization Engine console.

► **IBM Director console and optional Virtualization Engine console**

The IBM Director console has full function through either a graphical interface or a command line interface. With the Virtualization Engine console installed, from a single web browser on your workstation, you can interface with several “product consoles,” in addition to the IBM Director console. As discussed in Chapter 4, “The Virtualization Engine console” on page 133, the Virtualization Engine console has optional “bridge” functions to the supported “other product” console functions. The Virtualization Engine console bridge to the IBM Director server, as shown in Figure 2-11, can perform the IBM Director console functions.

The IBM Director Bridge is an optionally installed component of the Virtualization Engine console, which allows you to perform IBM Director console functions without the IBM Director console component being installed.

From your browser you connect to the Virtualization Engine console “portlet” running under a WebSphere Application Server. The Virtualization Engine console has been installed knowing it can use the bridge to interface with IBM Director server.

Note: IBM Director server, console, and agent can be ordered, installed and run on the supported operating systems. These IBM Director components do not use a WebSphere Application Server instance to perform their functions.

IBM Director 5.10 CDs are also included in “packages,” such as Virtualization Engine console and IBM Director, that are associated with the newer releases of i5/OS - V5R3 or later, and AIX 5L - 5.3 or later dependent on the planned management server operating system. Many of the Virtualization Engine products (the Virtualization Engine console, in our context), require an HTTP server instance, a WebSphere Application Server instance and an associated WebSphere Portal server to be installed. All of these products/components are included in the Virtualization Engine package of CDs.

As part of the Virtualization Engine installation wizard process, the installation of any of the required products detects if one of these products is already installed and, if so, bypasses installation of that product. One of the choices you are asked to make during Virtualization Engine installation is what “bridges,” if any, to other products such as IBM Director or Enterprise Workload Manager, you want to include with the installation.

See Chapter 4, “The Virtualization Engine console” on page 133 in this presentation guide.

► IBM Director Server

This system is the central repository of all IBM Director information, such as collected inventory data, the definitions of “monitors” and so forth. The IBM Director console connects to this server to have the IBM Director function performed or to view the collected information. As you can see to the right of the IBM Director server ‘box” there is a separate box showing items such as BladeCenter, SMI-S Storage devices, and Simple Network Management Protocol (SNMP) devices.

BladeCenter is an xSeries-specific hardware configuration and IBM Director provides special functions for a BladeCenter configuration. A BladeCenter unit consists of a chassis, one or two management modules, one or more network devices (previously called switches, up to a total of four), and one or more blade servers (up to a total of 14, depending on the model).

The chassis is the physical enclosure that contains the blade servers. The chassis has one or two management modules that contain a service processor. IBM Director discovers the chassis and gathers information from the chassis by way of the management module. You cannot install IBM Director Agent or IBM Director Core Services on the chassis. However, IBM Director Agent or IBM Director Core Services can be installed on the operating system running on the blade server.

You also see the HMC and service processor and Hypervisor. Specific IBM Director capabilities for these items are planned for 1Q 2006 and further details were not available when this chapter was created.

Recall that any statements made about IBM plans are subject to change without notice.

► Agents (managed systems or managed nodes)

These are the hardware and operating system “nodes” that are managed by the IBM Director server under control of a specific IBM Director console. The agents are the IBM Director components that do what the IBM Director server has “told” them to perform.

Starting with IBM Director 5.10, the single agent component of an earlier release has become a 3-levels of function agent:

- IBM Director Agent Software is installed for level 2 system

- IBM Director Core Services is installed for level 1 system
 - No IBM Director component is installed for level 0 system
- ▶ SNMP Device
 - like discussed before (see Figure 2-2 on page 42), there is a support for SNMP devices with IBM Director 5.10
 - ▶ SMI-S Storage
 - SMI-S is a industry standard and means *Storage Management Initiative Specification* and is for discovering and managing storage devices
 - IBM Director 5.10 uses this standard SMI-S for example for IBM entry-level disk systems IBM TotalStorage DS300 and DS400, more about both see:
<http://www-03.ibm.com/servers/storage/disk/ds/index.html>

2.8 IBM Director 5.10 cross platform functions

ibm.com/redbooks | International Technical Support Organization **IBM**

IBM Director 5.10 Cross Platform Functions

- **Discovery, Inventory** Discover and Inventory my environment
- **Monitor, Hardware Status** Let me know if my servers are healthy
- **Event Action Plans** Define events and actions to notify me if there is a problem
- **Process Manager** Monitor my important OS and application processes
- **Group Management** Create and manage dynamic and static groups of systems
- **Remote Session/Remote Control** Establish either a graphical remote session or command line sessions to all of my remote systems
- **File Transfer** Transfer files and manage directories on systems and groups
- **Task Scheduler** Schedule and monitor non-interactive management tasks
- **Software Distribution/Update Assistant** Deploy firmware, device driver, SW packages to systems or groups

Redbooks Workshop
IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 2-12 IBM Director 5.10 Cross Platform Functions

In this topic we describe IBM Director Cross Platform Functions in more detail. While these functions are named “cross” there are few differences between implementation for different operating systems and / or hardware environments, for which we give examples. The whole list of IBM Director Cross Platform Functions is shown in Table 2-1.

Table 2-1 IBM Director 5.10 Function summary

Function	Descriptions
Discovery	Automatically discover the systems and devices in your environment. You customize the rules for discovery and can also “manually discover” a specific host name or IP address to discover. Discovery rules include “discover IBM Director agents” or discover “SNMP devices.”
Inventory	Gather and store in a central database information about the hardware, software, and fix components on your managed systems. The database can be viewed, exported, and customized. Some of the inventory collected is displayed regardless of hardware platform and installed operating system, while some of the information displayed is unique to the hardware platform and installed operating system. If you had the extended functions for BladeCenter support, BladeCenter unique information is located here.

Function	Descriptions
Resource Monitors	Monitor the various resources on your systems such as CPU and disk utilization and specify to let you know of any problems or unusual situations. "Threshold values" can be specified as an "event" to trigger an action such as highlighting the occurrence on the IBM Director console or take an action, such as sending a message, or calling a pager device. See the Event Action Plan entry later in this table. Some monitors are "the same" across all operating systems, such as CPU utilization value, while others are unique to each hardware platform and operating system. i5/OS unique monitors include file changes, job queues, and message queues.
Process Management	Monitor the important operating system and application processes and let you manage them. For example, you can view, control, and monitor i5/OS jobs, subsystems, and server jobs. There is also a Process Task function to run a program (on Windows, Linux or i5/OS).
Event Action Plan	Define events and actions to notify you if there is some "problem" monitored by IBM Director. A simple example is a CPU utilization threshold has been triggered. There are many predefined actions that you can take in response to events. These include: run a program on the agent, run a program on the server, send an E-mail, send a page, send a Tivoli event, display the event with an associated icon, and others.
Group Management	Create and manage dynamic and static groups of managed objects, so that appropriate tasks can be run against all of the managed objects or only a specific one. Static groups are collections remote systems that you manage. Dynamic groups are defined based on inventory attributes such as operating system type, installed application, and/or installed hardware. Agents dynamically join or leave the group as their inventory attributes change.
Remote Session, Remote Control	Establish either a graphical remote session or a command line session to your remote managed system so that you can centrally manage them. Remote Session is typically a telnet session to the managed operating system. Remote Control is only supported for Windows systems.
File Transfer	Transfer files and manage directories on systems. This data transfer is considered an interactive task and, as such, is not available to the job scheduling IBM Director support. If you get the extended functions with the Premium Edition of Software Distribution, you can import multiple files into a package that can be scheduled for later transfer.
Scheduler	Schedule and monitor non-interactive management tasks or jobs (operating system specific constructs). Any non-interactive IBM Director task can be run immediately or scheduled to run later.
Update Assistant (base) and Software Distribution (Premium Edition)	Update Assistant enables the deployment of firmware, device driver, fixes, and software packages, or files to systems or groups. Update Assistant does not provide the "package build function." A package built by IBM, for example the IBM Director Agent software for an i5/OS system, comes pre-built when IBM Director Server is installed in an i5/OS system (depends on IBM Director agents selection). If you get the IBM Director Premium Edition, you get the capability to create packages for AIX installation, i5/OS object, library, licensed program installation, InstallShield, and Windows Installer RPM. This "advanced package support" includes building a simple list of data files or fixes into a package. All of these packages can be scheduled to be sent to multiple groups at different times.

Function	Descriptions
Hardware and operating system specific extensions ^a	There extended IBM Director functions. Some are free downloads and some are additional cost capabilities.
Simple Network Management Protocol (SNMP)	Discover and manage SNMP devices such as printers or an SNMP server running on i5/OS.
Administration Management	Manage IBM Director administrator accounts to control which administrators can perform which functions on which agents.
Blade Center Management	Discover, configure, and manage BladeCenter chassis and their blades.
Hardware Management Console (HMC) Integration	Discover and manage Hardware Management Console (HMC) systems for pSeries and iSeries. This function will be fully functional during 2006.
Upward Migration	Provide hardware alerts and inventories for IBM and third party systems management products such as Tivoli Enterprise™, Tivoli Netview, CA Unicenter TNG, HP Openview, and Microsoft SMS and MOM. This is available several operating systems use the operating system's Common Information Management Object Model (CIMOM) support.

a. See 2.9, "IBM Director extensions" on page 65 for more information.

2.9 IBM Director extensions

ibm.com/redbooks | International Technical Support Organization **IBM**

IBM Director 5.10 Extensions

- **Installation base extensions**
 - BladeCenter Management
 - Rack Manager
- **Free Web Download**
 - System Availability
 - Virtual Machine Manager
 - ServerRAID manager
 - Web-based Access
 - z/VM Center
 - IBM Real Time Diagnostics
 - Update Express
 - HMC Manager
- **Fee Based Extension**
 - Capacity Manager
 - Software Distribution Premium Edition
 - Remote Deployment Manager
 - Virtual Server Deployment and Server Complex (Linux on zSeries)

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 2-13 Separately Available IBM Director 5.10 Extensions

A large set of functions come with the IBM Director commonly ordered with an associated operating system. These can be considered “base functions”. There are two extensions that can be optionally installed as part of the base functions of IBM Director. This section describes these base extensions as well as “extended functions” that must be downloaded or additionally ordered.

A common term used to collectively refer to these extended functions is *Extension Task Functions*.

Many of the extensions have functions unique to one or two of the supported hardware platforms or operating systems. The Software Distribution Premium Edition has the most cross operating system functions among these extensions.

Note that many, but not all, of the extensions listed on this slide were previously described under Section 2.6, “Comprehensive platform management with IBM Director” on page 55.

Base IBM Director install optional extensions

- ▶ **BladeCenter Management:** The BladeCenter Management extension is actually a pair of tools that work together and are focused on managing the IBM Eserver BladeCenter infrastructure. BladeCenter Configuration Manager allows you to create and edit profiles that include configuration settings for the components of BladeCenter chassis, such as ethernet and fibre channel switches, and management modules. These profiles can easily and quickly be applied to BladeCenter chassis in your environment. The Network Device Manager starts vendor software to manage your switches or other network devices.

Depending on the device, a Telnet window, Web interface, or other software interface is started.

- ▶ **Rack Manager:** The Rack Manager is an extension that allows you to group your equipment in rack suites. Using Rack Manager, you can create virtual racks by associating equipment, such as managed systems and devices, networking devices, power devices, and monitors, with a rack in order to visually represent an existing rack in your environment.

You can use Rack Manager to view hardware-status alerts that occur on managed systems or devices in a rack. If a rack component has a hardware-status alert, the rack component is outlined in red, blue, or yellow, depending on the severity level.

Free Download Extensions

- ▶ **Electronic Service Agent:** Allows IBM Director to interface with IBM's support to open support tickets in the case of hardware failures (including Predictive Failure Alerts)

www.ibm.com/support/electronic/

- ▶ **ServeRAID Manager:** The ServeRAID Manager task manages IBM ServeRAID controllers or adapters that are installed locally or remotely on servers. You can view information that is related to controllers, arrays, logical drives, hot-spare drives, and hard disk drives. Also, you can view configuration settings and events and locate defunct hard disk drives.

IBM ServeRAID controllers are typically installed on IBM xSeries systems.

- ▶ **Web-based Access:** This enables browser-based view of managed system information, changing alert standard format (ASF) alerts, changing system settings and configurations, and more. When you install Web-based Access on a managed system, you can access IBM Director Agent and view real-time asset and health information about the managed system from a Web browser.

This feature is supported only on Windows 32-bit operating systems.

- ▶ **System Availability:** This analyzes the availability of a managed system or group. You can view statistics about managed-system uptime and downtime through reports and graphical representations.

This feature is supported only on Windows and Linux operating systems on xSeries platforms.

- ▶ **IBM Real Time Diagnostics:** Real Time Diagnostics is systems management software from IBM that runs on your IBM eServer xSeries system while the operating system is active. It can help prevent downtime; thereby increasing your system's availability.

It runs with Windows and Linux operating systems.

- ▶ **z/VM Center:** This provisions Linux systems on virtual hardware that is based on real System z9 and zSeries hardware and the z/VM hypervisor.

To be able to use z/VM Center, you must purchase the IBM Director Extensions, V5.10 feature of IBM Virtualization Engine and Infrastructure Services for Linux on System z9 and zSeries, V2.1.

- ▶ **Virtual Machine Manager:** This provides management of both physical and virtual machines from a single console. With Virtual Machine Manager (VMM), you can manage both VMware ESX Server and Microsoft Virtual Server environments using IBM Director. VMM also integrates VMware VirtualCenter and IBM Director for advanced virtual machine management.

There are instructions for use of the Virtual Machine Manager in the IBM Virtual Machine Manager Installation and User's Guide, available on the Virtual Machine Manager Web site.

- ▶ **UpdateXpress:** Update Xpress can be used to help reduce the cost of computing by providing a method to update server firmware and firmware of supported options contained within the server on most of your xSeries products. If you have purchased an IBM xSeries server, UpdateXpress is available for download at no additional charge.
- ▶ **HMC Manager Tools:** This is a suite of tools that integrates many Hardware Management Console (HMC) functions into IBM Director. With this extension installed, IBM Director can discover and authenticate with an HMC that is managing IBM eServer iSeries or pSeries systems, then present all Central Electronic Complexes (CECs) and Logical Partitions (LPARs) that are running on these servers. In this manner, the IBM Director console can be used as a single point of management, showing topology and hardware status, as well as providing power and basic hardware control for these advanced server solutions. If more extensive management is required, IBM Director can perform an in-context launch of the HMC directly.

These tools are to be made available originally during early 2006. At the time this chapter was written the tools were not yet available. You cannot perform any IBM Director to HMC functions until the set of tools become available during 2006.

Fee-based download extensions

The following extensions are fee based. Please review the information at the websites listed below. You may need to contract IBM sales representative for additional information:

- ▶ **Capacity Manager:** This provides proactive performance monitoring and bottleneck identification and resolution.
- ▶ **Remote Deployment Manager:** This enables automated, remote system configuration and bare-metal OS deployments.
- ▶ **Software Distribution (Premium Edition)**
Enhances the base Software Distribution task by enabling customer created distribution packages, including some operating system specific distributions.
- ▶ **Application Workload Manager:** Application Workload Manager (AWM) extends the systems management capabilities of IBM Director to provide a comprehensive solution for the allocation, management and control of system resources on IBM xSeries and IBM BladeCenter servers, running a Windows or UNIX operating system.

Designed to eliminate system resource contention and improve performance and reliability in demanding enterprise environments, AWM integrates tightly with IBM Director to enable the precise allocation of CPU and Real and Virtual Memory to specific applications.

This product is provided by Aurema, Inc.¹.

For information on downloading, go to:

http://www-03.ibm.com/servers/eserver/xseries/systems_management/ibm_director/Select_downloads

Table 2-2 shows the list of extended tasks and agent levels required.

Table 2-2 *Extended tasks summary: Agent levels required*

IBM Director Extension Task	Agent Level 0	Agent Level 1	Agent Level 2
BladeCenter Management ^a	NA	NA	NA
Rack Manager			X

¹ More on Aurema at the following web site: www.aurema.com.

IBM Director Extension Task	Agent Level 0	Agent Level 1	Agent Level 2
Electronic Service Agent			X
Hardware Management Console ^b		X	X
ServeRAID Manager			X
IBM Remote Deployment Manager	X	X	X
System Availability			X
Virtual Machine Manager			X
Capacity Manager			X
Software Distribution (Premium Edition)	X	X	X

a. No agent required, Director communicates directly with the BladeCenter management module.

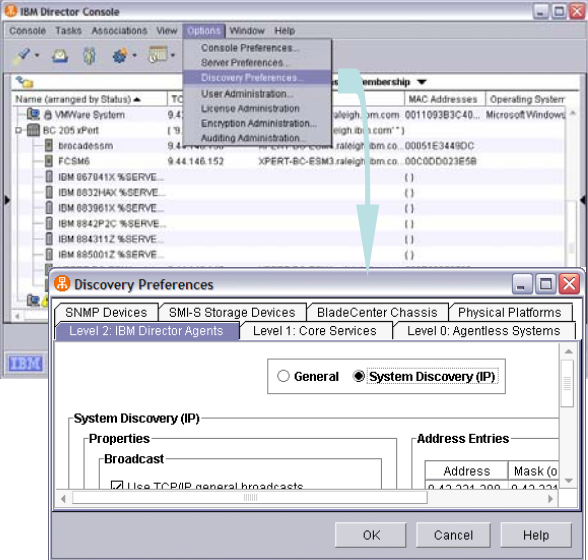
b. For use with iSeries and pSeries systems only. Support planned for 1H 2006. IBM plans are subject to change without notice.

2.10 Discovery cross platform functions

ibm.com/redbooks
International Technical Support Organization

Discovery of Managed Objects

- ❑ Configure discovery settings for all levels of IBM Director systems, SNMP devices, BladeCenter Chassis, and Physical Platforms
- ❑ Automatically secure newly discovered Agents
- ❑ Automatically “discover” Agents which contact the Director Server
- ❑ Support for SNMPv1, SNMPv2c, and SNMP v3 devices
- ❑ Enables renaming of BladeCenter Chassis and Physical Platforms based on available parameters



IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 2-14 Discovery of Managed Objects

Two major functions to get an IBM Director environment up and running with managed systems) are *Discovery* and *Inventory*.

Discovery preferences are defined in IBM Director Console.

As discussed earlier, the highest term used for “a thing to be managed” is a “managed object”. A managed object could be an SNMP device or an IBM Director managed system (running an IBM Director agent. When you drill down to specific management functions often you see the term system to distinguish the managed object from an SNMP device.

So throughout this next set of functions explanations we, in most cases use the term managed object to be generic, but frequently the examples show specific “object types,” such as Level 2 Agent, SNMP devices, and so forth.

Examples of managed object types that can ben discovered are listed below:

- ▶ IBM Director Systems (Deployed Operating Systems)
 - This represents an operating system within a physical platform. In this context the managed system object would be a partition and its operating system on a hardware platform with more than one partition.
 - These would be IBM director agents: level 2, level 1 (Core services), and level 0 agentless. See Figure 2-5 on page 47 for more about managed system level 0,1, and 2.
- ▶ Physical or virtual platforms (blade servers or virtual machines)

- Represents the physical server enclosure that can host an operating system
 - For example, discovery of HS20/JS20 processor blades or xSeries 440 server *prior* to operating system install
- Used by the RDM extension to deploy an operating system to physical platforms or operating system containers
- Used by the RDM extension to deploy an operating system to physical platforms or operating system containers

RDM stands for *Remote Deployment Manager* and is an additional cost IBM Director extension.
- ▶ BladeCenter Chassis
 - Represents *entire* BladeCenter chassis
 - Discovery, management and some inventory performed out-of-band via management module
 - BladeCenter Management and BladeCenter Configuration are customized versions of Server Management and Management Processor Configuration
- ▶ SNMP Devices
 - Represents a SNMP Device like switch module, printer
 - Support for SNMP version 1,2,3

In Figure 2-14 on page 69 we show a Level 2 agent discovery preference for manual discovery of a managed system by specifying one or more IP addresses to discover. In the next topic we show an example of automatic discovery preferences.

2.10.1 Discovery: automatic preferences

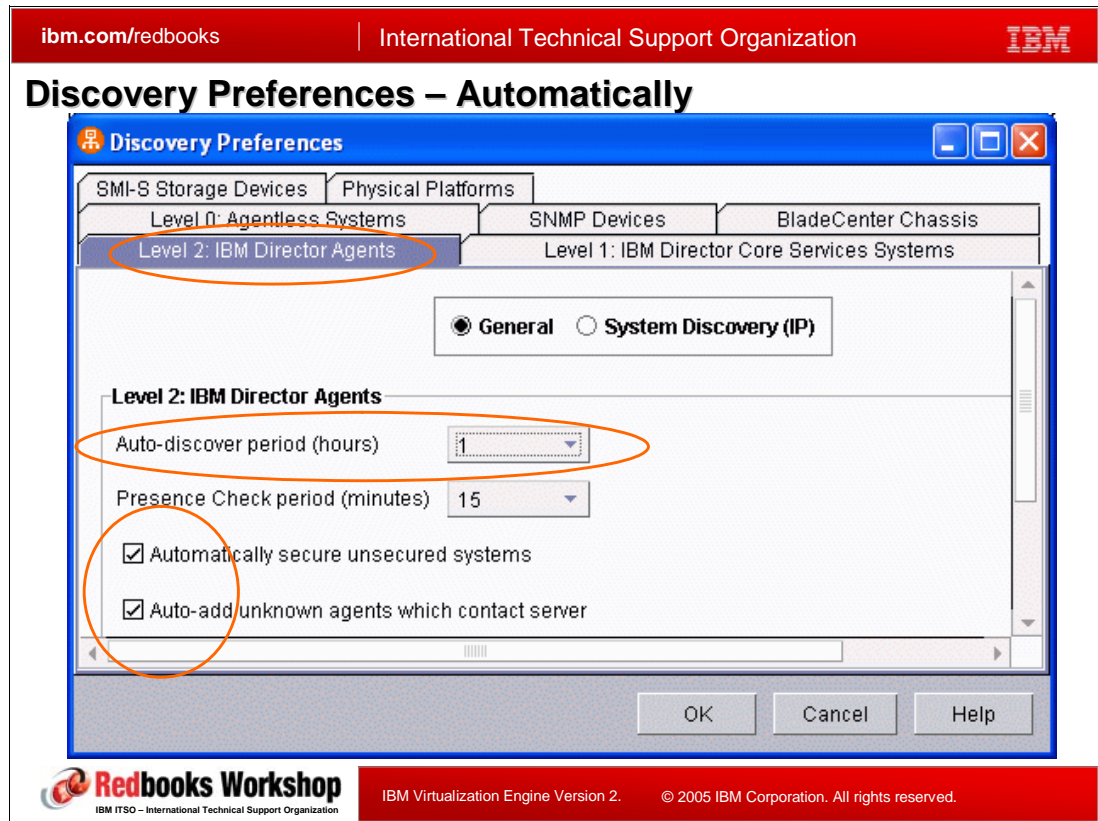


Figure 2-15 In Discovery Preferences - Automatically...

In Figure 2-15 we show a panel of automatic discovery preferences by first clicking the General radio button as shown. Specifying General brings up the lower panel shown with options to specify how frequently to attempt automatic discovery and:

- ▶ Automatically discover IBM Director Agents, which contact the IBM Director Management Server

Select this check box to have the IBM Director Server automatically add to the Group Contents pane unknown agents that contact the server. This process occurs after the discovery task is performed by the server. Unknown agents are those not previously discovered.

- ▶ Automatically secure newly detected Agents

Select this check box to instruct the IBM Director Server to automatically secure any unsecured system that it finds and can manage. When this setting is enabled, the IBM Director Server delivers its public keys to any system it finds, even if the system is secured by another server. The IBM Director Server then secures the client, instructing it to not trust any server for which it does not already have public keys.

2.10.2 Discovery preferences, naming templates

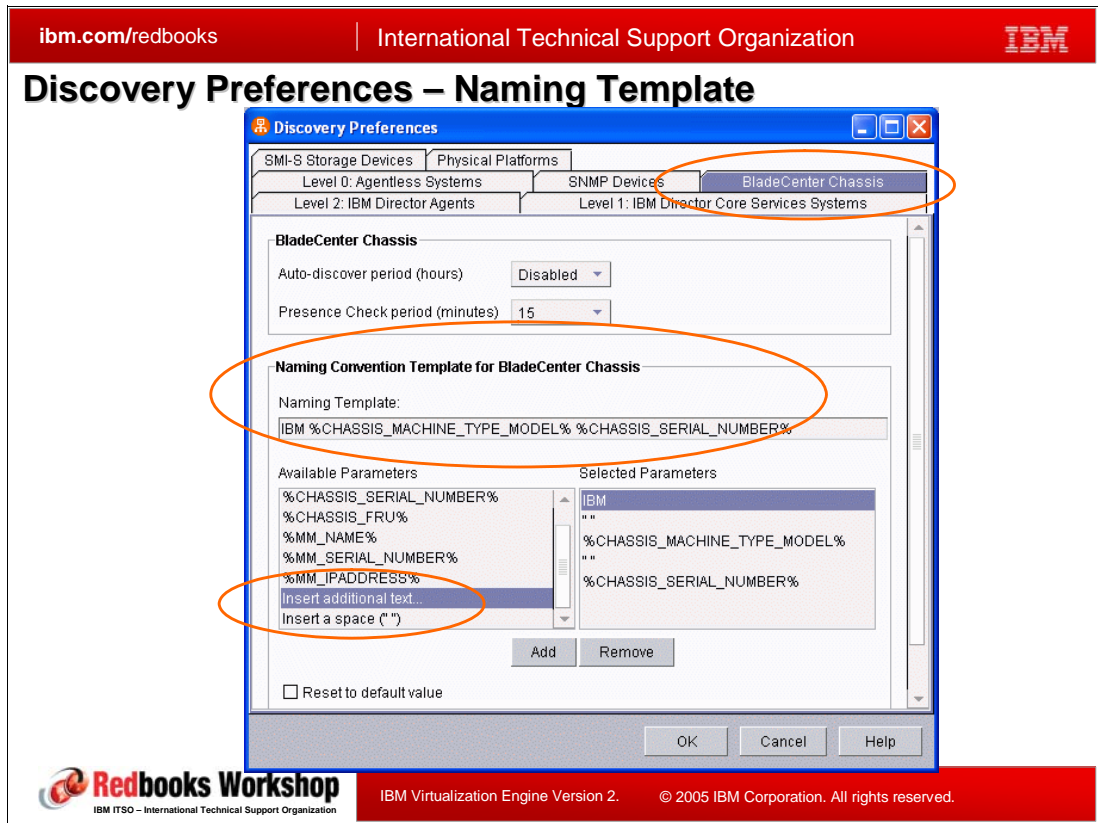


Figure 2-16 Discovery Preferences - Naming Convention Template

Another example of a discovery preference is the use of a Naming Convention Template for a BladeCenter Chassis. Further discussion of BladeCenter naming conventions and available parameters is beyond the scope of this redbook.

For further IBM Director BladeCenter details, go to IBM Director Information Center.

Important: When a managed system has been discovered for the first time, the system appears on the IBM Director console with a lock symbol associated with it. You can perform very little functions with this locked managed system until it is “unlocked.” This system is considered locked until you right click the system and select Request Access in the drop down menu.

2.11 Inventory cross platform functions

ibm.com/redbooks		International Technical Support Organization				IBM
Overview of Inventory Collected by Platform						
Shaded grey cells indicate new with 5.10						
Director Inventory Query	Linux/Windows xSeries	Power Linux iSeries	Power Linux pSeries	Linux on zSeries	AIX, I5/OS	
Base Inventory via operating system: •Disk and Logical Drive •Installed Memory, Processor •Installed Patches and Packages •Network Adapter & IP Address •Operating System/System Information •Physical Enclosure	Yes	Yes	Yes	Yes	Yes	
SNMP Agent Configuration	Yes	Yes	Yes	Yes	Yes	
Serial Number Information	Yes	Yes	Yes	Yes	Yes	
Asset ID/Personalized Data	Yes	Yes	Yes	Yes	Yes	
Lease/Warranty Information	Yes	N/A	N/A	N/A	N/A	
FRU Service Numbers	Yes	Yes	Yes	Yes	Yes	
Memory Modules	Yes	No	No	No	No	
Firmware	Yes	Yes	Yes	Yes	No	
BIOS Details	Yes	N/A	N/A	N/A	N/A	
Processor Details	Yes	Yes	Yes	Yes	Yes – AIX Only	
System Board Configuration	Yes	Yes	Yes	Yes	Yes	
Memory Cache	Yes	Yes	Yes	No	Yes – AIX Only	
Port Connectors	Yes	Yes	No	No	i5OS only	
DISK Partition, Geographic Information	Yes	Yes	Yes	Yes	Yes	

Figure 2-17 Overview - inventory collected by hardware platform

The Inventory function (also called the inventory task) collects information about installed hardware and / or software in your IBM Director environment for all managed systems (level 0, 1, 2), SNMP devices (“agents”), and the IBM Director Management server itself. The inventory information is logically similar across all managed objects but, of course, unique based upon SNMP, agent level, and in some cases, operating system.

Figure 2-17 represents the kind of hardware information you get from the various supported hardware and operating systems.

For our SNMP printer device - an IBM Infoprint® 1120, we viewed our inventory and could see the device was an BM Infoprint 1120 device with, its IP address and host name.

For hardware platform and operating system supported inventory functionality depends on installed level of managed system

- ▶ For level 0 = Agentless Management *Base Inventory* is supported
- ▶ For level 1= Core Services and level 2 = IBM Director Agent *Platform specific Inventory* is supported in addition to *Base Inventory*, see Figure 2-13 on page 65.

For software inventory you get operating system unique information. For a Windows XP workstation, you can view the set of device drivers installed. Device driver information includes driver name, driver vendor name, the directory path in which the driver is installed, and more.

Collected Inventory information is stored in database, which you have defined during IBM Director Server installation.

2.11.1 Inventory subtasks

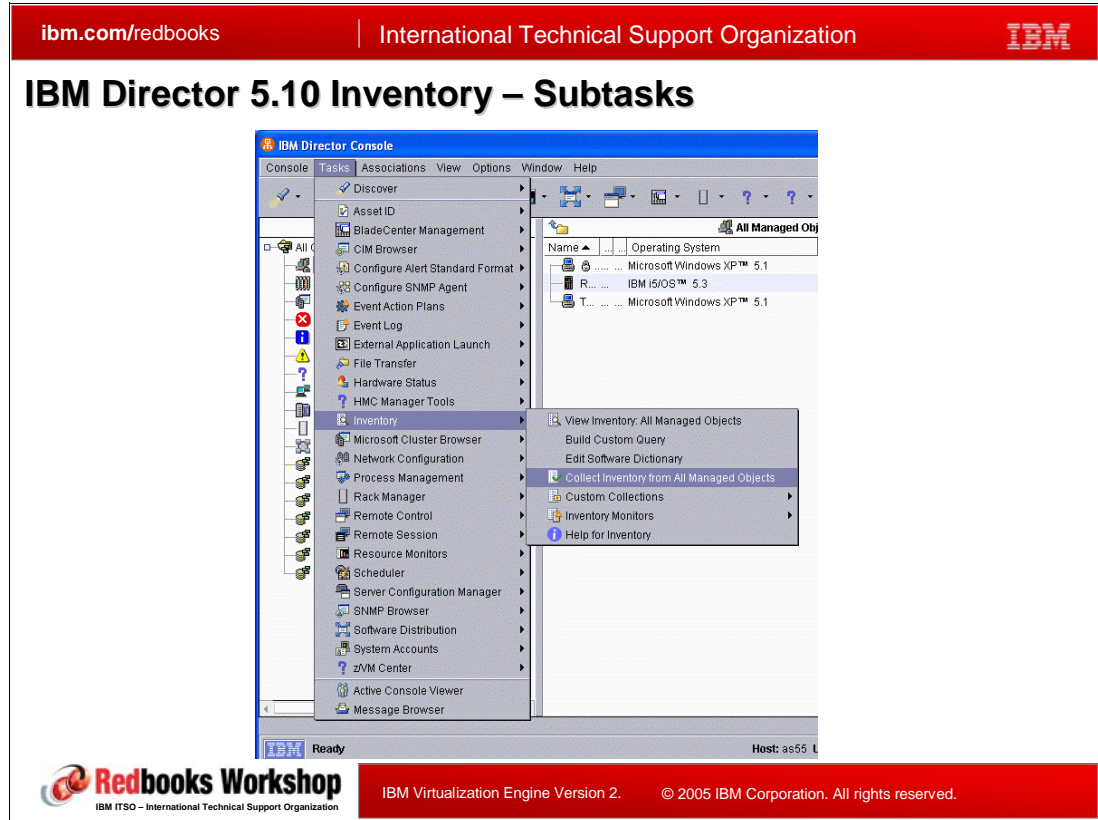


Figure 2-18 IBM Director 5.10 Inventory - Subtasks

Expanding the Inventory task in IBM Director Console 5.10 under Tasks (see Figure 2-18) shows following subtasks:

- ▶ Collect inventory from all managed objects
- ▶ Custom collections
- ▶ Inventory monitors
- ▶ Build a custom query
- ▶ Edit software dictionary
- ▶ View inventory from all managed objects

If you have never collected inventory, you see three subtasks only:

- ▶ Collect Inventory
 - Perform a collection on an object or group. Specify at first under Server Preferences your settings for Inventory Collection and then you requests can be sent to collect inventory. You can monitor the status of that collection.
- ▶ Custom Collections
 - Allows you to specify a subset of inventory items for which to collect data.
- ▶ Inventory Monitors

Allows you to monitor for a change in inventory data, both hardware and software, and generate an event when a change is detected.

► Edit Software Dictionary

You can use the *inventory-software dictionary* to track the software that is installed on your managed systems. When adding a new entry, the file is specified and you do not specify drives or directories that you want the Inventory task to search during the software-inventory collection process. The software-dictionary file contains predefined software profiles that recognize most standard software packages after they are installed. When you install software applications on servers, computers, or devices, the inventory-query browser displays the new software after the next inventory collection.

If there is an existing inventory -that is, there are records in the IBM Director server's database, you see the additional subtask View Inventory, and the option to build your own query of the inventory.

Note: IBM Director Console user must have been given authority to collect inventory data from all managed objects - for immediately or scheduled collection.

IBM Director can collect inventory data when a managed object is discovered initially and at regular intervals, or, rather than collecting inventory upon initial discovery, you can schedule an inventory collection at a more convenient time using the Scheduler feature.

The default interval for refreshing the database is every 7 days. You can change the refresh interval and other inventory-collection parameters using the Inventory Collection Preferences page in the IBM Director Console Server Preferences window. You also can collect inventory data on a managed system or group immediately or schedule an inventory collection using the Scheduler task.

2.11.2 Inventory: custom collection, inventory monitors, and custom query

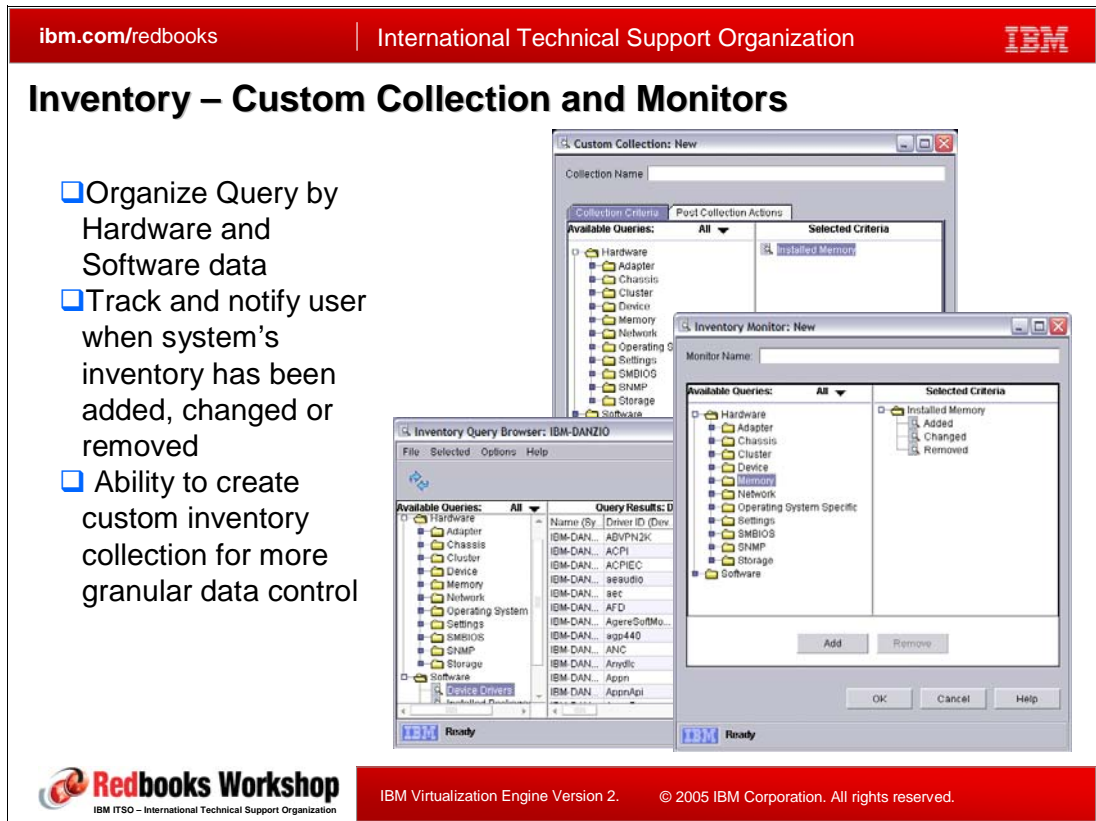


Figure 2-19 Inventory - Custom Collection and Monitors

In Figure 2-19 we show window examples of a custom collection (selecting to collect only installed memory), inventory query browser (device drivers for a PC workstation), and an inventory monitor (detect when memory is added, changed, or removed).

Using the default IBM Directory query you can query the inventory database to display details about properties of a managed system. When you select to view the collected inventory you get 3 folders to select (not shown any figures in this chapter):

- ▶ Custom
- ▶ Hardware
- ▶ Software

We could spend hours just covering all the ways to look at collected hardware inventory data. This is beyond the scope of this chapter, but we can tell you that, for hardware, you get a very long hierarchy tree of hardware categories and some of the tree folder hierarchy may not show the hardware you expected for a specific hardware platform - operating system combination. This is dependent how each operating system responds to the inventory request from the server. You can find the hardware information but it may require selection of a few of the hierarchy tree folders before you find the complete information for that specific hardware platform - operating system combination.

As stated previously, you can use a standard query that is provided or create your own custom query. See Figure 2-20 for an example of a result from a custom query of iSeries Hardware Inventory

2.11.3 Hardware Inventory query - i5/OS example

ibm.com/redbooks
International Technical Support Organization
IBM

Example: i5/OS related Inventory enhancements in Director 5.10

Software – added product feature and option values
Hardware – added physical locations

- Added Auxiliary Storage Pool (ASP) inventory
- ScreenCapture shows result of iSeries Hardware Inventory (custom query)

Physical Resource Name...	Description (iSeries Hardware)	Type (iSeries...	Model (iSer...	Part Number (iSeries...	Serial Number (iSeries...	Physical
CEC01	Main card enclosure	3406	550		10-F17AD	
DSP002	Work station device	3487	0HC		00-00000	
DSP003	Work station device	3487	0HC		00-00000	
DSP007	Work station device	3487	0HC		00-00000	
DSP004	Work station device	3487	0HC		00-00000	
DD004	Disk unit	4326	074	97P3030	21-0128DAD	
DD003	Disk unit	4326	074	97P3030	21-013348A	
DD001	Disk unit	4326	074	97P3030	21-0133E75	
DD020	Disk unit	4326	074	97P3030	21-0133214	
DD019	Disk unit	4326	074	97P3030	21-013355A	
DD010	Disk unit	4326	074	97P3030	21-0133053	
DD015	Disk unit	4326	074	97P3030	21-0133FA1	
DD002	Disk unit	4326	074	97P3030	21-0132DED	
DD012	Disk unit	4326	072	97P3030	21-013364F	
DD009	Disk unit	4326	072	97P3030	21-0133FF9	
DD017	Disk unit	4326	072	97P3030	21-0133581	
DD016	Disk unit	4326	072	97P3030	21-0131489	
DD018	Disk unit	4326	072	97P3030	21-013400E	
DD011	Disk unit	4326	072	97P3030	21-0133AFA	
DD013	Disk unit	4326	072	97P3030	21-0133CB5	
DD014	Disk unit	4326	072	97P3030	21-0133D9B	
DSP005	Work station device	3179	002		00-00000	
CTL04	Work station controller	6A59	002	000000000000	00-00000	C1
OPT01	Optical storage unit	6331	002	0000097P3692	00-00000	
CTL04	Work station controller	6A59	002	000000000000	00-00000	C1
CMN11	Ethernet port	2892	002	0000097P4465	53-4186021	C01
LIN08	Unknown	2892	002	0000097P4465	53-4186021	C01
DSP005	Work station device	3179	002		00-00000	
CC01	Communications processor	2892	002	0000097P4465	53-4186021	C01
LIN01	Communications adapter	6B03	001		00-00000	C0
CMN01	Communications port	6B03	001		00-00000	CU

Redbooks Workshop
IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2.
© 2005 IBM Corporation. All rights reserved.

Figure 2-20 i5/OS related Inventory enhancements in Director 5.10

Figure 2-20 shows hardware inventory information in terms an i5/OS customer would understand. For example:

- ▶ Physical resource name: Depending on the *Description (iSeries Hardware)* column value, the resource name may appear in i5/OS screens or be used in a “device configuration” command to associate, an i5/OS object name to the actual hardware (resource) card.
- ▶ Model (iSeries Hardware): Depending on the *Description (iSeries Hardware)* column value this could convey a LAN adapter capacity or, for a disk unit configured as part of a RAID-5 set:
 - 070: Non-parity member of a parity (RAID) set. Full capacity. Data compression is inactive.
 - 071: Parity member of a parity (RAID) set with sixteen parity members. Fifteen-sixteenths capacity. Data compression is inactive.
 - 072: Parity member of a parity (RAID) set with eight parity members. Seven-eighths capacity. Data compression is inactive.
 - 074: Parity member of a parity (RAID) set with four parity members. Three-fourths capacity. Data compression is inactive.
 - 078: Parity member of a parity (RAID) set with two parity members. Half capacity. Data compression is inactive.
 - 080: Non-parity member of a parity (RAID) set. Full capacity. Data compression is active.

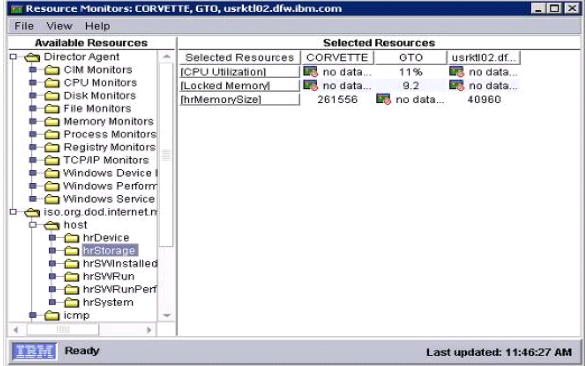
- 082: Parity member of a parity (RAID) set with eight parity members. Seven-eighths capacity. Data compression is active.
- 084: Parity member of a parity (RAID) set with four parity members. Three-fourths capacity. Compression is active.
- ▶ Physical card slot: As circled in Figure 2-20, this refers to the actual card slot used for the device or adapter on an iSeries or IBM System i5 family system.

2.12 Monitors and Hardware Status cross platform functions

ibm.com/redbooks
International Technical Support Organization

Resource Monitors

- Alerts based on user-defined System or Group thresholds
- Data can be recorded and graphed at a later time / date
- Supports both Director Systems (level 2 on all operating systems) and SNMP devices
- Ability to monitor files / directories for changes
- CPU Utilization integrated with Event Action Plan Wizard



Selected Resources	CORVETTE	OTO	usrkt02.df...
[CPU Utilization]	no data...	11%	no data...
[Locked Memory]	no data...	9.2	no data...
[hrMemorySize]	261556	no data...	40960

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 2-21 Resource monitors

All Monitors are used to provide, for a defined time frame or “forever,” unattended monitoring of a certain condition. You can get an idea of the monitors available by looking at the left pane tree structure. In the right pane we show monitors for CPU utilization, locked memory, and memory size.

You can define resource monitors for: CPU, disk, files, directories, memory usage, processes, TCP/IP activity, and Windows operating systems devices, performance, and services. Starting with 5.10 defining and managing a CPU utilization monitor is assisted by the Event Action Plan wizard.

You can define a monitor to be associated with a specific system or a group of systems (CORVirtualization EngineTTE, GTO, and so forth). You can also monitor some resources of one or more SNMP devices,;

A monitor can record data collected at a specific time that can be reviewed and, depending on the defined monitor object, graphed at a later time.

You associate an IBM Director Event with the monitor to highlight an icon, or “send a message” or take some other action, such as sending an alert to a Tivoli monitoring products.

Sometimes monitors are combined with activities like run a command - all in unattended mode.

With some resource monitors, you can set thresholds to detect potential problems with managed systems or devices. When a threshold is met or exceeded, an event is generated. You create event action plans to respond to resource-monitor events.

Example of resource monitor: CPU utilization on an i5/OS partition

Resource monitors and thresholds are configured for a managed agent, using the IBM Director console connected to an IBM Director server. When the monitored conditions occurs the agent generates an event and reports the condition back to the IBM Director server.

The next set of slides and figures provide an example of setting up and using a CPU resource monitor for an i5/OS partition:

1. Select the system or group whose resource(s) you want to monitor
2. Select the resource monitor - i5/OS CPU utilization, in our example.

You can define a single resource monitor with multiple resource types being monitored within a single monitor. The resource monitor can be assigned to a single system or group of managed systems that you created. Multiple operating systems can be monitored within the resource monitor, but some “available resource types” may be unique to a specific operating system

In our CPU utilization example, note that the i5/OS CPU utilization resource type is distinct from a CPU utilization resource for an xSeries system running, for example a Windows operating system. We also show only a single managed system being monitored. When starting out with IBM Director’s resource monitoring we recommend defining a monitor with a single resource type (available resources) for a single system being monitored. Then add an event action plan to experiment with automated actions.

After you gain experience you can progress to defining a resource monitor with more than one resource type and over a group of managed systems running the same operating system. You can progress to more complex resource monitoring, depending on your particular network and types of operating systems supported.

3. Define a threshold for a selected resource type

In our example we selected an i5/OS resource - CPU % utilization and specify an Above or Equal value of 70%. We also specify how long CPU % utilization must last before signalling an event. In a real environment you would specify a longer time value than the 1 minute value we use in our example. You can also specify how long to wait before another occurrence of the event is signalled.

4. Start the monitor by associating it with a managed system or group. In our example we do not show defining an event action plan and associating it with the threshold and resource type being monitored. We merely trigger an event. Later in this chapter we show setting up an event action plan to take some automated action.
5. Look at the IBM Director console all managed objects pane and detect a visible icon that a threshold event has occurred
6. View the event log to see all the defined events that have occurred over a period of time. Note that events in addition to the ones for our CPU % utilization threshold are shown in our example. In our example some IBM Director default events are shown such as “user xxxx has logged on.” You can turn these off.

The example starts by showing selection of an i5/OS CPU percent utilization resource.

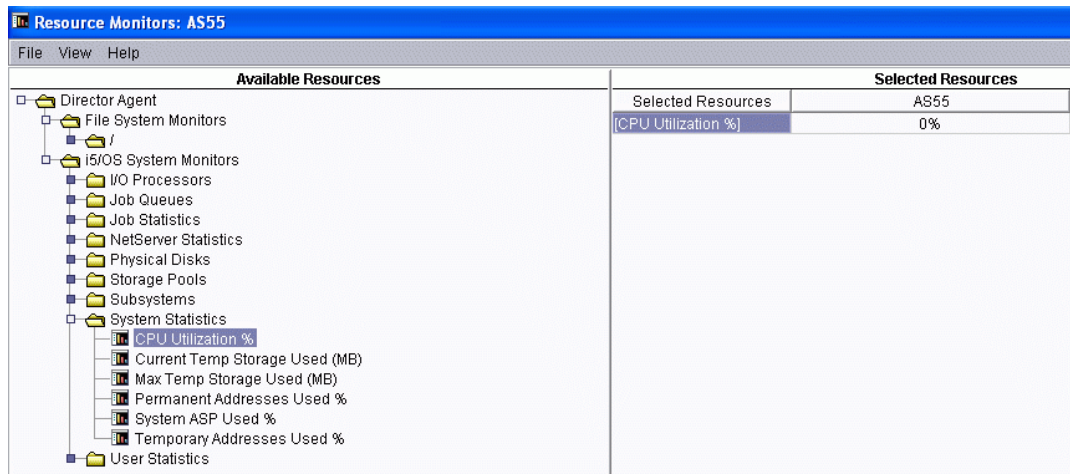
Step 1: create a resource monitor

Figure 2-22 Resource Monitor: CPU Utilization i5/OS environment (AS55)

Using Figure 2-22 as a base:

1. Create a Resource Monitor for CPU Monitoring in a i5/OS environment
 - a. In *IBM Director Console*, select the system (AS55, in our example) in the view **All Managed Objects**
 - b. Select **Tasks** → **Resource Monitors** → **Monitor Resources**.
 - c. The New **Resource Monitors: AS55** window appears.
 - d. In order, expand **Director Agent** → **i5/OS System Monitors** → **System Statistics**. Double click on **CPU Utilization %** . You get on the right pane of Figure 2-22 as shown under **Selected Resources** [CPU Utilization. Figure 2-22 on page 81
 - e. Click File → Save As. A new window pops up. Enter a name meaningful to you for this monitor. In our example we use AS55CPUmon.

Step 2: setting up a CPU utilization threshold

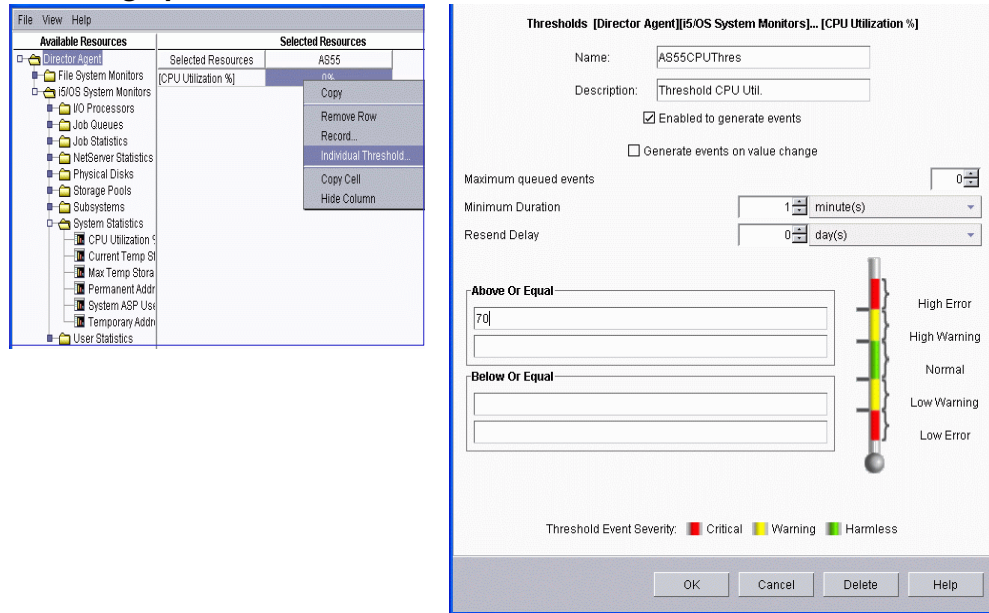


Figure 2-23 Resource Monitor: Individual Threshold

2. Now, define a specific Threshold inside your created Monitor AS55CPUmon
 - a. Right click in cell under **Selected Resources** in the right part of window on our defined monitor. The menu list shown on the right of Figure 2-23 appears. This takes you to a new window shown on the right pane of Figure 2-23.
 - b. In this simple example we set up above 70 (percent) for a High Warning threshold. We selected a minimum duration value of 1 minute, just to capture the windows used in this example as quickly as possible. In real life you would not want this threshold triggered for normal short spikes of high CPU utilizations

You can select to additional thresholds, such as High Error or Low Warning or Low Error. You decide.

Note the Record option in the same list containing the Individual Threshold option. The Record option lets you specify how long to retain threshold event occurrences. You can later look at this information over a period of time to observe peaks and valleys

- c. Click on **OK** in same window and the window is closed

Step 3: Starting and reviewing a resource monitor with a managed object

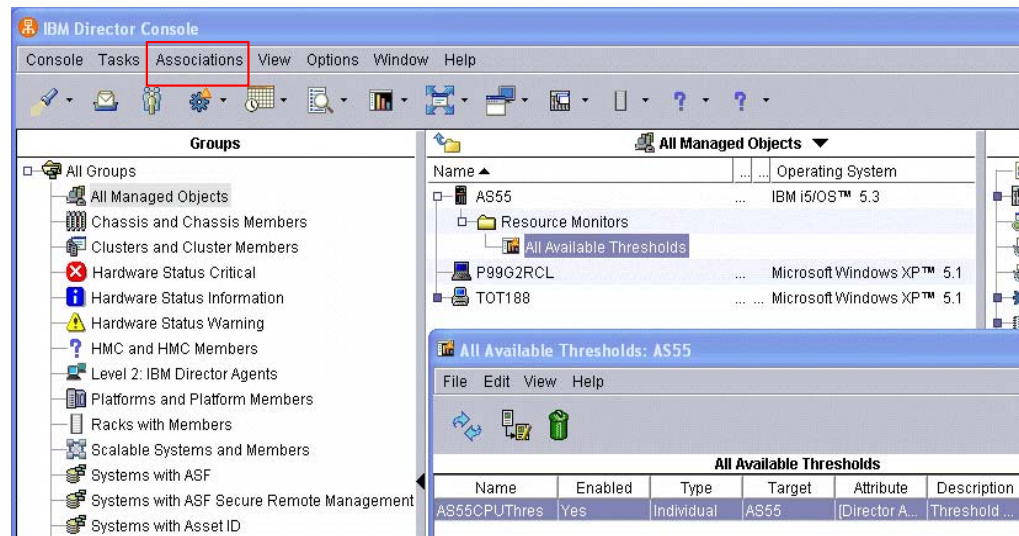


Figure 2-24 Definition Association

- d. From the IBM Director Console shown in Figure 2-24, from the top menu bar select **Associations** → **Resource Monitors**. In the right pane showing **All Managed Objects**. For AS55, expand Resource Monitors → All Available Thresholds. Open (double click). You can start the monitor here. You could also have scheduled the monitor to run only on certain dates and time. We have not shown this here.
- e. As shown, AS55CPUThres has been started (enabled). From here, if you wish, you can further customize your defined threshold. In our scenario we return to the IBM Director console to keep an eye on our monitor.

Step 4: Monitoring your resource monitor

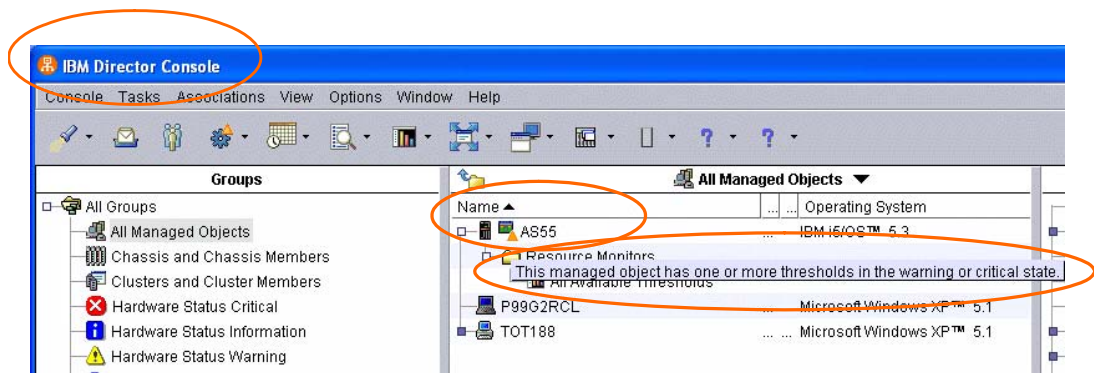


Figure 2-25 Message shown after Monitor and Threshold is up and running and condition reached

3. From the IBM Director console shown in Figure 2-25, you can see by the icon next to AS55 and the “bubble text,” that one or more “thresholds are in the warning or critical state.”

To keep this chapter from becoming a redbook we did not specify any automated actions in our example. We do, however, show the “events log” so you can get an idea of the logged information available to you.

Step 5: A resource monitor event log

Events (9) - Last 24 Hours					
Date	Time	Event Text	System	Severity	Category
10/4/2005	3:28:45 PM	User 'TOT188' (TOT188) logged on server from 's...	AS55	Harmless	Alert
10/4/2005	3:20:23 PM	User 'TOT188' (TOT188) logged off server from 'si...	AS55	Harmless	Alert
10/4/2005	3:08:00 PM	Monitor 'AS55CPUThres' High Error: 'CPU Utilizati...	AS55	Critical	Alert
10/4/2005	11:27:34 AM	System 'AS55' is online	AS55	Harmless	Resolution
Monitor 'AS55CPUThres' High Error: 'CPU Utilization %' has been above or equal to 70 for 0:01:00. Value reported is 98.					
10/4/2005	10:59:46 AM	System 'AS55' is online	AS55	Harmless	Resolution
10/4/2005	10:59:34 AM	User 'TOT188' (TOT188) logged on server from 's...	AS55	Harmless	Alert
10/4/2005	10:44:54 AM	System 'AS55' is offline	AS55	Harmless	Alert
10/4/2005	8:14:08 AM	User 'JCOOK' (JCOOK) logged on server from '9...	AS55	Harmless	Alert

Figure 2-26 Event Log AS55 with entry about critical situation

One of the highlights of IBM Director's resource monitor support is that you can go back and look at past events. You can also specify how long to retain recordings. For certain resources you can graph the retained records.

In our example in this section we look at event logs.

4. Select your managed system (in our example AS55) in the **All Managed Objects** view. From the menu bar click **Tasks** → **Event Log** → **Event Log: AS55**

A New window pops up, which shows all event log entries for AS55, note a critical event is included. By using your mouse pointer on this message you get more details, as shown in Figure 2-26

You find more examples of using Resource Monitors in *IBM Director 5.10*, SG24-6188-02, including:

- CPU Monitoring
- Disk Storage Monitoring
- File Monitoring
- Process Monitoring

Tips:

Process Monitoring as sub function of *Resource Monitors*, and gives you more options like information about:

- ▶ Total execution time
- ▶ Current active processes
- ▶ Maximum running at once (today or yesterday)

Process Management, not Process Monitoring, gives you information about process state changes.

2.13 Inventory monitors



Figure 2-27 Inventory monitor - hardware example

Inventory Monitors are available as a subtask under Inventory (Figure 2-18 on page 74).

An Inventory Monitor allows you to track changes in collected inventory data, both hardware and software. In the Inventory subtask select to create this kind of monitor.

Select **Inventory Monitor** → **Create**.

in Figure 2-27 we have already selected to monitor hardware inventory memory. we have also expanded the memory classifications on the right pane and selected to monitor memory that is added, changed, or removed.

We do not show any more details on monitoring inventory. After saving our monitor JCMemMon you have to find this monitor on the IBM Director console and drag it to a system or a managed group in order to associate the defined monitor to the system or systems in the group.

To get any information from you monitor you also need to define and associate an Event action plan with the monitor.

Then you can run the monitor and observer any memory additions, removals, of changes.

2.14 Hardware status

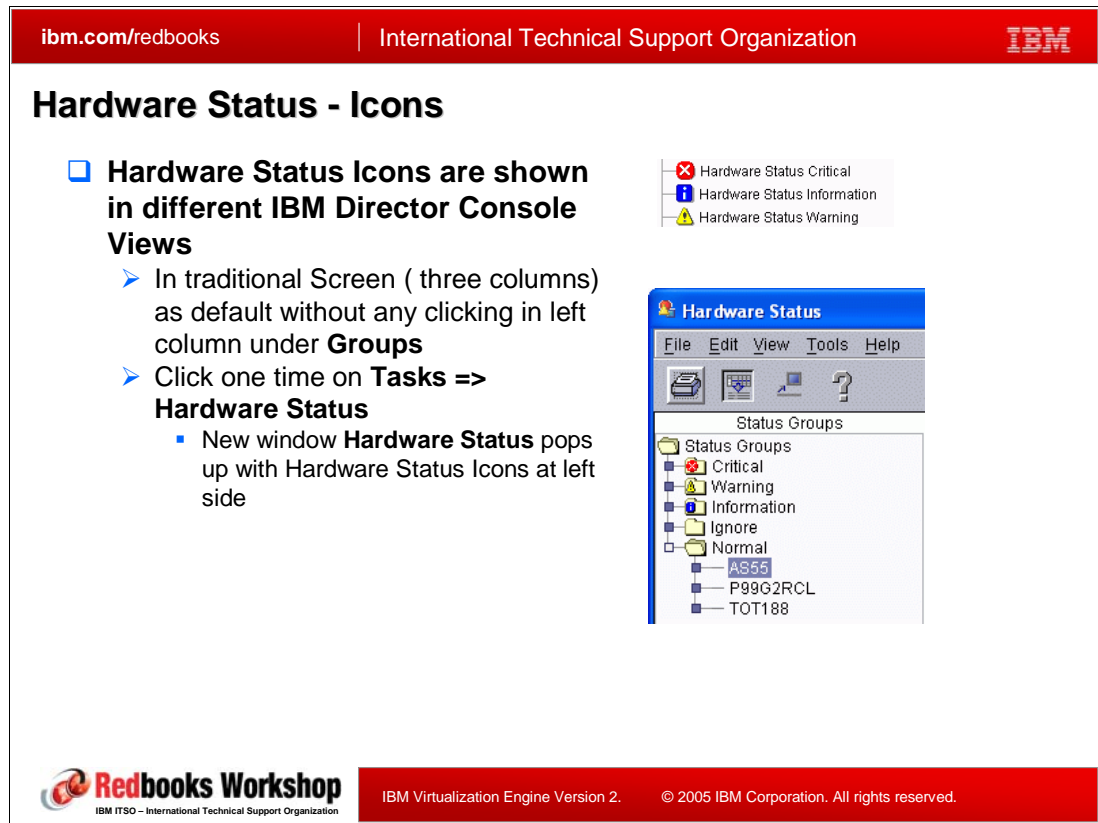


Figure 2-28 Hardware Status - Icons

The hardware status function (task) of IBM Director is able to show you, at a glance, possible troubles with you managed object. This function is separate and distinct from resource monitors

There are different severity level errors and for each severity there is a special icon. For example:

- ▶ red = critical error
- ▶ yellow = error with warning status
- ▶ blue = error with informational status

as shown in Figure 2-28.

If one system has more than one error, for example one critical error and two warnings, then than highest error in this example - critical error (red icon) will be shown and yellow icons for both warnings are not shown. You must drill down to see all the error/status details.

In addition to this at a glance view with icons there is a hardware status screen with details about each error. We do not show an example in this chapter.

Note: To prevent getting unnecessary messages, there is a option for *Ignore Events*.

A typical situation could be an IBM Director managed system has two Network Interface Cards (NICs). The first one is used (for example the typical Ethernet card) and second is not used and offline. An example on many PC is the Wireless) NIC. By default you get a yellow warning icon and message, saying "NIC 2 is offline." Of course this is "normal" in many office environments.

You can ignore this event by selecting: IBM Director Console **Tasks** → **Hardware Status**. Select your workstation and then **Communications**. Your NIC interfaces will be shown. Right click the wireless NIC and select **Ignore Events**

Please note, this Ignore does not survive reboot of the managed system or workstation. The error indication will probably reappear after your reboot your workstation.

The Hardware Status can also be launched from the Hardware Status "Dashboard" in the lower right corner of IBM Director Console.

Please note two points:

- ▶ The Hardware Status task can be applied to individual objects or to groups to view the status
- ▶ Even though the Hardware Status task is an IBM Director base Task and will be used with IBM Director console, you need for managed systems to have installed the IBM Director software, as shown below:
 - Hardware Status is supported on managed system with level 1 or 2. This means, you need one of the following installed.
 - for level 1: Core Services have to be installed on managed system
 - for level 2: IBM Director Agent has to be installed on managed system

When an agent level 1 system is discovered and unlocked, hardware status gets the initial status. All subsequent updates to the Hardware Status GUI for the system are made as a result of asynchronous events sent to the Director server by the system. Initial status for a system is retrieved:

- ▶ When an already discovered agent level 1 system goes to Online from offline state
- ▶ When the Director server managing the system is restarted
- ▶ When a new agent level 1 system is discovered and unlocked
- ▶ When already unlocked agent level 0 system is promoted to agent level 1

2.15 Event Action Plans cross platform functions

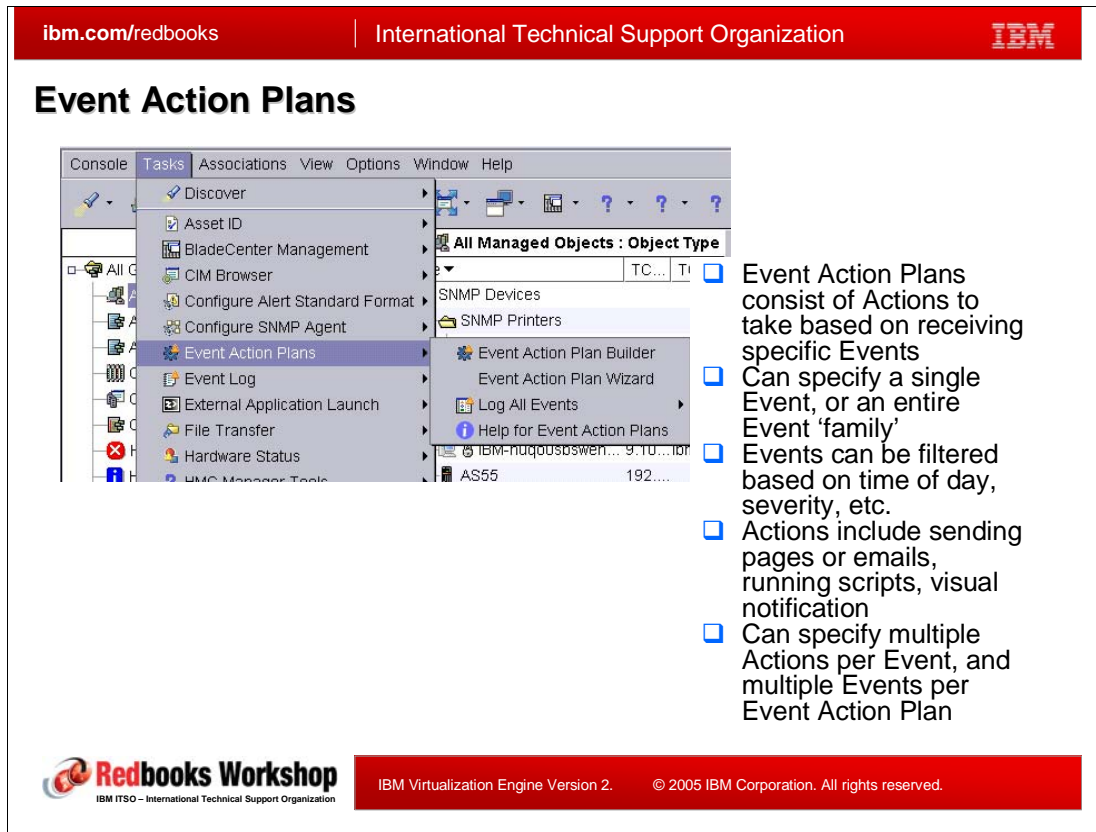


Figure 2-29 Event Action Plans - overview

Use Event Action Plans to specify actions that are performed in response to events that are generated by a managed object. Managed objects include, but are not limited to, Level-0, Level-1, Level-2 managed systems, SNMP devices, BladeCenter management modules, platforms, and switches.

An event action plan is composed of two types of components:

- ▶ One or more event filters, which specify event types and any related parameters
- ▶ One or more event actions, which occur in response to filtered events

You can apply an event action plan to an individual managed object (an example is shown in Figure 2-32 on page 93), several managed objects, or a group of managed objects.

By creating event action plans and applying them to specific managed objects, you can be notified by e-mail or pager, for example, when a specified threshold is reached or a specified event occurs. Alternatively you could also choose to specify an event action plan to start a program on a managed object (IBM Director Server or any managed system) and change a managed-object variable when a specific event occurs. You can use any event, including process-monitor and resource-monitor events, to build an event action plan.

When you install IBM Director, a single event action plan is already defined, in addition to any that you create using the Event Action Plan wizard. There is an event log that records the history of all event action plans. We show an example of this later in

Event Action Plan Wizard

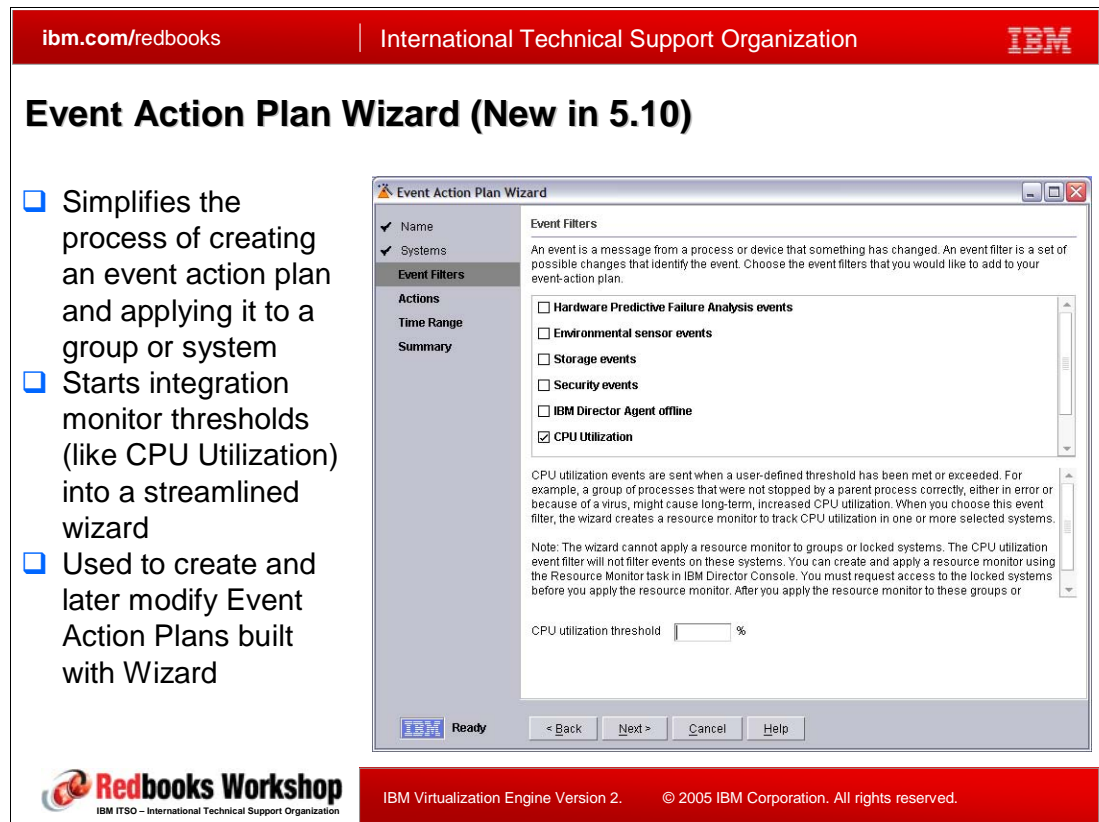


Figure 2-30 Event Action Plan Wizard

You can build an event action plan by selecting **Tasks** → **Event Action Plans** → **Event Action Plan Wizard**. As shown in Figure 2-30 in the left “navigation area,” we have already named the event action plan and passed the step to assign the Event Action Plan to systems. We could have specified systems during that step but we can also associate the defined Event Action Plan later to a managed system or a group.

You can combine event filters and event actions to easily create an event-action plan.

In the Event Action Plan Wizard Event Filters page, **select the check boxes** adjacent to the types of events you want to monitor. You can select the following event filters:

- ▶ **Hardware Predictive Failure Analysis® (PFA) events**

These events are sent when failure of a computer subsystem is imminent. Some of the subsystems for which IBM Director sends PFA events include hard disk drives, voltage regulation modules, power supplies, and thermal sensors.
- ▶ **Environmental sensor events**

These events are sent when environmental sensors built into a system board detect that a manufacturer-defined threshold has been exceeded. For example, a fan shaft might have stopped.
- ▶ **Storage events**

Storage events are sent when there has been a change in the status of a storage subsystem. Examples might be the failure of a hard disk drive or a logical drive that has been rebuilt.
- ▶ **Security events**

These events are sent when there has been a change in the status of physical security. An example might be that someone has removed the system from the LAN or opened the system enclosure.

► IBM Director Agent offline

An IBM Director Agent offline event is sent when IBM Director Agent stops running because the program either failed or was intentionally stopped.

► CPU Utilization

CPU utilization events are sent when a user-defined threshold has been reached or exceeded. For example, a group of processes that were not stopped by a parent process correctly, either in error or because of a virus, might cause long-term, increased CPU utilization. When you choose this event filter, the wizard creates a resource monitor to track microprocessor use in one or more selected systems.

Note: The wizard cannot apply a resource monitor to groups or locked systems. You can create and apply a resource monitor using the Resource Monitor task in IBM Director Console. You must request access to the locked systems before you apply the resource monitor. After you apply the resource monitor to these groups or previously locked systems, the event filter will start to filter these events.

► Memory use

Memory-use events are sent when a user-defined threshold has been met or exceeded. For example, a memory leak, caused by a process that has been stopped unexpectedly without freeing resources, might occur. When you choose this event filter, the wizard creates memory-used and memory-available resource monitors to track the memory use in one or more selected systems.

Note: You cannot apply memory-used and memory-available resource monitors to systems running NetWare or i5/OS .

Click **Next** to go to the wizard's event actions window. We do not show the event actions window or the succeeding ones in this chapter. We do, however, describe the steps to complete the definition of the event action plan.

You have three options for event actions:

- Select the **e-mail check box**, if you want to be notified by e-mail when an event occurs. Then, configure the e-mail notification, necessary fields pop up.
- Select the **Pager check box**, if you want to be notified by pager when an event occurs, in the Select the notification window. Then, configure the pager notification, necessary fields pop up.
- If you want to start a program as a result of the event, select the **Start program check box**. Then configure the start program instructions, necessary fields pop up.

Click **Next**.

The next wizard window is the Time Range page. In this window choose the period of time over which you want to collect the events. You can select **All day** to enable the plan to be active all the time. Or you can select **Custom** to choose the time range for the plan to be active during specific days of the week.

Click **Next**.

In the Event Action Plan Wizard, on the Summary page, verify the details of the event action plan. If you need to make changes, click Back.

Click **Finish**.

The event action plan is saved.

To locate the new plan, in the IBM Director Console, click **Tasks** → **Event Action Plans** → **New plan**. The new plan is the name you previously assigned in the wizard. The plan also appears in the Event Action Plan Builder, and under the systems or groups that the plan is applied to when Event Action Plans is selected in the IBM Director Console Associations menu.

Tip:

It is a good idea, to create your first Event Action Plan (EAP) using following steps:

1. Create an EAP using Event Action Plan Wizard
It is important to note, not all possible actions are shown and selectable when defining an event action plan using the wizard.
2. Test your first EAP by applying it to one or two managed systems and verify that the EAP is working.
3. After verifying the EAP is working, you can enhance your EAP using the Event Action Plan Builder
4. In the EAP Builder you have ALL actions available. There are several and covering them is beyond the scope of this chapter.
5. Test your modified EAP again with few managed systems
6. Apply your EAP to all systems you want.

The next slide shows the full, complete, all options available for event action plans using the Event Action Plan Builder.

Event Action Plan Builder

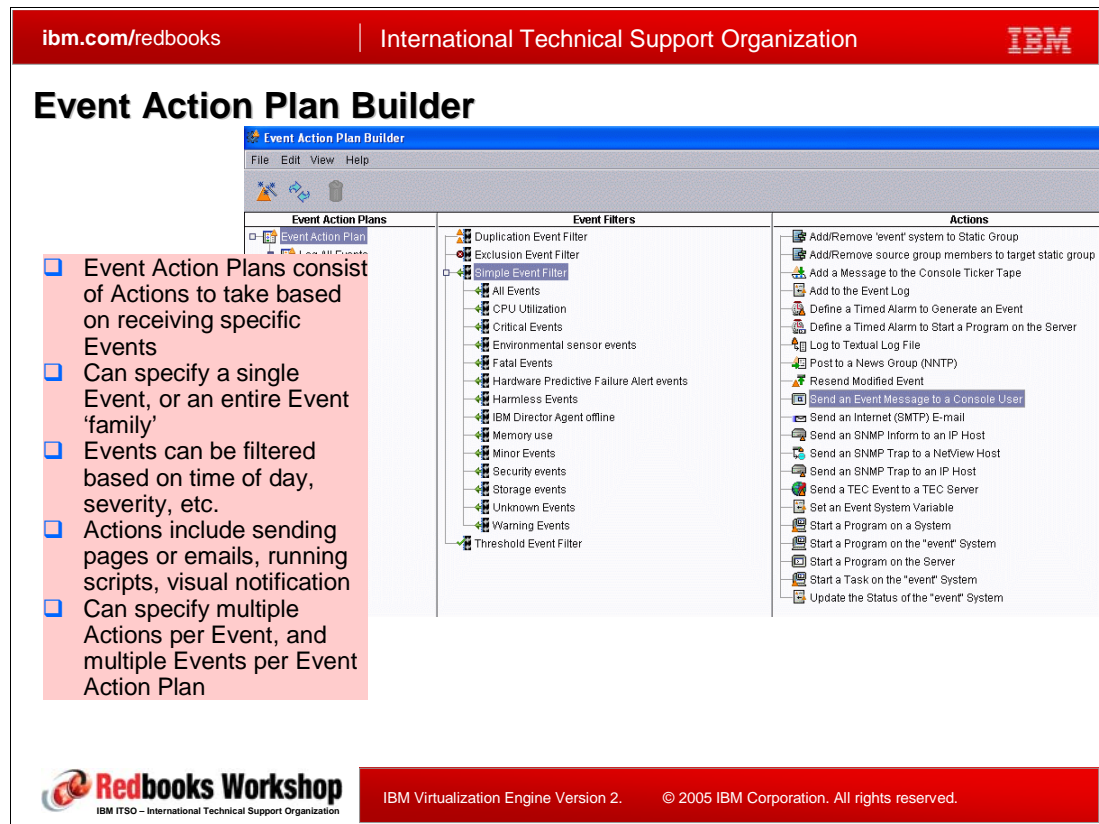


Figure 2-31 Event Action Plan Builder

You get to the Event Action Plan Builder using the IBM Director Console Tasks menu bar:

Tasks → Event Action Plans → Event Action Plan Builder

Looking at Figure 2-31, you can see the impressive set of event action plan capabilities:

- ▶ Logging options
- ▶ Event filters
- ▶ Event actions

Some of the capabilities worth noting include:

Event filters:

- ▶ Simple event: This is the default. IBM Director provides many predefined filters. These predefined filters are useful for creating generic event action plans. Several of these filters use the severity of the event as criteria to allow events to pass through. The Critical Events filter is concerned only with events of critical severity. The Fatal Events filter is concerned only with events of fatal severity, and so on.

The All Events filter passes through any events that occur on any managed system. Carefully consider the use of the supplied Simple Event Filters, as many of these are quite broad in what events will be caught, leading to potentially numerous alerts being generated. For example, there could be a critical CPU utilization. event, and a critical storage event, and if both those systems have an event action plan that filters for critical events, then the defined action would occur for both.

- ▶ Duplication event: Allow for duplicate events to be ignored, in addition to the options given by the Simple Event Filters.

- ▶ Exclusion event: Allow the exclusion of selected event types, in addition to the options given by the Simple Event Filters
- ▶ Threshold event: Allow the selection of an interval or count threshold that must be met, in addition to the options given by the Simple Event Filters.

As shown in Figure 2-31 the Actions list is quite impressive. You can see they include the actions available when creating the event action plan with the wizard but the list also shows many more. Note the new with 5.10 there is TEC (Tivoli Enterprise Console) event to a TEC console.

Event Action Plan Association

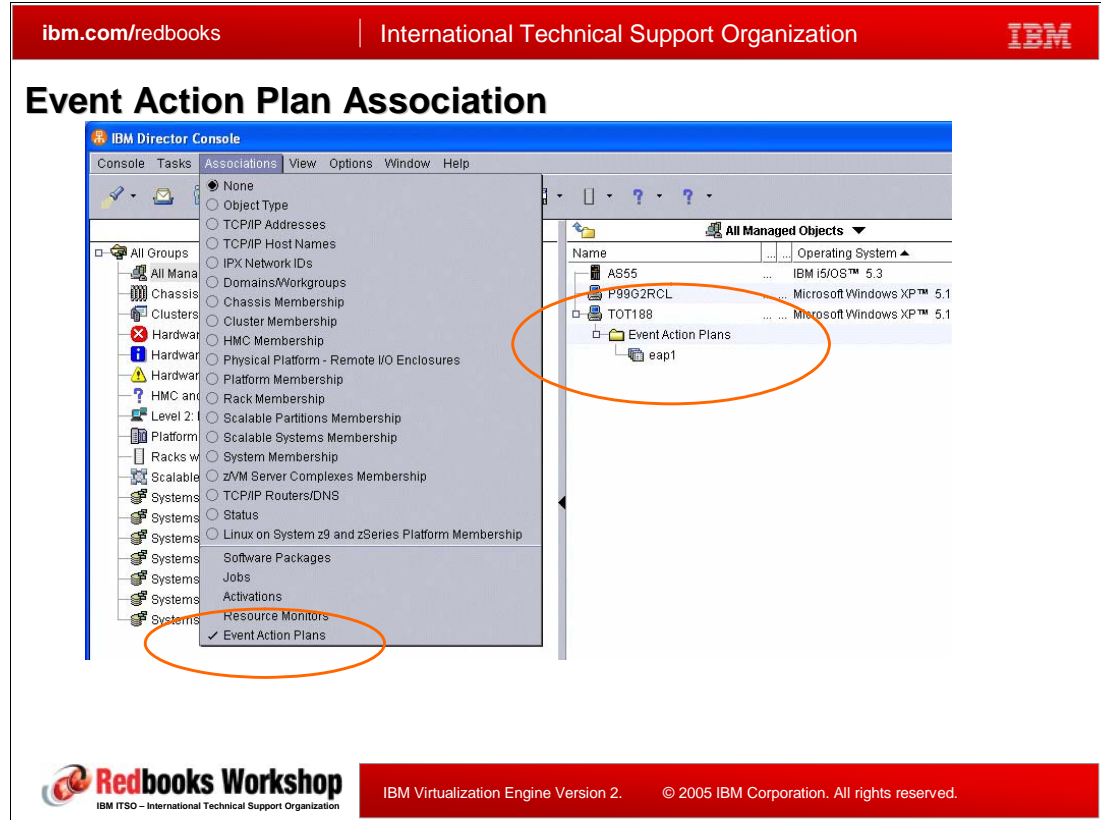


Figure 2-32 Event Action Plan Association

Note: Similar to resource monitors you must associate an EAP with a managed object. After you have created an EAP you must drag it to a managed object or group to associate Event Action Plan with a certain set of machines. This step “activates” your EAP.

You can see established association in IBM Director Console under “All managed objects” like shown in Figure 2-32 on page 93, after you select Event Action Plans under Associations

Event Action Plan up and running

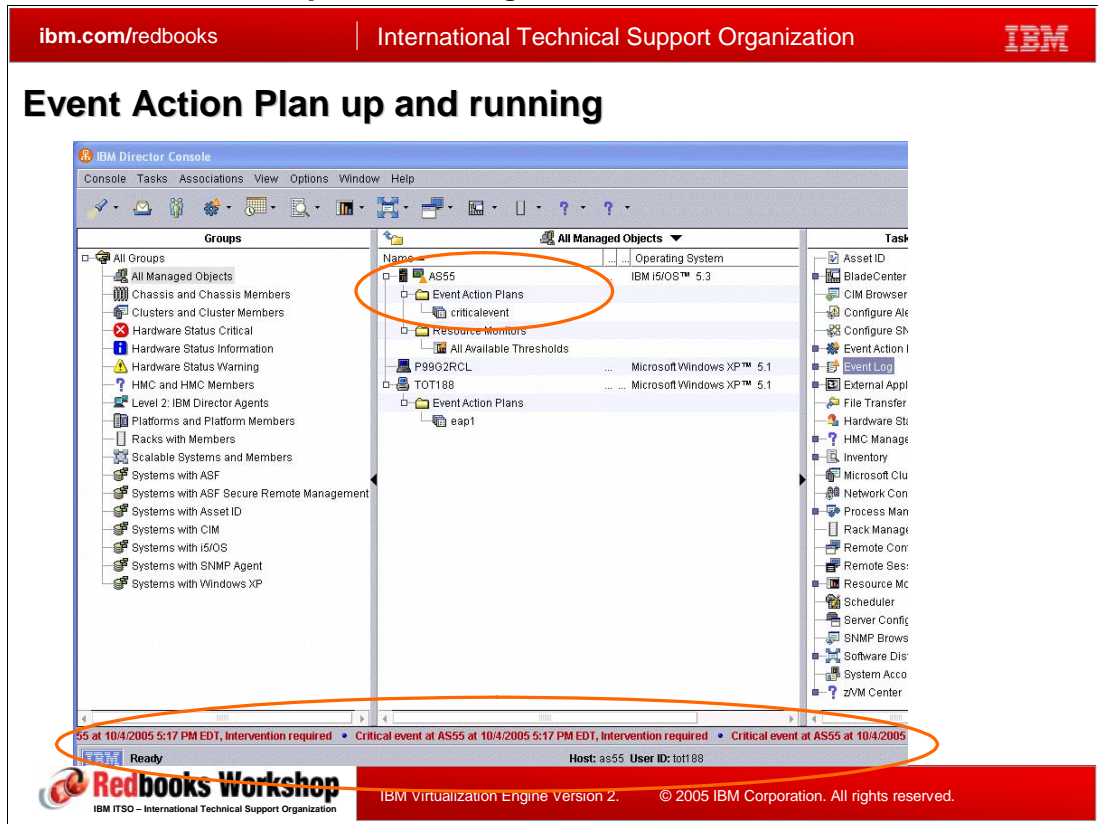


Figure 2-33 Event Action Plan up and running

An example for an activated Event Action Plan is shown in Figure 2-33. Our action plan is to display the event occurrence in the console’s “ticker tape area” at the bottom of the window.

The ticker pops up, if a critical event is detected at system AS55. In our example a critical event is CPU utilization higher than 70%. We defined this in “Monitors and Hardware Status cross platform functions” on page 79.

Event Action Plan Event Log example


ibm.com/redbooks
International Technical Support Organization
IBM

Default Event Log = Log All Events

Event Log
[Min] [Max] [Close]

File Edit View Options Help

Events (15) - Last 24 Hours					
Date	Time	Event Text	System ...	Severity	Category
10/4/2005	4:50:28 PM	User 'TOT188' (TOT188) logged on server from 's...	AS55	Harmless	Alert
10/4/2005	4:35:29 PM	User 'TOT188' (TOT188) logged on server from 's...	AS55	Harmless	Alert
10/4/2005	4:34:48 PM	User 'TOT188' (TOT188) logged off server from 'si...	AS55	Harmless	Alert
10/4/2005	4:27:59 PM	User 'TOT188' (TOT188) logged on server from 's...	AS55	Harmless	Alert
10/4/2005	4:27:35 PM	User 'TOT188' (TOT188) logged off server from 'si...	AS55	Harmless	Alert
10/4/2005	4:25:00 PM	Monitor 'AS55CPUThres' Informational: 'CPU Utili...	AS55	Harmless	Resolution
10/4/2005	3:28:45 PM	User 'TOT188' (TOT188) logged on server from 's...	AS55	Harmless	Alert
10/4/2005	3:20:23 PM	User 'TOT188' (TOT188) logged off server from 'si...	AS55	Harmless	Alert
10/4/2005	3:08:00 PM	Monitor 'AS55CPUThres' High Error: 'CPU Utilizat...	AS55	Critical	Alert
10/4/2005	11:27:34 AM	System 'AS55' is online	AS55	Harmless	Resolution
10/4/2005	11:15:20 AM	System 'AS55' is offline	AS55	Harmless	Alert
10/4/2005	10:59:46 AM	System 'AS55' is online	AS55	Harmless	Resolution
10/4/2005	10:59:34 AM	User 'TOT188' (TOT188) logged on server from 's...	AS55	Harmless	Alert
10/4/2005	10:44:54 AM	System 'AS55' is offline	AS55	Harmless	Alert
10/4/2005	8:14:08 AM	User 'JCOOK' (JCOOK) logged on server from '9...	AS55	Harmless	Alert



IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 2-34 Event Log examples

In our simple example, we selected to display all logged events as shown in Figure 2-34. The Log All Events event action plan has the following characteristics:

- ▶ It uses the event filter named All Events, a simple event filter that processes all events from all managed objects.
- ▶ It performs the action Add to the Event Log, a standard event action that adds an entry to the IBM Director Server event log.

There are several ways to get to the event log window. One way to get this from the IBM Director console is to select **Tasks** → **Event Log** → **Event Log** → **Open**.



Resource Dependency Service

Resource Dependency Service provides the connection of business processes to resources within the IT infrastructure by automatically discovering both IBM and non-IBM resources. The resources are then grouped together to create a topology. A customer may then use this topology to identify resources in the infrastructure, discover side effects of problem resources or operator actions, as well as plan for disaster recovery.

As part of the process of building the topology, these resources, their dependencies and their interactions need to be expressed in a consistent and standardized way. The standardization allows different components from different sources to be used according to a uniform type of description. Standardization allows different vendors to develop implementations that exploit the same interfaces; thus solutions can be build with components from different sources. Resource Dependency Service creates an interface to the resources which allows them to be accessed in this standardized way.

Resource Dependency Service (Resource Dependency Service) provides different tools that acquire the characteristics of infrastructure resources and, using these characteristics, it will be able to present the resources in a uniform and standardized way, called the *Resource Model*.

Resource Dependency Service is a new component of the IBM Virtualization Engine Version 2. This chapter:

- ▶ Defines what is a manageable resource
- ▶ Describes the Resource Dependency Service available tools
- ▶ Describes the Resource Model.

3.1 Modeling the I/T Infrastructure with virtual resources

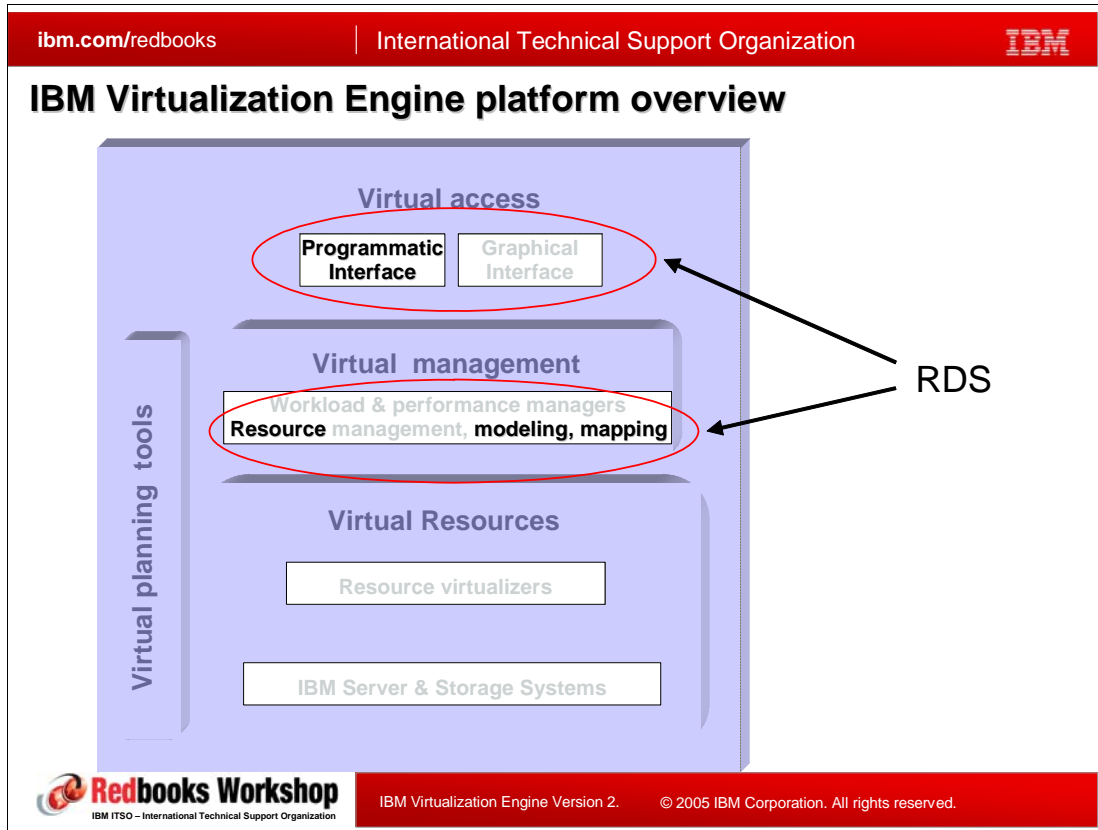


Figure 3-1 IBM Virtualization Engine platform overview

The traditional approach to managing an organization's I/T infrastructure consists of using a set of disparate methodologies and tool sets for various heterogeneous and disjointed proprietary systems. Void of unifying shared standards, models, and frameworks, the I/T infrastructure has grown very complex and its management has become very resource-intensive and error prone.

The complexity inherent in the typical organization's I/T infrastructure has significantly hindered the implementation and automation of many essential services to simplify its management and use. These services include:

- ▶ Workload management
- ▶ Business resilience management
- ▶ Infrastructure orchestration and provisioning
- ▶ Resource use optimization
- ▶ Heterogeneous resource inter operability
- ▶ Consolidated and comprehensive resource inventory, monitoring, access, and management
- ▶ Problem management
- ▶ Asset management
- ▶ Change and configuration management

In order to facilitate the implementation of advanced services such as those listed above, organizations need to build two fundamental enabling pillars:

- ▶ The first one is to create a mechanism by which the actual I/T infrastructure can be accurately documented and viewed. Because of the sheer complexity and size of most I/T environments, such mechanism must facilitate the automatic discovery and documentation of all relevant elements of the infrastructure (e.g. storage, servers, operating systems, networks, applications).
- ▶ The second fundamental building block that needs to be provided is the creation of a unifying abstract representation of the I/T infrastructure so it can be more easily understood, accessed, and managed.

The overall goal for the implementation of Resource Dependency Service is to create an environment where the infrastructure elements are abstracted into virtualized *resources*, which are managed, and accessed in virtual form. Resource Dependency Service does provide some essential elements of this virtualized environment; it provides:

- ▶ Facilities (tools) for modeling and mapping of the infrastructure (infrastructure model).
- ▶ A software framework to allow for standards-based programmatic access to the infrastructure model.

From a user's perspective, Resource Dependency Service provides technology support to facilitate the following activities:

- ▶ Automatic discovery of IBM and non-IBM I/T infrastructure resources
- ▶ Definition of identity properties for a manageable resource
- ▶ Identification of resource dependencies and relationships
- ▶ Logical grouping of resources to reflect business objectives
- ▶ Definition and creation of I/T infrastructure topologies that can be managed and accessed programmatically
- ▶ Dynamic extension of the overall interface and behavior of the manageable resources
- ▶ Leveraging of industry standards related to management function and pervasive manageable resource models
- ▶ Use of an abstract model to represent resources, types, and relationships for the I/T infrastructure
- ▶ Use of a standards-based software framework to access and manage the abstract infrastructure model in a coherent and consistent way

The listed technical capabilities enabled by Resource Dependency Service create value for organizations in three tangible ways:

1. They facilitate the assessment of the organization's current I/T environment for exploitable resources and the relationships between themselves and to business functions. Resource Dependency Service helps improve the accuracy of inventory records for the I/T infrastructure.

This infrastructure information can be used to estimate the impact of planned or unplanned resource outages of various I/T resources on business operations.

2. They create a connection between the organization's current I/T environment and a more flexible and standard service oriented infrastructure that can be used as a foundation for a flexible and efficient business.

This is done by providing access to the I/T resources and their management via technology standards such as Web services interfaces, Web Services Resource

Framework (WS-RF), Managing Using Web Services (MUWS), and others (see 3.2, “A manageable resource” on page 101).

An service oriented infrastructure enables and supports a flexible organization whose business processes can be more easily changed to reflect business environment changes. Moreover, a service oriented infrastructure leverages the use of infrastructure elements from heterogeneous vendors via the adoption and use of the mentioned industry standards.

3. They provide essential building blocks that readily enable the organization to take advantage of other advanced technology services that can exploit the service oriented infrastructure. These include automated workload management, enhanced business resilience, resource optimization, and automated asset provisioning.

Resource Dependency Service provides an essential building blocks for these management disciplines and services. For example, organizations could build business function level granularity for recovery, with differentiated levels of service for workloads, based on business impact of resource outages. The automated provisioning of infrastructure resources can be made more simplified and efficient by removing the complexities of vendor proprietary technologies and encapsulating them away from the business logic.

The Virtualization Engine platform propose a view where I/T resources are virtualized by resource virtualizers. These resource virtualizers are implemented by IBM technologies and offerings. The resource virtualizers create logical (virtual) resources and these become part of the I/T infrastructure along with the physical resources. The virtual and physical resources are modeled and mapped into a standards-based model of the I/T infrastructure through the Resource Dependency Service component.

IBM plans to make all the resources in the standardized model accessible to any client application via a standardized set of programmatic interfaces, that can be built based on the resource model, provided by Resource Dependency Service¹. In this release these interfaces are not externalized.

¹ IBM plans are subject to change without notice and this information represents goals and objectives only. Such information is not intended as a definitive statement of a commitment to specific levels of function or delivery schedules.

3.2 A manageable resource

Any component which has a need to be managed must be a manageable resource. A manageable resource is represented in a WS-RF compliant component that expresses the manageability of the resource.

A manageable resource is a resource that has a management interface with operations that control the state and the function of the resource. Managed resources are represented in Web services; managed resources are managed through a Web service and the exposed state of the resource is represented in a WSDL interface; manageable resources are made accessible through Web service communication protocols.

Note that in this release, the manageable resources created by Resource Dependency Service cannot be used to control state and function; the manageable resources created by Resource Dependency Service in this release are mainly for visualization.

3.2.1 Anatomy of a manageable resource

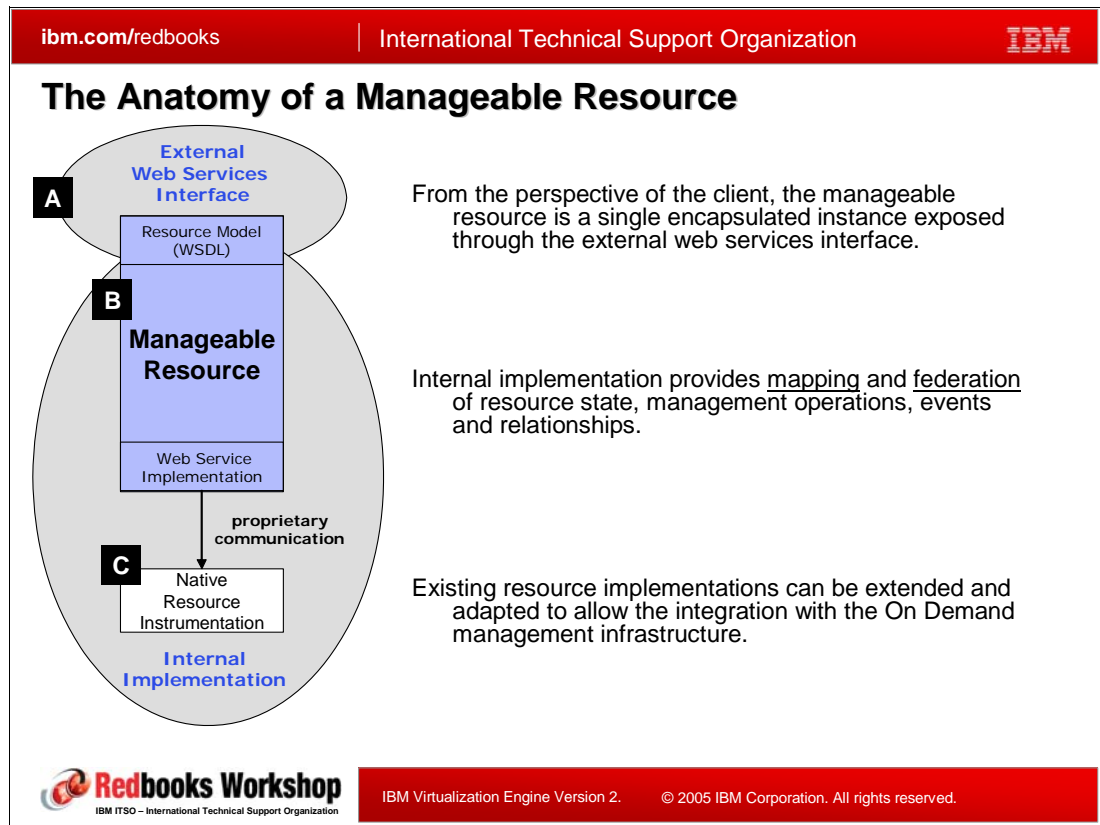


Figure 3-2 The Anatomy of a Manageable Resource

Figure 3-2 shows how a *native* I/T infrastructure element such as a server or a storage device is wrapped with code using the *adapter* design pattern. This adaptation federates the special (and usually proprietary) access to the native element and externalizes services (via WSDL) for its standardized access. The internal implementation of the element is not exposed externally to the client applications. Client applications use the externalized Web services interface. This essentially creates what we call a modelled *Manageable Resource* (MR).

Figure 3-2 represents the basis of the manageable resource:

- ▶ **A** represents the external interface of the resource, through which a resource can be used by an external user. This interface expose the resource model externally through Web services, meaning the interface is deployed in an operating environment, in this case WebSphere. Any user with the Web services knowledge is able to reach the resource.
- ▶ **B** implements the framework which can be defined as a set of available properties and functions defined with standards that can be used by the web services. The framework will map the native interface to a standardized model.
- ▶ **C** represents the native resource which is not modified by the model. The native resource itself will still be targeted internally by using its own usual interface.

Resource Dependency Service as part of the Virtualization Engine, is built upon industry standards and other technologies and trends, which are likely to become standards. The basic set of standards is described in Section 1.4, “Standards: from OGIS to WSDM” on page 13. These standards include Web services, WS-RF, WSDL and WSDM, which are developed by groups like the Organization for the Advancement of Structured Information Standards (OASIS) and the World Wide Web Consortium (W3C). Aside from these foundational technologies and standards, Resource Dependency Service uses additional ones, which were not described before in this book. We will cover them in this section using Figure 3-2 and Figure 3-3. More information on OASIS and W3C can be found at the following URLs:

<http://www.oasis-open.org/who/>
<http://www.w3.org/>

3.2.2 Exploiting standards

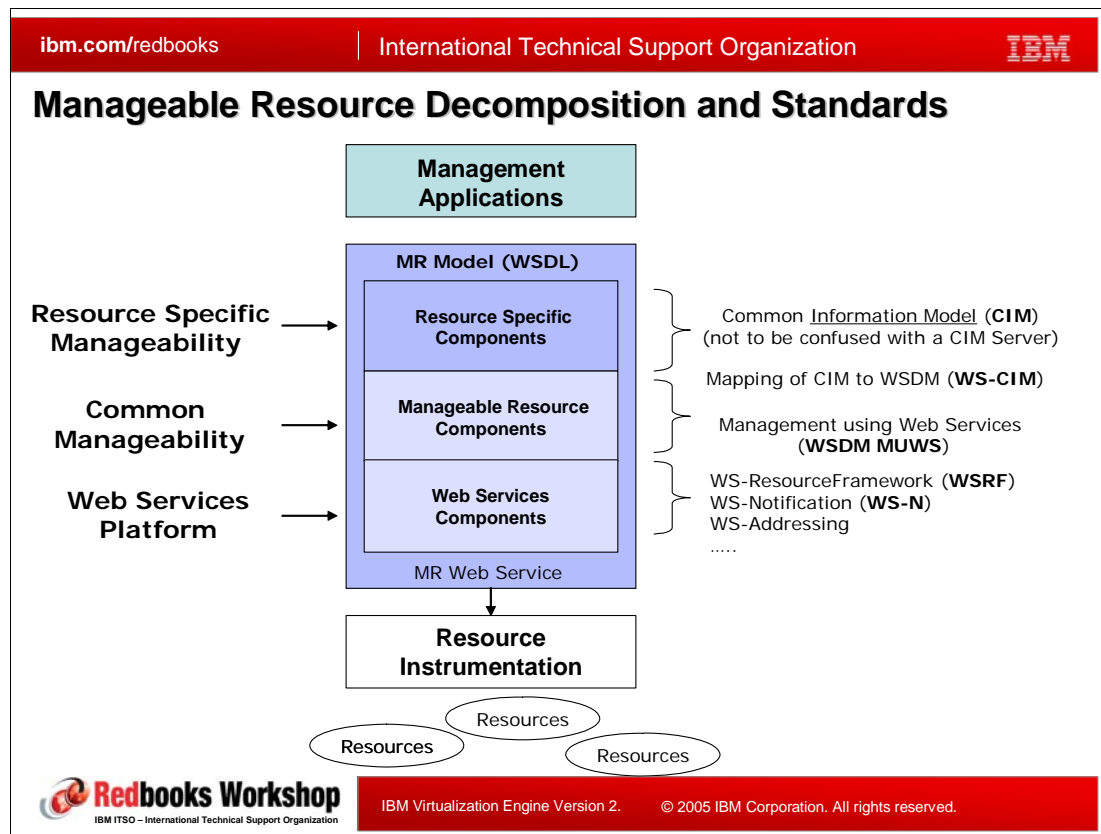


Figure 3-3 Manageable Resource Decomposition and Standards

An essential characteristic of the service oriented infrastructure is its ability to allocate, assign, monitor, and adjust the use of the resources in the infrastructure in accordance with business policies. The extent to which a resource in the system can be managed depends on its ability to represent itself through an interface. Resources types include hardware (processors, network devices, storage, etc.), software components (database managers, application server middleware, message queues, etc.), business application components, as well as data and users. This interface is called the *manageability interface*. Resources can be divided into resource providers (e.g. Server, Storage, Network and Software) and resource exploiters (e.g. EWL, TIO, etc.).

Resource Dependency Service introduces a base set of interfaces, consistent with those being standardized by the OASIS WSDM work. Manageable resources are expressed as WS-RF compliant Web services . Existing resource implementations must be extended and adapted to allow the integration with the On Demand management infrastructure in compliance with the Resource Modeling Framework.

From an implementation perspective, the resource instance may be composed of multiple components, each potentially residing in its own hosting environment

Figure 3-3 extends Figure 3-2 and illustrates how the adaptation of the native element is achieved using base standards such as WS-ResourceFramework (WS-RF), WS-Notification, WS-Addressing, and others. Once the element is adapted and exposed using Web services, its management framework is implemented using WSDM, MOWS and MUWS standards. The resulting manageable resource is mapped into a standard Common Information Model (CIM).

More on WS-RF

The role of WS-RF has been explained in Section , “WS-RF” on page 15. In particular, in regards of this chapter, WS-RF defines and standardizes the following functionality used in the resource model:

- ▶ WS-ResourceProperties, which defines the mechanism for describing and exposing state of a resources as properties expressed using XML and XML schema and defines the operations to read and query the state of the resource.
- ▶ WS-ResourceLifetime, which defines the operations for explicit and scheduled resource destruction.
- ▶ WS-ServiceGroups, which defines the grouping and the registration semantics for WS-RF Web services. It is used for resource discovery in a SOA.
- ▶ WS-BaseFaults, which defines the base fault XML schema and the rules for its use.

The Common Information Model

The Common Information Model (CIM) is a resource modeling schema built using a set of specifications (and standards) developed by the Distributed Management Task Force, Inc. (DMTF²). It provides common definitions of management information for I/T infrastructure elements, and it allows for vendor extensions. CIM's common definitions enable the exchange of semantically rich management information between systems throughout the organization's network.

The use of CIM makes it possible to model I/T infrastructure manageable resource such as physical virtual servers, storage volumes, files, and network routers as stateful services according to the WS-RF specifications. In particular, the manageable resources provide a common interface in the form of the WS-Resource service - the base class of all

² The Distributed Management Task Force, Inc. (DMTF) is the industry organization leading the development of management standards and integration technology for enterprise and Internet environments. More information about DMTF on the following Web site: <http://www.dmtf.org/home>

resource services in a virtualized I/T infrastructure environment. Each manageable resource has a *Resource Type* associated with it. Each manageable resource also has a *Resource Factory* associated with it to manage the creation and disposal of its instances. Additionally, the created manageable resource instances are registered in CIM-based repositories. One of the Resource Dependency Service's repositories is, indeed, a CIM-based repository.

CIM defines the following elements:

CIM Client:

The CIM Client is a management application and can reside anywhere in the network, because it uses HTTP to talk to CIM Object Managers and Agents.

CIM Managed Object:

A Managed Object is a hardware or software component that can be managed by a management application by using CIM.

CIM Agent:

The CIM Agent is embedded into a device and interfaces with the management application. It interprets CIM requests and responses to it.

CIM Object Manager (CIMOM):

common conceptual framework for data management that receives, validates and authenticates the CIM requests from client application. It then directs the requests to appropriate component or CIM provider.

CIM Server:

A CIM Server is the software that runs the CIMOM and the CIM provider for a set of devices. It is used when the devices do not have an embedded CIM Agent. When people use the term CIMOM they often mean the CIM Server.

Device Provider:

A CIM Provider is the element that translates CIM calls to the device specific commands, it is like a device driver. A provider is always closely linked to a CIM Object Manager.

More information on CIM can be found on the DMTF web site at the following URL:

<http://www.dmtf.org/standards/cim/>

The WS-CIM protocol

There is an ongoing work in DMTF to provide a mapping that allows CIM-modeled resources to be exposed as WSDM manageable resources: the WS-CIM protocol, which enables the DMTF's infrastructure protocols to take advantage of Web Services. WS-CIM specifies how resources modeled using CIM are exposed, described, managed and discovered via Web services. In other words, WS-CIM makes it possible to represent CIM-modeled manageable resources as WSDM-manageable resources via Web services. It describes how to represent and access the content of the CIM model using Web services, WS-RF and WSDM.

WS-CIM is leveraged in the implementation of Resource Dependency Service. For instance, when CIM-modeled resources have properties that can be accessed, WS-CIM specifications allow Resource Dependency Service to access them using WS-Properties, rather than having to use other lesser-known message protocols such as CIM-XML. More information on the WS-CIM specification can be found at the following URL:

<http://www-128.ibm.com/developerworks/webservices/library/specification/ws-wsdm/>

CIMOM and WBEM

The Storage Network Industry Association (SNIA) used CIM and created the Java *Common Information Model Object Manager* (CIMOM), which is a CIM-based resource server. CIMOM implements a CIM schema and is, essentially, a container of manageable resources. This work was later turned over to the Open Group for its management and it was used by this organization to create a new Java open source CIM-based resource server called *Pegasus* (also known as OpenPegasus). More information can be obtained on the Open Group and OpenPegasus at the following URLs:

<http://www.opengroup.org/>
<http://www.openpegasus.org/>

OpenPegasus uses another specification developed by DMTF called *Web Based Enterprise Management* (WBEM). WBEM also provides facilities for the management of systems, networks, users and applications across multiple vendor environments.

CIMOM-based resource servers can exist anywhere on an organization's I/T infrastructure. They can be implemented by IBM such as in the TotalStorage Productivity center (TPC) and by other vendors in order to expose a standard framework with which to manage I/T infrastructure devices.

CIM and WBEM together enable end-to-end multi-vendor interoperability. The OpenPegasus CIM server can be seen as an implementation of the WBEM standards provided by the Open Group.

Examples of IBM products implementation of CIM

OpenPegasus has been chosen by IBM as the strategic CIM server and it is being ported to the major IBM platforms. The CIM infrastructure is shipped with all IBM eServer Operating Systems: since i5/OS V5R3 for iSeries, since AIX 5.2F for pSeries, System p5, and iSeries, since z/OS 1.7 for zSeries and System z9.

The implementation of Resource Dependency Service has taken all of these maturing specifications into consideration. In this release, Resource Dependency Service includes a CIM-based repository. It is envisioned that, as they continue to evolve, this set of specifications will continue to shape Resource Dependency Service's implementation.

Many IT resources are already currently instrumented using the CIM schema defined by DMTF. The following list is not exhaustive and wants to provide a few examples of CIM implementation:

- ▶ The IBM Director component of the Virtualization Engine can connect to and manage resources hosted by CIMOM servers existing in an organization's environment.
- ▶ The pSeries HMC uses CIMOM to model the hardware resources of the pSeries server. It can provide information about its CIM objects to remote CIM clients.
- ▶ CIM on AIX 5L V5.2 and V5.3 includes the following:
 - An Open Source implementation of the Common Information Model Object Manager called Pegasus Version 2.3.
 - A CIM schema, Version 2.7 that defines an information model for representing system management resources.
- ▶ The CIM Agent for DS Open (API) is middleware that provides a CIM-compliant interface for applications that require it or can make use of it. The Agent code provides a CIM API into IBM TotalStorage DS8000, DS6000, and ESS
- ▶ In z/OS V1.7 a subset of z/OS resources and metrics is mapped into the CIM standard data model to allow its use for system management functions. In the release following

z/OS V1.7, IBM plans to make the CIM services suitable for applications that require high request rate processing.

- ▶ Many Tivoli products (Tivoli Monitoring, Tivoli Inventory, Tivoli Business Systems Manager, and many others) provide a CIM interface.

3.3 Resource Dependency Service overview

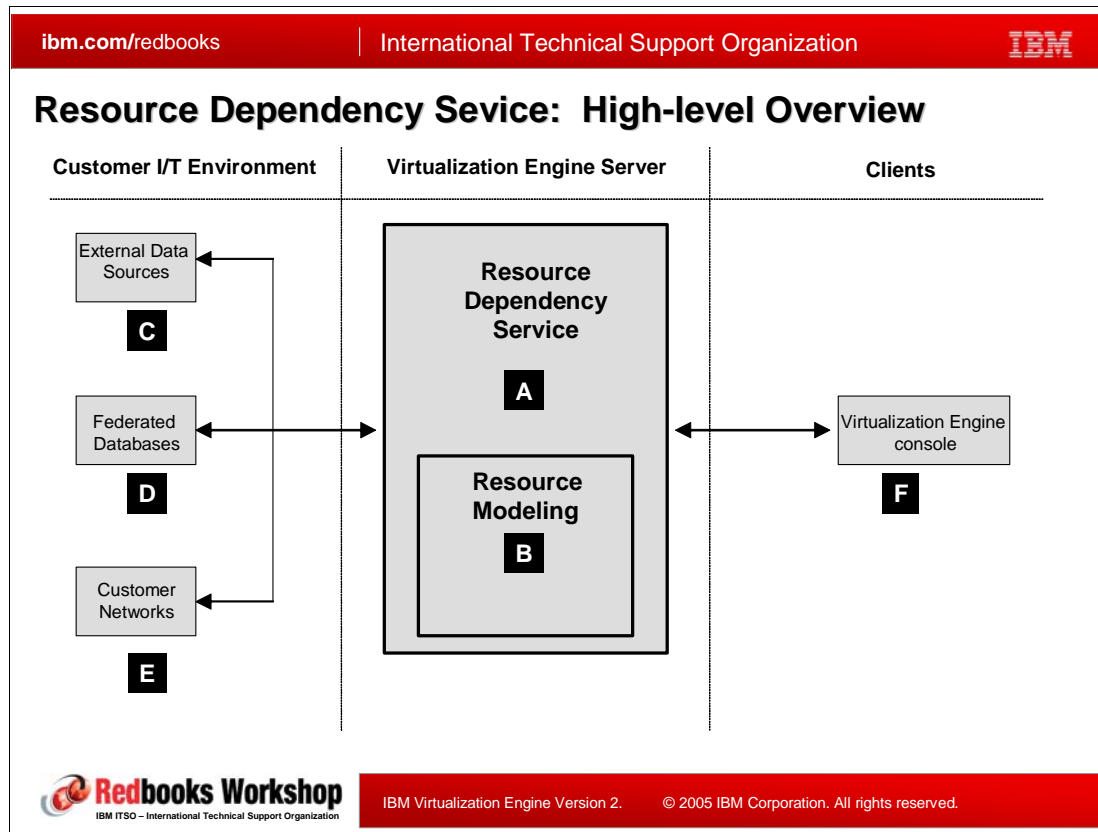


Figure 3-4 Resource Dependency Service High-level Overview

Resource Dependency Service focus is the creation of a catalog of resource instances and a representation of their relationship.

Resource Dependency Service (A) is packaged as one of the components of IBM's Virtualization Engine, utilizes an important subcomponent called Resource Modeling (B), and executes as part of the Virtualization Engine Server runtime. Resource Dependency Service provides a software framework, which exposes services and allows client applications such as the Virtualization Engine console (F) to access the virtualized I/T infrastructure in a standardized way. In turn, Resource Dependency Service scans existing customer networks (E) and detects their IP-addressable elements so they can be easily mapped as part of the standardized model. Moreover, Resource Dependency Service includes functionality that allows databases external to Resource Dependency Service (D) to be federated as part of the Resource Dependency Service repositories. Resource Dependency Service provides built-in mechanisms that access, extract, and analyze data coming from many other external sources (C) that may contain relevant I/T infrastructure information.

Essentially, Resource Dependency Service represents a set of technologies designed to discover, collect, define, and model an organization's I/T infrastructure in order to present that converged information in a persistent data store for its orchestrated access and management. The I/T infrastructure resources, their interrelationships, and their relationships to business processes are stored, accessed, and managed via a set of standards-driven mechanisms.

3.4 Resource Dependency Service architecture

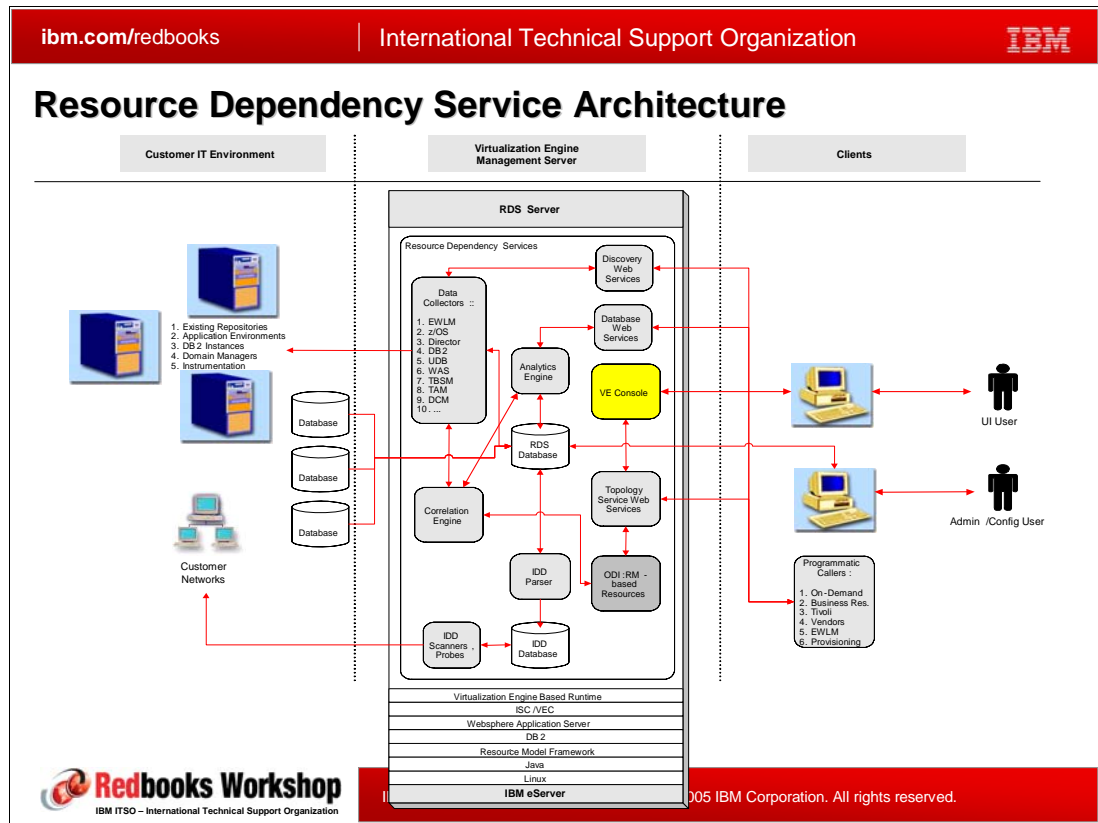


Figure 3-5 Resource Dependency Service architecture

If we drill down deeper into the Resource Dependency Service architecture, we can get a glimpse into all of the high-level components of Resource Dependency Service. While the Virtualization Engine console is shown inside the context of Resource Dependency Service, it is a separate Virtualization Engine component as discussed in Chapter 4, “The Virtualization Engine console” on page 133.

Figure 3-5 shows how the external data sources that can be mined by Resource Dependency Service include the following:

- ▶ Existing repositories such as the Tivoli Business Service Manager (TBSM) data sources, which are DB2 databases.
- ▶ Application environments such as WebSphere
- ▶ DB2 instances
- ▶ Domain Managers such as EWLM
- ▶ Instrumentation such as IBM Director.

Customer networks can be scanned by Resource Dependency Service to find IP-addressable infrastructure elements automatically. External information repositories such as relational databases used by other resource managers from IBM and other vendors will be federated to become part of the Resource Dependency Service repositories³.

³ This capability may become available after Resource Dependency Service general availability.

There are graphical user interfaces accessible via the Virtualization Engine console for human user access to those resources and for Resource Dependency Service administration. Additionally a programmatic interface through which client applications can access the Resource Dependency Service services and the WS-RF infrastructure model may become available⁴.

From an end-user perspective, here is the high-level process by which you take advantage of Resource Dependency Service:

1. Use Resource Dependency Service's Intelligent Device Detection (IDD) technology to scan and discover I/T infrastructure elements and automatically register them in Resource Dependency Service's repositories.
2. Use Resource Dependency Service data collectors and its datasource federation functionality to augment the information stored in the Resource Dependency Service repositories. External data stores can be mined to get a good base set of resources. The Resource Dependency Service Analytics/Correlation engines will make sure that each discovered resource is singly defined. Data collectors can also be used for mining extensive resource information from server platforms which may be present in the network including all the IBM @server.
3. Schedule data collectors to execute automatically on a periodic basis. This will keep the Resource Dependency Service repositories synchronized as the I/T environment changes.
4. Use Resource Dependency Service specific user interface panels of the Virtualization Engine console to define additional attributes and relationships on the stored resources, when needed.
5. Use Resource Dependency Service-specific user interface panels of the Virtualization Engine console to establish relationships between infrastructure resources and the organization's business processes.
6. Use Resource Dependency Service-specific user interface panels of the Virtualization Engine console to access topology and query services, which facilitate the analysis and management of the organizations's I/T infrastructure resources and business processes.

Figure 3-6 shows that the ultimate goal for the use of Resource Dependency Service is to create an environment where infrastructure elements such as servers, storage, and networks, as well as their corresponding instrumentation, can be mapped into manageable resources, which are subsequently exposed using Web services.

- ▶ *Autonomic Managers* can be built to have access to these manageable resources and their services via the ESB in order to guarantee certain levels of business resilience governed by established SLAs and other business policies. Autonomic managers are applications built to use autonomic computing features based on the implementation of MAPE (monitor, analyze, plan, execute) logic. For more information on autonomic managers, please refer to the IBM Autonomic Computing Kit and its documentation, available at the following Web site:

<http://www-128.ibm.com/developerworks/autonomic/books/fpu1mst.htm>.

- ▶ Furthermore, *Workload Managers* such as EWLM could also access the manageable resources as needed in order to decide how to properly allocate workloads to use them.
- ▶ *Resource Managers* such as the Tivoli Intelligent Orchestrator (TIO) can have access to the manageable resources so it can make decisions regarding their allocation, commissioning, and decommissioning.

⁴ This capability is not part of this release.

The Virtualization Engine console can be used to view and manage topologies of the I/T infrastructure dynamically. The Virtualization Engine console user interface can also be used to establish and manage the relationships between the infrastructure resources and the organization's business processes.

All communications between external applications such as autonomic managers, workload managers or resource managers, and the services exposed by the MRs is done via Web services interfaces. The fabric for these Web services connections is provided by IBM's WebSphere Application Server via its support for the Enterprise Service Bus (ESB). WebSphere Application server provides the support framework and the implementation of the Web services infrastructure.

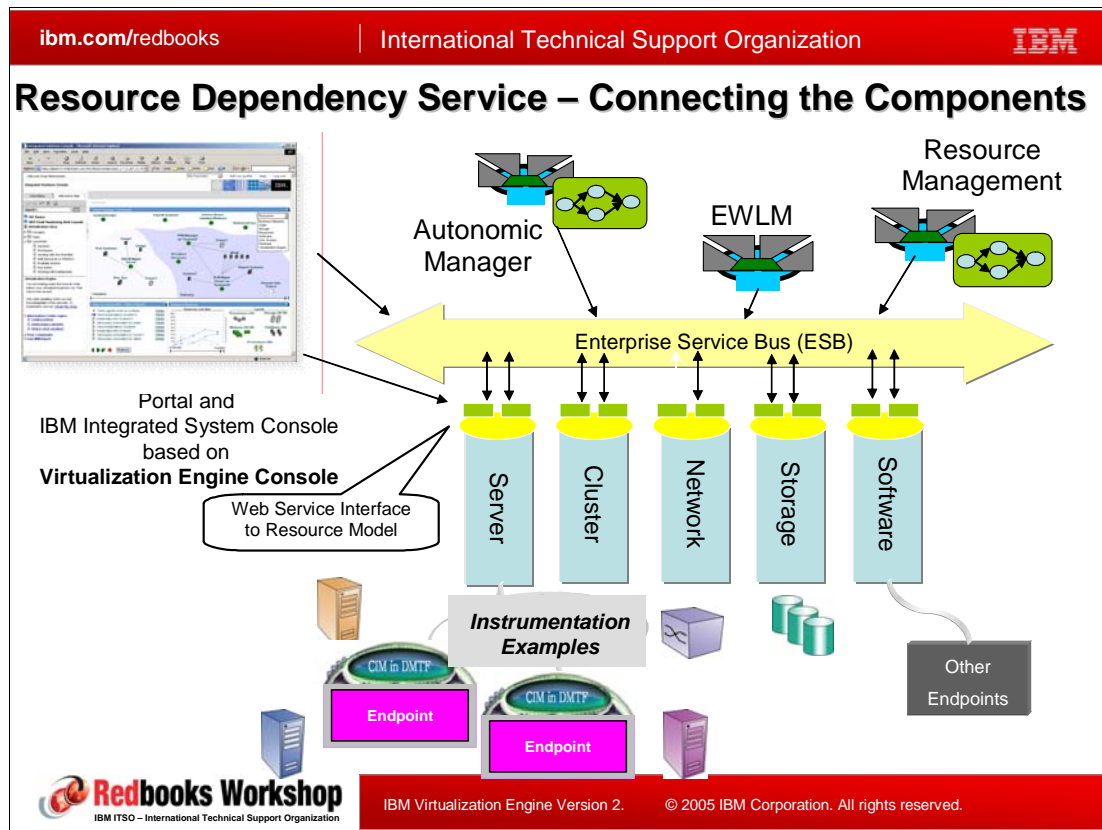


Figure 3-6 Resource Dependency Service - Connecting The Components

Each component of Resource Dependency Service will be described in more detail in the next sections of this chapter.

3.5 Intelligent Device Discovery function

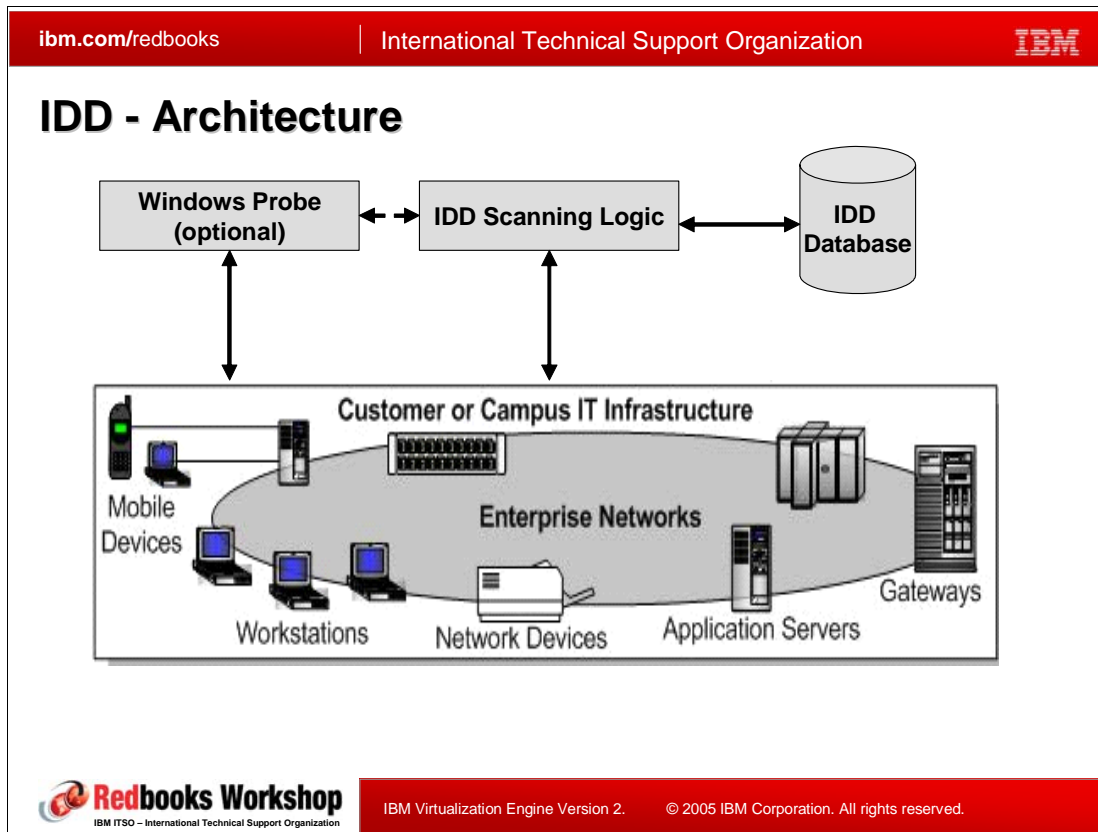


Figure 3-7 IDD Architecture

Intelligent Device Discovery (IDD) is a network-based IT asset discovery and categorization tool. It combines various device discovery, network and security scanning programs with enhanced data collection, mining techniques, and distributed automation to form a single application.

IDD is the technology component used by Resource Dependency Service to discover and generally classify I/T infrastructure elements. This discovery is done within parts of the organization's network (as configured by the user) using low-intrusion methods on the scanned devices. The network areas to be scanned can be specified by the user by supplying which IP-address ranges to scan.

Scanning levels

IDD's resource scanning process can operate at different levels of increasingly detailed discovery. The scanning levels are as follows:

Level 1 General information about any IP-addressable (TCP/IP) active device is discovered. This is done with absolutely no software installation or security privileges needed on the scanned devices.

Level 1 Enhanced (Level 1+)

More specific information is obtained for Microsoft Windows devices from the Windows Registry. Attempts are also made to obtain additional information from physical computer servers that operate under the Unix operating system. This level of scanning does not require any software installation on the scanned

devices, but does need administration credentials (e.g. userids, passwords, community strings, etc.) for the relevant scanned resources.

- Level 2** This level of scanning captures detailed HW/SW Configuration information from scanned Microsoft Windows Servers. In order to do this, a special probe agent is required to be installed on a Microsoft Windows server within the network to be scanned. No level 2 scanning is done with Resource Dependency Service.

Architecture

Figure 3-7 shows a high-level architecture of the IDD discovery mechanism. As was described before, IDD includes logic to perform scans of the existing I/T infrastructure. Additionally, IDD includes its own database to store the discovered information. As an option, users can take advantage of the external Windows auxiliary server to allow for deeper levels of information discovery.

IDD can find many types of resources such as servers, network devices (e.g. load balancers, routers, switches, etc.), file systems, programs, and physical disks. It also finds relationships between them, if they exist. The scanning is done using a variety of open-source and proprietary techniques including IP ping sweeps (not the same as broadcasts), Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) ports scans, and remote Windows SMB/Registry/WMI fingerprinting.

IP ping sweeps are targeted scans of pre-selected Internet Protocol (IP) ports or ranges of IP ports to find network-attached devices. The TCP and UDP protocols are subsets of the Internet's ubiquitous IP protocol. TCP and UDP ports represent internet addresses used by many device applications on the network that can be discovered and investigated by Resource Dependency Service. Server Message Block (SMB) is a protocol for sharing files, printers, serial ports, and communications abstractions such as named pipes and mail slots between computers. Microsoft uses this protocol to provide file and printer sharing services in various versions of Windows. Resource Dependency Service also uses the Windows Registry to mine resource information since it is a database of nearly all the settings for Windows servers and their installed applications.

Windows Management Instrumentation (WMI) is the Microsoft implementation of Web-Based Enterprise Management (WBEM). WMI provides integrated support for the CIM schema and includes a CIM-compliant object repository that can be mined by Resource Dependency Service for resource information.

As an example, IDD can find the relationship between a server and an ethernet switch. In order to do this, IDD uses the Simple Network Management Protocol (SNMP) to read the switch's Management Information Bases (MIB). This shows that a given Media Access Control (MAC) address is connected to a given switch port. A match can be made between the MAC address and its related IP address using router information such as the Address Resolution Protocol (ARP) tables. The connection can be established by saying that a specific server has specific IP and MAC addresses, and is connected to a specific switch using a specific port.

The following standards are used by IDD:

- ▶ ICMP and TCP for the ipscan
- ▶ TCP/UDP packets for the hostscans
- ▶ SNMP/HTTP/WMI for the follow-ons services

The following is a list of resources that IDD will accurately find and represent if they are operational, reachable, and the relevant credentials are provided for access, when required:

- ▶ zSeries (including z/OS, Linux OS, DB/2, and CICS®)

- ▶ iSeries (including i5/OS, Linux OS, AIX OS, and DB/2)
- ▶ xSeries (including Windows OS, Linux OS, and applications)
- ▶ pSeries (including AIX OS, Linux OS, and applications)
- ▶ Blade Center, if Director is managing it, and the Director data collector is configured
- ▶ HP-UX
- ▶ Sun (including Solaris and Linux)

Note: Resource scanning and discovery services may also be a feature in other IBM products such as the IBM Director and the TotalStorage Productivity Center (TPC). However, such functionality offered by those products is more specialized and limited than IDD's. For instance, TPC's scanning is targeted to and specialized for storage devices. IBM Director's scanning does not detect certain resources such as applications running on the found host servers.

More detailed information on the technologies and operation of the IDD can be found at the following URL:

<http://www.zurich.ibm.com/csc/ibi/idd.html>

3.6 Data Collectors function

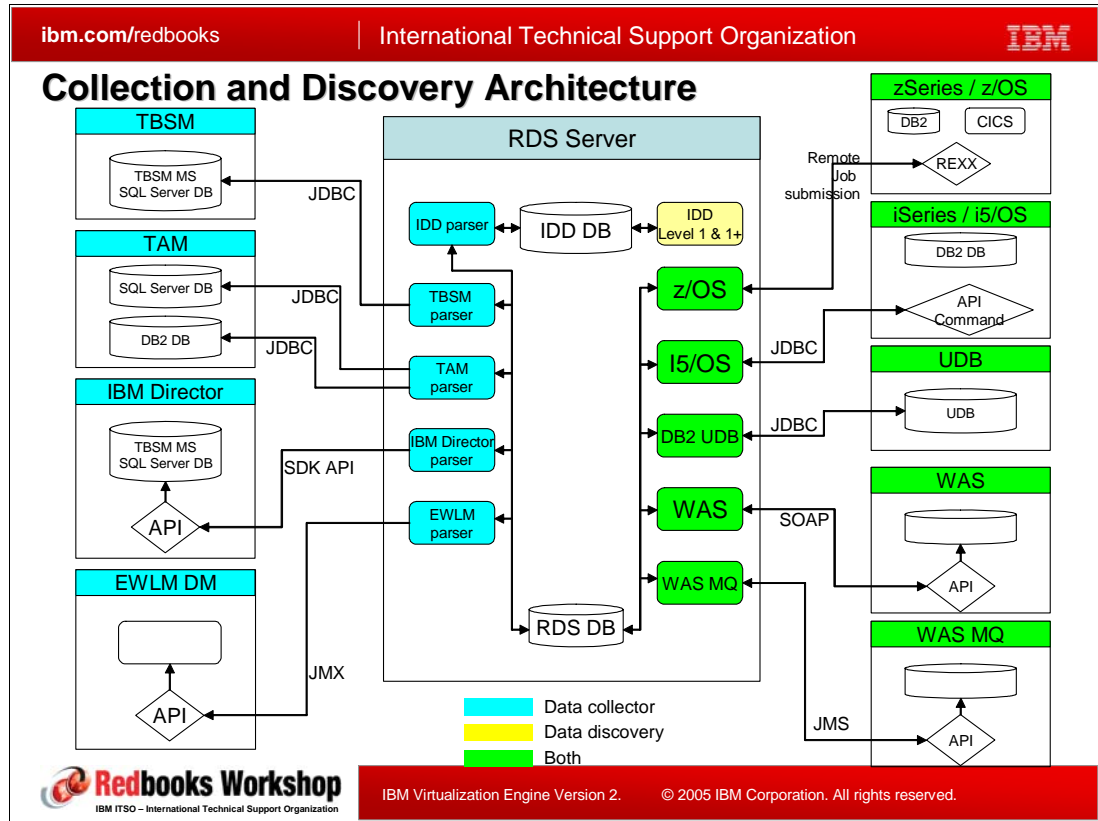


Figure 3-8 Collection and Discovery Architecture

Resource Dependency Service can get information from other products repository databases which will complete the information it discovered with the IDD; all the information will then be analyzed and correlated.

In this release, the Resource Dependency Service repositories contain static information, which is not automatically refreshed via an event mechanism by external or internal sources. Collectors included into Resource Dependency Service can be scheduled to run periodically to keep Resource Dependency Service repository information up to date.

Figure 3-8 shows how the data collection and discovery tasks are related and how they are implemented in Resource Dependency Service. For each type of external data source, there is a specialized data collector (sometimes called as parser). In some cases, the collection of data is assisted and complemented by data discovery.

In general, the functionality of the Resource Dependency Service Data Collectors follows this logic flow:

1. Connect with the existing remote target. These targets can be:
 - External repositories. These are typically accessible via database connectivity technologies such as JDBC™.
 - Other applications. Access to these is obtained via corresponding APIs.
 - External domain managers such as EWLM. Access is typically obtained via APIs.

- Specific platform or OS implementations. Access is obtained via OS-specific supported interfaces such as remote job submissions.
2. Discover and acquire data specific to Resource Dependency Service functionality
 3. Analyze, transform, and correlate the data into a resource model
 4. Store appropriate information in the Resource Dependency Service repositories.
 5. As resources are stored, the correlation engine will check to see if there is sufficient data collected about each resource. If there is, it will create the resulting manageable resource in the Resource Modeling Type Registry (more on this in section “Resource Modeling Type Registry” on page 121).

Figure 3-8 shows that IDD has its own database, which is DB2-based. There is also a parser associated with IDD, which will collect relevant data from that IDD database and transfer it to another database, the Resource Dependency Service Datastore, as required.

As can be inferred, the Resource Dependency Service data collectors use a variety of methods and techniques for inspection, depending on the type of data source. It is important to note that none of the listed products (and data sources), which have data collectors associated with them are prerequisites for the use of Resource Dependency Service.


Data collectors exist for the following sources:


- ▶ DB2 UDB; to gather data, the DB2 UDB data collector uses a JDBC connection to the DB2 UDB database. The DB2 UDB database can physically reside remotely within the organization’s I/T environment. Information like database properties and data container properties can be retrieved by this collector.
- ▶ Data Center Model (DCM); information like VLAN information or subnet IP information can be retrieved by this collector.
- ▶ IBM Director; the IBM Director data collector utilizes the IBM Director SDK to remotely communicate with the IBM Director server, and indirectly with the Director database. Since the IBM Director SDK encapsulates the database access within its APIs, a remote connection to the database does not need to be established by the data collector itself. Information like hardware and operating system properties can be retrieved by this collector.
- ▶ Intelligent Device Discovery (IDD); the IDD data collector uses a JDBC connection to the IDD and Resource Dependency Service repositories. Both of these repositories are DB2 databases. Information about hardware, operating system, network devices, storage devices can be retrieved by this collector. This collector runs automatically when a user runs an IDD campaign to pull information out of the IDD database.
- ▶ Enterprise Workload Manager (EWLM); the EWLM data collector employs JMX calls to the EWLM domain management server to gather data. Information about the domain manager can be retrieved by this collector.
- ▶ Tivoli Business Service Manager (TBSM); the TBSM data collector employs a JDBC connection to the TBSM Microsoft SQL Server database. The SQL Server database can physically reside remotely within the organization’s I/T environment. Information about hardware, operating system, database, network device can be retrieved by this collector.
- ▶ Tivoli Configuration Manager (TCM); the TCM data collector employs a JDBC connection to the TCM RDBMS server. The TCM RDBMS server can physically reside remotely within the organization’s I/T environment. Information about the operating system and the application can be retrieved by this collector.
- ▶ WebSphere application server; the WebSphere Application server data collector uses calls into WebSphere’s wsadmin interface and its related scripts. A file is created by

WebSphere Application server with data and it is parsed to obtain the gathered data. Information about the hosting container can be retrieved by this collector.

- ▶ WebSphere MQ; the WebSphere MQ data collector uses the WebSphere MQ JDK classes for Java to remotely extract resource (e.g., Queue Managers, queues, etc.) information from the Message Queuing Administration Interface (MQAI). Information about the application and the hosting container can be retrieved by this collector.
- ▶ The iSeries OS/400 data collector employs the IBM Toolbox for Java facility on iSeries to communicate with it and gather data. The IBM Toolbox for Java must be installed and configured on the Resource Dependency Service Server. The IBM Toolbox for Java package provides a JDBC connection to the iSeries DB2 database so that it can be read using the JDBC 2.0 specifications. Information about the operating system, the network interface and the data container can be retrieved by this collector.
- ▶ The zSeries data collector uses FTP to submit jobs on the zSeries host machine. The submitted jobs, in turn, run REXX scripts. Consequently, the zSeries host must have FTP and REXX installed and running to support the data collector. If the zSeries host machine has DB2 installed and running, DB2 data will be collected. If the zSeries host machine has CICS installed and running with CICSplex System Manager, and has the REXX run-time interface installed, CICS data will be collected. If any the following technologies are installed, configured, and running on the zSeries host: a) Batch Job JCL, b) IMS , c) CICS and WebSphere Studio Asset Analyzer with Inventory MVS Assets performed, then information about them will be collected. If the IMS Transaction Manager or Database Manager is installed, then the DBRC API and IMS commands are used to discover all IMS region types (TM/DB, DBCTL, DCCTL). Information about the operating system, database, storage unit, hosting container can be retrieved by this collector.

Figure 3-9 documents the different type and relevancy of the general information gathered by the Resource Dependency Service data collectors.

ibm.com/redbooks International Technical Support Organization	
	
Discovered/Collected Resource Types	
Applications	An application is a program that runs within a hosting container or an operating system instance.
Data Containers	Data containers are like file systems, containing collections of files and attributes associated with those files.
Database Instances	A database instance is a collection of interrelated or independent data items stored together to serve one or more applications.
Hosting Containers	Hosting containers are programs or applications that host other applications (e.g. WAS).
Network Devices	Network devices are special purpose, dedicated computers like routers and load balancers that help manage network traffic.
Network Interfaces	A network interface (also known as a network interface controller or NIC) provides the interface control between a system and an external high-speed network.
Network Subnets	A subnet is an interconnected, but independent segment of an entire network that is identified by its Internet Protocol (IP) address.
Operating Systems	An operating system is a collection of system programs that control the overall operation of a computer system.
Services	A service is a process or sub-process used to automate common tasks such as verifying security credentials or calibrating system clocks.
Storage Volumes	A storage volume represents a medium used for storage of data on a computer. A storage volume can be an entire physical hard disk, or portions of a hard disk.
Systems	A system represents physical computers or virtual computers, such as logical or physical partitions.


Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 3-9 Discovered/Collected Resource Types

As information is obtained from the I/T environment by Resource Dependency Service using the data collectors, sometimes assisted by the automatic data discovery, a wide variety of resource types are encountered and mapped into the Resource Dependency Service repositories. These resource types vary from physical or virtual (logical) computer servers called *Systems* to *Hosting Containers* like WebSphere Application server and *Applications* that run within them.

As was explained before, all of the collected and discovered resources will be stored in the Resource Dependency Service repository. Additionally, if a resource has enough attributes to warrant induction into the Resource Modeling Type Registry, it is stored there as well along with its relationships to other resources. More will be explained about these resources as they become part of the Resource Dependency Service repositories in the Section 3.7, “Resource Dependency Service repositories” on page 118.

3.7 Resource Dependency Service repositories

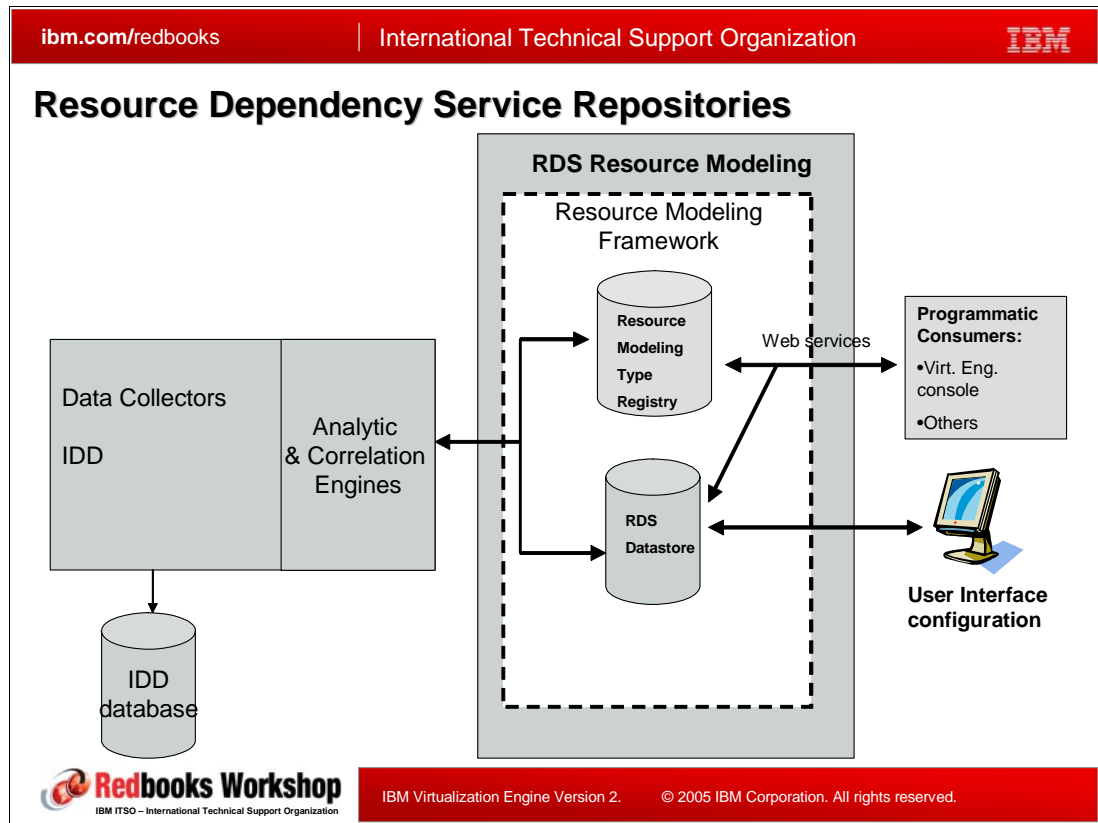


Figure 3-10 Resource Dependency Service Repositories

All information detected and collected by Resource Dependency Service for the I/T infrastructure needs to be stored. Resource Dependency Service has different repositories:

- ▶ The first one is a DB2 database and is called the *Resource Dependency Service Datastore*. The Resource Dependency Service Datastore is used to house all complete or incomplete data gathered for all the resources created by IDD detection, via the data collectors.

During the process of discovery and collection for resource information, Resource Dependency Service stores all this information in this Resource Dependency Service Datastore. At some point in time, there will be enough information gathered so that all of the attributes for its resource type as defined using CIM specifications can be defined. At that time, Resource Dependency Service creates a new manageable resource and stores it in the Resource Modeling Type Registry database.

- ▶ The *Resource Modeling Type Registry (RMTR)* is built using the CIM specifications and other related industry standards as explained in “A manageable resource” on page 101. Manual input by the user, via the Virtualization Engine console can be done; they are registered into this repository. It is important to note that, unless a resource has been stored in the RMTR, it will not be displayed as part of infrastructure topologies viewed using the Virtualization Engine console. It is possible, however, to drill down and see these incomplete resources using other panels of the Virtualization Engine console interface (refer to the Section 4.7, “The topology view” on page 162 for more information on this topic).
- ▶ Note that IDD has its own database. As it scans a network, it puts data into that database. Then the Resource Dependency Service IDD data collector pulls information out of that

database into the Resource Dependency Service database, where it is correlated with data from other data collectors before an MR is created. There is no prereq to install the IDD database and the Resource Dependency Service repositories on the same server.

Over time, the RMTR is destined to become the authoritative source for resource data. The Resource Dependency Service Datastore will be used only for the correlation and federation of data collected and discovered by the various collectors and mining tools in Resource Dependency Service. The Resource Dependency Service Datastore can be used as a “staging” internal repository.

The term *Resource Dependency Service Resource Modeling* refers to the set of two things:

- ▶ The two Resource Dependency Service repositories (Resource Dependency Service Datastore and RMTR) that hold infrastructure resource information.
- ▶ The software frameworks included to expose, access, and manage the stored resources.

3.8 Resource Modeling framework

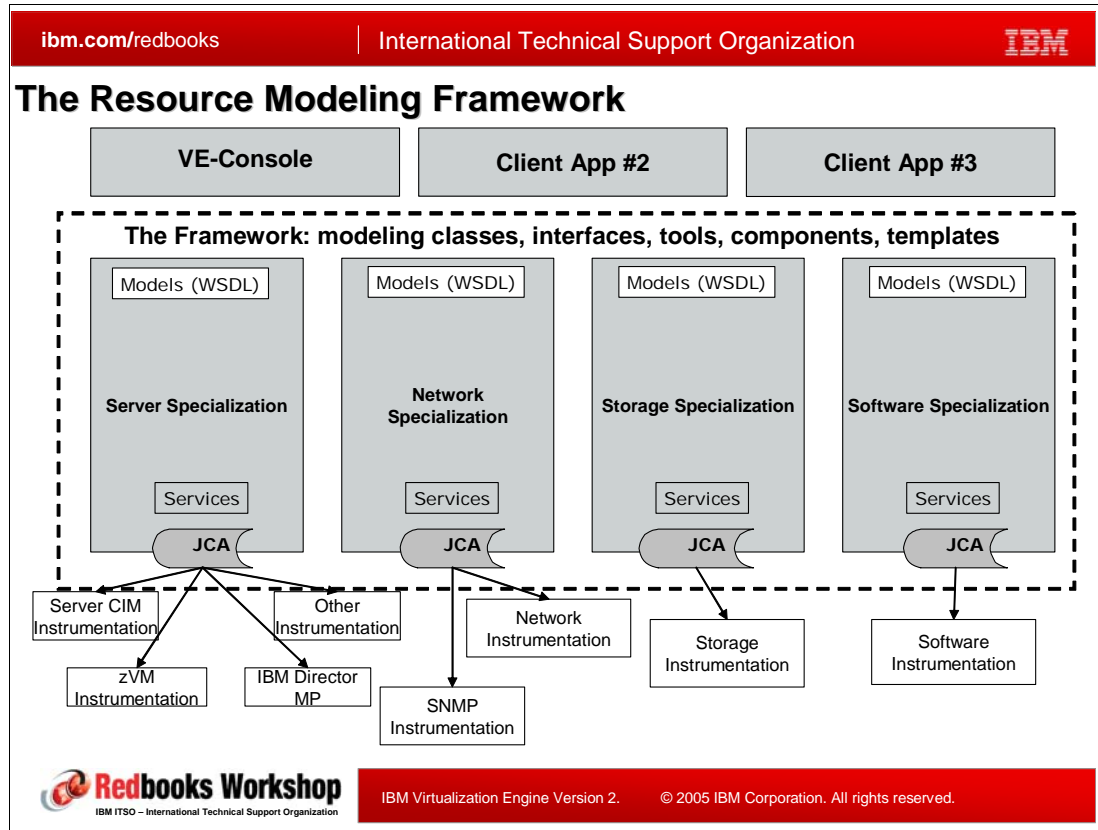


Figure 3-11 The Resource Modeling Framework

The design, planning and deployment of domain-specific models and manageable resources (servers, storage, network, etc.) can be simplified by leveraging the use of a common framework such as the Resource Modeling Framework. This allows domain experts to focus on domain-specific aspects while minimizing the need to understand the entire I/T infrastructure and or aspects which are unique to each resource origin or vendor.

The Resource Dependency Service Resource Modeling Framework is comprised of software code built using industry standards in order to expose access to and manipulation of I/T infrastructure resources managed by Resource Dependency Service. This software code is, essentially, an implementation of a set of Web service specifications on top of an IBM WebSphere foundation. The Resource Modeling Framework uses the WS-RF and WS_Addressings implementations provided in WebSphere.

The functionality provided by the Resource Modeling Framework includes the following:

- ▶ Services to define Web Service interfaces for manageable resources and their exposed management functions in a standard and interoperable way
- ▶ Services to define interfaces and usage patterns to publish, discover, compose, integrate, and manage resources within the Resource Dependency Service repositories
- ▶ Services to define the properties that define the unique identity of the managed resource
- ▶ Services to allow the overall interface and behavior of the manageable resource to be extended dynamically

- ▶ Services to define the interfaces to express and manage relationships between manageable resources
- ▶ Services to define patterns for manageable resource composition and service level policy association
- ▶ Services to help exploit the use of industry standards related to infrastructure resource management functions and pervasive manageable resource models.

Figure 3-11 shows how the Resource Modeling Framework includes the use of Web services and their corresponding WSDL interfaces to expose the model's manageable resources. Other standards such as the use of the J2EE Java Connector Architecture (JCA) are included as a formal method for accessing the native instrumentation (outside of the WebSphere Application server J2EE container) associated with the manageable resources. The Resource Modeling Framework also makes use of many modeling classes, tools, components (EJBs, Session Beans, etc.) to create the standard representation of the manageable resources by domain (storage, servers, etc.). It can be seen on Figure 3-14 on page 126 how the Resource Modeling Framework also provides services to allow for manageable resources to be associated to an organization's business processes and operations.

Resource Modeling Type Registry

Provisioning and orchestration are key functions of a service oriented infrastructure. In order to provision a resource, the system must understand the constituent components of each resource, their pre-requisites and their dependencies with other resources and the state of each available resource in the environment. The resource modeling framework needs to provide the necessary information to understand the characteristics of each component, its state, and how it interacts with other resources.

The infrastructure is made of multiple types of resources. Various resources have to be combined with the relationships among them. To do so a list of the components, their characteristics and their connections has to be built; this includes information about the resource interface and its properties as well as about the relation description to other resource types.

This catalog is expressed as a Resource Type Registry: the Resource Type Registry is a catalog of resources that contains information about resource types as well as possible relationships to other resource types.

The type registry is represented through a WS-Resource that contains the properties about the resources, their types and their relationships. The resource type registry simply contains the name of a type and any client is able to query for a resource type service using the type name as search criteria.

The Virtualization Engine Version 2 provides different types of manageable resources, for servers, network, or storage; for example:

- ▶ The *OSContainer*, which represents a logical collection of resources necessary to run an operating system. An OSContainer may represent a standalone server or some type of virtualized server (e.g., LPAR, virtual machine).
- ▶ The *OperatingSystem* manageable resource represents an operating system instance that has been booted in an OSContainer.

Figure 3-12 illustrates, a theoretical view of the CIM/CIMOM-based resource model implemented in Resource Dependency Service. Resources such as OS Container and OperatingSystem are represented along with their interrelationships. For instance, the OS Container represents any environment platform capable of hosting an operating system of

any kind. It has attributes (e.g. manufacturer, machine type, model, number of processors, etc.) and methods. The possible relationships that the OS Container MR has include Hosts, HostedBy, and LastBootedOS. The OperatingSystem MR represents a booted operating system. It has properties such as Name, OSType, and OSVersion. It has methods that include Reboot, and Shutdown. One of its possible relationships is OSInstalledOn. Another example is the StorageVolume, which represents a physical storage disk.

Resource Dependency Service's RMTR houses, both, resource model definitions and resource relationships. All resources types are modeled as stateful WS-Resource services. Access to the resource and relationship information is obtained only via Resource Dependency Service's exposed Web services interface, which is used by the Virtualization Engine console. Direct programmatic manipulation of the RMTR data is not allowed in this release of the Virtualization Engine.

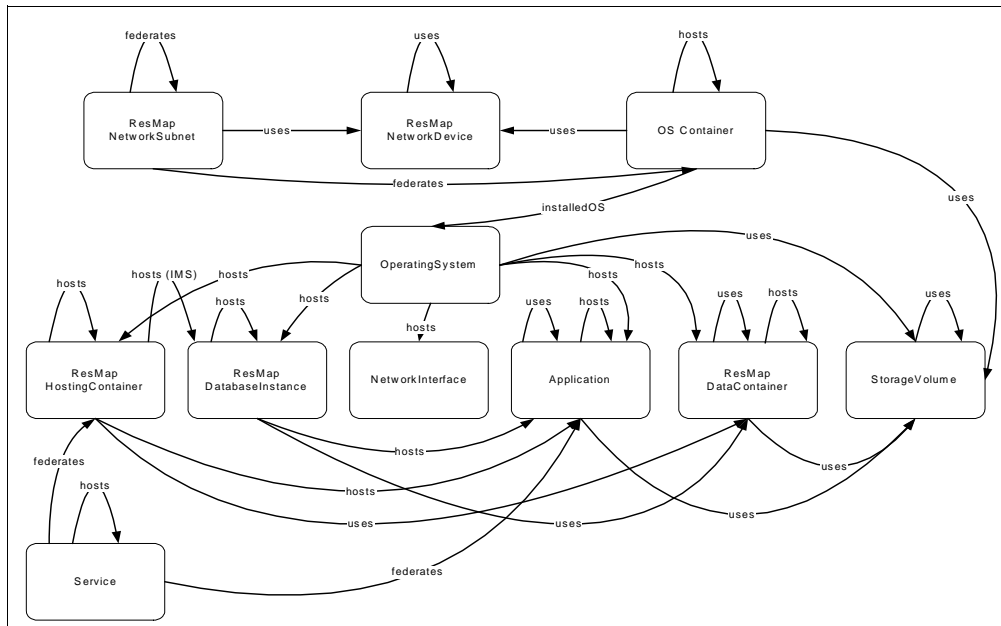


Figure 3-12 The Resource Model

Many of the resources displayed in Figure 3-12 are described in Figure 3-9. Neither of these figures is meant to be a complete representation of the modeled resources and relationships, but only a representative subset.

It is important to understand that the proposed resource model does not introduce another resource model to the industry: this model reflects and depends upon the DMTF ongoing work to define resource model profiles derived from the CIM resource model schema.

Data Source Federation

An interesting feature of Resource Dependency Service is the ability to discover resource data and mine information from third-party resource repositories such as Configuration Management Database (CMDB) and DCM. This way, external repositories can be mined for resource data, as needed, in order to create MRs and store them in the Resource Modeling Type Registry (when their attributes are considered complete).

Note that this capability will be available after Resource Dependency Service general availability.

The CMDB is a term used by vendors in the industry to refer to their repositories for infrastructure assets and their associated configurations. The CMDB can be a single relational database or a set of federated data stores created to behave as a single repository. The Data Center Model (DCM) is a component of TPM and it is used as a centralized repository that contains physical and logical assets and their relationships. Federation could also provide a mechanism by which repositories from other vendors such as HP's OpenView can be mined for infrastructure resource information.

3.9 Positioning Resource Dependency Service and the Common Configuration Management Database

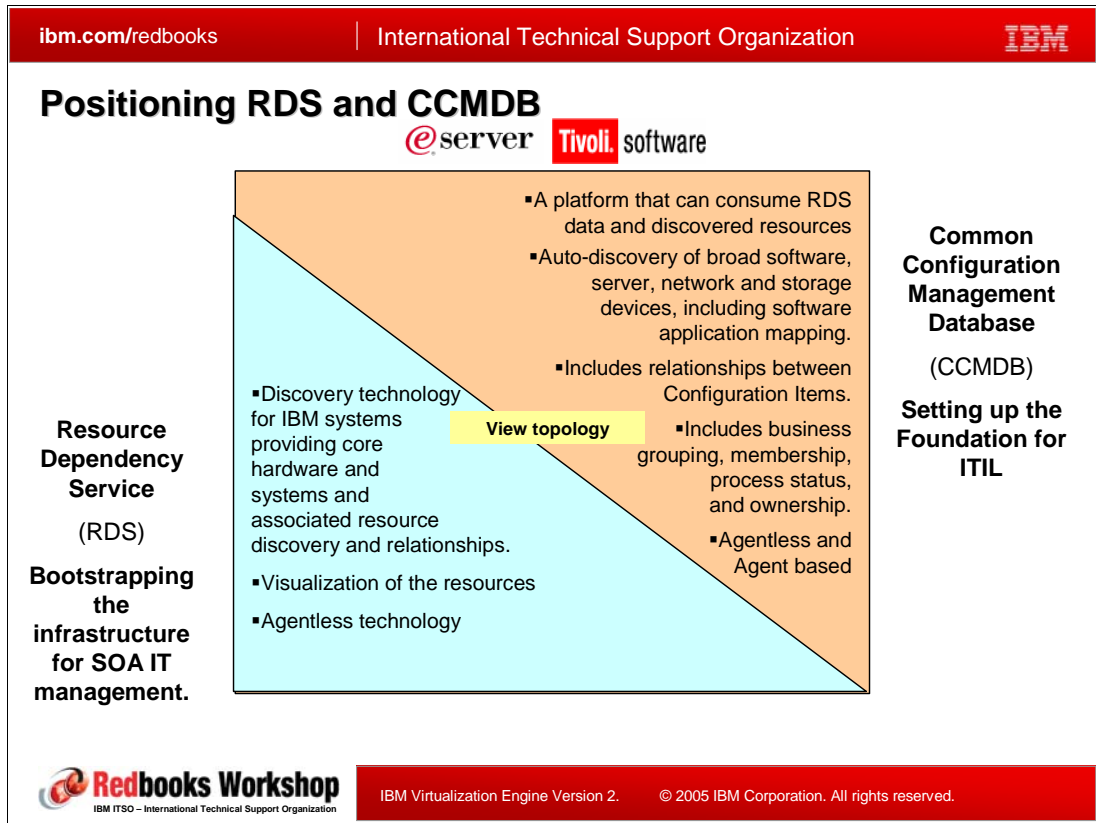


Figure 3-13 Resource Dependency Service versus CCMDB

In July 2005, IBM announced the IBM Tivoli Change and Configuration Management Database (CCMDB) product which provides a “virtualized” database that federates IT information spread across multiple databases. It provides a single view of an application running on multiple servers. What differentiates Resource Dependency Service from CCMDB? The following summarizes the complementarity of these components.

Resource Dependency Service:

- ▶ Is a discovery technology for servers, storage, network, and installed software applications and Operating Systems, providing discovery and relationships.
- ▶ Provides tools for visualization of these resources and relationships.
- ▶ Is based on agentless technology.
- ▶ Provides a foundation for WSDM based manageable resources.
- ▶ Provides the foundation of web services for the IT infrastructure management on which service oriented infrastructure solutions are based.

IBM Tivoli Change and Configuration Management Database:

- ▶ Provides the foundation for the IT Infrastructure Library (ITIL) based infrastructure management; ITIL⁵ is a very widely accepted approach to IT service management in the world.

⁵ More information about ITIL can be found at the following Web site: <http://www.itil.co.uk/>

- ▶ Provides auto-discovery of broad software, server, network and storage devices, including software application mapping.
- ▶ Includes relationships between configuration items, but is expanded to include business grouping, membership, process status, and ownership.
- ▶ Can be agentless and agent based.
- ▶ Includes process management user interfaces, providing for human and tool interaction in process-centric manner, increasing efficiency and effectiveness.
- ▶ Will provide a WSDM MUWS access standard through which resources can be managed through a common web services resource model.

3.10 Analytics and correlation engines

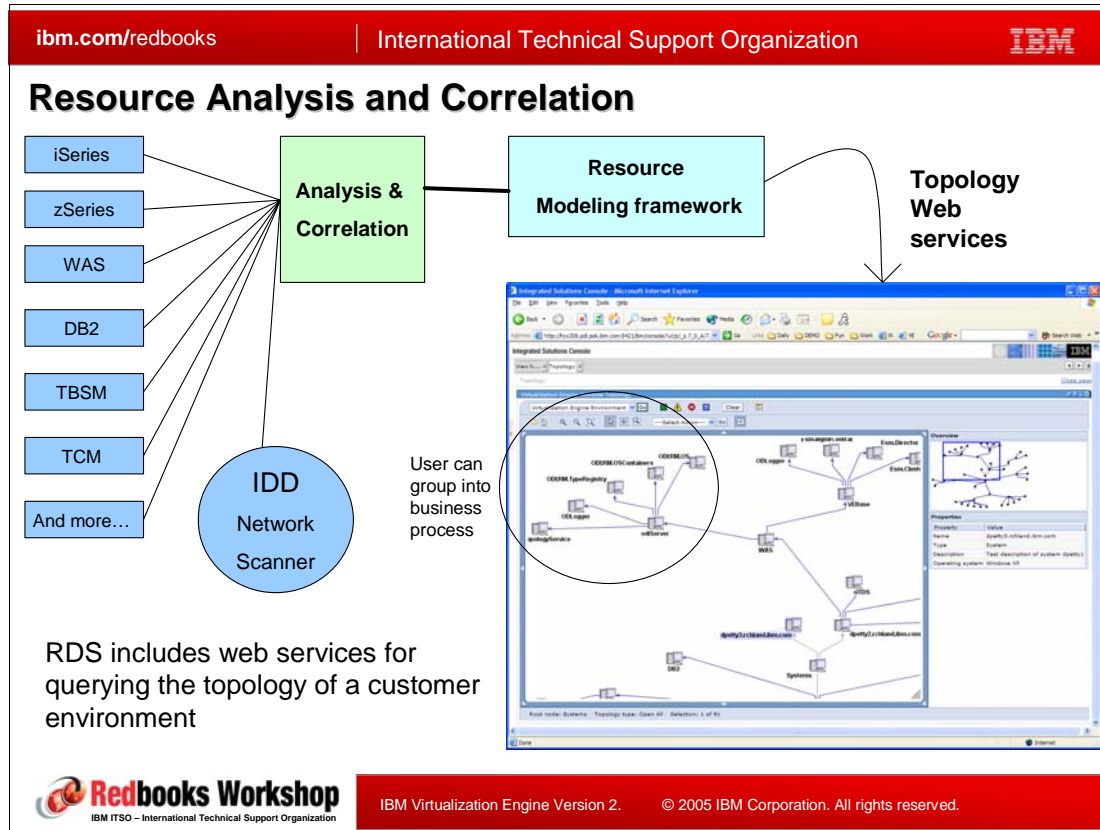


Figure 3-14 Resource Analysis and Correlation

The resource modeling framework must describe the relationships among resources: Resource Dependency Service provides the tooling to discover and model these relations.

As the name implies, the purpose of the Analytics and Correlation engines is to analyze, transform, and correlate the gathered and discovered environment data so it can be properly stored in the Resource Dependency Service repositories (Resource Dependency Service Datastore and the Resource Modeling Type Registry).

Every time that a data collector finishes executing, it will start an instance of the Correlation Engine; only one instance of the correlation engine can run at one time.

The correlation engine has three main functions:

- ▶ Gather data from the mined data sources.
- ▶ Put the consolidated data (combining “like” resources) into the Resource Dependency Service Datastore.
- ▶ Call the manageable resource factory to update the RMTR, if the found resources are complete enough to reside there.

Many times, the same resource can be discovered multiple times using different data collectors and discovery methods. In these cases the correlation engine detects this situation and makes sure that the resource is singly defined in the Resource Dependency Service repositories and the aggregated data is consolidated correctly.

Figure 3-14 shows how the Analysis and Correlation engines are used to process the resource information discovered and collected by Resource Dependency Service. The resource information is, then, stored and manipulated using Resource Dependency Service' Resource Modeling and exposed via Web services to external applications such as the Virtualization Engine console for user manipulation.

3.11 Topology services

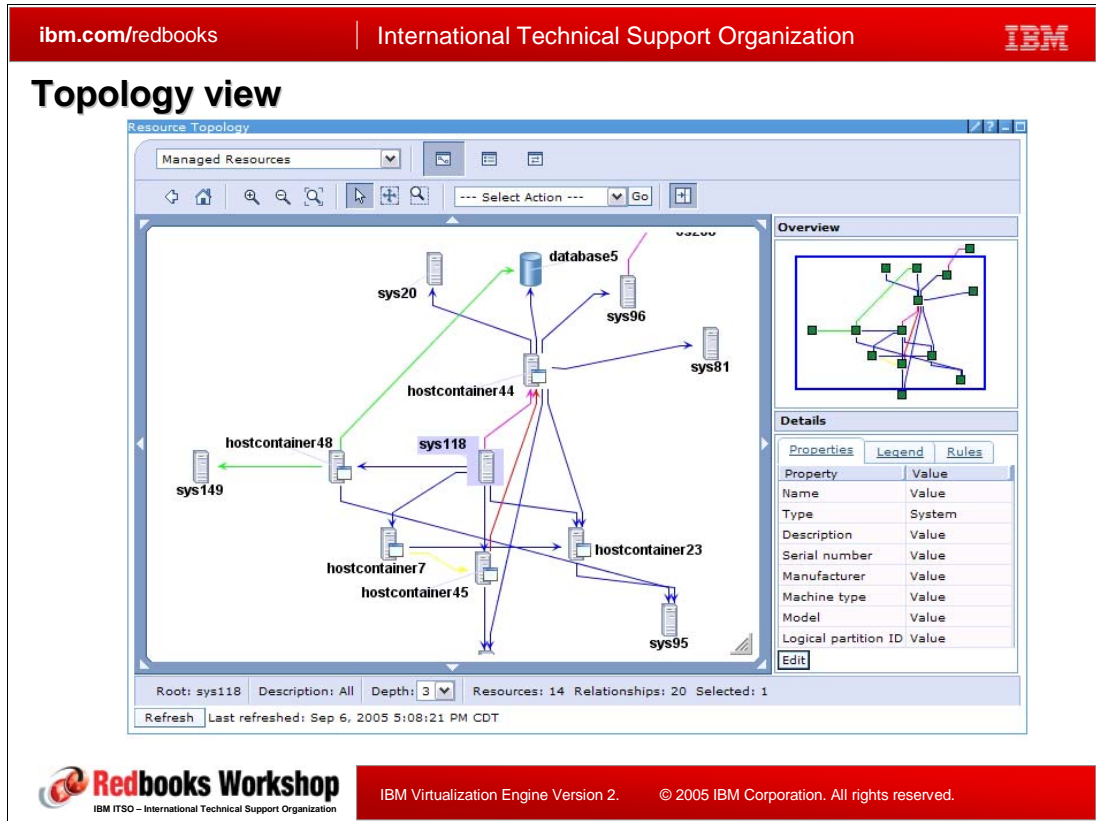


Figure 3-15 Topology view

The functionality exposed by Resource Dependency Service to programmatic clients is made available via Web services (and their WSDL descriptions). These Web services could be used to programmatically query, view, and update the contents of the Resource Dependency Service repository (e.g. resources, relationships, topologies, etc.); in this release these interfaces are not externalized.

Because of their relative importance, this section covers the Resource Dependency Service topology services, which are offered to allow programmatic clients to access resource topology information. In this release of the IBM Virtualization Engine, the only consumer of these services is the Virtualization Engine console. The Virtualization Engine console uses these web services to build resource topologies that are displayed within its topology portlet. Through the use of the Virtualization Engine console, the user selects a topology resource to work with. The Virtualization Engine console will, then, use the Resource Dependency Service topology services to allow the user to query and visualize the properties of that resource, all the resources that have a relationship to that resource, and the nature of each relationship.

The topology view is described in detail in Section 4.7, "The topology view" on page 162.

Note: It is important to note that, in this release of the IBM Virtualization Engine, the intent is to provide all end-user access to the Resource Dependency Service services through the Virtualization Engine console user interface. All of Resource Dependency Service's exposed Web services are accessed via the Virtualization Engine console and not directly in a programmatic way. The Virtualization Engine console will, in turn, serve as a proxy and will federate the use of the Web services, as required on behalf of its users.

3.12 Runtime, packaging, installation and configuration

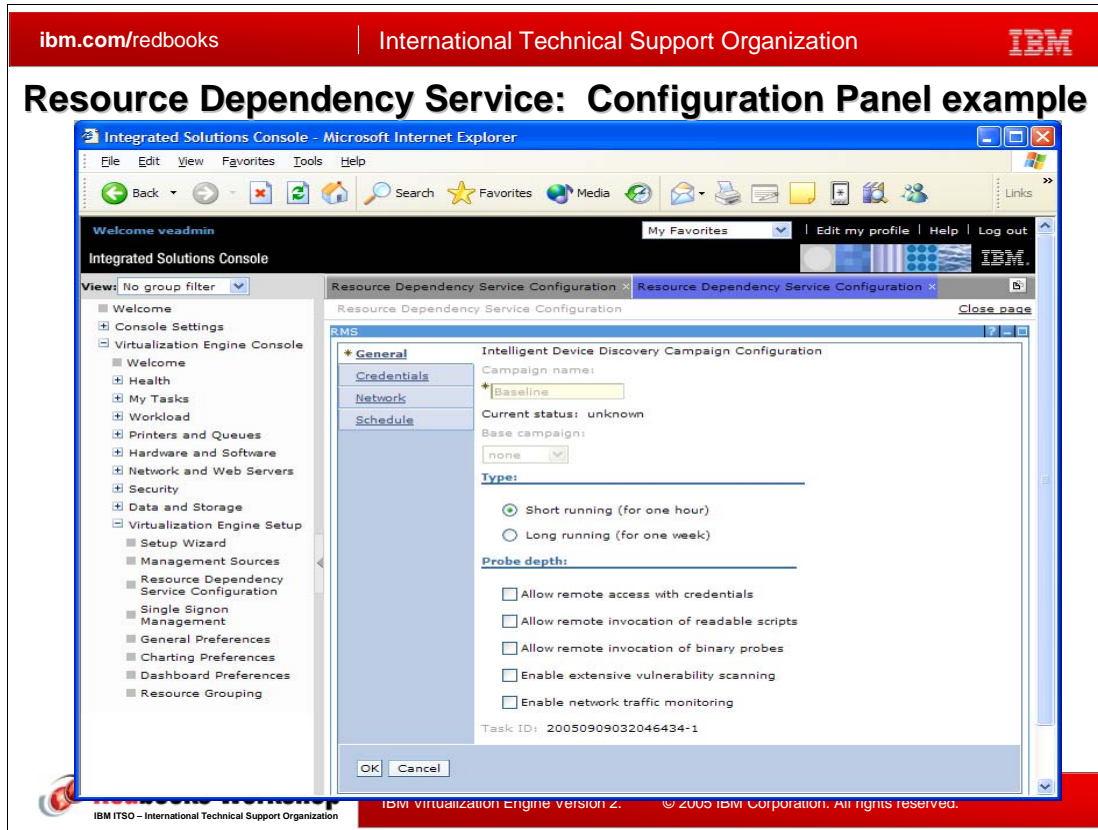


Figure 3-16 Sample Resource Dependency Service Configuration Panel

Runtime

The version of Resource Dependency Service included in the Virtualization Engine version 2 can only execute on the Linux environment. It currently supports the following Linux environments:

- ▶ SUSE Linux Enterprise Server (SLES) version 9
- ▶ Redhat Enterprise Linux (RHEL) version 4

Note: The supported versions of Linux can be executing on xSeries, iSeries, pSeries, or as zLinux.

Resource Dependency Service executes as a set of J2EE applications running on top of WebSphere Application server. For instance, the Resource Modeling components can be viewed and monitored as three separate applications: ODIRelationshipStore, ODITypeRegistry, and RVSServer.

Event logging for Resource Dependency Service is enabled using the standard WebSphere logging mechanisms. Resource Dependency Service's Resource Modeling uses java package names as the component logging string. Logs are created in the location specified on the WebSphere Diagnostic Trace Service Configuration page.

The Resource Dependency Service data collectors always run asynchronously. While simple execution status is available from the Resource Dependency Service administration panel,

WebSphere Application server logs must be viewed for detailed status and messages. In order to prevent server memory overruns and performance issues, only one of each type of data collector is allowed to execute at once. A data collectors of a given type will be queued up, if a collector of the same type is detected as already running. The queued collectors will run after currently executing data collector is complete.

Packaging and Installation

Resource Dependency Service is packaged as an installable component of the Virtualization Engine version 2 and uses the same installation facilities. It is packaged as “EAR” files for deployment into WebSphere Application server. If Resource Dependency Service is selected for installation, it will also install the Resource Modeling components. The installation process will do things like WebSphere Application server configuration for Resource Dependency Service services, DB2 instance/table creation, and some automated Resource Dependency Service configuration. The runtime environment for Resource Dependency Service is referred to as the *Resource Dependency Service Server*.

If the optional IDD Microsoft Windows helper probe is required for deeper resource information discovery, it must be installed on a single Microsoft Windows server that has connectivity to the Resource Dependency Service Server. The connectivity between the two servers is done securely over SSL. The Windows helper probe server must have the Java Runtime Environment (JRE) installed. Therefore, the JRE is installed during the probe’s installation if not detected as present.

Configuration

Resource Dependency Service is configured via the Virtualization Engine console using a variety of graphical user interface panels. The basic configuration is done automatically during installation. The following Resource Dependency Service components require specific configuration:

► IDD

The IDD configuration areas include:

- Type of scanning to be done - short (1 hour) or long (1 week) duration. Long duration allows for the discovery of resources that may not be up and running constantly.
- Range of IP addresses to scan and hosts to exclude
- Whether or not to use the Microsoft Windows helper probe
- Campaign names. A campaign name is used to identify a set of scanning objectives and configuration parameters.
- Credentials to use, when required (for probe, SNMP, WMI, SSH, etc.)

► Resource Dependency Service Data Collectors

- Scheduling for periodic execution
- Credentials needed by data collectors, if applicable
- JDBC and other database connection parameters, if applicable
- SSL, if applicable
- Target hostnames, if applicable
- Port and queue names, if applicable
- FTP, JES, JCL, CICS CMAS, WSAAs Subsystem, IMS database and transaction manager subsystems, REXX parameters for zOS data collector

Note: This book is not intended to provide detailed Resource Dependency Service installation and configuration instructions. The reader should refer to the Virtualization Engine InfoCenter and other available Resource Dependency Service documentation for that type of assistance.

Figure 3-16 illustrates one of many graphical user interface panels used to configure Resource Dependency Service. This one is used for the configuration of general IDD parameters such as scan duration and type.

Common Component Dependencies

Resource Dependency Service uses and depends on the following foundational components:

- ▶ IBM WebSphere Application Server (WAS). Resource Dependency Service uses the Web service foundational framework provided by WebSphere Application server. This release of the Virtualization Engine depends on WebSphere Application server version 6.0.2 + Virtualization Engine iFixes.
- ▶ IBM DB2. The IDD database and the Resource Dependency Service Datastore are implemented using DB2. This release of the Virtualization Engine depends on DB2 UDB version 8.2 SP1.
- ▶ IBM Tivoli Directory Server. ITDS is used to store configuration and security information as part of the GCR. This release of the Virtualization Engine can use the following versions of ITDS:
 - ITDS 6.0 SP1
 - ITDS 5.1, 5.2, or IBM Directory Server for iSeries
- ▶ Java Runtime Environment (JRE). This release of the Virtualization Engine depends on JRE version 1.4.2 SP2. Resource Dependency Service was written and is packaged as a set of J2EE applications. Therefore, it requires the Java runtime.

Security

Resource Dependency Service is governed by and follows the security framework and policies defined in Chapter 7, “Security” on page 253. Nonetheless, we will briefly touch on some security features of particular importance to Resource Dependency Service.

The Single Signon (SSO) functionality built into Virtualization Engine manages most user security functions including userid/password management, federation, and mapping. However, some additional security requirements exist:

- ▶ IDD scanning levels 1+ require the setup and use of separate credentials (userid, password, etc.) via the Virtualization Engine console. This is done using the panels to configure the use of IDD and during the configuration of the optional Microsoft Windows IDD probe.
- ▶ Many of the Resource Dependency Service data collectors require the setup and use of separate credentials.

During the installation and configuration of the Resource Dependency Service components, an LDAP repository (ITDS) is used to hold configuration and security information such as the location of Resource Dependency Service repositories, URIs for important Web services, userids, passwords, etc. LDAP groups are mapped to roles in support of the security mechanism. This LDAP repository for general Virtualization Engine information is known as the Global Configuration Repository (GCR).

LTPA Tokens and Username/Password pairs are used by Resource Dependency Service as the supported and configured authentication mechanism. SSL can be setup and used for communications with the Resource Dependency Service data collectors. Additionally, the DB2 credentials used at install time are configured as J2C Authentication Entries. The Web services interfaces implemented by the Resource Modeling Framework follow the WS-Security standard.



The Virtualization Engine console

The goal of the Virtualization Engine console is to provide a converged view of the user's entire environment. The ultimate goal is to deliver one continuous console experience to a user: the user will be able to display its environment and to launch commands to adjust or to make corrections from one single screen.

This chapter describes the Virtualization Engine console, a single portal for administering all virtual resources such as resource health, monitoring, and integration access to planning tools. The role of this console is to gather, analyze and interact with the resources in your infrastructure.

In this chapter we discuss:

- ▶ The role of the Virtualization Engine console
- ▶ The new usability features such as
 - The welcome page
 - The health center wizard
 - The Virtualization Engine updates mechanism
- ▶ The health center
- ▶ The launch-in-context
- ▶ The business resource topology view

The IBM Virtualization Engine console is based on the IBM Integrated Solutions Console framework to provide a consolidated view for managing your virtualized enterprise resources.

4.1 Positioning the Virtualization Engine console

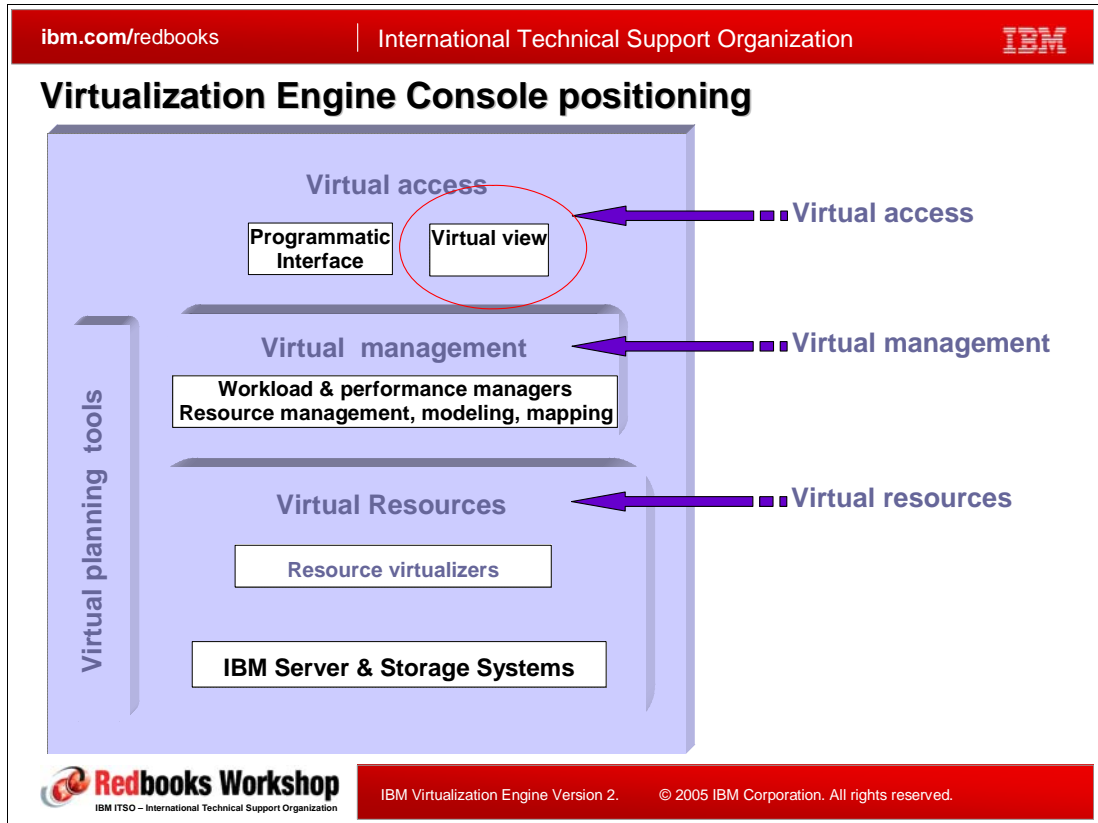


Figure 4-1 Virtualization Engine console positioning

In Section 1.5, “The Virtualization Engine platform” on page 20, we have presented the three layers of a virtualized environment. As shown in Figure 4-1, the Virtualization Engine console is positioned in the virtual access layer. From this Virtualization Engine console, the users have a complete view of the resources, virtual or physical, which are deployed in the infrastructure, and can interact with the management tools used to adapt the resources to the business needs. The goal is clearly to have the Virtualization Engine console be the default administration point for any administrators independently of the type of servers they have to manage. They will be assisted in determining the state of the infrastructure and the appropriate tools will be launched each time a correction is necessary.

This virtual view is the main function of the Virtualization Engine console, however some of its functions overlap the virtual management layer since it is possible from the console to manage resources. It is especially true for some of the operations that can be started directly through portlets, and where no external applications are used to modify the environment.

4.2 Virtualization Engine console functions

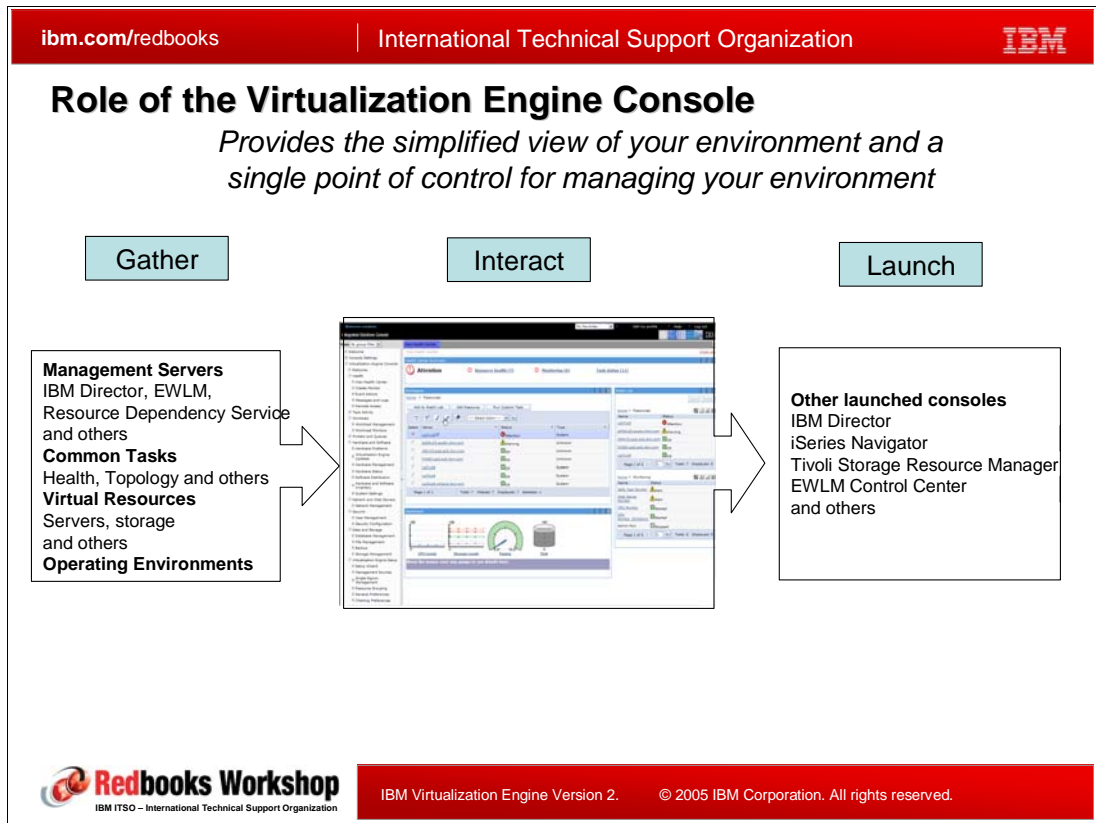


Figure 4-2 Role of the Virtualization Engine console

The role of the Virtualization Engine console is to let the administrators have a clear view on their infrastructure and assist them in taking any action in order to correct an error or adapt the infrastructure to new needs. In order to achieve such a goal, the function of the console must be threefold:

- ▶ Gathering information
- ▶ Displaying information
- ▶ Launching commands

Gathering information

Before thinking of managing resources, you must first know which resources exist that you have to manage. The Virtualization Engine console links to several external sources to gather information on the existing infrastructure. In particular, the Virtualization Engine console relies on:

- ▶ IBM Director server (more information about IBM Director in Chapter 2, “IBM Director 5.10” on page 37) retrieves information on hardware components and operating system installed on IIBM and non IBM servers.
- ▶ IBM Cluster System Manager retrieves information on AIX servers part of a cluster.
- ▶ Resource Dependency Service (more information about Resource Dependency Service in Chapter 3, “Resource Dependency Service” on page 97) retrieves information about hardware servers, storage as well as operating systems or middle ware components.

- ▶ Enterprise Workload Manager (more information about EWLM in Chapter 5, “Enterprise Workload Manager” on page 173) retrieves information about operating systems involved in end to end transactions.
- ▶ Management Central, Using Management Central, an administrator or operator may perform an operation on a System Group instead of repeatedly performing the same operation on individual iSeries systems.

These tools allow the console to monitor hardware including: xSeries, iSeries, pSeries, zSeries, Hardware management console (HMC), non IBM x86 servers and Storage systems and devices, as well as the following operating environments: Windows, Linux, i5/OS, AIX, z/OS and Tivoli.

Displaying Information

As described before, the Virtualization Engine console gathers data from many management points on the network, the next step is to consolidate this data and to display the useful elements.

The useful elements can be either the failing components, or the resources that need a special focus. The health center, described in Section 4.5, “The health center function” on page 147, is responsible for this role within the console. Its summary window lets administrators know immediately whether the infrastructure is healthy or if there is an abnormal situation. Its dashboards let the administrator closely follow the state of resources, they have chosen to monitor.

Another role of the Virtualization Engine console is to show the relationship between the resources under its control. The topology view, described in Section 4.7, “The topology view” on page 162, provides a way to display the resources identified in the infrastructure and let the administrator choose the level of information to be displayed.

Figure 4-3 shows the left frame of the console, from where you launch the actions to display the various components.

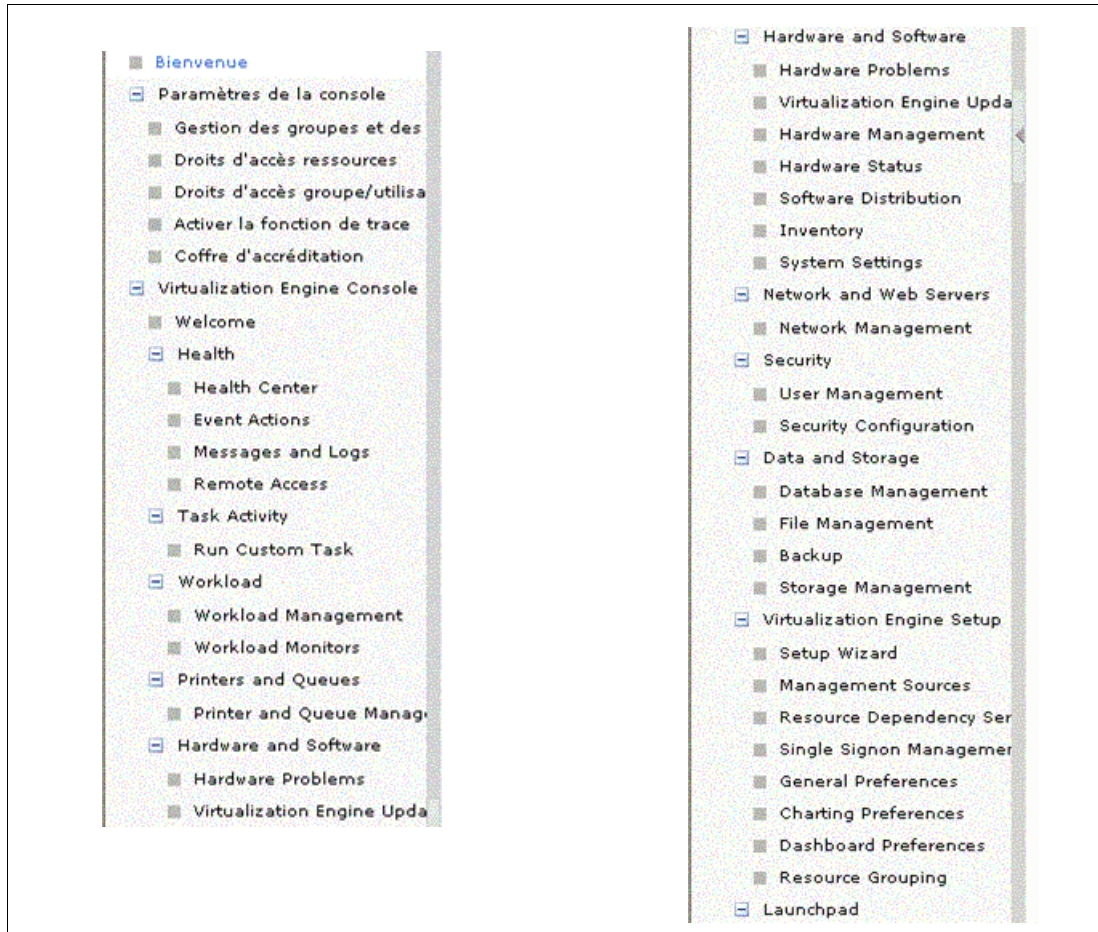


Figure 4-3 The master Virtualization Engine console panel

Launching consoles

The information being gathered and displayed, the last step is to interact with the resources. There are many ways to interact with your virtualized infrastructure. The applications you can launch from the Virtualization Engine console include:

- ▶ DB2 control Center: a graphical user interface (GUI) allows you to perform most of the tasks necessary to setup, configure, and modify the federated system. The DB2 Control Center uses panels--dialog boxes and wizards--to guide you through a task. These panels contain interactive help when your mouse hovers over a control such as a list box or command button. Additionally, each panel has a help button that provides information about the panel task, and links to related concepts and reference information.
- ▶ IBM Director console, as described in Chapter 2, "IBM Director 5.10" on page 37
- ▶ I5/OS Extensions: iSeries Navigator is a no-charge feature of iSeries that is bundled with i5/OS. iSeries Navigator is a graphical interface for the iSeries server that gives end users and administrators a simple way to work with iSeries resources by integrating the iSeries environment with the Windows desktop.
- ▶ iSeries navigator Console.
- ▶ Web-based System Manager (includes the HMC) which is a client-server application that gives the user a powerful interface to manage UNIX systems. Web-based System Manager uses its graphical interface to enable the user to access and manage multiple remote machines.

- ▶ Tivoli Storage Area Network Manager: IBM Tivoli SAN Manager provides effective discovery and presentation of SAN physical and logical topologies and provides multiple views of the SAN. Through its interface, it can be configured to show historical and real-time monitoring of SAN fabric devices.
- ▶ Tivoli Storage Resource Manager: IBM Tivoli Storage Resource Manager is a comprehensive capacity management solution for heterogeneous storage environments. It includes enterprise-wide reporting and monitoring, policy-based management and automated capacity provisioning for DAS, NAS, and SAN environments.
- ▶ Multiple Device Manager: Multiple Device Manager is designed to reduce the complexity of managing SAN storage devices by allowing administrators to configure, manage and monitor storage from a single console.
- ▶ EWLM Control Center which is fully described in Section , “Control center” on page 176.

4.3 Virtualization Engine console architecture

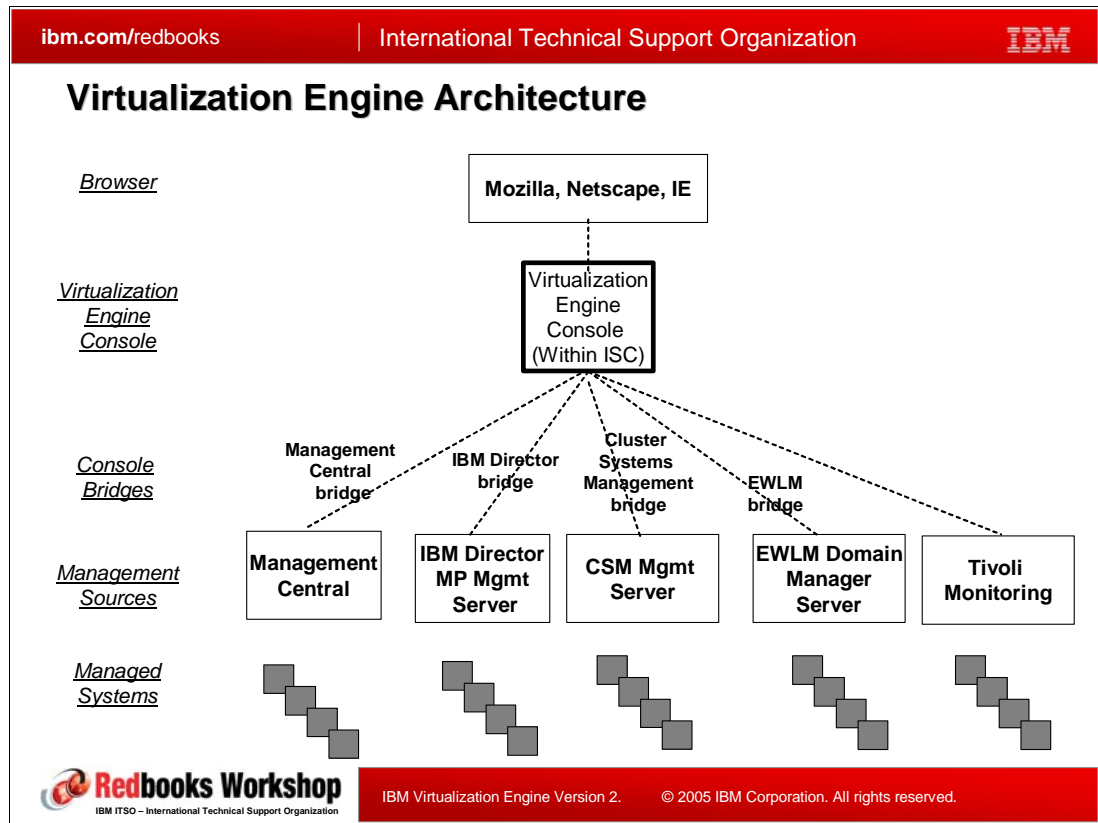


Figure 4-4 Virtualization Engine Console components

The console support is comprised of five elements:

- ▶ A Web browser,

The browser is used to access the Virtualization Engine console. Because the Virtualization Engine console is a Web-based application, you need to use a Web browser to access the console. The Virtualization Engine console works with Internet Explorer, Netscape and Mozilla) and does not require any additional plug-in. However, if you are planning to use full graphical features such as the dashboard, you need to install an SVG plug-in for your browser. The SVG support is currently available only on Internet Explorer 4.0 or higher.

- ▶ The Virtualization Engine console itself
- ▶ The IBM Virtualization Engine console bridge

We use sometimes the *Virtualization Engine console bridge* wording though there is no such thing: there are separate installable bridges depending on what product we want to bridge to; this expression, technically wrong, is generically used to refer to the bridge concept, whatever the product we want to bridge to. The Virtualization Engine console bridge is a software piece that allows the Virtualization Engine console to communicate via Web services with consoles. The Virtualization Engine console bridge provides interfaces to the Virtualization Engine console for accessing the Management Sources. The Virtualization Engine console bridge is running in a WebSphere Application Server. Therefore, when you install the bridge, a WebSphere Application Server is installed with it. Usually, the Virtualization Engine console bridge must be installed on the same machine

as the management source. However, if you are planning to use IBM Director, you can install it on a system other than that where IBM Director is installed.

- ▶ The management source

The management sources collect, distribute, and exchange data with the managed systems. The management source usually communicates with the Virtualization Engine console through a bridge. The management source is a previously deployed software server management application solution that the Virtualization Engine console is able to interact with, usually via a bridge. All information that you can see on the Virtualization Engine console are provided by the management sources.

- ▶ A set of managed resources

Managed systems are the endpoint systems, devices, or resources that are supported and managed by the management source, such as Cluster Systems Management, Management Central, Tivoli Monitoring, or IBM Director for Multiplatform). There is no Virtualization Engine unique code that is installed on the managed resources. However, managed agents typically run on the managed resources (for example, IBM Director agent and EWLM Agent).

The Virtualization Engine console requires a WebSphere application server and an LDAP server for its user management.

4.4 The welcome page

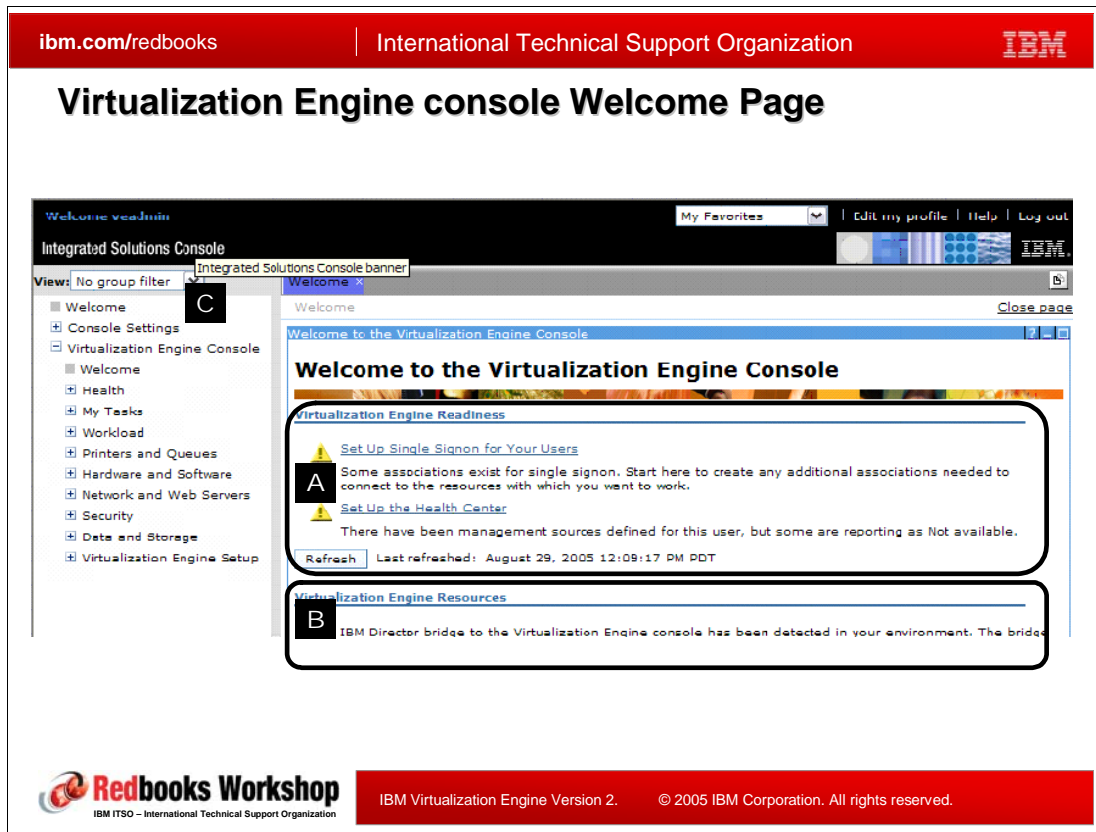


Figure 4-5 Virtualization Engine Welcome Page

The Virtualization Engine Welcome page is a new addition of the Release 2 of the Virtualization Engine console. It helps answer the questions: “Now that I have installed everything, how can I make sure that it is working, how do I start and what can I do?”

The users/administrators can access this welcome page from the left frame regrouping the tasks oriented application. You can see the welcome page menu item as being the first one listing in the Virtualization Engine console entries.

Attention: The very first entry listed in the left frame is also named/titled Welcome. It is the welcome page for the Integrated System Console on which the Virtualization Engine console is based. It provides information on the level of the code installed. Do not confuse with the entry for the Virtualization Engine console.

The Welcome page is divided in two parts:

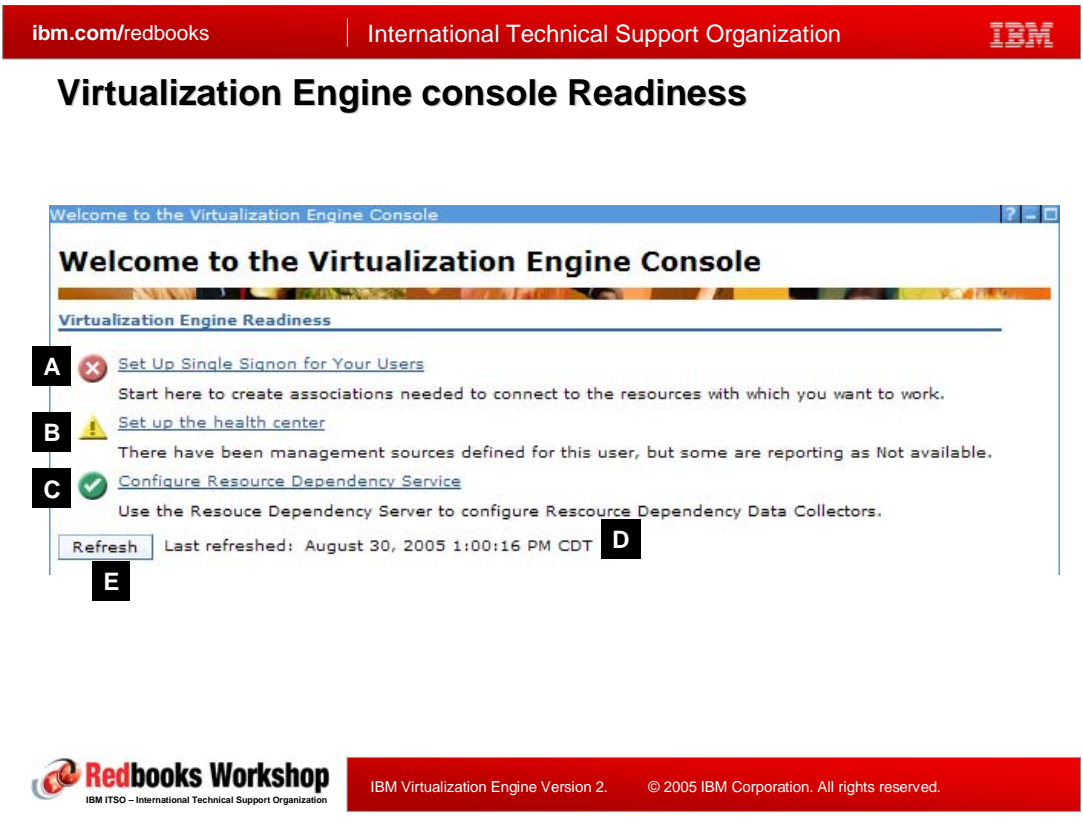
- ▶ The top panel (zone **A** on Figure 4-5): is called the *Virtualization Engine Readiness* and helps you find out if the components of the Virtualization Engine console are installed and correctly working.
- ▶ The lower panel (zone **B** on Figure 4-5): is called the *Virtualization Engine Resources* and focuses on the setup of the resources, such as IBM Director, or EWLM.

The **View** (**C**) allows to filter users authorizations. The Virtualization Engine console user groups can be:

- ▶ The *Virtualization EngineMonitor* represents a systems administrator that needs to know the status of Virtualization Engine console health and data but cannot update or delete any state or data.
- ▶ The *Virtualization EngineManager* represents a systems administrator in charge of multiple platforms who uses the Virtualization Engine console to get his job done. This role provides the ability to monitor and run tasks, but not the ability to set up or define how the virtualization environment is laid out.
- ▶ The *Virtualization EngineAdministrator* represents a lead systems administrator or systems planner in charge of part or all of the IT planning in the company. This role includes the support of the Virtualization Engine User and in addition has the ability to set up and tweak the Virtualization environment (setting up what resources can be virtualized, setting up single sign-on capabilities, etc.).
- ▶ A *Virtualization EngineAnonymous* represents any unauthenticated user.
- ▶ A *Virtualization EngineUser* represents any authenticated user.

In this version, the Virtualization Engine console only supports the Virtualization EngineAdministrator and the Virtualization EngineUser; the other groups are not used yet.

The Virtualization Engine Readiness panel






ibm.com/redbooks | International Technical Support Organization IBM

Virtualization Engine console Readiness

Welcome to the Virtualization Engine Console

Virtualization Engine Readiness

- A**  [Set Up Single Signon for Your Users](#)
Start here to create associations needed to connect to the resources with which you want to work.
- B**  [Set up the health center](#)
There have been management sources defined for this user, but some are reporting as Not available.
- C**  [Configure Resource Dependency Service](#)
Use the Resource Dependency Server to configure Resource Dependency Data Collectors.

Last refreshed: August 30, 2005 1:00:16 PM CDT **D**

E

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 4-6 Virtualization Engine Readiness

The virtualization Engine Readiness panel focus on readying the Virtualization Engine console. It browses your virtualized environment, discovers resources and helps you finalize their customizing. For each of the resources it finds, it provides a status colored icon:

- ▶ Red (icon type **A**): an action is required in order to complete the setup of that specific component
- ▶ Yellow (icon type **B**): some configuration is done, more may be needed

- ▶ Green (icon type **G**): all configuration is done, and the component is ready to use

The red and yellow icons are links to the necessary action plans needed to finalize the setup of the given components.

In Figure 4-6, we see the item **Configure Resource Dependency Service** as being green. This specific item shows up as soon as the console detects an Resource Dependency Service installation and, unless the installation failed leaving the Resource Dependency Service manager in an broken state, will be green independently of the resources being collected or not.

The last line includes the time stamp (**D**) indicating when the last assessment of the environment has been achieved. The **refresh** button (**E**) lets you force a new discovery of resources, and provides the latest status on all components.

Note that the links in the readiness section depend on what the user has installed: for example the Resource Dependency Service link will not show up if Resource Dependency Service is not installed in the environment.

Virtualization Engine Readiness: an example

The screenshot shows the 'Virtualization Engine console Readiness' page. At the top, there is a red banner with 'ibm.com/redbooks', 'International Technical Support Organization', and the 'IBM' logo. Below the banner, the title 'Virtualization Engine console Readiness : Example' is displayed. The main content area is titled 'Welcome to the Virtualization Engine Console' and 'Virtualization Engine Readiness'. It lists three items: a red 'X' icon for 'Set Up Single Signon for Your Users', a yellow warning icon for 'Set up the health center', and a green checkmark icon for 'Configure Resource Dependency Service'. A red arrow points from the red 'X' icon to a separate window titled 'Single Signon' which shows the 'Welcome to Single Signon Setup' page with instructions and links for managing user identity associations and configuration. At the bottom of the screenshot, there is a red banner with 'redbooks WORKSHOP', 'IBM ITSO - International Technical Support Organization', 'IBM Virtualization Engine Version 2.', and '© 2005 IBM Corporation. All rights reserved.'

Figure 4-7 Virtualization Engine Readiness: Example

As an example, Figure 4-7 shows what happens when you click on the red icon **Set up Signon For Your Users**. In our scenario, the red icon indicates that nothing has been done yet and that a full configuration must start. The **Welcome to Single Signon Setup** window shows up and guides you through the steps needed to let the Virtualization Engine console users connect to the remote systems without having to log in again. More on the security can be found in the Chapter 7, “Security” on page 253.

The second icon, **Set up the health center** would bring up the wizard which let you add management sources, groups and monitors to the health center. The details for this wizard are described in the Section , “The health center setup wizard” on page 148.

The Virtualization Engine Resources panel

ibm.com/redbooks | International Technical Support Organization IBM

Virtualization Engine Resources

Virtualization Engine Resources

The IBM Director bridge to the Virtualization Engine console has been detected in your environment. The bridge allows you to work with IBM Director through the Virtualization Engine console.

[Set up IBM Director](#) **A**

Enterprise Workload Manager (EWLM) has been detected in your environment. Ensure that you complete these EWLM tasks:

[Configure EWLM domain manager](#)

Find information on the different methods to configure the domain manager.

[Install EWLM managed servers](#)

Find information on the different methods to install the managed servers.

[Configure EWLM managed servers](#)

Find information on the different methods to configure the managed servers.

[Ensure that you install the latest Virtualization Engine updates](#) **B**

View, download, and install your Virtualization Engine software updates.

[Get to know the Virtualization Engine console](#)

Find information on how to navigate and use the Virtualization Engine console.

Redbooks Workshop
IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 4-8 Virtualization Engine Resources

The lower panel of the welcome page is dedicated to resources. It provides links to additional setup for discovered resources. In the example shown in Figure 4-8, the installation of bridges to IBM director and Enterprise Workload Manager (EWLM) have been detected, and corresponding actions are being proposed.

If you click on the IBM Director link (**A**), you launch a wizard that allows you configure the IBM director manager in your infrastructure. Note, that there might be several such managers and that you can choose which one you want to interact with. Once the director manager is selected, you can decide to start the director console to discover systems managed by this server, or start its software distribution manager to distribute and install agents.

The next choice you have is to install a local console for the IBM Director. If you choose this option you are prompted to accept the download and execution of a Java Networking Language Protocol (JNLP) file, that will guide you through the installation steps for the IBM Director Console. This operation is painless, since you have a status window letting you know the progress of the installation, and only a panel asking you to authorize this installation will show up. The remaining download, install and launch of the console is done without your intervention.

You also have a link that brings up the infocenter documentation for the IBM Director and let you research additional information on this subject.

In our example, an EWLM installation has been detected, as such we are being shown up the link to the InfoCenter that documents the additional steps to complete the customization of this environment. Those links include:

- ▶ Configure the EWLM domain manager
- ▶ Install EWLM managed servers
- ▶ Configure EWLM managed servers

Once again, the links are only shown if the products are installed in the environment.

Virtualization Engine Updates panel

The screenshot displays the 'Virtualization Engine Updates' interface. At the top, there is a navigation bar with 'ibm.com/redbooks', 'International Technical Support Organization', and the IBM logo. Below this is the title 'Virtualization Engine Updates'. The main content area has three tabs: 'Updates' (labeled 'C'), 'Software' (labeled 'B'), and 'Systems' (labeled 'A'). The 'Updates' tab is active, showing a table of updates. A context menu is open over the first row, with 'View Update Details' selected. The table has the following data:

Select	Update	Version	Software	Platform	System	Size	Status
<input checked="" type="checkbox"/>	DAIx602		Director Agent	Linux on xSeries	hcx148.pdl.pok.ibm.com	4.33 KB	Downloaded
<input type="checkbox"/>	DAIx603		Director Agent	Linux on xSeries	hcx148.pdl.pok.ibm.com	4.33 KB	Downloaded
<input type="checkbox"/>	DAIz201		Director Agent	Linux on zSeries	hcx148.pdl.pok.ibm.com	4.27 KB	Downloaded
<input type="checkbox"/>	DAIx601	5.10.10	IBM Director Agent	Linux on xSeries	hcx148.pdl.pok.ibm.com	4.33 KB	Downloaded
<input type="checkbox"/>	DAIx203	5.10.21	IBM Director Agent	Linux on xSeries	hcx148.pdl.pok.ibm.com	4.31 KB	Downloaded
<input type="checkbox"/>	DAIp202	5.10.20	IBM Director Agent	Linux on Power	hcx148.pdl.pok.ibm.com	4.32 KB	Downloaded
<input type="checkbox"/>	DAI5201	5.10.10	IBM Director Agent	i5/OS	hcx148.pdl.pok.ibm.com	3.55 KB	Downloaded
<input type="checkbox"/>	DAIp203	5.10.21	IBM Director Agent	Linux on Power	hcx148.pdl.pok.ibm.com	4.32 KB	Downloaded
<input type="checkbox"/>	DAI5202	5.10.20	IBM Director Agent	i5/OS	hcx148.pdl.pok.ibm.com	3.55 KB	Downloaded

At the bottom of the page, there is a 'Redbooks Workshop' logo and a footer with the text 'IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.'


Figure 4-9 Virtualization Engine Updates

The next link (field **B** in Figure 4-8) is an addition of the version 2 of the Virtualization Engine console. Given that the console has been developed to simplify the management of your environment, it is out of the question that its administration, in terms of identifying and applying new fixes for all of its components, becomes a nightmare. This link guides you to the window where you will be able to select the systems and software, part of your virtualized environment, that you want to consider when the console will check for new fixes directly to the IBM site.

- ▶ The Systems tab (field **A**) shows a list of all the systems included in your virtualized environment, their hostname, type (AIX, i5/OS, Linux, or Windows), their internet connection method, whether or not you select the system for update, and a status message. If some of the systems can not be accessed, you receive an error message showing there has been a problem, and if this a connection problem, you also have a link leading you to the single signon setup task.
- ▶ The software tab (field **B**) brings up a window listing for each considered systems (checked in the systems tab):

- The Virtualization Engine software components installed
- The path where this component is installed
- Its current version
- Its last version
- The last time it has been updated

A search toolbar helps you find specific components name based on their name their and their location in the table.

- ▶ The updates tab (field ) lists the updates for the selected software which have been downloaded (or are being currently being downloaded). For each update, you have:
 - The name of the update
 - The version
 - The name of the component it applies to
 - The platform it refers to
 - The platform on which it can be installed
 - The size of the update
 - The status (installed, downloaded, install failed)

When you select an update, you get its readme file as well as additional information including :

- Type: update only or install package ?
- Reboot required: is a reboot needed for this update to be active ?
- Log-in required: is a login required to install this update ?
- Agent restart required: do you have to restart the program ?
- Build date: the date of the build ?
- Pre-requisites: level of software needed for this update ?

If the status for the selected update is **downloaded**, the contextual menu lets you distribute and install this update on every selected servers for which this update applies.

If the status for the selected update is **install failed**, the contextual menu let you view the error report as well as log and trace for this installation which might help understand what went wrong.

You also have the choice of deleting this update from the server, whether the distribution and install succeeded or not.

The last item is status line indicating when was the last time the search for new fixes has occurred, and pressing the **refresh** button let you force a new scan for updates.

4.5 The health center function

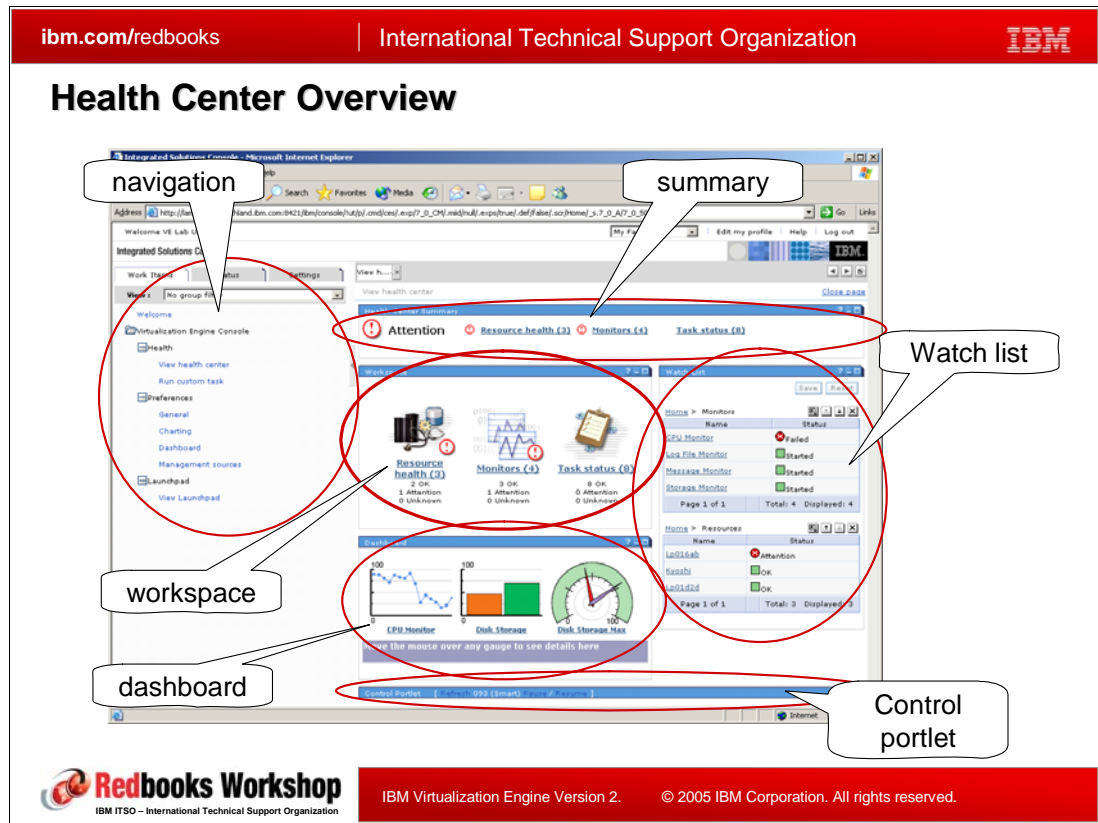


Figure 4-10 Health Center overview

The Health Center provides a single user interface in an heterogeneous environment by aggregating information where you can perform key server (AIX, Linux, i5/OS, and Windows) and storage monitoring and management tasks. The Health Center simplifies your work as a system administrator by providing a Web console that displays the health of your server and storage resources, monitors, and tasks. You can identify problems and take actions to solve them from the health center by simply clicking on the resource, monitor, or task that needs attention. The Health Center gives administrators a bird's eye view of system resource health across a converged system and storage management environment. The health center provides the following functions:

- ▶ Displays and analyzes the status and health of resources.
- ▶ Displays real-time resource data and analyses through the dashboard display.
- ▶ Monitors resources across a distributed cross-platform environment.
- ▶ Runs actions against resources in response to conditions detected by the health center.

The health center setup wizard

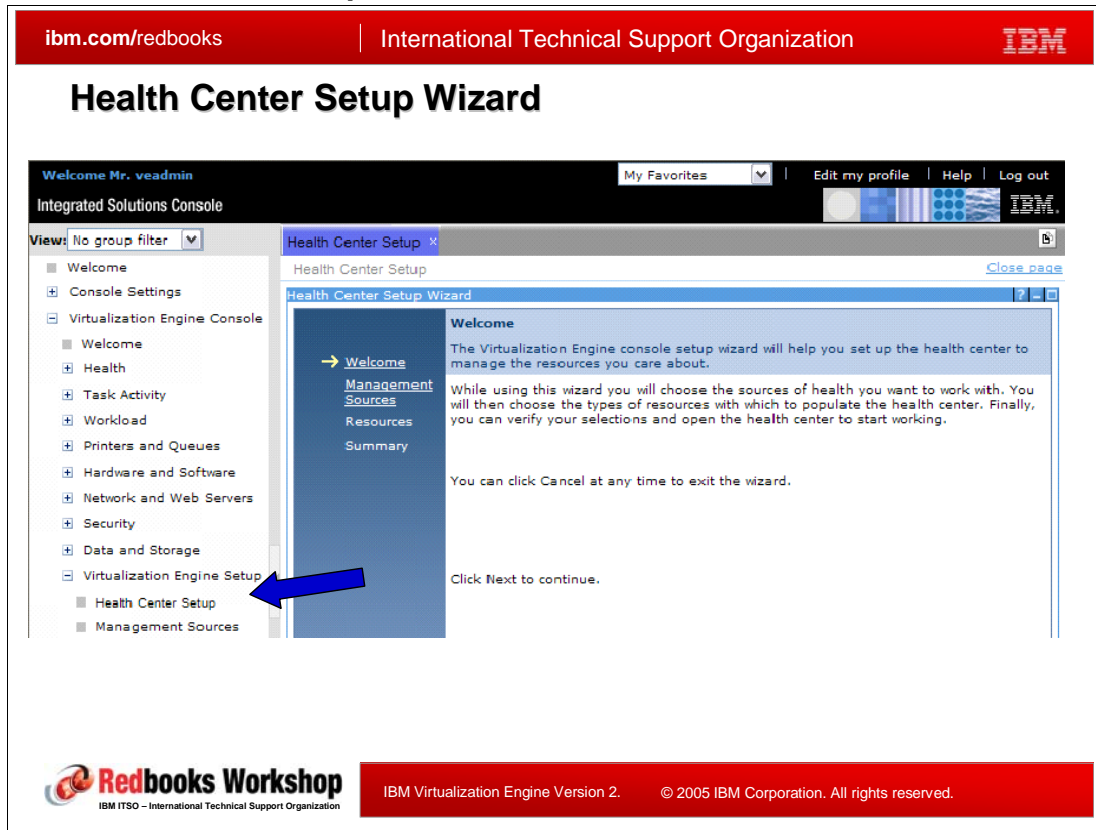


Figure 4-11 Health center Setup wizard

The health center setup wizard is new with Version 2. It basically helps you start the configuration needed for the health center.

As shown in Figure 4-11, the link to this wizard is located in the left frame under the Virtualization Engine Setup category.

The setup wizard guides you through the steps identified in the center frame.

Step 1

The first step is the usual welcome page, a presentation panel which describes the function of this setup wizard.

Step 2

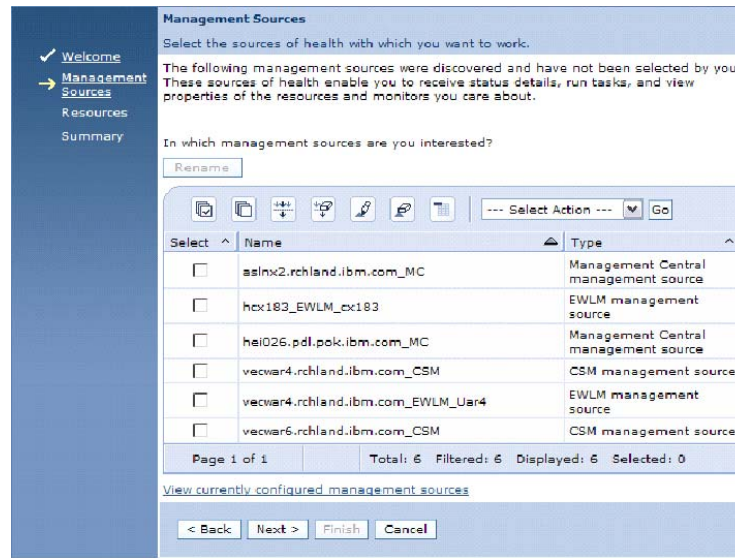


Figure 4-12 Setup Wizard - Management Sources

This step allows you to select the management sources. When you arrive on this health center wizard screen, a discovery of your network has been done and potential management sources have already been identified; they are proposed. You can then decide which of them you want to consider for the health center for this Virtualization Engine console. The possible sources of management are:

- ▶ An EWLM management source
- ▶ A CSM Management source
- ▶ An IBM Director Management source
- ▶ A Management Central management source
- ▶ An IBM Tivoli Monitoring using Tivoli Managing Environment

When identified by this auto-discovery mechanism, the name of the management sources is made of the hostname of the management server with an extension providing information on its role (that is `_DIR` for an IBM Director server, `_EWLM` for an EWLM manager, `_CSM` for a CSM manager). This name can be changed so it is easier for you to remember what this server is about. To change its name, just select a resource and click the **rename** button. You can select many lines and start the renaming operation, there will then be one form per server when you can choose the new name for this management source.

The **view currently configured management sources** link brings up a table which list already configured management sources as well as their status. From there, you can also add a new management source, with the usual menu.

Step 3

The third step, that you reach in pressing the **Next** button, will let you decide the type of resources you want to add to your health center environment. This type can be monitors or groups, a group being the aggregate of resources.

Step 4

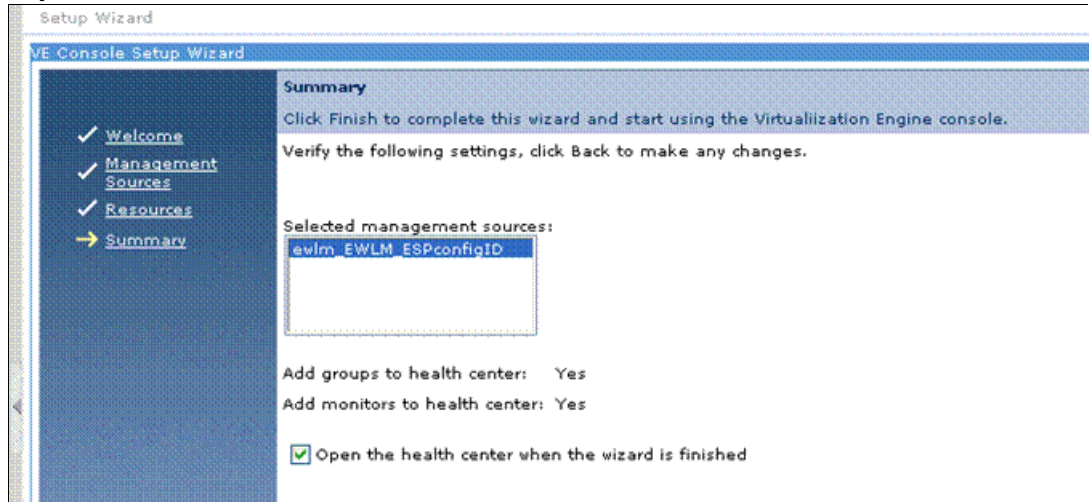


Figure 4-13 Health Center setup wizard summary

The last step is a summary of your choices, management sources, and type of resource. If you confirm these choices, just push the **finish** button and the health center will be populated. Since this operation is contacting each selected server, this last step can take some time.

You have the choice to automatically launch the health center. If selected, the result of this operation is the display of the health center.

The health center panels

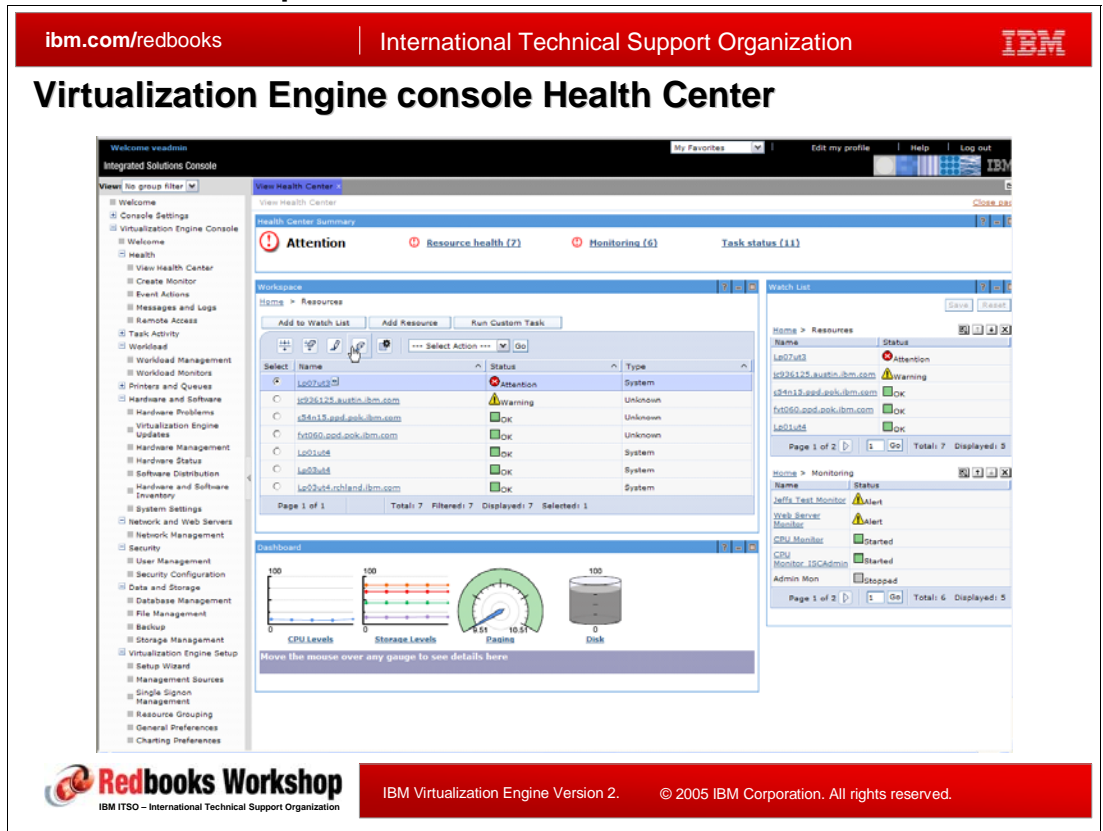


Figure 4-14 Virtualization Engine console health center

The Health Center gets its underlying information from *Management Sources* and consolidates health from a variety of management sources.

The Health Center is a *portlet-based console*: it is a “portal” which comprises a set of related “portlets.” Portlets are little pieces of the user interface which are customizable. A portlet is a web-based interface similar to a client-based dialog box or window. A set of portlets is a portal. A portal is just a single web page broken into multiple portlets or sections. A user can work with multiple portlets at a time, all available and running in the Health Center.

As the name “Health Center” implies, it gives you a way to check the pulse and other vital signs of the managed systems in your IT infrastructure using a set of pre-installed management sources such as:

- ▶ IBM Director
- ▶ EWLM)
- ▶ Cluster Systems Manager (CSM)
- ▶ Management Central (MC) for iSeries
- ▶ Tivoli Monitoring (ITM)

The Health Center checks, displays, and analyzes the status and health of resources. It displays a resource's real-time data, analyzes it using charts and graphs, and adds it to a dashboard display. It monitors resources across a distributed cross-platform environment.

The three main health center functions are the resource health, the monitors and the task status.

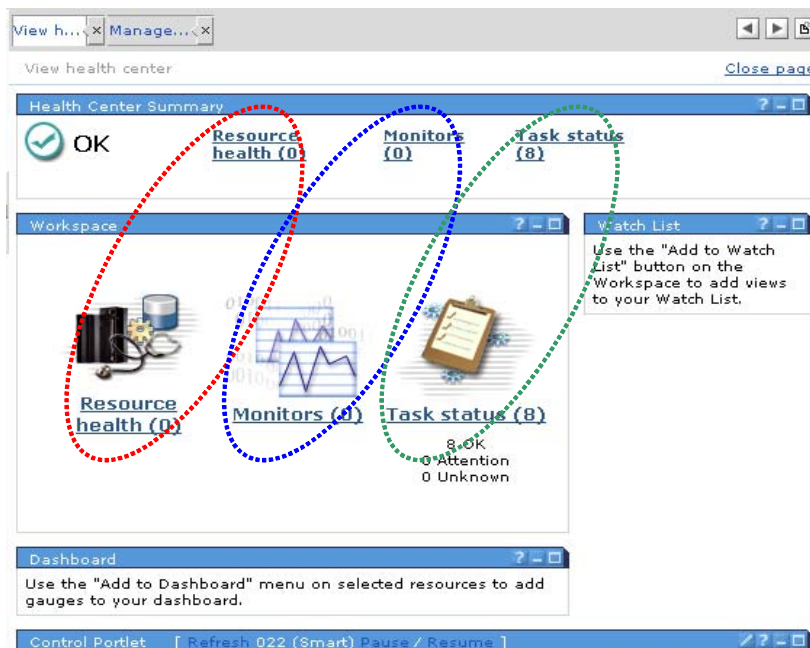


Figure 4-15 Health center functions

The Resource health

Resource health lets you look at managed systems, such as servers, devices and groups of systems. You can quickly navigate through several levels of resources and take action on a resource that may need fixing. When a user opens the Resource Health view, he will see a list of systems and groups with status icons. This list of systems is configured by the user in his preferences. The underlying data in all cases flows from the managed system (like a Director Agent running on a server) to the management source (like IBM Director), through a

Virtualization Engine console Bridge to the Health Center in the Virtualization Engine console.

For example, you could keep track of CPU utilization, events, logs or processes on any of the servers managed by multiple IBM Director management sources. Resources include BladeCenter. The Virtualization Engine Console provides status and health information for each resource.

If you open Resource Health window and click “Add Resources”, the list of resources are reported by management source. We can add/remove resources which are configured on management sources.

The Monitors

Monitors let you work with monitors you’ve set up on a management source (like IBM Director), you then will be notified when a critical event is triggered and you can view a topology of the managed systems. There are several ways to make the data from Monitors easy to view and organize:

- ▶ Real-time graphs and charts.
- ▶ Dashboard.

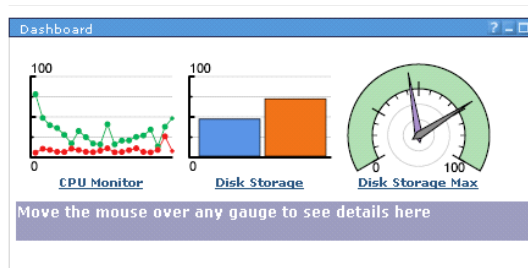


Figure 4-16 Dashboard

The Dashboard portlet in the health center displays a near real-time graphical representation of your managed systems. The metrics can be displayed in a variety of formats so you can pay attention to whatever you’re most focused on at the moment. Of course, it’s something you can customize to meet your needs. You can add and remove gauges and change their formats at will.

- ▶ Watch List.

Home > Monitors	
Name	Status
CPU Monitor	Failed
Storage Monitor	Failed
Log File Monitor	Started
Message Monitor	Started
Page 1 of 1	Total: 4 Displayed: 4

Home > Resources	
Name	Status
Rchasccon	Attention
Remote Systems	Attention
Lp03ut4	OK
Web Servers	OK
Page 1 of 1	Total: 4 Displayed: 4

Figure 4-17 Watch List

The Watch List portlet in the Health Center displays views of resources, monitors and tasks. You can add any data items that you want to monitor closely to the Watch List. The information in the Watch List persists even as you navigate away from the health center view, change to another page in the Virtualization Engine console or log off the Virtualization Engine console. The Watch List enables you to watch selected views of resource health and navigate through the resources to watch a subset of data items contained in that resource view.

The kind of resources which can be monitored depends on the management source. In order to view monitors in the Virtualization Engine console, we need to create them first on the management source. For example, we need to configure “Threshold” in IBM Director management source to monitor it from Virtualization Engine console. If you want to add monitors, click “Add Monitor”. The list of monitors configured on management source comes up. Check radio button and click “OK” to add a monitor.

In addition, we can display a graph for monitors. To get a dynamic refreshed picture, you need to install SVG plug-in our web browser to display a graph. Click the monitor name and select “Display graph” on the action box.

We can configure Dashboard and Watch List for a specified device. Dashboard provides a rough image to monitor resources visually. On the other hand, the watchlist is a shortcut for any resource and monitor. Once we register a resource or a monitor in the watchlist, we can check each status from the top page. If you want to add a resource or monitor in Watch List, click “Add Watchlist” button. If you want to add a monitor in dashboard, look for the “Add to Dashboard” action on the action box.

Task Status

Task Status will let you run simple or customized tasks and check their status. The Health Center lets you perform simple actions like starting and stopping resources and monitors. But you can also use custom tasks to run more complex actions against your resources and monitors. Custom tasks are predefined on the management source; to run a custom task

from the Health Center, you need to configure that task on the management source first (e.g., IBM Director or iSeries Navigator).

The bridges

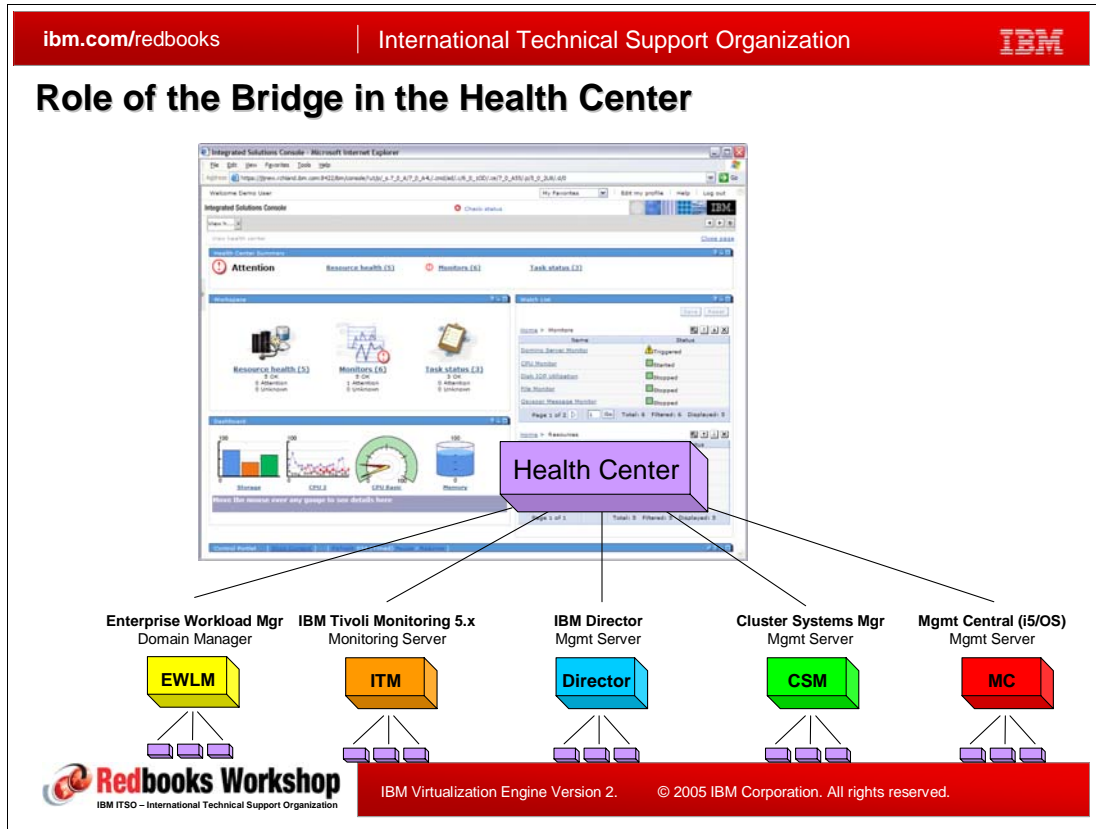


Figure 4-18 Role of the bridges in the health center

The magic behind the health center resides in Virtualization Engine console *bridges*.

A Virtualization Engine console bridge is the piece of software that allows the Virtualization Engine console to communicate with a management source. In most cases, the management source has an associated Virtualization Engine console bridge that communicates via web services with the Virtualization Engine console system.

There are four bridges supported in the Release 2.

- ▶ For the Management Central management source, the Management Central bridge to Virtualization Engine console must be on the same system as the Management Central management source.
- ▶ For the Cluster Systems Management (CSM) management source, the Cluster Systems Management bridge to Virtualization Engine console must be on the same system as the CSM management source.
- ▶ For the Enterprise Workload Management (EWLM) management source, the Enterprise Workload Management bridge to Virtualization Engine console must be on the same system as the Enterprise Workload management source.
- ▶ For the IBM Director management source, the IBM Director bridge to Virtualization Engine console may be the same system as the IBM Director management source or the IBM Director bridge to Virtualization Engine console bridge may be on a different system than

the IBM Director management source. If the IBM Director bridge is not on the same system as the IBM Director server, further security setup (for Single Sign On) is required (documented in the InfoCenter).

The code for the Virtualization Engine console bridges runs within on a WebSphere application server profile that communicates to a management source. When the console sends a request to one of its bridges, it passes along the lightweight third party authentication (LTPA) key associated with the session.

Linking with Tivoli Manager

The connection to the IBM Tivoli Manager is a bit different since there is no Virtualization Engine console bridge. It interfaces directly with the Virtualization Engine console health center. The Virtualization Engine console health center will communicate to the management source via a Tivoli Management Environment (TME) framework client that is packaged with the Virtualization Engine console.

In terms of security, the connection to the IBM Tivoli Monitoring management source works slightly differently. The TME framework client, running in the Virtualization Engine console, retrieves the user profile login ID and password stored in the portal server credential vault packaged within the IBM Integrated Solutions Console. The TME framework client connects to the ITM management server through the TME framework to validate the user profile login ID and password.

The Virtualization Engine console bridge to Tivoli Enterprise Management Server and the Tivoli Enterprise Server

The Tivoli Enterprise Portal Server (TEP Server) and the Tivoli Enterprise Management Server (TEM Server) were previously titled Candle Net Portal Server (CNPS) and Candle Management Server (CMS).

This new Virtualization Engine console bridge to TEP/TEM, servers is specifically being implemented to exploit z/OS systems and sysplexes managed through the TEP Server, the TEM Server and the Omegamon z/OS Management Console Agent (z/MC Agent). The TEP Server, the TEM Server and the z/MC Agent are currently available within IBM Tivoli Monitor 6.1 (ITM 6.1).

The Virtualization Engine console bridge to TEP Server will retrieve the z/OS resources, monitors and tasks from the TEP Server for integration into the Virtualization Engine console health center function:

- ▶ The z/OS systems and sysplexes being managed by the TEP Server through the Omegamon z/MC Agent, will be displayed as resources within the health center.
- ▶ Similarly, Omegamon z/MC Agent situations monitored by the TEP Server, will be available as monitors within the health center.
- ▶ Omegamon z/MC Commands, available within the TEP Server, will be available in the tasks and health center resource action panels.

Figure 4-19 describes the implementation of this new bridge.

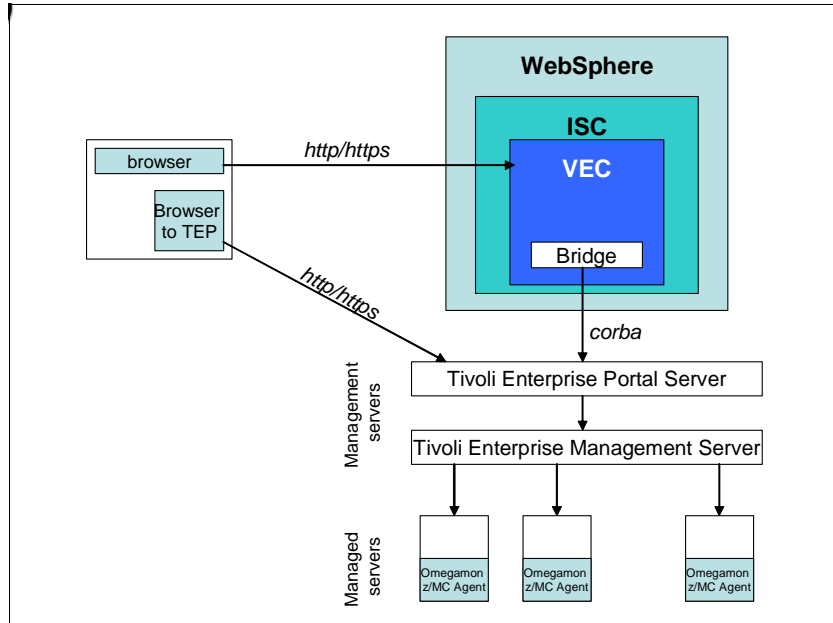


Figure 4-19 TEP/TEM bridge

At the time of writing, this new bridge is expected to be available in 1Q2006¹.

¹ IBM plans are subject to change without notice and this information represent goals and objectives only. Such information is not intended as a definitive statement of a commitment to specific levels of function or delivery schedules.

4.6 Interacting with the environment

When comes the time to issue a command against one of the virtualized components, the Virtualization Engine console propose three ways to interact with the environment:

- ▶ Portlets
- ▶ Launchpad
- ▶ Launch-in-context

Portlets

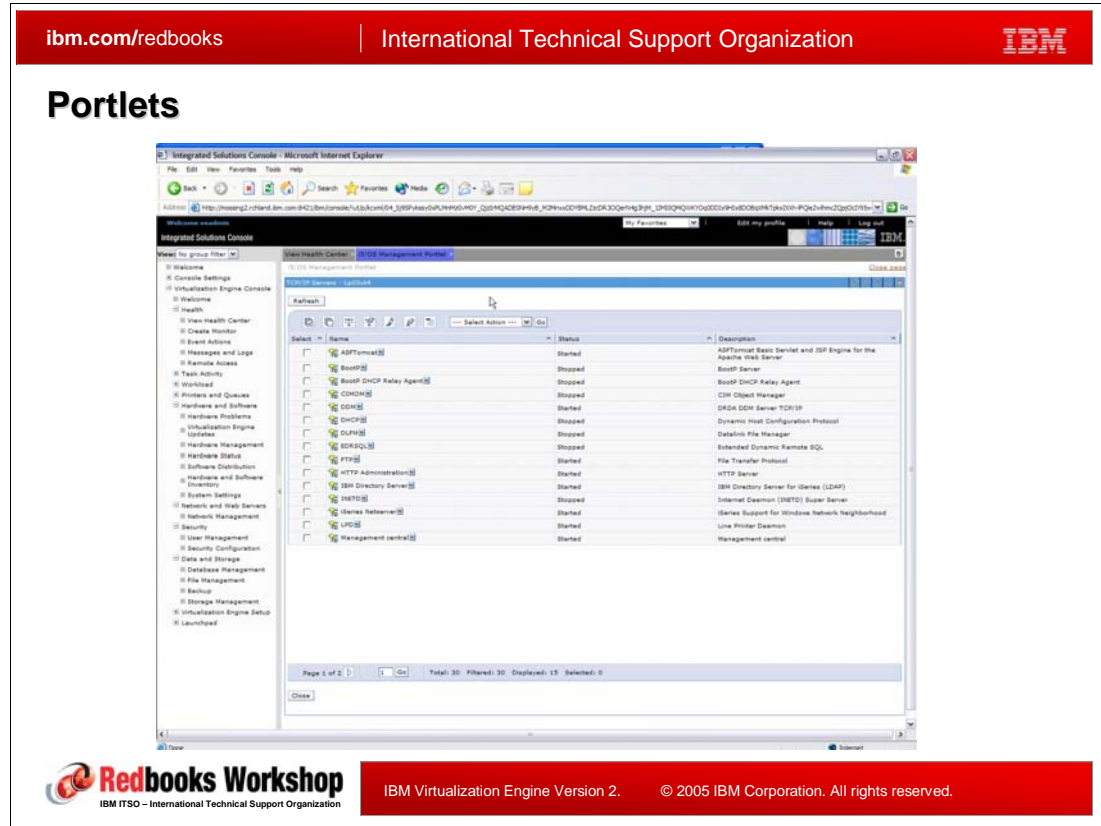


Figure 4-20 Portlets

A portlet is a Web-based interface similar to a client-based dialog box or window. Portlets are components of a portal; users can work within multiple portlets at the same time, all available and running on a single page in the health center. The health center consists of the following working areas:

- ▶ Health center summary
- ▶ Workspace
- ▶ Watch list
- ▶ Dashboard

Additional portlets are included in the wizards and tasks that can be started from the Virtualization Engine console.

The example in Figure 4-20 shows within a portlet the TCP/IP daemons and their status for an i5/OS environment.

Launchpad

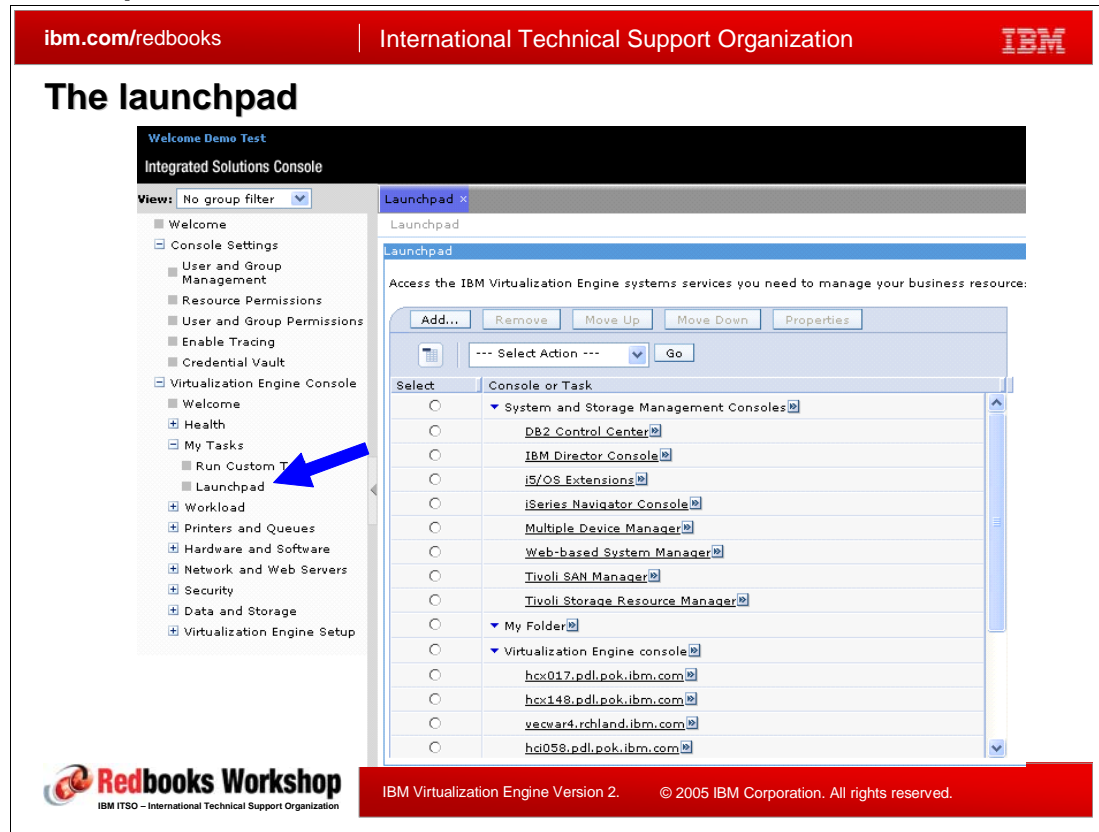


Figure 4-21 The launchpad

The Virtualization Engine console launchpad provides a single location to access all the Virtualization Engine interfaces. It can be started from the left frame under the **My tasks** category.

The Virtualization Engine console automatically detects the virtualization engine web consoles in the environment, such as the EWLM Control Center, and automatically creates links for them.

By default, the following products, if installed in the infrastructure, can have their consoles linked in the launchpad:

- ▶ DB2 control center
- ▶ IBM Director
- ▶ i5/OS extensions
- ▶ iSeries Navigator
- ▶ Multiple Device Manager
- ▶ Web-based System Manager (including the Hardware Management Console)
- ▶ Tivoli SAN Manager
- ▶ Tivoli Storage Resource Manager
- ▶ Launch-in-context

You can also decide to manually add additional consoles that you use for the infrastructure administration. Any web-based console, that you can describe by its URL, can be added to the launchpad.

A new usability addition included in Version 2 is the possibility for you to reorder the listing and let you put on top the console that you most frequently used by example. Click on a console title and select the **move up** or **move down** button to reorder the list.

Launch-in-context

Launch-in-Context

- ❑ Console utility to launch specific-tasks in other consoles.
- ❑ Launches the most-used functions from
 - IBM Director Console
 - WebSM Console
 - iSeries Navigator Console
- ❑ Launching is done from within the flow of a VME Console scenario.
- ❑ Launch-in-Context is a tool for:
 - Coping with functions that have not yet migrated to the ISC
 - Leveraging function that will never migrate to the ISC
- ❑ Customers prefer to have access to an entire scenario, even if it launches another console within its flow.

Figure 4-22 Launch-in-Context

The launch-in-context concept, introduced in version 2, is a way to efficiently run tasks that are effectively being executed by another application without disrupting the apparent scenario for the user.

In the example given in Figure 4-22, we have the case where from the health center (top screenshot, **A**), we find out that an action is necessary on an iSeries system.

- ▶ The right click on the system name opens a contextual menu including all possible actions for this systems.

Let say you want to run a remote session on this iSeries, this task can be launched in the IBM Director console from the Virtualization Engine.

- ▶ A popup window, such as shown, will appear letting the user know what is happening and the real IBM Director console will appear pointing directly positioned on the menu from which the remote session can be started. Had an IBM director console been opened already, this window would have been reused to display the appropriate menu. Once the operation accomplished, the users transparently go back to the place where he left in the Virtualization Engine console and can continue the action they had started.

The majority of tasks available in Virtualization Engine Release 2 are Launch-in-Context tasks that exploit function within:

- ▶ IBM Director

- ▶ iSeries Navigator
- ▶ Web-based System Manager

The long term view for the Virtualization Engine console is to become the only administrative console for the whole infrastructure, and the launch-in-context is a way to launch tasks in consoles that have not yet migrated to the Virtualization Engine console or for other that will never be migrated.

Also being able to keep the same controlling tool during the whole scenario, from the warning detection, to the troubleshooting operation and up to the corrective action, reduces the time between the time a problem arises and the moment when everything is back to normal.

Launch-in-Context is built on top of Java Network Launching Protocol (JNLP). JNLP is defined as a mechanism for transferring a server-resident application over a network and across firewalls, and launching it on a client. JNLP is the protocol utilized by JNLP clients such as Java Web Start (JWS).

JWS is a technology for simplifying deployment of Java applications by allowing users to launch full-featured applications with a single click from a Web browser without going through complicated installation procedures. The JNLP protocol provides a means to securely transfer a Java application from a server to the client. JWS is bundled with J2SE starting with version 1.4.

JWS existence on the client can be detected with the use of Javascript; the browser can determine whether JWS is installed on the client. If JWS is not installed on the client, the user will be presented with a warning message indicating the need for JWS in order to perform certain functions within the Virtualization Engine console. The Virtualization Engine launchpad addresses this issue in this release by presenting a warning message the need of JWS along with a hyperlink pointing to Sun's Java Web Start product page in order to download a JRE that contains JWS.

A task consists of a series or map of key/value pairs that make up its actual context. This map communicates everything that the client console needs to know in order to launch a respective task. The Management Source predefines all of the available tasks for each system. In the Virtualization Engine console, the user selects the Remote Control task—but the key/value pairs that make up the task were already defined. The user's selection causes the Remote Control key/value map to be sent down to the Director console via JNLP.

Additional Console

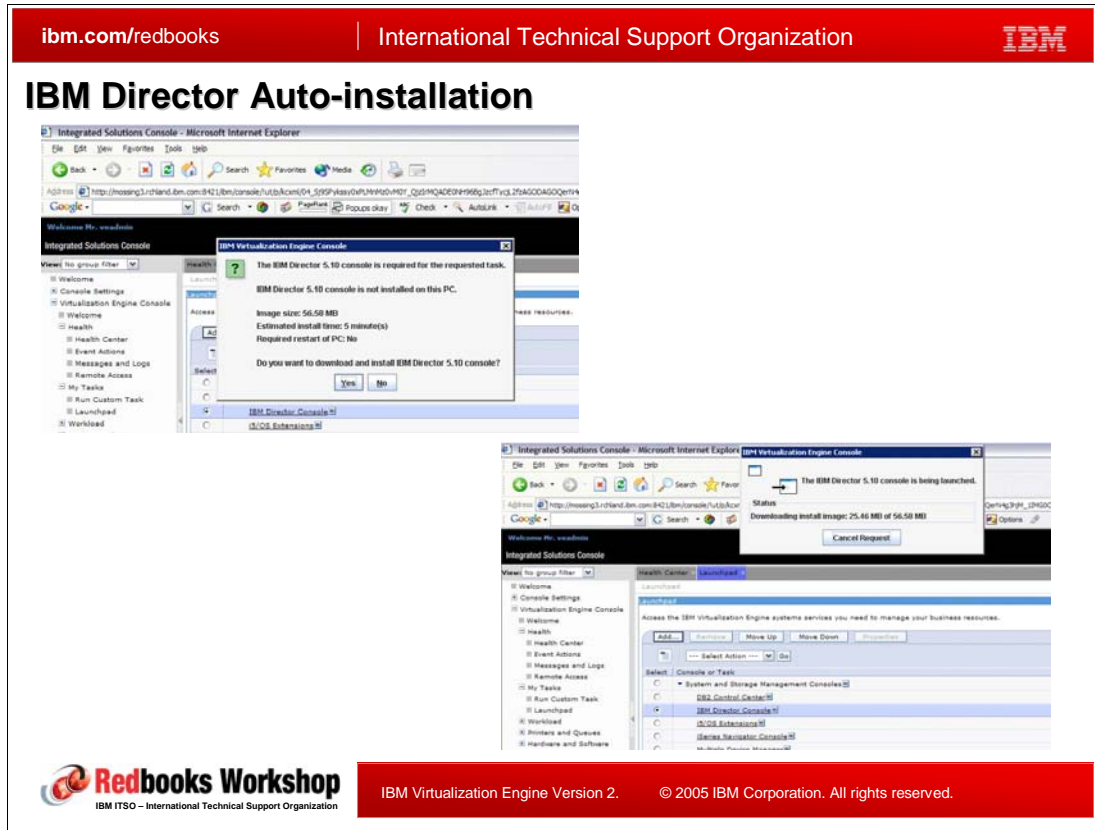


Figure 4-23 IBM Director - Auto Installation

Since a problem can arise at any time, an additional functionality of the Launch-in-context can speed up even more the resolution of a problem. The Virtualization Engine console can be started from any navigator on the network, so it is possible for an administrator to start the console from a workstation on which no additional consoles have ever been installed. During the investigation process, if it appears that some of the tasks required must be started from an external application, the launch-in-context will automatically propose the installation of the additional console. It will pop up a window asking whether you agree for the installation of the console on this system, download the necessary program from the Virtualization Engine console host machine, and assist you in the remaining installation. Once the additional console is installed, the launch-in-context goes on with the required task.

Such installation is available for the Web-based System Manager or the IBM Director, but not for the iSeries Navigator.

4.7 The topology view

The Resource Dependency Service, described in Chapter 3, “Resource Dependency Service” on page 97, is a new component of the Virtualization Engine Version 2, that lets you discover resources on your network and creates relationship between this resources.

The topology view included in the Virtualization Engine console is the display of the information gathered by Resource Dependency Service. Note that some topology can be seen even if Resource Dependency Service is not installed, based on what’s installed in the customer’s environment.

Launching the topology view

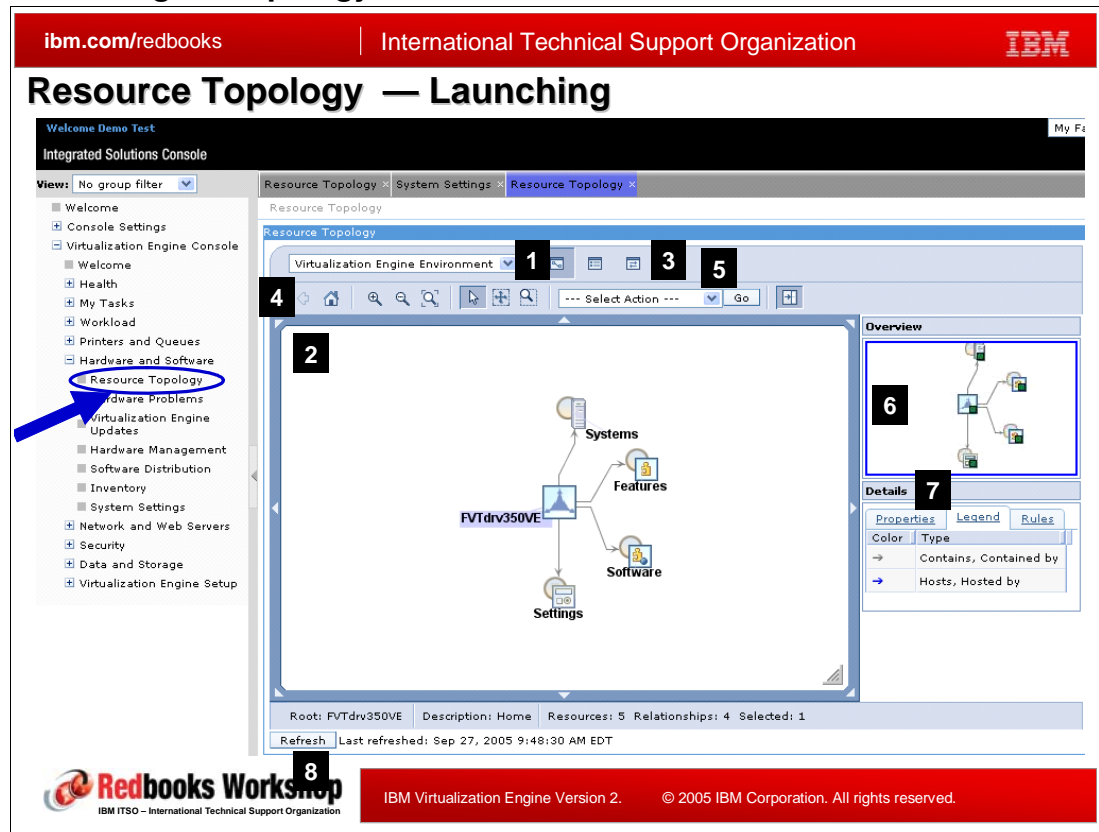


Figure 4-24 resource Topology launching

The access to the topology is located in the left frame under the **Hardware and Software** item, as shown on the Figure 4-24. This link brings the window on the right made of various default panels:

1. Environment selection combo box: the place where you select the environment you want to explore. The choice is today between the systems included in your virtualized engine component and those discovered by Resource Dependency Service. More about this is described in , “Selecting the data source” on page 163.
2. Workspace area: the place where you selection is being displayed and from where you can drill down, and get detailed information about your infrastructure.
3. Toggle view button: let you choose between icon, resource table, or relationship table view for the workspace area.
4. navigation area: let you navigate in the workspace area. More about this in the next section.

5. Action scroll list: let you execute action on a selected object.
6. Overview window: Let you view what part of the complete environment is being displayed in the workspace area.
7. Details table: Display information on the selected object and the workspace area.
 - The **Properties** tab contains at least the name, type, description of the selected object.
 - The **Legend** tab provides information on the symbols utilized on the workspace area.
 - The **Rules** tab lists the rules used to establish relationships between objects displayed in the workspace area.
8. Refresh button: let you know when the data have been retrieved and let you force a new exploration of your environment.

Selecting the data source

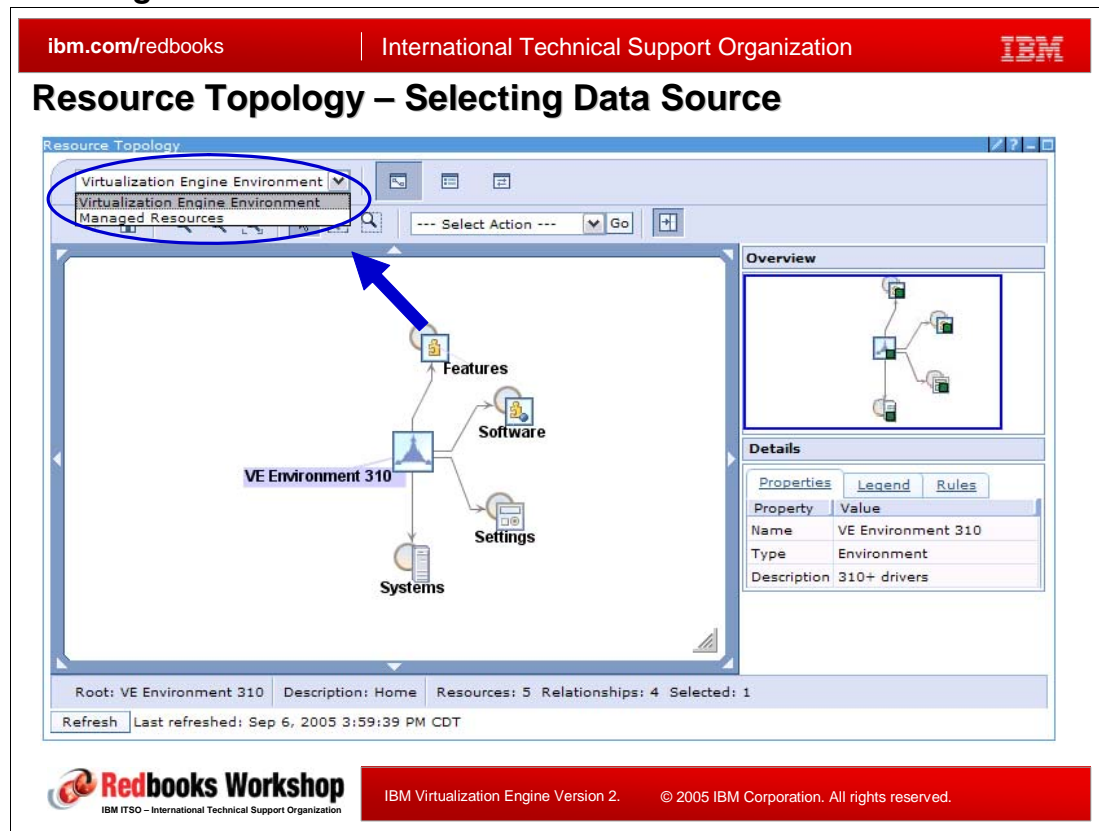


Figure 4-25 Resource Topology - Selecting Data Source

When you start the topology view, the first choice you have to make is to select which environment you want to explore. The choice is between the **Virtualization Engine Environment**, which includes all the management servers and their resources, or the **Managed Resources**, which includes all the systems and resources gathered by Resource Dependency Service.

In the example shown in Figure 4-25, we have chosen to explore the virtualization engine environment. The starting point is represented by the icon **Virtualization Engine Environment 310**. As you can see, this environment contains:

- ▶ **Systems:** as expected, if you choose the **Open** action on this icon, the display is being changed to represent all the systems part of our environment. You can then drill down on

each icon, to find out which components of the Virtualization Engine is being installed on the system, down to the websphere application being deployed on the system.

- ▶ **Software:** Similarly, if opening the software, you will be presented with a list of the software installed in our environment (can be DB2, WebSphere Application Server, ISC, EWLM, Resource Dependency Service..), and a drill down let you find out what systems and where this software is being installed on.
- ▶ **Settings:** let you explore the settings for the components in your environment. In our example exploring the settings icon would lead to second screen including Virtualization Engine security, WAS security, and SSoConfig. Opening the Virtualization Engine console EWLM icon let you change the reportDataInterval settings, while exploring the WAS security icon let me find out that the Listening port for the LDAP server is being set at 389 (its default value).
- ▶ **Features:** displays all the features in the environment, a feature could be:
 - a web service list such as the bridges to the Virtualization Engine console
 - a web application list such as the virtualization engine console itself
 - a software list such a WebSphere Application Server or DB2

Creating resources and relationships

ibm.com/redbooks
International Technical Support Organization

Managed Resources — Creating Resources

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

IBM ITSO - International Technical Support Organization

Figure 4-26 Managed Resources - Creating resources

When you select the **Managed Resources** environment, the same screen and action described previously apply. The major difference is that the topology view is extracting this information from the Resource Dependency Service database.

It also allows an additional functionality which is the creation of additional resources and relationships. During the discovery process described in chapter XXX, Resource

Dependency Service will populate this database, however it is possible that you want your own customized entry.

We can take an example, where the Resource Dependency Service discovery mechanism has detected a server but couldn't find out what its operating system was (being an unsupported version). you can decide to enter this information manually. For that, you have to unselect all the objects, start a **New Resource** action, and select the **Operating System** item, this will open a form where you enter the name and various properties (such as name, description, version, vendor and so on) for this operating system. select the **OK** button, and your new resource is being added to your topology and to the Resource Dependency Service database.

Of course, it is also possible to delete resources. For that, select the desired resources and apply the **Delete** action. Once confirmed, these resources will be permanently destroyed.

Resources that you can manually add include:

- ▶ Application
- ▶ Data container
- ▶ Database instance
- ▶ Group
- ▶ Hosting container
- ▶ Network device
- ▶ Network interface
- ▶ Network subnet
- ▶ Operating system
- ▶ Service
- ▶ Storage volume
- ▶ System

To manually create a relationship between two resources, the process is easy, select two resources in the workspace area, and execute the **New Relationship** action.

In the previous figures, we have chosen the graphical representation of the topology, since it is easier to understand the relationship between resources when it is materialized by a drawn link. however, it might be sometimes more practical to display the same information in a spreadsheet format. That is what is shown in Figure 4-26, the resources are being displayed, and the little arrow button behind resources names indicate that you can either open the properties menu for this resource or drill down to the next level.

The same table view is also available for the relationship view. For each selected object, the table lists the relationships link it with the other resources in the environment.

This table views can also be used to export the data to an external tool. From this table, copy the contents of the table, and paste in a spreadsheet for example.

Defining a logical group

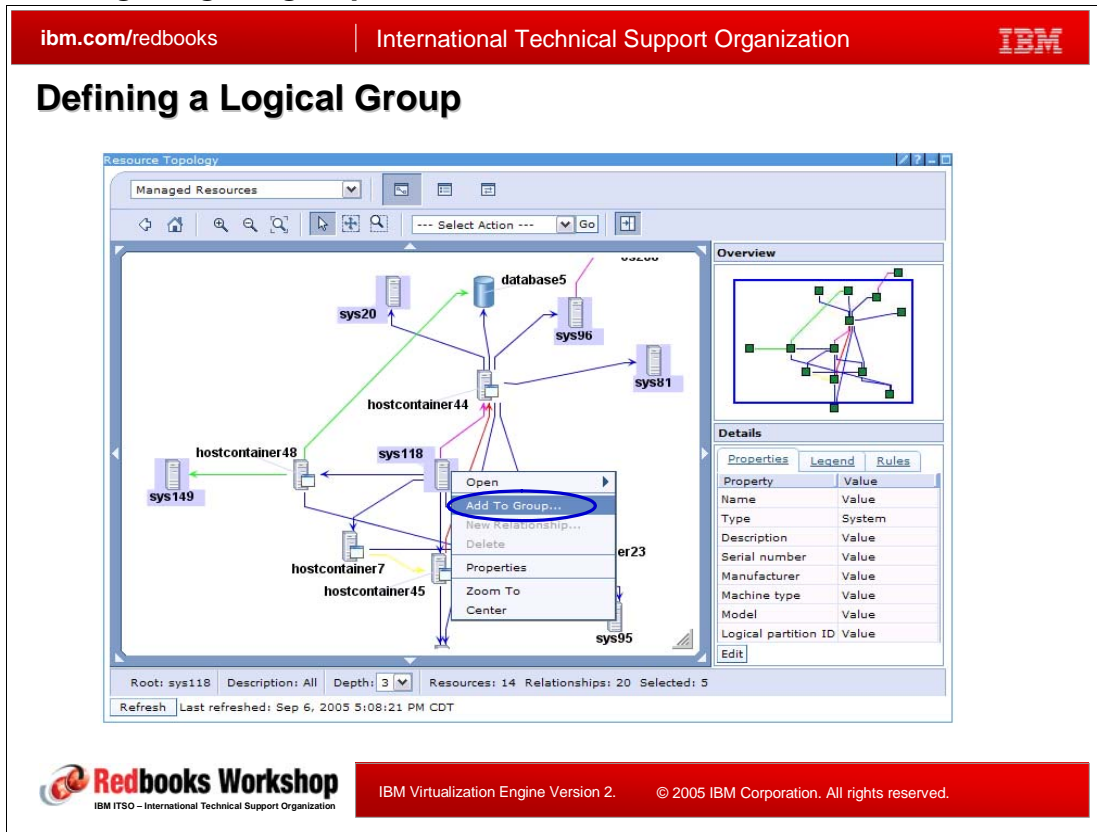


Figure 4-27 Defining a logical group

We have seen before how to create relationships between resources. To extend this notion, it is also possible to create logical groups. Let's suppose that several systems represented in a graph, although not having any relationship (to the Resource Dependency Service point of view) but are still part of the same business process, say payroll. It might be interesting to link those together, and that is what logical group has been created for.

From now, if you detect any problem with a resource, you can easily detect which business process it belongs to and which incidence can this problem have.

Navigating in the topology view

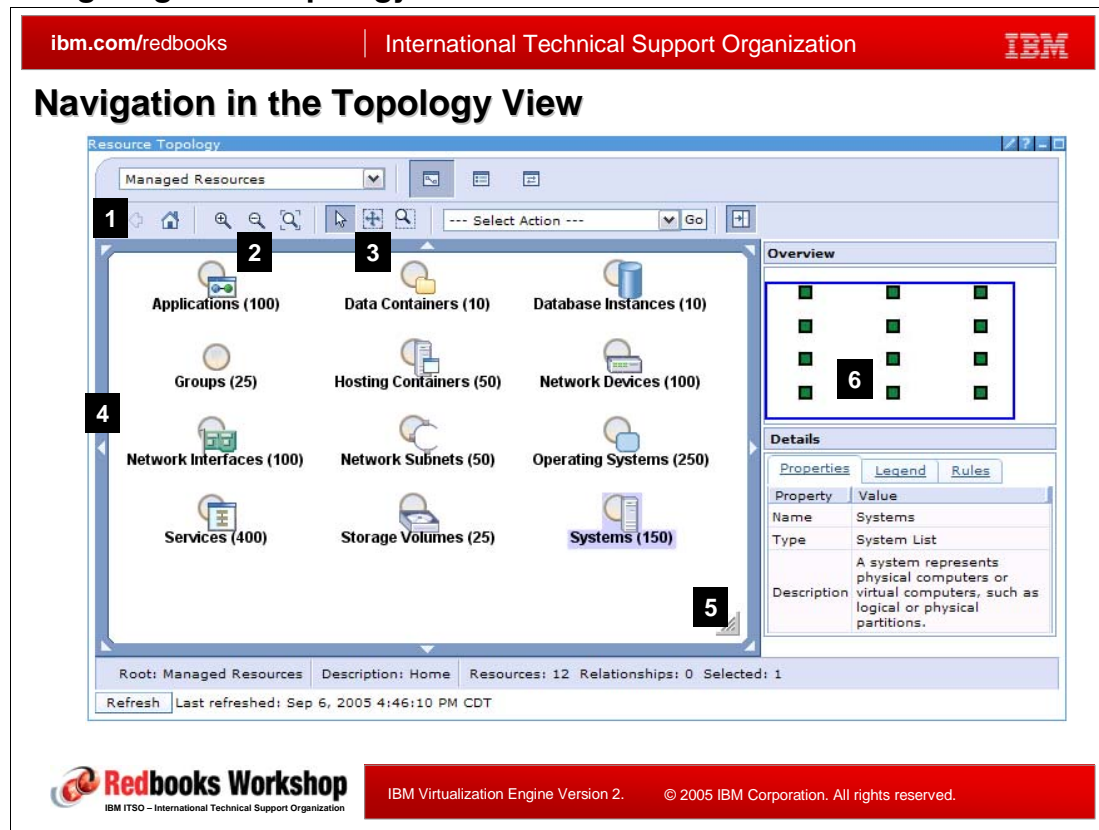


Figure 4-28 Navigation in the Topology View

In a complex environment, the numbers of resources and relationships to display can be extremely high. The example shown in Figure 4-28, includes more than 1200 resources, for each icon the number in parenthesis give the number of occurrences of each resources types.

Clearly, 1200 icons can not be displayed in a window without ways to manage this complexity :

1. Navigation bar: it includes two buttons
 - The **Go back one page** arrow let you return to the previous screen
 - The **Home** button returns you to the starting point
2. Zooming bar: let you zoom in, zoom out, or resize the contents to fit the viewport.
3. Pointer selection bar: let you switch between the default pointer, the moving area pointer, or the cropping pointer.
4. handles: the four handles let you move the displayed contents within the workspace area.
5. resizing handles: let you modify the size of the workspace area
6. overview window: let you understand what part of the complete infrastructure is being displayed in the workspace area.

Scheduling Resource Dependency Service

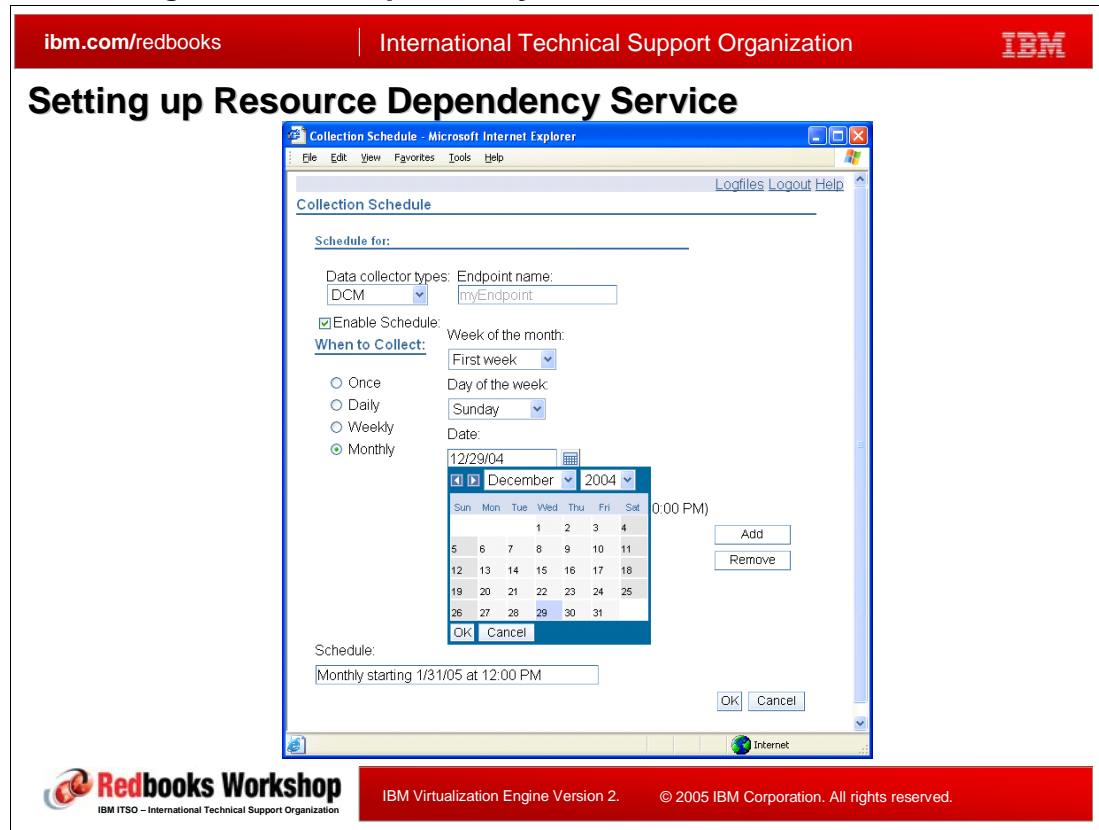


Figure 4-29 Setting up Resource Dependency Service

The window in Figure 4-29 shows the various parameters that can be modified to configure the Resource Dependency Service data collections:

- ▶ The data collector type and, if adequate, the name of the endpoint to start the collection
- ▶ The dates and time at which the collection must take place. You can select the frequency (between once and monthly), the week of the month, the day of the week, or specify on the calendar which day to start. Your choice is summarized on the last line, you have to press the **OK** button to confirm.

4.8 The Virtualization Engine console under the hood

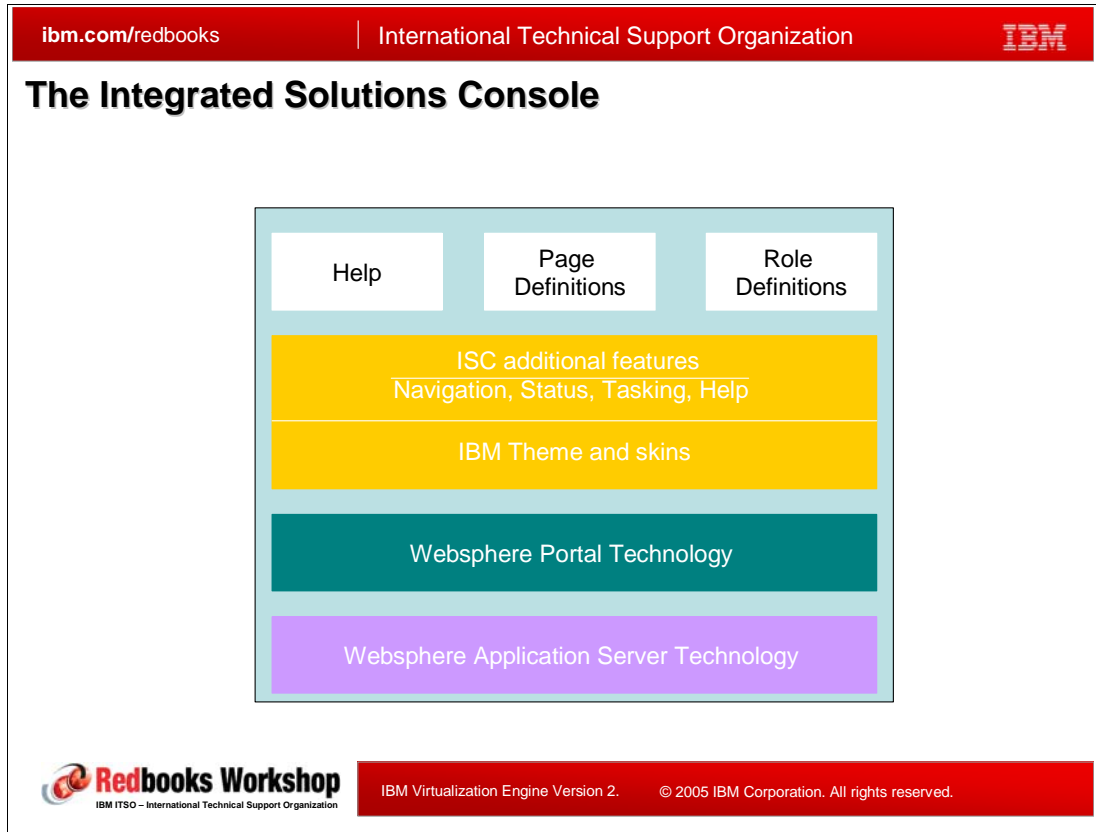


Figure 4-30 The Integrated Solutions Console

The Integrated Solutions Console

The Virtualization Engine console is based on the Integrated Solutions Console (ISC) work part of the Autonomic Computing initiative.

The primary goal of the ISC is to provide a single platform that can host all the administrative console functions built by IBM server, software, and storage products in a manner that allows customers to manage solutions rather than specific IBM products. Administrative console functions range from setup and configuration to solution runtime monitoring and control.

The role of the ISC being to display many customized elements of information, navigation menus, or help screen, an obvious choice is to use a portal technology. The ISC is being built on top of the Websphere Portal technology, itself using the Websphere Application Server technology. This version 2 of the Virtualization Engine console is based on the version 6.0.2 of the ISC requiring the same level of the WebSphere Application Server.

The second important point here is standards, if the ISC is to host every IBM products but also third party vendor consoles, the development and integration of such consoles within the ISC must be based on standards. As a matter of fact, the ISC is based on the following industry standards:

- ▶ J2EE
- ▶ Java
- ▶ XML
- ▶ Eclipse

- ▶ JSR168 portlet API

Since the ISC code base is based on WebSphere infrastructure, the ISC will make use of future functional enhancements and standards in the application server and portal server technology, as it has been done in this version going from WebSphere Application Server 5.0 to WebSphere Application Server 6.0. These features will be exploited to achieve tighter integration and increased consistency.

4.9 Summary

ibm.com/redbooks
International Technical Support Organization

Virtualization Engine console User Experience Themes

- ❑ **Health of my environment**
 - > Health center
 - > Hardware status and problems
- ❑ **Relationships between my resources**
 - > Resource topology
 - > Understanding resource dependencies
- ❑ **Apply my expertise to new areas**
 - > Converged tasks
 - > Launch in context
- ❑ **Drill deep into OS management**
 - > OS extensions for i5/OS
 - > Launch in context for other OS's
- ❑ **Keep Virtualization Engine environment running smoothly with minimal effort**
 - > Software Updates (for the Virtualization Engine)
 - > Virtualization Readiness on welcome page
 - > Setup wizard, RDS configuration
 - > Single Sign-on mgmt

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 4-31 Virtualization Engine console User experiences themes

This version 2 of the Virtualization Engine includes the second edition of the Virtualization Engine console. It enhances the functionalities included in the previous version.

- ▶ The first function already present in release 1 is the health center. It let the administrators find out on our single display whether the systems they to look after are operating correctly or if any trouble is arising. It starts with a single icon turning from green to red as soon as a system in the infrastructure is being detected as malfunctioning. The administrators are then guided up to the point where they discover the origin of the alert.
- ▶ The second function, new in version 2, is the business resource topology view exploiting and displaying the data collected by the Resource Dependency Service (RDS). This product will find the resources on the network and establish relationships between those different resources. If we go on with our previous example, the administrators, having found the source of the problem, can from the topology view determine which resources will be missing if the server stops and which line of business will be affected.
- ▶ If additional troubleshooting is to be performed on the failing server, the Virtualization Engine console provides guidance to launch additional investigation tools, letting operators performing basic tasks to help analyze the problem even on operating systems they are not familiar with.
- ▶ The next step might be to call the expert administrators on board, from the Virtualization Engine console, they can either interact with the systems directly from the console (for example for the OS extensions for I5/OS) or launch the full blown administrative consoles, such as the IBM Director console, or the Web-based system manager for AIX systems). The new launch-in-context functionality will let them automatically download and install the

necessary tools on the local systems, there is no need anymore for the administrators to go/run back to their desk to start a corrective action.

- ▶ This version of the Virtualization Engine console includes now a welcome page that let the users find out whether their virtualized environment setup is complete, or will help them in accomplishing the last needed steps.
- ▶ New also is the health setup wizard which guides you through the configuration of the health center by automatically discovering the management servers in your infrastructure as well as the resources and monitors that can enter the health center.
- ▶ The last point is the new Virtualization Engine updates strategy being adopted in this version. Once properly configured, the console will automatically compare the installed level of all the components part of the virtualization engine environment to the latest level available from IBM. If instructed, it will automatically download new updates required by the infrastructure, and the administrator will just have to choose when they want the console to distribute and install these updates on the servers running the virtualization engine functions, making the maintenance of the environment really easy.



5

Enterprise Workload Manager

This chapter contains the technical introduction of Enterprise Workload Manager (EWLM), a Virtualization Engine component. EWLM is a product which can dynamically monitor and optimize distributed heterogeneous workloads to achieve user-defined business goals.

In this chapter, the following topics are described:

- ▶ EWLM overview
- ▶ ARM instrumentation and IBM products
- ▶ EWLM domain policy
- ▶ EWLM partition management
- ▶ EWLM load balancing

5.1 Enterprise Workload Manager overview

ibm.com/redbooks
International Technical Support Organization

EWLM overview

- ❑ 'Virtualization Engine Enterprise Workload Manager'
- ❑ A performance management tool
- ❑ Supports total business environment consisted of heterogeneous servers
- ❑ Monitoring and automatic optimization of servers

Platforms/Middleware

- Windows
- Linux / CKRM (New in V2)
- AIX
- i5/OS
- Solaris
- z/OS
- HP-UX (New in V2)
- WebSphere
- DB2 UDB
- Apache
- IIS

Resource Optimization:

- Algorithms engine
- P5/i5 LPAR CPU management
- Load balancing
- TIO integration possibility

IBM Virtualization Engine Version 2.
© 2005 IBM Corporation. All rights reserved.

Figure 5-1 EWLM overview

Enterprise Workload Manager (EWLM) is a product, a component of the IBM Virtualization Engine which can monitor and dynamically manage and optimize distributed heterogeneous workloads to achieve user-defined business goals.

EWLM is a product for performance management which supports heterogeneous environments consisting of multiple types of servers, multiple types of middlewares, and multiple operating systems.

EWLM can monitor the performance of each server, each process, each application as well as the entire end-to-end infrastructure. An application environment is usually made of multiple servers which contain middlewares such as http server, application server, or database. When a work request arrives at the first entry point server, it is processed by the first middleware, then the request is passed to the next server or middleware, then passed to forth and back, and after all, the request leaves the system as a response to the requester. The end-to-end response time which EWLM supports as the performance goal, means the total elapsed time from the request arrival to the system, to the response departure from the system. The mechanism which enables the response time measurement of each server is possible through the Application Response Measurement instrumentation (ARM).

In addition to response time, EWLM supports another performance goal which is called velocity goal. Velocity goal means such as processing speed of a batch job or a daemon process, or the performance of an LPAR partition in a server.

EWLM monitors and collects the performance data of a system. Then the next step is to optimize the system using the collected performance data. EWLM has three major performance management facilities:

- ▶ One is the load balancing of servers which is obtained by sending commands to load balancer hardware which is in front of the system.
- ▶ The automatic assignment of system resource of each LPAR partition is the another facility. At the writing time, automatic CPU allocation of IBM eServer i5/p5 systems is supported.
- ▶ EWLM performance management can also be integrated with TIO (Tivoli Intelligent Orchestrator).

5.2 Enterprise Workload Manager terminology

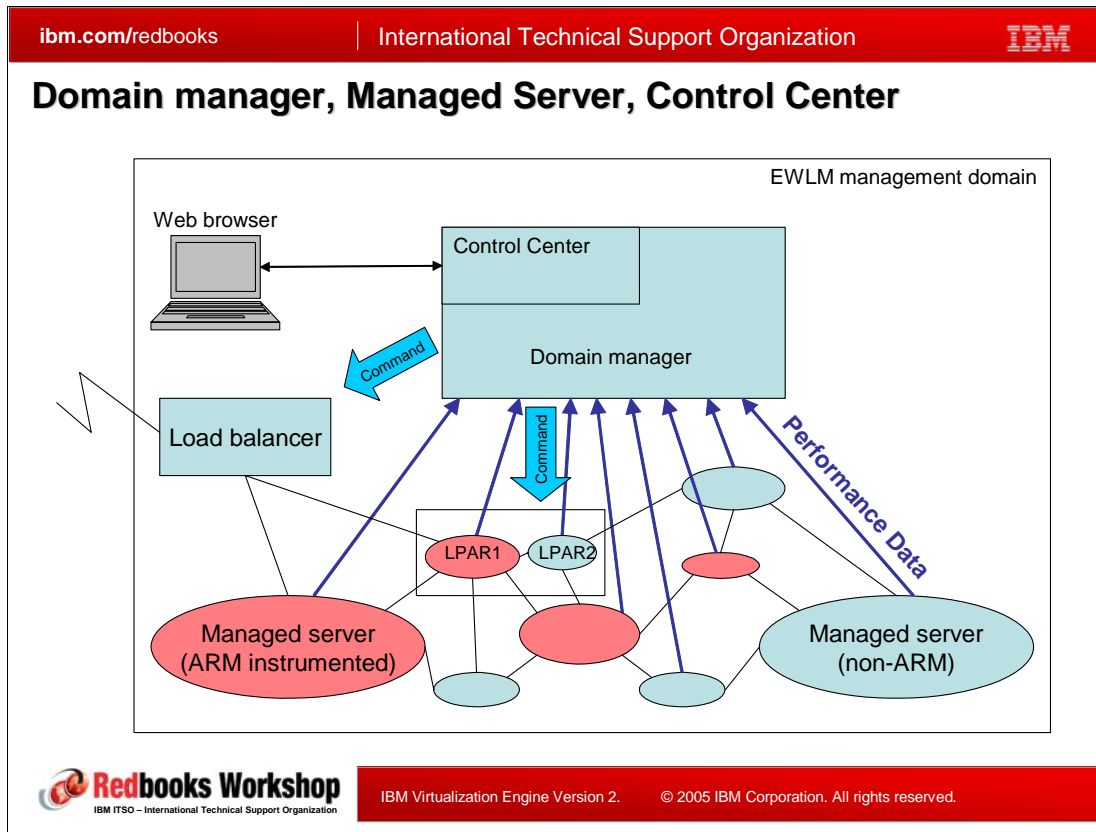


Figure 5-2 EWLM terminology

Management domain

EWLM manages servers as a logical group called EWLM management domain. A management domain is the unit of EWLM performance management where any monitoring reporting and optimizing activity of EWLM is done. Therefore, any logically related servers such as the same application, same group of users, or all servers in a system environment, should consist in one EWLM management domain. For example it would make sense that all the servers which consist in a two tier or three tier application are located in the same management domain.

A management domain is controlled by a EWLM server facility called EWLM *domain manager*. The domain manager manages and controls the all servers in the management domain. It gathers the performance data from each servers, reports the information, and can send load balancing commands to load balancers or LPAR partitions.

Control center

The EWLM *control center* is the user interface of EWLM. It is a component of the domain manager and it runs as an instance of the WebSphere Application Server included in the Virtualization Engine common runtime. Users can access the control center from a web browser and can do all the EWLM work from there such as, configuration, setting, management, or view reports.

Managed server

The server managed by EWLM is called *managed server*. Each managed server must have an EWLM agent software, which is also called “EWLM managed server”, installed. Each managed server monitors the performance of the server continuously and reports the aggregated performance data to the domain manager every 10 seconds. Some of the servers might run ARM instrumented applications but servers which do not have ARM instrumented application can also exist in an management domain.

Figure 5-2 depicts a load balancer which distributes requests to two servers with balancing. An EWLM domain manager can send commands to load balancers to change the dispatching balance based on the information reported by the managed servers.

EWLM also can send commands to a server which has LPAR partitions, and change the resource allocation of the server. In EWLM V2R1, CPU allocation of IBM eServer p5/i5 is supported by this facility. EWLM sends command to Power hypervisor of these servers.

5.3 ARM instrumentation

ibm.com/redbooks
International Technical Support Organization

ARM instrumentation

- ❑ **Application Response Measurement 4.0**
 - A standard of Open Group
 - <http://www.opengroup.org/tech/management/arm/>
- ❑ **Enables application to report its property/status**
 - Managing application (EWLM) can use
- ❑ **Provided as dynamic link library**
 - libarm4.a(AIX), QSYS/LIBARM4(i5/OS *SRVPGM), libarm4.dll(Windows), libarm4.so(Linux), etc.
 - Same API set

```

graph LR
    subgraph ARM_Instrumented_Server [ARM Instrumented Server]
        Application[Application] --> ARM_library[ARM library]
        ARM_library --> EWLM_Managed_Server[EWLM Managed Server]
    end
    EWLM_Managed_Server --> Domain_Manager[Domain Manager]
    
```

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

EWLM response time measurement is based on Application Response Measurement (ARM) 4.0. ARM is an open standard formulated by the Open Group, which measures service levels of single or multiple systems.

ARM is implemented through dynamic link libraries in each platform. A program can call ARM APIs when linking to these libraries. The application programs which sends response time data to EWLM domain manager must be ARM instrumented. Actually, many of IBM middleware are already ARM instrumented and can be enabled to use ARM with appropriate setting. As far as you use these ARM instrumented products, the user programs do not have to call ARM APIs explicitly.

ARM standard defines a set of APIs and its parameters, related variables and buffer layouts. ARM API can be called by C programs or a Java programs. Java program uses Java Native Interface (JNI) mechanism to call ARM library.

Table 5-1 lists the available ARM APIs.

Table 5-1 ARM 4.0 APIs

API name	Description
arm_bind_thread()	Indicates the transaction thread is identified by the start handle
arm_block_transaction()	Indicates transaction is blocked waiting on an external transaction
arm_destroy_application()	Indicates the registered application data is no longer needed

API name	Description
arm_discard_transaction()	Discards the transaction started by arm_start_transaction
arm_generate_correlator()	Generates a correlator, the token transferred to subtransactions
arm_get_arrival_time()	Reports the arrival time of the transaction rather than arm_start_transaction() executed time
arm_get_correlator_flags()	Reports specified correlator data
arm_get_correlator_length()	Returns correlator length
arm_get_error_message()	Get error messages
arm_is_charset_supported()	Indicates whether an ARM library supports the specified character encoding
arm_register_application()	Informs the ARM library of metadata about the application. Application name, ID, etc.
arm_register_metric()	Informs the ARM library of metadata about the metric the application provides. Application ID, buffer pointer, integer type, etc.
arm_register_transaction()	Informs the ARM library of metadata about the transaction measured by the application. Application ID, transaction ID, etc.
arm_report_transaction()	Reports statistics of a completed transaction
arm_start_application()	Indicates an instance of an application has started
arm_start_transaction()	Indicates a transaction is beginning execution
arm_stop_application()	Indicates that the application instance has finished making ARM calls
arm_stop_transaction()	Indicates the end of a transaction
arm_unbind_thread()	Indicates the transaction thread is no longer performing on behalf of the transaction
arm_unblock_transaction()	Indicates that the blocked transaction is no more waiting a external transaction
arm_update_transaction()	Indicates the transaction is still processing

As you can see from the table, all APIs are able to manage or report the transaction related information such as the attribute of the transaction, the application name in which the transaction runs, or the thread information of the transaction. The time stamp data is also captured in many of these APIs.

Correlator is the data which is passed around from a parent transaction to the child transaction.

For a complete description of ARM 4.0 APIs you can refer to the following Web site:

<http://www.opengroup.org/tech/management/arm/>

5.3.1 ARM 4.0 API execution

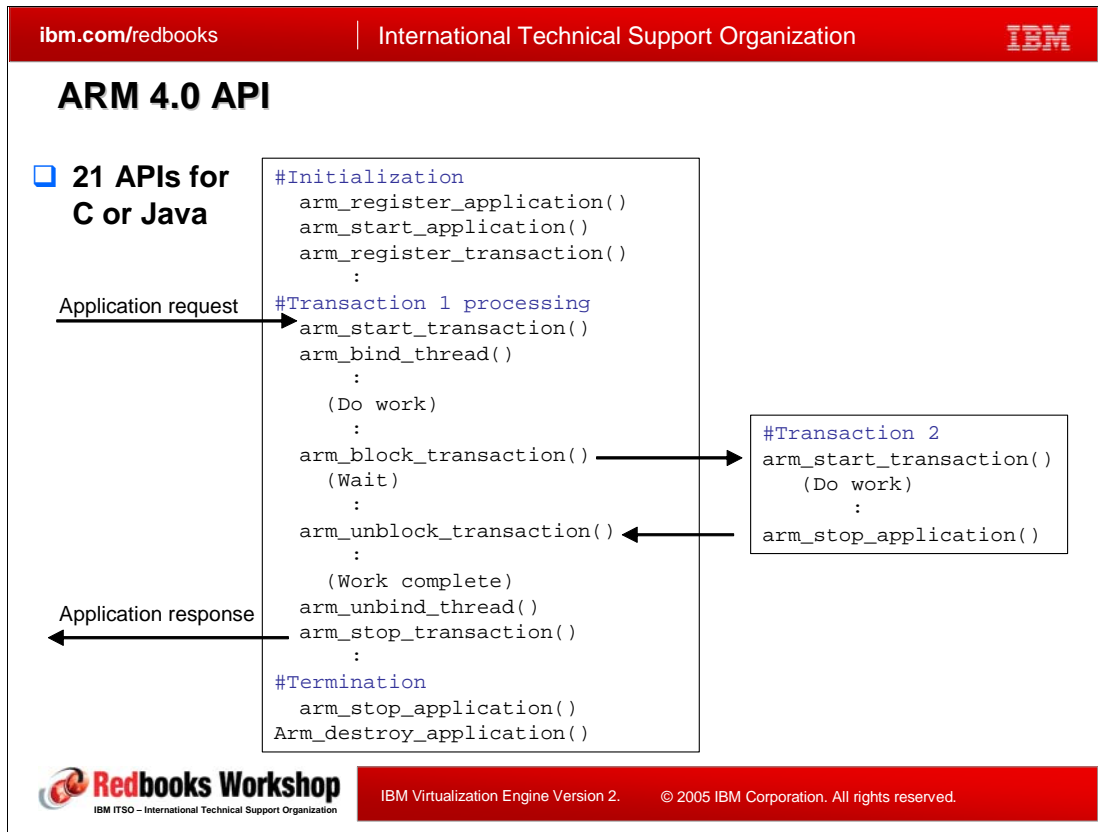


Figure 5-3 ARM 4.0 API

In Figure 5-3, a typical ARM API calling sequence is shown. When you use ARM instrumented IBM products, all ARM API calls are done in the IBM middlewares internally.

- ▶ An `arm_register_application()`, `arm_start_application()`, `arm_register_transaction()` sequence is issued in the initialization step. “application” means the name of the middleware such as WebSphere.
- ▶ When a transaction comes to the server application, the application issues `arm_start_transaction()` and `arm_bind_thread()`, which is the trigger of the measurement of the response time.
- ▶ When the application makes a child transaction call (transaction 2 in the foil), the application issues `arm_block_transaction()`, then issues an `arm_unblock_transaction()` when the control is returned from the child transaction.
 - If the invoked child transaction runs in an ARM instrumented environment, the same mechanism is used to measure the performance of the child transaction. In this case, the two levels of transaction response time, the original one, and the child transaction, are measured.
 - If the child transaction runs in a non-ARM instrumented environment, the performance of the child transaction is not measured in the response time, however, the total (end-to-end) response time is measured since the first transaction is run in an ARM instrumented environment.

Though non-ARM instrumented applications can exist in EWLM management domain, at least the entry point application must be ARM instrumented to measure the end-to-end

response time. To measure the elapsed time of the transaction in each server or application, all middleware (or applications) must be ARM instrumented.

To monitor the performance of non-ARM instrumented applications, EWLM supports a different metrics rather than response time, that is called 'velocity', which is described later.

5.3.2 Correlator and hop

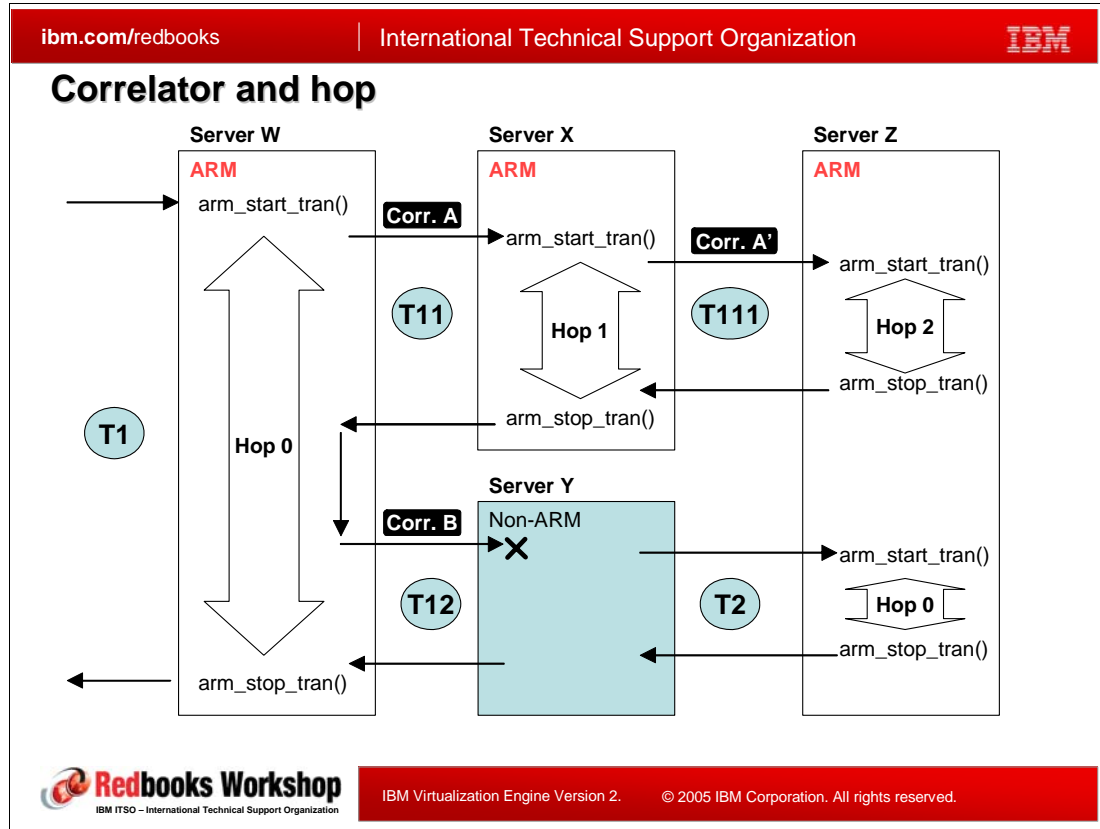


Figure 5-4 Correlator

The *Correlator* is a data token passed around the parent and child transactions. The purpose of the correlator is to keep and to identify the parent-child transaction relationships of ARM instrumented applications.

The correlator is defined in the ARM 4.0 standard and you can generate a correlator by yourself using an ARM API. However, in an EWLM environment, all correlators are managed by EWLM internally and the user does not control. It is worth to know the mechanism of the correlator to understand the EWLM transaction performance monitoring.

Figure 5-4 shows a sample of relationships of transactions in multiple server environment. There are four servers. Server W is the entry point of the application. Server X and Y are the tier 2 servers and server Z is the tier 3. All applications in the servers except for server Y are ARM instrumented. T1 is the parent transaction. T1 contains T11 and T12 as its child transactions. T111 is the child transaction of T11 and the “grand child” of T1. T2 is the child of T12.

In an ARM instrumented environment, a transaction starts with an `arm_start_transaction()` API and ends with an `arm_stop_transaction()` API. The elapsed time between these two

APIs is recognized as the response time of the transaction. A child transaction is started by the parent with issuing an `arm_block_transaction()` and the parent issues an `arm_unblock_transaction()` when the child is finished. The block/unblock APIs are not shown in the chart. EWLM records all the timestamps of start/stop, block/unblock transactions.

Correlators are used to keep parent-child relationship of a transaction. When a child transaction is invoked, the parent generates a correlator and passes it to the child. To keep the relationship, this process must be done between the two servers without EWLM domain manager. In actual implementation, ARM instrumented IBM products sends correlator in various way such as, including the correlator in http request header, or using JDBC control transfer protocol. EWLM managed server software monitors, summarizes, then sends the correlator information to the domain manager.

In the chart, **Corr.A** is the correlator of the parent transaction T1, which is generated in the server W, for the child transaction T11. **Corr.A** includes any control information to identify that T11 is the child of T1. When T11 invokes a child transaction (T111), a new correlator **Corr A²**, which has the **Corr.A** information and the additional information of server X, is generated and sent to server Z. Thus EWLM can identify the parent-child relationships and the topology of the transaction passed around.

Correlator has an important role in the EWLM facility “transaction classification”, which is described in ref.

Hop count means the depth of the transaction in parent-child relationship. The parent transaction which generates a correlator is hop0 and its child is hop1, the grandchild is hop2, and so on. The parent-child relationship is shown as the hop count in any of EWLM performance report. Hop0 application is also called *entry application*.


Note that correlators can be handled only by ARM instrumented applications. In Figure 5-4, transaction T12 is invoked from ARMed server W with **Corr.B**, but the correlator is ignored since server Y is not ARM instrumented. The parent-child relationship is broken here. Logically, T2 is the child of T12. However, since server Y is not ARM instrumented, T2 is not recognized as the child of T12 by EWLM. For EWLM, T2 is a brand new transaction for server Z. Therefore, T2 is handled as hop0. While EWLM can know the end-to-end response time of the transaction T1 (the elapsed time between `start_tran()` and `stop_tran()` in server W), it cannot know the response time of T12 exactly.

5.3.3 ARM instrumented IBM products

ibm.com/redbooks | International Technical Support Organization **IBM**

ARM instrumented products

- DB2 Universal Database (V8.2 or later)**
- WebSphere Application Server (V5.1.1 or later)**
- http servers with WAS Web serving plug-in**
 - Apache http server
 - IBM HTTP server
 - Microsoft Internet Information Services (IIS)
- http servers with independent plug-in**
 - Independent plug-in for Apache http server 2.0.47 or later
 - Independent plug-in for IBM HTTP server 2.0.47 or later
 - IBM HTTP server for iSeries
 - EWLM supplied plug-in for Internet Information Services (IIS) 5.0 and 6.0

 **Redbooks Workshop**
IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 5-5 ARM instrumented products

Using ARM library, you can write ARM instrumented application programs using Java or C by yourself. However, in most cases, the programs which should be ARM instrumented are the middlewares such as http servers, application servers, or database products, which are deployed as the logical nodes of multi-tiered application environment.

IBM currently provides ARM-instrumentation support on the middleware products named in Figure 5-5. These products can be ARM enabled by setting up the environment properly.

For http servers, ARM instrumentation is enabled by using the ARM instrumentation framework of WebSphere Application Server (IBM Web Serving plug-ins).

IBM also provides non-WAS Web serving plug-ins (Independent plug-ins) for some http servers. They can be used in environment of, such as Tomcat, JBoss, CGI, which does not have WebSphere Application Server.

At the writing time, EWLM supplied plug-in is for static web page only and is not recommended to use in production environments.

5.3.4 ARM enablement

The screenshot shows a document page with a red header containing 'ibm.com/redbooks', 'International Technical Support Organization', and the IBM logo. The main title is 'ARM enablement of WAS, DB2, http servers'. The content is organized into four main sections, each with a blue square icon:

- Prerequisite tasks**
 - Install EWLM managed server software
- DB2**
 - Set DB2 environment variable
 - capabilities = CAP_ARM_APPLICATION,CAP_PROPAGATE
- WebSphere Application Server**
 - Enable PMI (WebSphere Performance Monitoring Infrastructure)
 - Check 'Enable EWLM' check box
 - Create JVM custom properties
 - Set 6 properties for WebSphere JVM
- http servers**
 - Webserver plug-in
 - Configure plug-in from the administration console
 - Independent plug-in
 - Copy the independent plug-in to a directory
 - Edit httpd.conf

At the bottom, there is a 'Redbooks Workshop' logo and footer text: 'IBM ITSO - International Technical Support Organization', 'IBM Virtualization Engine Version 2.', and '© 2005 IBM Corporation. All rights reserved.'

Figure 5-6 ARM enablement of WebSphere Application Server, DB2, http servers

Enabling ARM in each middleware is a simple process on any platform. Figure 5-6 shows the general procedure to enable ARM for each middleware. The detailed process is different for each platform and the chart does not intend to describe full detailed process.

For WebSphere application server, you can enable ARM by checking some check boxes from the administration console. Setup of http servers which utilizes the WebSphere Application Server Web serving plug-in can also be done from the WebSphere Application Server admin console. Independent ARM plug-in for http servers which don't utilize the WebSphere Application Server plug-in can be installed as well as the other http server plug-in of the servers.

Please note that there are some prerequisite tasks before enabling ARM on each server. It includes the installation of managed server software, setup of EWLM domain manager, configuration of EWLM control center, etc.

5.3.5 Non-ARM instrumented servers support

Non-ARM instrumented applications are also supported by EWLM. For non-ARM instrumented applications, the serving “velocity” of the application must be defined as the performance goal. Velocity goal is intended to be used for long running application such as batch jobs, server daemon processes, or non-ARM instrumented application.

Velocity goal can also be used to measure the whole LPAR partition. Velocity goal is defined as one of the following:

- ▶ Fastest
- ▶ Fast
- ▶ Moderate
- ▶ Slow
- ▶ Slowest

You have to specify one of these five velocity goals for each processes (jobs) which are monitored by EWLM. EWLM managed server monitors them in many aspects of the process, such as CPU utilization or IO wait time. A *performance index* calculated from the data measured is used to decide whether the goal is met.

The velocity goal measurement or evaluation is limited within the process or a server LPAR partition. You cannot measure the combined or end-to-end application performance using velocity goal.

5.4 Enterprise Workload Manager domain policy

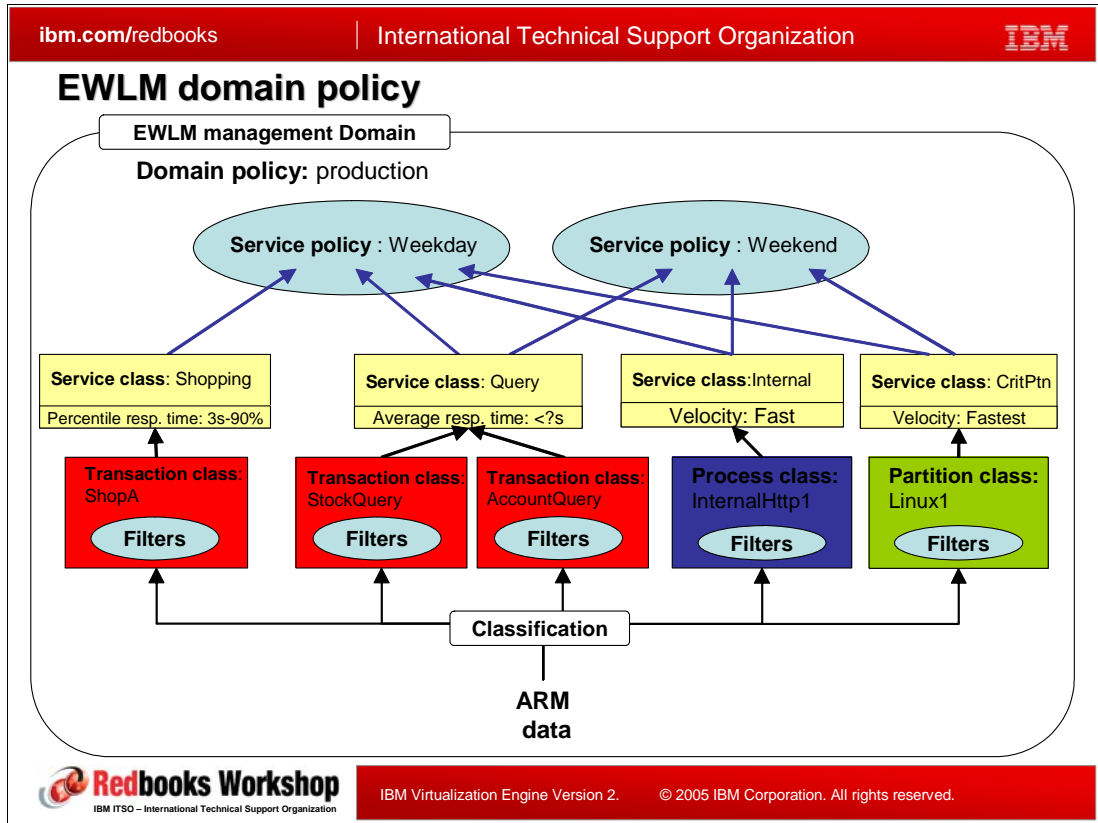


Figure 5-7 EWLM domain policy

To take full advantage of EWLM, you need to understand the concepts of EWLM domain policy. Figure 5-7 shows the top level view of the EWLM components.

EWLM does performance monitoring and evaluation by comparing the collected data to the user defined performance goals, such as response time or velocity. In order to do it, all performance raw data monitored by EWLM must be categorized identifying which application, process, or server the data come from and which goal the data should meet. This process is called *classification*. The classification is done by filtering the data. This process is shown in the lower part of Figure 5-7.

Transaction class, process class, and partition class are the EWLM objects defined by the user which represents an application transaction, a process in a operating system, or a whole LPAR partition.

Filters are set in *transaction class*, *process class*, or *partition class* and, as the result, each performance data is classified to any of these classes.

Transaction, process, and partition classes are registered to a *service class*. A service class, the key component of EWLM domain policy, is the object defined by user in which the performance goal is defined.

Service classes are registered to *service policy*. A service policy is a set of service class which includes all the service classes used in entire system at a time. Only one service policy can be activated at a time in an EWLM management domain.

Service policies are registered to *domain policy*. Domain policy is the top level component which contains everything. Only one domain policy can be deployed at a time in an EWLM management domain.

The detailed description of these EWLM components and the structure is shown in the following sections.

5.4.1 Domain policy and service policy

ibm.com/redbooks
International Technical Support Organization
IBM

Domain policy and service policy

- ❑ **Domain policy**
 - The top level component of the EWLM component structure
 - Multiple **domain policies** in one **EWLM management domain**
 - Example: production, test, development environment
 - Only one **domain policy** is deployed at a time
 - Includes a set of all EWLM components deployed at once
- ❑ **Service policy**
 - The second level component
 - Multiple **service policies** in one **domain policy**
 - Example: Policy for weekday, weekend, backup

EWLM management domain

Domain policy: production


Service policy:
Weekday

Service policy:
Weekend

Domain policy: development

Service policy:
Weekday

Service policy:
Weekend



Redbooks Workshop
IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 5-8 Domain policy and service policy

Domain policy

A domain policy is a collection of EWLM components such as service policies, service classes, transaction classes, process classes, and partition classes, for an entire EWLM management domain. All EWLM component used in an EWLM management domain must be defined in a domain policy. An EWLM management domain can have one or more domain policies such as domain policy for production environment, development environment, and for test environment, however, only one domain policy can be deployed at a time in an EWLM management domain

Service policy

A service policy is the second level component of an EWLM domain policy. A service policy is included in a domain policy which is the top level component. Service policies are usually defined for each category of system operation such as weekday, weekend, or backup. You can define one or more service policies in a domain policy, however, only one service policy can be activated in a domain policy.

5.4.2 Service class

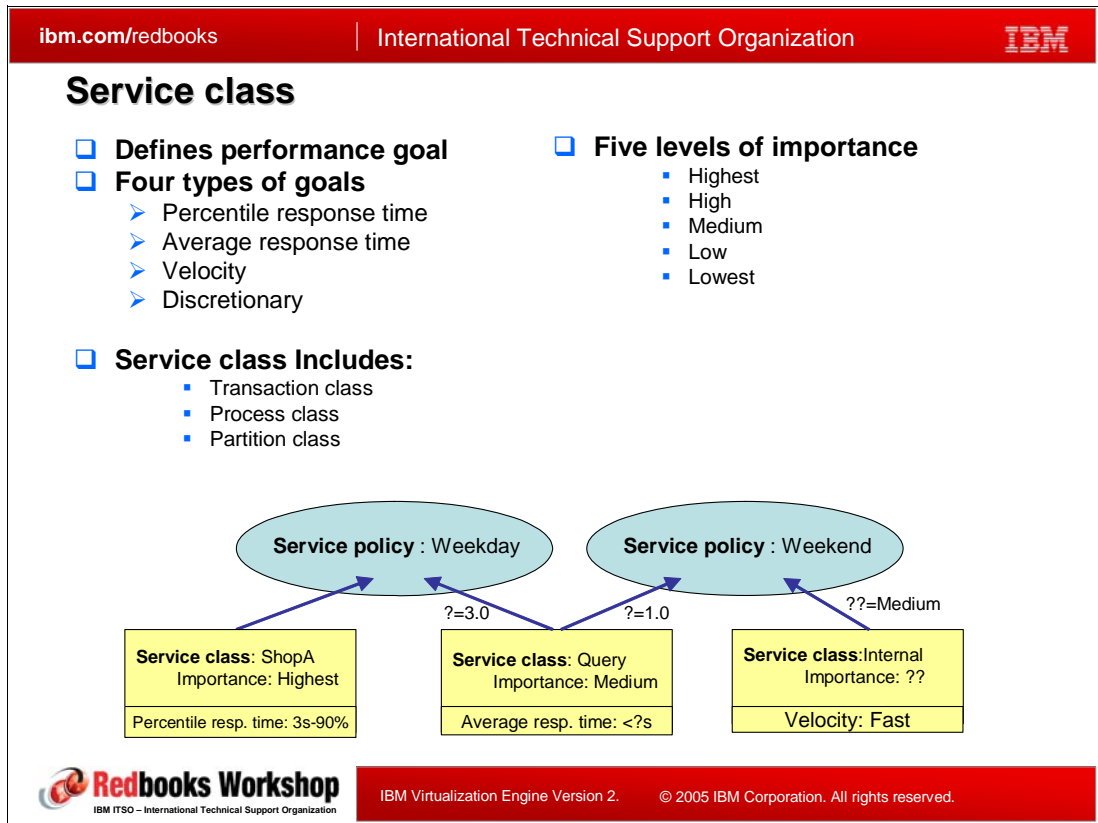


Figure 5-9 Service class

Service class plays the central role of the EWLM domain policy since the “performance goal” is defined in the service class. The service class is a component which specifies the expected service level for a performance entity (transaction, process, or partition). All these performance entities monitored by EWLM are classified to any of the service classes. The expected service level is defined as the performance goal in a service class.

Service classes are service policies. A service class can be included in more than one service policy. A service policy can contain multiple service classes. A service class defines a goal value and a goal type. The goal value of a service class, such as the actual admissible average response time, as in the chart, can be different in each service policy.

Service class includes transaction classes, process classes, and partition classes.

Performance goals

In EWLM, there are four types of performance goals. Only one type of goal must be specified for each service class. They are:

- ▶ Percentile response time

Percentile response time goal indicates what percentile of transactions should complete in a specified amount of time. The specified values are given as a percentage and the time. The time value can be in hours, minutes, seconds, and microseconds. Only ARM instrumented applications can be classified to this type of goal.

- ▶ Average response time

Average response time goal specifies the average amount of time in which work in the service class should complete. It is specified in microseconds, seconds, minutes, or hours. Only ARM instrumented applications can be classified to this type of goal.

► Velocity

A velocity goal defines how fast work should run without delays due to processor constraints, storage problems, and I/O delays. Velocity goal is used for work in which response time goals are not appropriate, such as service process, daemons, and long-running batch work. Applications which do not implement ARM also use a velocity goal. A velocity goal must be set as one of the following:

- Fastest
- Fast
- Moderate
- Slow
- Slowest

The evaluation whether the service speed is fast or slow is done according to the internal logic of EWLM. EWLM uses the CPU utilization, storage and IO delay data for the evaluation. If there is no other active process in the server and the process can utilize all resources it requires, the “Fastest” goal is met. If there is an amount of resource restriction for the process, because of contention with other processes or any other reason, the lower level of goal is met or even the “slowest” goal might not met in the worst case. EWLM automatically evaluates whether the velocity goal is met or not.

► Discretionary

Discretionary goal indicates that the performance entity that uses this goal is processed only when the spare system resource is available or when other performance entities do not utilize resources. This goal does not require a specific measurement value of goal. This goal is used for low priority work that does not require a particular performance goal. EWLM processes discretionary work using resources which are not required to meet goals of other service classes.

Importance

In addition to goal types, you must also specify the importance level of the service class. Importance plays a role only when a service class is not achieving its goal. For instance, resources may be taken away from a less important service class in order for a more important service class to achieve its goal. You can specify five levels of importance settings:

- Highest
- High
- Medium
- Low
- Lowest

Importance does not affect EWLM performance monitoring. It only affects the *action* of EWLM performance adjustment such as LPAR CPU management or load balancing.

As well as goal value, the importance of a service class can also vary in each registered service policy.

Workload

The EWLM term *workload* defines a set of service classes grouped together for reporting purposes. This is an optional definition.

You can make a workload by grouping the logically related service classes such as the service classes to classify an applications of a software vendor. Then you can view reporting data that is specific to the work requests processed for the vendor’s application.

5.4.3 Domain policy, service policy, service class example

Domain Policy: Development		Service Policy: Weekday	Service Policy: Weekend
Domain Policy: Production		Service Policy: Weekday	Service Policy: Weekend
Workload: Shops	Service class: ShopA	Goal: 90% in 3s Importance: Highest	(not registered)
	Service class: ShopB	Goal: 90% in 4s Importance: High	(not registered)
Workload: Query	Service class: Query	Goal: Average response time 3s Importance: Medium	Goal: Average response time: 1s Importance: High
Workload: Internal	Service class: Internal	Goal: Slow Velocity Importance: Low	Goal: Fastest Importance: Medium

Figure 5-10 Domain policy, service policy, service class example

Figure 5-10 shows an example of a domain policy, a service policy, and a service class settings.

There are two domain policies in the management domain example: “production” and “development”. The “Production” domain policy has two service policies: “Weekday” and “Weekend”. There are four service classes in the “production” domain. The service classes “ShopA” and “ShopB” are defined as the same workload “Shops” and they are registered only on service policy “Weekday”. (We assume that the shops are closed in weekend.) Service classes “Query” and “Internal” are registered to both “Weekday” and “Weekend”.

EWLM can monitor the performance of service classes which are registered to a service policy at a time. With this configuration, you can monitor the performance of those four service classes on weekday, however, you cannot monitor any performance information of “Shops” workload when “Weekend” service policy is activated even if the shops are open in the weekend temporarily and generating heavy workload. (In that point, this is not a good example!) Thus the performance monitoring of EWLM is absolutely “application based” and not system based. It is the user’s responsibility to select properly which application has to be monitored by EWLM (An exception is the service classes associated with partition class).

Service classes has goal type and importance as its attributes but the values of goal and importance can be varied in each service policy. The goal of the service class “Query” is average response time less than 3s in “Weekday” policy but it is less than 1s in “Weekend”.

5.4.4 Workload classification

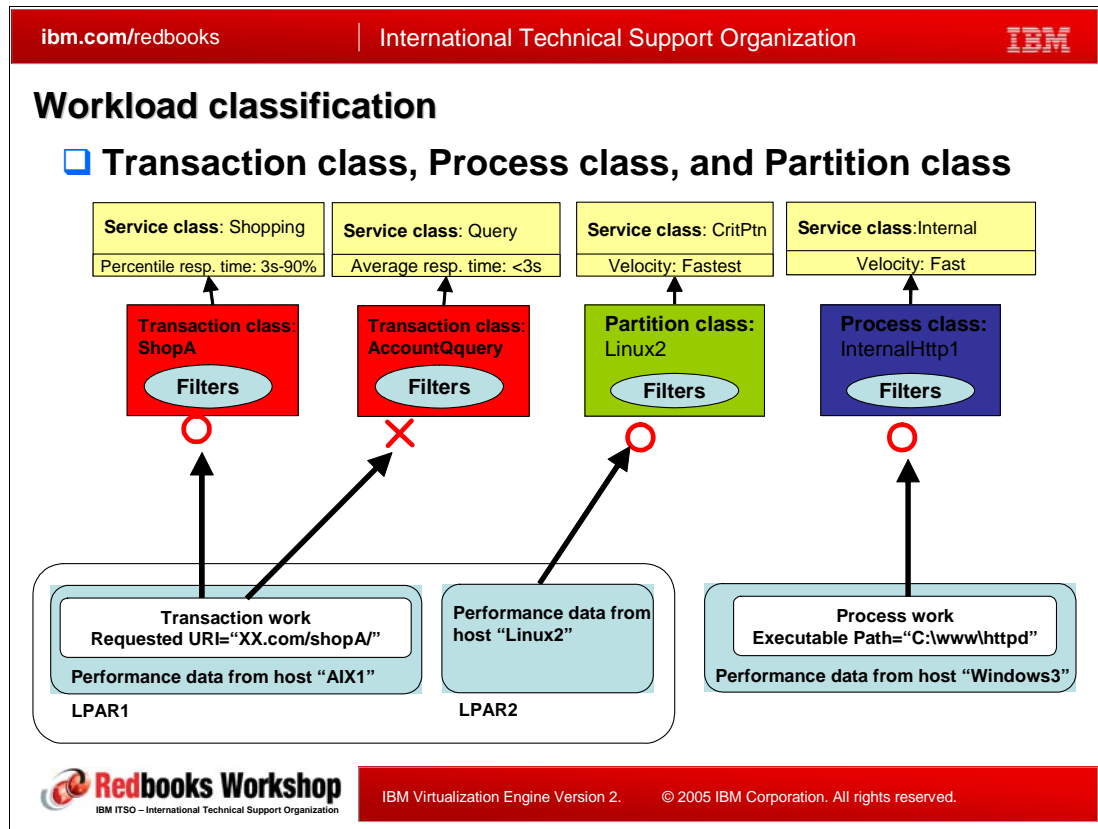


Figure 5-11 Workload classification

A service class contains transaction, process or partition classes. These three components are used by EWLM for the classification workloads. Classification means the process of associating a particular workload with an existing transaction, process or partition class. To associate a particular workload to the correct class the user needs to design the classification filters correctly. Transaction workloads are classified when the work request first enters the EWLM domain.

- ▶ *Transaction class* is used for transactions associated with ARM instrumented applications.
- ▶ *Process class* is used for classifying an operating system process (long running program).
- ▶ *Partition class* is used for classifying work performed by an entire partition.

Work to be classified must be classified to either a transaction class, a process class or a partition class. Partition class enables EWLM to collect performance data of a entire LPAR partition in addition to each transaction and process.

EWLM partition management is a facility to balance the CPU utilization of each LPAR partitions in a server. In order to do it, EWLM must collect all the performance data for each partition, accumulate the data in each partition, then evaluate each partition performance to understand whether the goal is met or not. This process is done additionally to the usual

transaction or process performance monitoring of EWLM. The performance data from an LPAR partition is collected via:

- ▶ Transaction class which includes any performance data of the LPAR partition.
- ▶ Process class which the process is running in the LPAR partition.
- ▶ Partition class defined for the LPAR partition.

The performance data that EWLM handles has many attributes. For example:

- ▶ A transaction data from ARM instrumented application has attributes such as the requested URI, the application server instance name of the application running, the transaction name, etc.
- ▶ A data from an operating system process has attributes such as the executed program name of the process, the OS platform name, the user name of the process, etc.
- ▶ An LPAR partition has attributes such as the TCP/IP hostname of the partition, the system name of the partition, etc.

EWLM identifies each performance data using a set of matching rule for these attributes. The matching rules are called *filters*.

For example, in Figure 5-11, we assume three managed servers have reported performance data:

- ▶ The managed server "AIX1" is an LPAR partition of a p5 or i5 server. The data from host AIX1 includes transaction information from an ARM instrumented application. The requested URI of the transaction is XX.com/ShopA. XX.com is also the host name of AIX1 server and /ShopA is the http server document path. At the classification, the requested URI matches the filter of the transaction class "ShopA" and it is classified to the service class "Shopping". The service class Shopping has a percentile response time goal: 90% of the transaction has to have the response time less than 3s.

Since AIX1 is an LPAR partition the data from the transaction class is also used to EWLM LPAR CPU management facility.

- ▶ The managed server Linux2 is an LPAR partition of the same server as AIX1. There are no ARM instrumented application running in Linux2 partition. To collect the performance data from this partition, there are two ways. One is to create a process classes for specific processes in this partition. The other is to create a partition class for the entire partition. In Figure 5-11, the latter example is shown. The whole Linux2 partition is classified to the partition class "Linux2" which has the host name or IP address filter. The performance data collected via the partition class is used to EWLM LPAR CPU management facility.

Though you can define a partition class to a partition in which ARM instrumented application is also running, The typical usage of partition class is to just define only a partition class for the partition in which no ARM instrumented application running.

- ▶ The managed server "Windows3" is a stand-alone Windows server. An http server, which is not ARM instrumented, is running in the Windows3 server. The http server is launched by executing the program C:\www\httpd. To collect the performance data of the http server, a process class is defined. It has the host name filter "windows3" and the executable path filter "C:\www\httpd", Via the process class, the performance data is classified to the service class "Internal" which has "Fast" velocity goal.

Transaction classification

ibm.com/redbooks
International Technical Support Organization

Transaction classification

- **Transaction class and ARM instrumentation**
 - Entry application (Hop0 application)
 - The start point of end-to-end performance measurement
 - Where EWLM looks up the filters
 - Entry application must be ARM instrumented

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 5-12 Transaction classification

The transaction is classified when the first work request arrives to the EWLM management domain.

Strictly speaking, a transaction classification happens when a transaction without a correlator arrives at the ARM instrumented application (middleware). The chart in Figure 5-12 shows an EWLM management domain which includes ARMed and non-ARMed applications. The arrows mean the direction of the transaction invocation. “Corr.” is correlator. The points where classification happens are shown by the small circles. Typically, transaction classification happens when a new transaction comes the first time into the domain. Additionally, when a non-ARM instrumented application invokes a child transaction in an ARM instrumented application, such as the rightward two points, the classification also happens.

Another example is Figure 5-4 on page 181, where classification happens at the beginning of transactions T1 and T2.

Though transaction classification happens at all these points, it is the user’s responsibility to classify or not the transactions to a service class. It depends on the filters definition in the transaction classes defined by the user. If you want to monitor the performance of a specific transaction, you must set the filters which matches the attribute of the transaction. A default transaction class and a default service class are provided by EWLM. It matches all the transactions which are not classified by any user-defined transaction classes. The default service class has a discretionary goal.

Filters

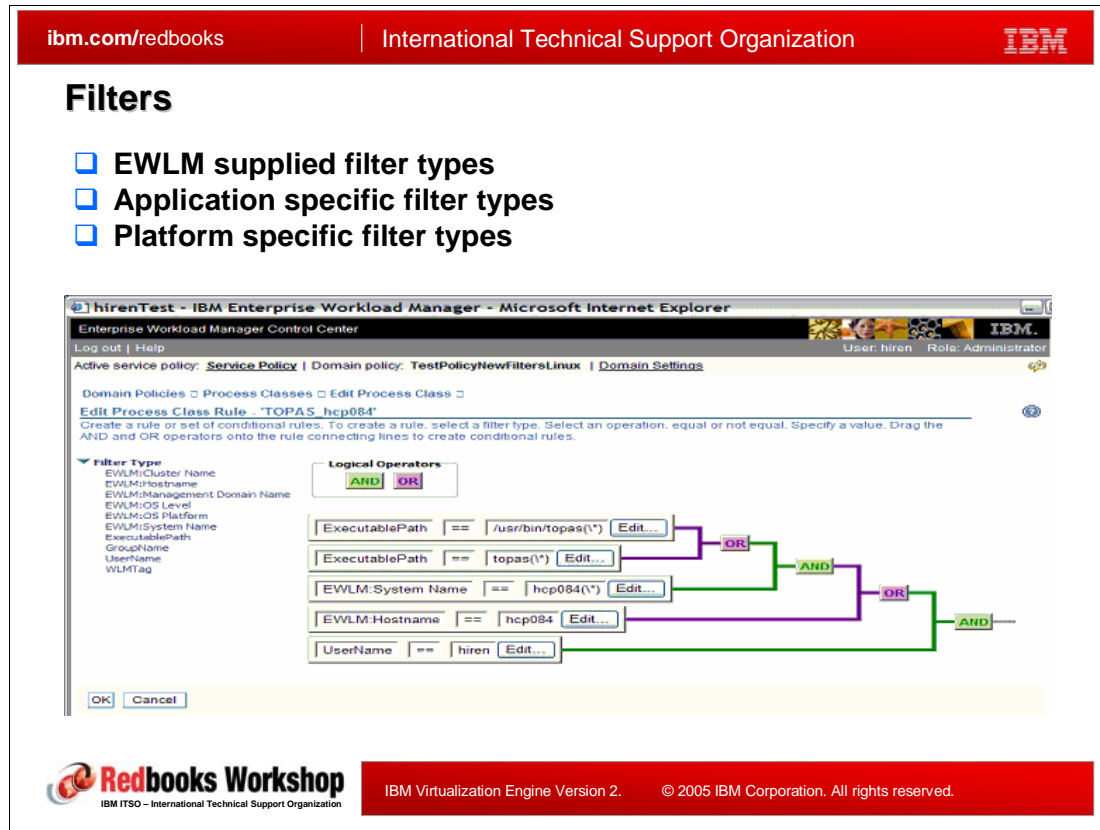


Figure 5-13 Filters

The workload classification of EWLM is obtained by applying filters to each performance data. Filters are set in transaction, process, and partition classes and tests various attributes of the performance data.

There are three categories of filter types in EWLM: EWLM supplied, application specific and platform specific filters, which are listed respectively in Table 5-2, Table 5-3, and Table 5-4. Though these tables contain all available filter types, note that the purpose of the tables shown are to have an outlook of applying filters. Supported filter types also depend on the class to be specified (transaction, process, or partition), and there are many additional constraints or concerns. For complete information, see the EWLM manual. [reference](#)

The picture shown in Figure 5-13 is the EWLM control center screen to edit filters for a process class. Multiple filters can be combined with AND or OR logic. A filter has an attribute named "Position", which is set as 1, 2, etc. The position specifies the order of filters to be tested.

In EWLM control center, filters are also referred as *rules*.

Table 5-2 EWLM supplied filter types

Filter type	Description
EWLM: OS Platform	The name of the operating system such as AIX or OS/400.
EWLM: OS Level	The release level of the operating system. For example, AIX 5.2.0.0 or z/OS 1.6.0.0.

Filter type	Description
EWLM: Hostname	A unique name by which a server is known on a network. The fully qualified host name consists of the host name and the TCP domain name. The TCP domain name specifies a group of systems that share a common suffix (domain name). For example, in the fully qualified host name <i>server1.us.ibm.com</i> , <i>server1</i> is the host name and <i>us.ibm.com</i> is the TCP domain name that contains the <i>server1</i> host name. The fully qualified host name uses a character-based format such as <i>server1.us.ibm.com</i> . However, some servers might use a numerical-based name. The numerical-based host name is not the same as the system's IP address.
EWLM: Cluster Name	The name of the operating system cluster if the operating system uses clusters. For z/OS, the cluster name is the sysplex name.
EWLM: System Name	The system's name for the operating system instance. For example, in AIX and i5/OS, this is the name of the node.
EWLM: Management Domain Name	The name of the EWLM domain.
EWLM: Application instance	The application instance name from an <i>arm_start_transaction</i> call from the product that first processes the work request. The application instance might be a server application that either routes the work request to another server or a server application that completes the work request.
EWLM: Transaction name	The name of the transaction. The name is generated by the middleware application. It contains alphanumeric characters.
EWLM: user name	The user name, as understood by the instrumented application, that initiates the work request. This is the user name specified in the user sub-buffer of an <i>arm_start_transaction</i> call.
EWLM: URI	The Uniform Resource Identifier (URI) is how a point of content is identified on the Internet. The URI syntax is depends on the scheme. Most follow this general format: <i><scheme>://<authority><path>?<query></i> where the parameters are defined as follows: <i><scheme></i> (required) specifies the type of scheme used such as "http", "ftp", "mailto", or "urn". <i><authority><path>?<query></i> represents the scheme-specific parts whose format is determined by the specifications of the scheme. The EWLM: URI value does not need to contain the entire URI value. You can identify a URI value by specifying only a portion of the value. The URI string value is specified in the transaction identity sub-buffer of an <i>arm_register_transaction</i> or <i>arm_start_transaction</i> call.

Table 5-3 Application specific filter types

Filter type	Description	Application
Application Hostname	The TCP/IP host name or the NetBIOS name of where the database resides.	IBM DB2 Universal Database
Application Id	The ID generated when the application connects to the database at the database manager or DB2 Connect™ Server specified by communication protocol (TCP/IP, IP address, NetBIOS or IPX/SPX) in three parts separated by periods.	IBM DB2 Universal Database

Filter type	Description	Application
Client Protocol	The communication protocol that the client application is using to communicate with the server. Use this filter for problem determination for remote applications.	IBM DB2 Universal Database
Database Name	The name of the database or the alias of the remote database on the server.	IBM DB2 Universal Database
Execution Id	The ID that the user specified when logging in to the operating system.	IBM DB2 Universal Database
Program Name	The name of the application running at the client as known to the database manager or DB2 Connect. You can use program name with appl_id to relate data items with your application.	IBM DB2 Universal Database
System Auth Id	The authorization ID of the user who invoked the application connecting to the database. On a DB2 Connect gateway node, this is the user's authorization ID on the host.	IBM DB2 Universal Database
HostInfo	The authority segment of the URI, not including the port information.	IBM Web serving plug-in
PluginType	The type of plugin, for example 'Websphere' or 'IHS/Apache' or 'IIS'.	IBM Web serving plug-in
Port	The TCP/IP port on which the request arrived, specified as a string representation of the decimal value.	IBM Web serving plug-in, IBM WebSphere Application Server
Protocol	The protocol the client used to connect to the plugin, for example HTTP 1.1. Protocol is specified as the scheme plus the level.	IBM Web serving plug-in
QueryString	The query segment of the URI string which is the portion of the dynamic URI that contains the search parameters of the request, not including the ? character. For example, Example: http://www.ibm.com/support?p1=a &p2=bQueryString = p1=a&p2=b	IBM Web serving plug-in, IBM WebSphere Application Server
RemoteAddress	The request client's TCP/IP address specified as a string.	IBM Web serving plug-in
RemoteUser	The user name associated with the request specified as a string.	IBM Web serving plug-in
Scheme	The webserver scheme associated with the segment of the URI, for example http or https.	IBM Web serving plug-in
ServerName	The name assigned to the server or virtual server to replace the actual host name of the machine. Applies only to the WebSphere plug-ins.	IBM Web serving plug-in
ServerVersion	The version of the Web server. This filter applies only to Internet Information Services (IIS).	IBM Web serving plug-in
BusName	The bus name for the Java Message Service (JMS).	IBM WebSphere Application Server

Filter type	Description	Application
ClassName	The class name for the work alarm or deferred run command.	IBM WebSphere Application Server
DestinationName	The destination name for the Java Message Service (JMS).	IBM WebSphere Application S
EJB™ name	The logical name of the enterprise bean associated with the request.	IBM WebSphere Application Server
InputMessage	The input message name associated with the Web service.	IBM WebSphere Application Server
MdbDiscriminator	The MDB discriminator for the Java Message Service (JMS).	IBM WebSphere Application Server
MethodSelector	The method selector for the Java Message Service (JMS).	IBM WebSphere Application Server
NameSpace	The name space associated with the Web service.	IBM WebSphere Application Server
Operation	The operation name associated with the Web service.	IBM WebSphere Application Server
Topic	The topic name for the Java Message Service (JMS).	IBM WebSphere Application Server
Transport	The transport name associated with the Web service.	IBM WebSphere Application Server
WsdIPort	The port name associated with the Web service.	IBM WebSphere Application Server

Table 5-4 Platform specific filter types

Filter types	Description	Available platforms
CommandLineArguments	The parameter of the command.	AIX, HP-UX, Linux, Solaris, Windows
ExecutablePath	The process as it was launched that specifies the entire path. Windows example: driveletter:directoryName\subdirectoryname\filename.suffix AIX and Solaris example: /directoryName/subdirectoryname/filename.suffix Note: For AIX and Solaris, the paths do not specify a drive letter, they use forward slashes (/) instead of backward slashes (\), and begin with a forward slash (/).	AIX, HP-UX, Linux, Solaris, Windows
GroupName	The group name that is processing the program.	AIX, HI-US, Linux, Solaris
UserName	The user name that is processing the program.	AIX, HP-UX, Linux, Solaris, Windows

Filter types	Description	Available platforms
WLMTag	The tag passed from an application to the AIX workload manager.	AIX
Effective Group Profile	The name of the group profile which is currently used by the job or <i>*NONE</i> . The value is a character string.	i5/OS, Linux
Effective User Profile	The name of the user profile which is currently used by the job or <i>*NONE</i> . The value is a character string.	i5/OS, Linux
Command Base Name	The base name of the command.	Linux, HP-UX
Job Accounting Code	The name of the job accounting code. The value is a character string.	i5/OS
Job Description Name	The job description name. The value is a character string.	i5/OS
Job Name	The simple job name. The value is a character string.	i5/OS
Job User Name	The user name part of a qualified job name. The value is a character string.	i5/OS
Subsystem Name	The name of the subsystem. The value is a character string.	i5/OS

Process and partition classification

ibm.com/redbooks
International Technical Support Organization

Process and partition classification

- ❑ **Process class**
 - To classify work request based on OS process to a Service classes
 - ARM non-instrumented application support
 - Filter work requests based on:
 - Host name of the server
 - System name of the operating system
 - OS Platform
 - Executable path
 - more..
- ❑ **Partition class**
 - To classify the whole server (partition) to a service classes
 - eServer i5/p5 LPAR partitions support
 - Filter work requests based on:
 - Host name of the server
 - System name of the operating system
 - OS level
 - more...

```

graph BT
    WR[Work request] --> PC[Process class]
    PC --- F([Filters])
    PC --- PNC[Partition class]
    PNC --- SC[Service class]
    SC --- VC[Velocity: Fast]
  
```

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 5-14 Process and service class classification

Classification by process class or partition class is simpler than classification by transaction class. These two classes are used for monitoring the performance of single operating system process or entire LPAR which has no relation to other performance entity.

- ▶ A process class is used to classify an operating system service process, daemons, and long-running batch jobs. These jobs are usually classified by the “ExecutablePath” filter (See Table 5-4 on page 197), which is one of a platform specific filter indicates the executable program name of the process. Any other filters such as hostname, system name can be used to distinguish the process in a specific server, OS, or partition.

Process class can only connect to the service class which has velocity goal.

- ▶ A partition class is used to identify whether the performance data collected by EWLM comes from a specified LPAR or not. The purpose of a partition class is to aggregate the all performance data which is from a specific partition, and use it as the base data of the EWLM partition management.

A partition class can only connect to the service class which has a velocity goal.

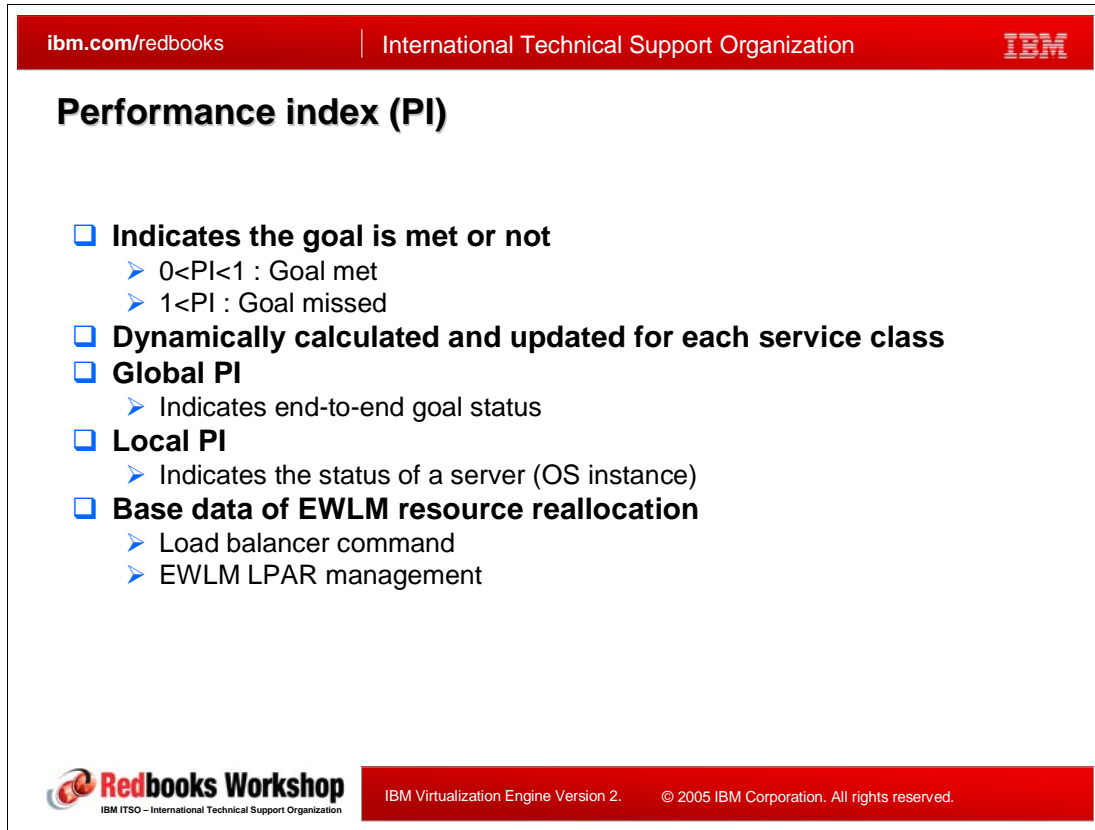
To classify to a partition class, Filters are set to distinguish a whole partition, such as host name or IP address.

The purpose of partition class is to monitor the whole partition performance and use the information to LPAR CPU management facility. Since the LPAR CPU management facility collects performance data also from transaction and process classes, you usually have to create a partition class when there are no ARM instrumented application in the partition, or when you are not interested in the performance of specific processes but want to monitor the whole partition performance.

The difference of data collection path (transaction, process, or partition class) is the difference of the possible performance goals to define. If you create only transaction classes to monitor LPAR partition, you define the end-to-end response time goal and the LPAR CPU management works to keep the performance of specified end-to-end transaction. If you create process class, the goal is the specific process performance. LPAR CPU management takes action only when the specific process is suffering performance degradation. If you create partition class, the goal is to keep the overall performance of the partition, which includes all processes in the partition. LPAR CPU management adjusts the CPU allocation of the partition to keep the overall performance.

Transaction, process, partition class should be defined according to the performance goal you want to keep. If necessary, partition class can be defined on an LPAR partition in which ARM instrumented transaction or any monitored process is running. In the case, all performance goal is monitored and LPAR CPU management works to keep all goals. If two or more goals are not met, LPAR CPU management compares the "importance" of each service class and works to keep the most important goal.

5.5 Performance Index



Performance index (PI)

- ❑ **Indicates the goal is met or not**
 - $0 < PI < 1$: Goal met
 - $1 < PI$: Goal missed
- ❑ **Dynamically calculated and updated for each service class**
- ❑ **Global PI**
 - Indicates end-to-end goal status
- ❑ **Local PI**
 - Indicates the status of a server (OS instance)
- ❑ **Base data of EWLM resource reallocation**
 - Load balancer command
 - EWLM LPAR management

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 5-15 Performance index (PI)

To evaluate whether the performance goal is met or not, EWLM uses a metric called *performance index (PI)*. EWLM calculates PIs for each service class automatically and periodically. Remember that the performance goal is described as an attribute of a service class. EWLM monitors various performance data such as response time, CPU utilization, or I/O delay and they are used to calculate PI internally in EWLM. After all, EWLM uses PI as the only performance index to optimize system resources.

PI is calculated for service classes with response time goals and velocity goals.

PI is given as a simple number such as 0.5, 1.0, 1.2.

- ▶ When the PI for a service class is equal or less than 1.0, the goal of the service class is met.
- ▶ If PI is greater than 1.0, the goal is missed.

There are two types of PIs in EWLM: global PI and local PI.

- ▶ Global PI is the calculated value of end-to-end system performance. Global PI is calculated from hop 0 response time of an ARM instrumented application (See:Figure 5-16 on page 202). Typically, global PI is calculated with the performance data from multiple servers. When we say just PI, it usually means the global PI.
- ▶ Local PI is the PI value calculated within one server (or one OS instance of LPAR). It is calculated only from the performance data of the server. It includes any process performance data, and any response time data of the transaction in the server including hop 1 or higher level, which are collected in a specific server.

In Figure 5-16, we assume that the server W, X, Z is actually the LPAR partitions of a server and there is a significant bias of CPU utilization by each partition, let's say, server X has very high CPU utilization. In the case, if only global PI is provided, EWLM can not sense that there is a performance bottleneck as far as the performance goal of the transaction T1 is met. However, if the local PI of server X is provided, EWLM can know that the server X is the bottleneck. This is the way that the local PI is used and EWLM partition management, which will be described later, is also based on this mechanism.

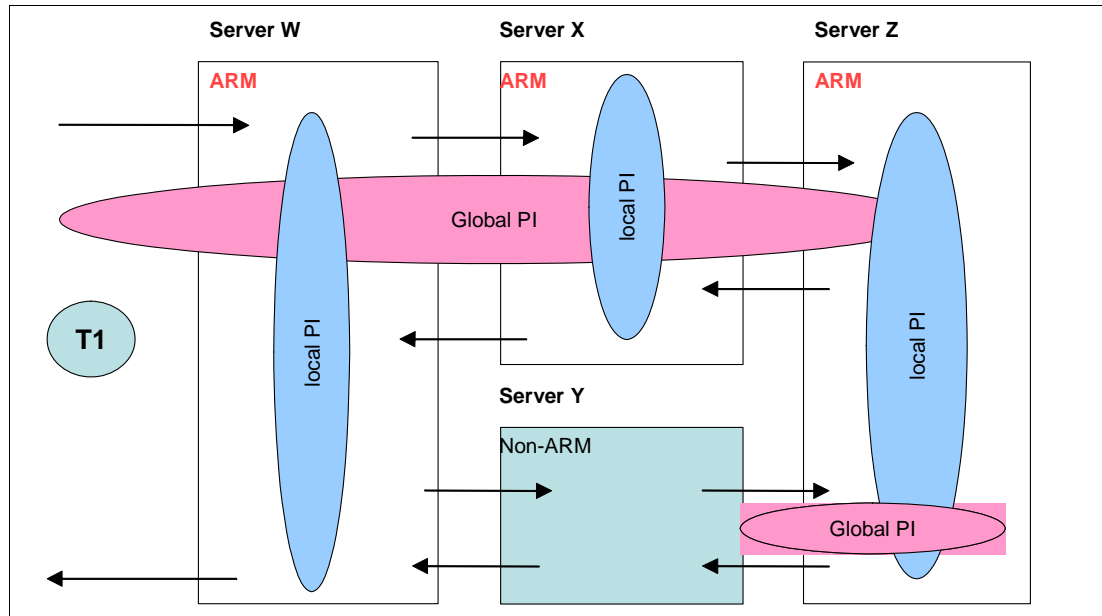


Figure 5-16 Performance index (PI)

EWLM uses PI and the “importance” of the service class to determine the service class which should be helped by the resource management function. Remember that the importance is also an attribute of a service class. The priority to find the service class to be helped by EWLM is shown in Table 5-5. There are 10 levels of priority and two or more service class might has the same priority. Global PI has higher priority than local PI.

Table 5-5 Priority order by PI and importance

Priority	PI	Importance
1(Highest)	Global	Highest
2	Global	High
3	Global	Medium
4	Global	Low
5	Global	Lowest
6	Local	Highest
7	Local	High
8	Local	Medium
9	Local	Low
10(Lowest)	Local	Lowest

5.6 Enterprise Workload Manager load balancing

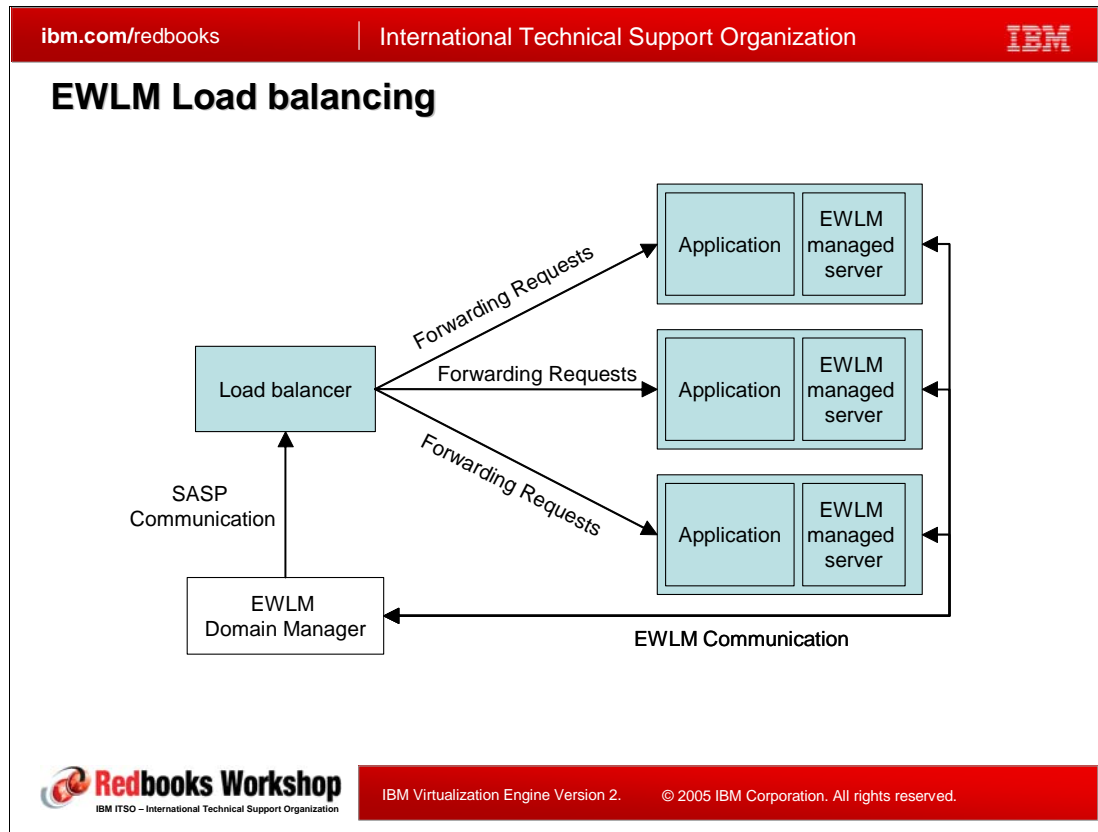


Figure 5-17 EWLM Load balancing

As stated in Section 5.1, “Enterprise Workload Manager overview” on page 174, EWLM has both monitoring and optimizing facility. Until Section 5.4, “Enterprise Workload Manager domain policy” on page 186, we discussed EWLM overview, concepts, and domain policy and they are mainly to understand how performance data is “monitored” by EWLM.

The next step is optimizing. The current version of EWLM, V2.1, has two performance optimizing facilities.

- ▶ Load balancing to network load balancer, already available in V1.
- ▶ eServer i5/p5 LPAR CPU management, new in V2.1.

The EWLM domain manager has a global view of all the servers and middleware technologies which make up the management domain that supports the business applications. The domain manager obtains performance information on each server and application instance, and knows how much time a request is spending at each instance. The domain manager can use this knowledge to influence network routing decisions made by a load balancer in order to achieve the end-to-end business response time goal.

Without EWLM, when incoming requests arrive at a load balancer, the load balancer typically has little information about the application’s health or the performance of the servers it is routing to. It can use pure algorithmic techniques such as round robin, or general system statistics represented by static, preconceived weights to route the incoming requests. Some load balancers have algorithms that allow them to sense network state and forward requests

to the most capable instance, like the instance with the least number of sessions or connections.

EWLM does not route the work itself, but provides recommendations to the routing entity using the IBM Server/Application State Protocol (SASP). Through SASP messages a load balancer can tell the domain manager which systems and applications it wishes to load balance and the domain manager can make recommendations to load balancers regarding how to distribute work. However, it is up to the load balancer to actually use EWLM's

recommendations to route incoming requests to the members. Today, the load balancers that exploit EWLM are from Cisco Systems Inc. and Nortel Networks.

5.7 Enterprise Workload Manager LPAR CPU management

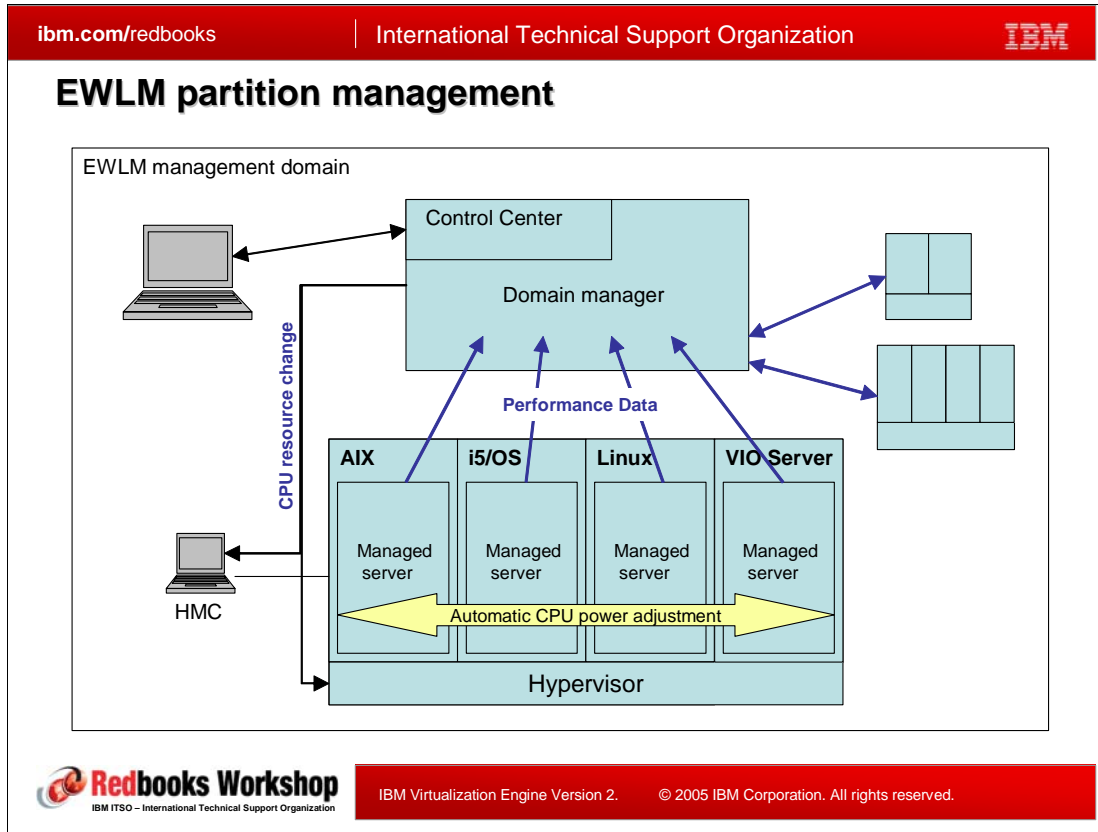


Figure 5-18 EWLM partition management

EWLM partition management is a new EWLM facility to automatically optimize the assigned processor power of each LPAR partition based on the performance data collected with mechanism described in Section 5.4, “Enterprise Workload Manager domain policy” on page 186. Any combination of AIX, i5/OS, Linux, and Virtual IO Server partitions on eServer p5/i5 are supported.

5.7.1 eServer p5/i5 LPAR concept

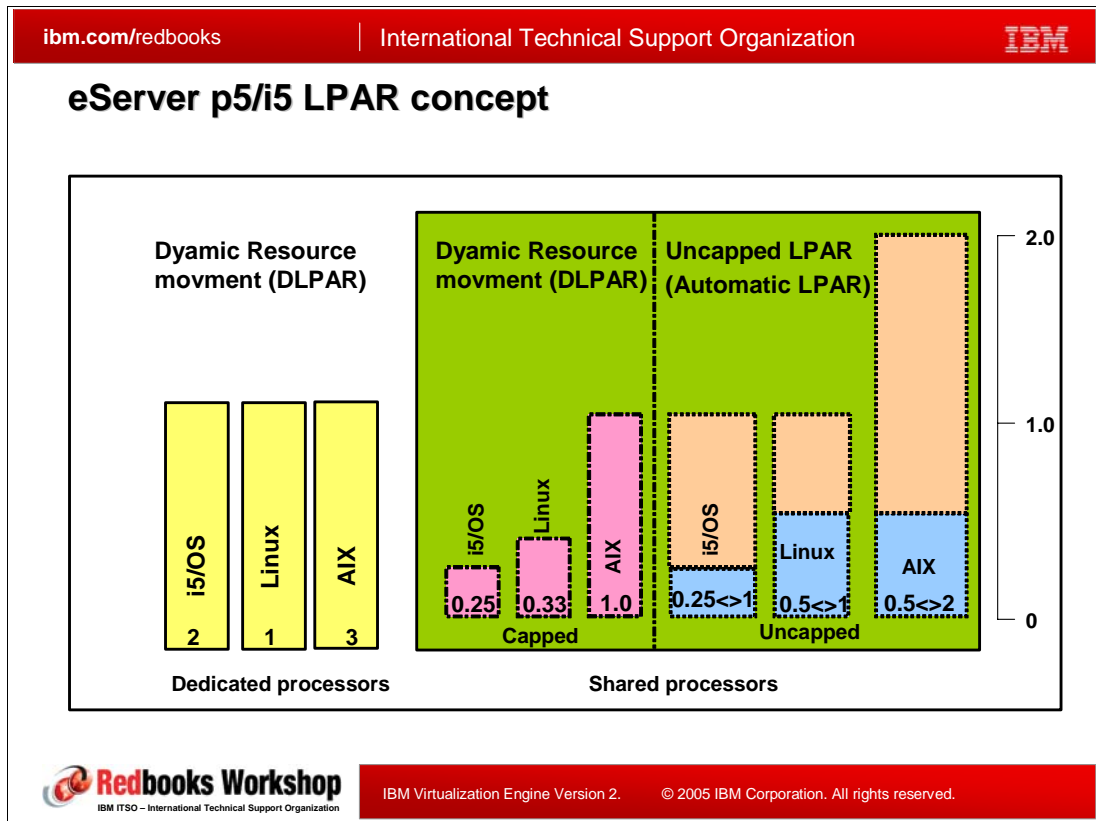


Figure 5-19 eServer p5/i5 LPAR concept

Before proceeding to EWLM LPAR management, we have to remind the eServer p5/i5 LPAR virtualization facility.

LPAR (Logical Partitioning) is a facility to have multiple operating system run in a single system. In p5/i5 LPAR implementation, an operating system runs in a *partition*, which is logically a complete machine which uses its own CPU, memory, and IOs.

On p5/i5, there are two types of CPU allocation to a LPAR: *dedicated* and *shared*. Each partition must utilize either type of CPU allocation.

- ▶ For dedicated processor partition, you can assign a whole number of processors such as 1, 2, 3, etc. The partition defined with dedicated processors use the assigned physical processors exclusively. If necessary, dedicated processors can be moved from one partition to other partition which is defined with dedicated processors by manual operation from the Hardware Management Console (HMC) with both partitions keep running. This facility is called Dynamic LPAR or *DLPAR*.
- ▶ Shared processor is a way to share a physical processor with two or more partitions. To utilize shared processor, you have to first create a *shared processor pool* by specifying the number of physical processors (whole number) to be included in the pool. At the writing time, you can create only one shared processor pool in a p5/i5 server. The processors which are not included in the pool can be used for dedicated processor partition. For shared processor partition, you can assign partial processor to the partition such as 0.1, 1.25, 2.00 from the shared processor pool. For the partition which has partial processor value, a physical processor is assigned to the partition with time-sliced. In Figure 5-19, six

shared processor partitions are shown. As the rule, at least 0.1 (one tenth) processor is required for one shared processor partition.

There are two types of shared processor partition. *Capped* and *uncapped*. You must specify either capped or uncapped to a shared processor partition and it can be changed at any time, dynamically.

- In capped partition, the processor power of the partition can not exceed the *entitled capacity*. The entitled capacity means the (partial) value user assigned from the HMC. Naturally, a capped partition which has 0.25 processor can use no more than 0.25 processors, which means the 25% of the CPU time slice assigned to this partition. The entitled capacity of a capped partition can be varied manually by the user from the HMC with all partitions still running. This facility is also called DLPAR.
- Uncapped partition is a option to utilize shared pool processors effectively. In uncapped partition, the processing power of the partition can exceed its entitled capacity if there is a spare power in the shared processor pool. When there is a processing power which is not assigned to any partition, or when the processor utilization of the other shared processor partitions are low, the spare power can be used by uncapped partitions which need more processor power. When two or more uncapped partitions compete for spare processing power, they are arbitrated according to the *uncapped weight* of the partitions. You have to specify the uncapped weight to each uncapped partition(0-255, bigger number is higher, default:128) from the HMC.

Uncapped LPAR is a function newly supported by p5/i5 POWER5 servers. It is integrated into the Power5 hypervisor. The redistribution of the spare processing power is done every 10ms, which is completely transparent to the partitions. Though DLPAR is also effective, it has less importance for uncapped partitions.

In iSeries wording, uncapped LPAR is sometimes called “automatic LPAR”.

Virtual processor

Virtual processor is an important concept to understand shared processor partitions. When you create a shared processor partition, you have to specify the (whole) number of virtual processors to the partition, which means “virtual N-way” of the partition. For example, a partition which is defined with two virtual processors, runs as a 2-way machine, which means that the partition uses two physical processors simultaneously even if the entitled capacity of the partition is less than 2.00.

The number of virtual processors must be a whole number and at least 0.1 (10%) processing power is needed for a virtual processor. The 2-virtual processors partition must have at least 0.2 processing power. The number of virtual processors must be more than the rounded-up number of the processing power. A partition which has a 1.20 processing power must have 2 or more (and equal or less than 12) virtual processors.

Generally, a small number of virtual processors results in better performance for the partition.

For capped partitions, setting the least number of virtual processors usually gives good performance. However, for uncapped partitions it is not always true since the maximum processing power which can be utilized by an uncapped partition is limited by the number of the virtual processors. The 2-virtual processors partition can use no more than 2.00 processors even when there 3 or more processors are in the shared processor pool.

In Figure 5-19 on page 206, the processing power of the i5/OS and Linux partition is limited to 1.0 since the number of virtual processor for these partitions are set to 1, while the AIX partition has 2 virtual processors.

The number of virtual processors of a partition can be changed dynamically with partitions still running, from the HMC. However, it always requires human decision and manual operation.

New with EWLM V2.1

EWLM partition management enables the automatic management of:

1. Entitled capacity
2. Number of virtual processors
3. Uncapped weight

Note that the facility is for CPU allocation of the shared processor partitions only. At the writing time, memory or IO management are not supported by EWLM LPAR management facility. Also, LPAR which have dedicated processors are not supported.

In the EWLM V2R1 release, the facility of LPAR management is only for p5/i5 Power5 servers.

5.7.2 Partition Workload group

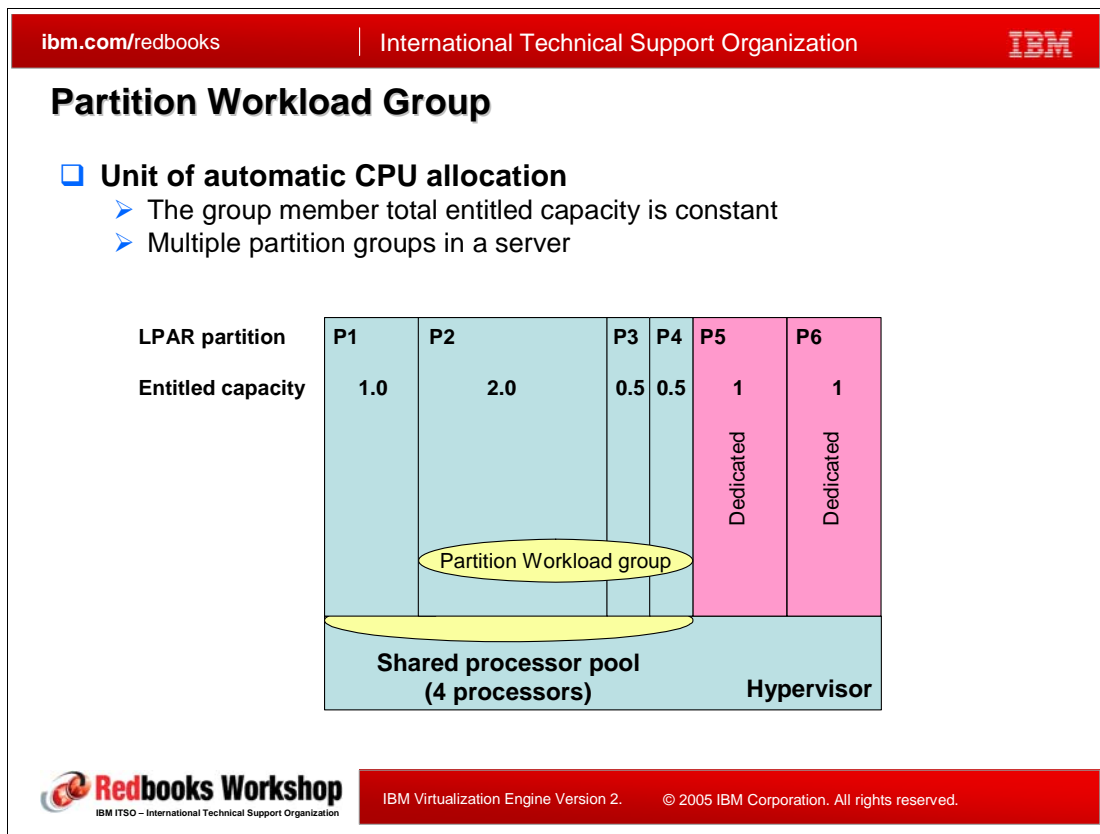


Figure 5-20 LPAR partition group

In EWLM partition management, a set of LPAR partitions is first defined as the *LPAR partition group*, which is a unit of performance management.

Figure 5-20 shows a 6-Way p5/i5 server which has two dedicated processor partitions and a shared processor pool which has four physical processors. There are four shared processor partitions. Note that the entitled capacity total of these four partitions are 4.0 which is the

same as the number of processors of the shared processor pool. In this example, we define P2, P3, and P4 as the LPAR partition group.

LPAR partition group is the hypervisor integrated LPAR partition attribute (?). The setup of the LPAR partition group is done from the HMC. Using the required version of the HMC, a partition has a setting parameter named "Partition workload group" which means LPAR partition group. You can set a number and the partitions which have the same number are defined as an LPAR partition group. Multiple LPAR partition groups can be created in a server.

Management behavior

EWLM partition management automatically optimizes the entitled capacities of the partitions in an LPAR partition group according to the monitored application performance of each partition. However, the total of the entitled capacities of the LPAR partition group is kept constant. EWLM partition management regards the initial total of entitled capacities of the partitions as the capacity of the LPAR partition group. In Figure 5-20, the initial total of entitled capacity of the LPAR partition group is 3.0 (2.0+0.5+0.5) where these values are set by a user from the HMC. The value 3.0 is always constant and can never be changed whenever the entitled capacity of these partitions are changed. All processing power must be assigned; when some partitions get more entitled capacity, the others are necessarily reduced.

If a user manually changes the entitled capacity of a partition which is a member of a LPAR partition group, the change is reflected to the LPAR group capacity since EWLM periodically monitors each entitled capacity of the managed partitions.

Adding or removing a partition from a LPAR partition group requires the partition powered off. When removing a partition from a LPAR partition group, the assigned capacity of the partition must be the same as the initial entitled capacity. You might have to change the entitled capacities of the member partitions manually from the HMC before removing the partition.

Additionally, EWLM partition management also changes the number of the virtual processors of each partition of the LPAR partition group. The number of virtual processors is set to ensure that a partition has enough virtual processors to take advantage of its full entitled capacity. For uncapped partitions, enough virtual processors are provided so that the partition can take advantage of any capacity above its entitled processor capacity that may be available to it. EWLM also ensures that a partition does not have too many virtual processors. When EWLM decides that missing performance goal is caused by the too many virtual processors in a partition, the number of virtual processor of the partition is reduced.

EWLM partition management also optimizes the uncapped weight of each uncapped partitions in the LPAR partition group. Basically it is adjusted proportionally to each partition's entitled capacity.

Figure 5-20 is a intentionally assumed complex sample for explanation. In typical case, all partitions in a server should use shared processors and are defined in one LPAR partition group.

5.7.3 LPAR CPU management mechanism

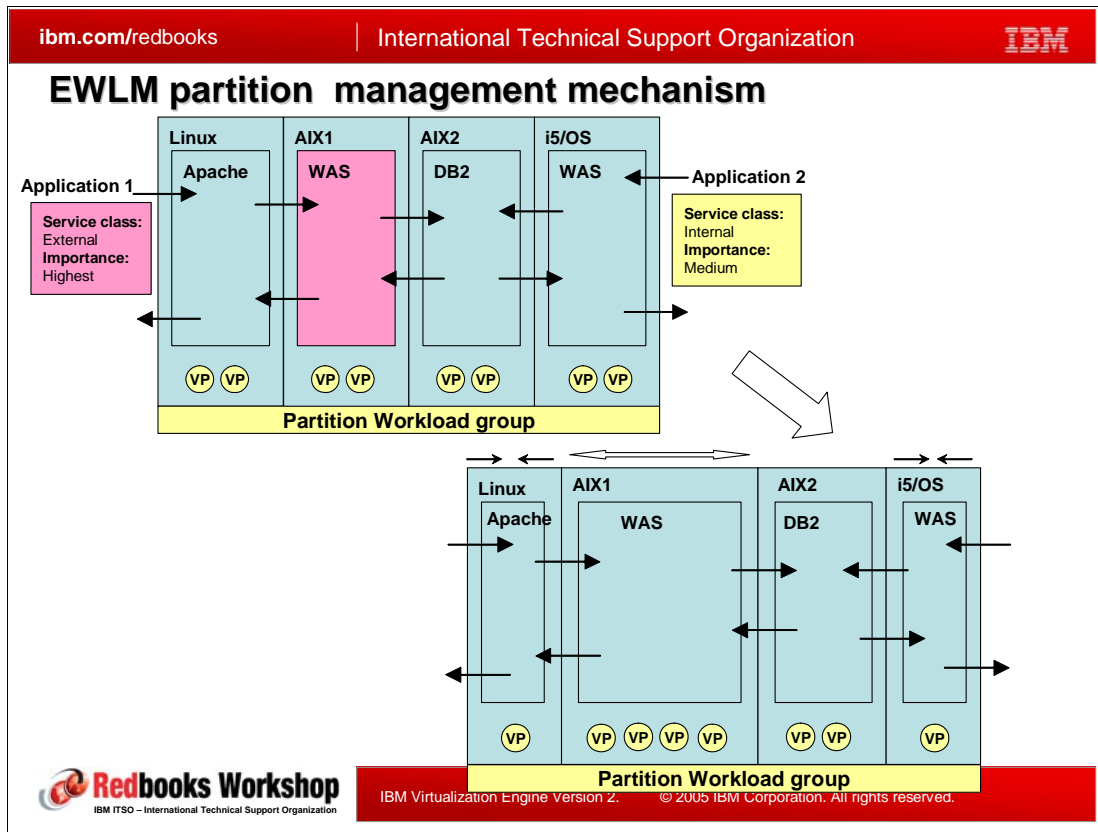


Figure 5-21 LPAR CPU management mechanism

The EWLM partition management is based on the *application performance* monitored by EWLM. The management occurs only when the performance goal of a service class is missing (PI > 1).

Figure 5-21 shows the mechanism of the EWLM partition management. There is an p5/i5 server which has four LPAR partitions. There is a 3-tier web application which is made of a Linux1 partition, an AIX1 partition and an AIX2 partition, which are classified into the service class External. There is another 2-tier application which consists of an i5/OS and an AIX2 partition, which are classified into the service class Internal. Both service classes have a response time goal and an importance, with External having a higher importance than Internal. All middlewares are ARM instrumented and the performance data is monitored via transaction class.

The EWLM partition management mechanisms works as follows:

1. In Figure 5-21, the entitled capacity of each partition is shown as the width of the partition rectangle, and the number of virtual processors are the same for all the partitions.
2. The application 1 workload increases and the service class External begins to miss the goal.
3. Then, using the global and the local PIs, EWLM detects that the cause of missing goal is the degradation in the AIX1 partition. EWLM finds the cause is especially CPU bottleneck. Actually, PI is not calculated only from the CPU utilization. When the PI indicates missing goals, EWLM checks whether the cause is CPU bottleneck or not using its monitored performance data.

4. Then EWLM takes the entitled capacity away from i5/OS partition in which the service class has less importance. And gives it to the AIX1 partition.
5. EWLM also determines that the service class External will do better by taking CPU away from Linux partition and giving them to AIX1 partition.

EWLM monitors many performance metrics of the managed partition including the CPU utilization and can estimate the total system performance after the CPU configuration change (change of entitled capacity, number of virtual processor, uncapped weight). The decision of which partition is to be removed/added CPU is done based also on this information.

6. Then the entitled capacities are moved and the number of virtual processors (VPs) is adjusted to match new resource allocations (lower right chart).
7. If these partitions are uncapped partitions, the uncapped weight parameters are also changed.

LPAR configuration APIs

To enable EWLM to change the LPAR configuration of p5/i5 server, two new APIs are supported in each operating system on p5/i5. These are public APIs:

- ▶ **lpar_get_info()** or **d1par_get_info()**: Retrieve partition information. Returns information about configuration and CPU utilization of the partition on which this API is called.
- ▶ **lpar_set_resources()** or **d1par_set_resources()**: Modify partition information. Modify the configuration attribute of the current partition.

Only service classes which has response time goal (ARM instrumented application) is shown in Figure 5-21. If there are service classes which has velocity goals, the mechanism works as similarly.

5.8 Enterprise Workload Manager and TIO integration

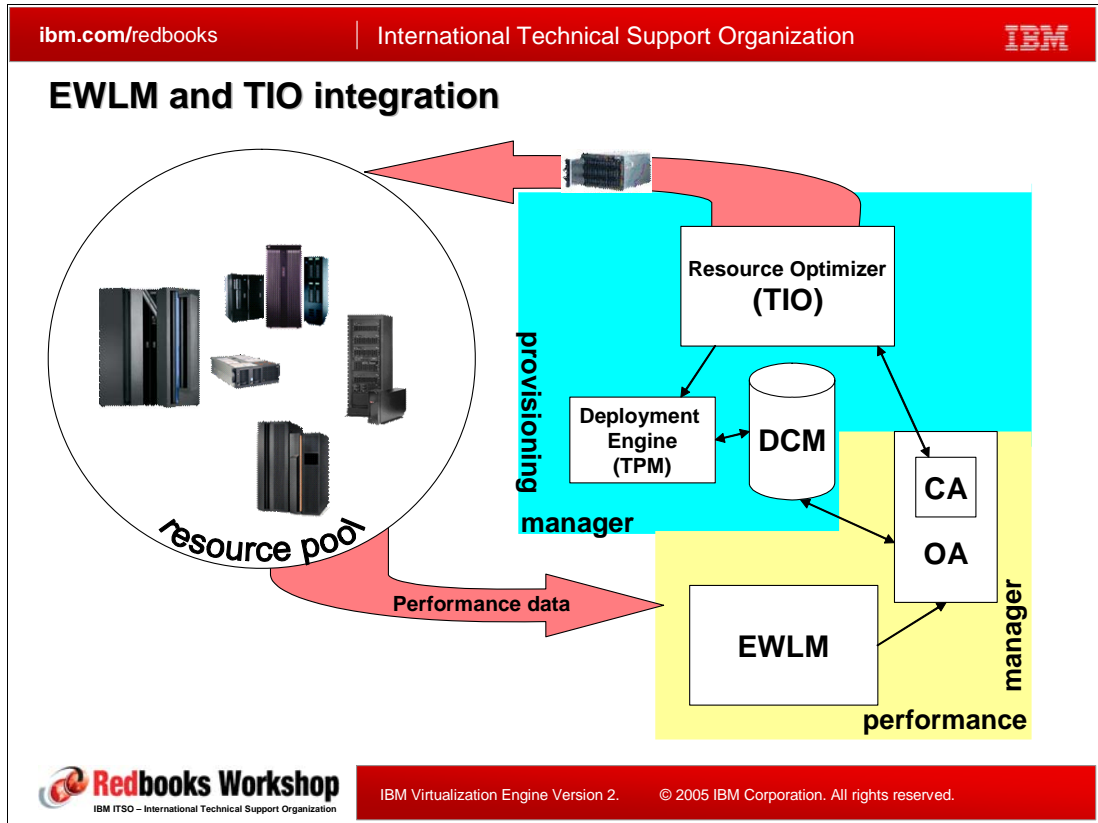


Figure 5-22 EWLM and TIO integration

IBM Tivoli Intelligent Orchestrator (TIO) is an automated resource management solution which automatically triggers the provisioning, configuration and deployment done by IBM Tivoli Provisioning Manager¹ (TPM). TPM is part of the TIO product and may be bought and installed as a standalone product as well. For more information on IBM Tivoli Intelligent Orchestrator (TIO), see:

<http://www-306.ibm.com/software/tivoli/products/intell-orch/>

Through orchestrated provisioning, TIO provides the ability to manage the IT environment in real time, according to defined business policies, which can be obtained from EWLM.

Functionally, there are three components to the provisioning solution:

- ▶ The provisioning manager which consists itself of 3 main functions:
 - The resource management function. Its role is to calculate the best allocation of servers for the current workload and to make capacity allocation decisions, using a pool of servers which have been defined to be used for the applications. In this case, it is provided by TIO. The resource optimizer sends decisions to TPM for execution.
 - The deployment engine. Its role is, depending on the resource optimizer (TIO) decisions, to provision a server from a resource pool, using workflows which define the steps of the provisioning function. A provisioning function can be the building of an

¹ TPM is part of the TIO product though it may be bought and installed as a standalone product as well. For more information on IBM Tivoli Intelligent Orchestrator (TIO), see: <http://www-306.ibm.com/software/tivoli/products/intell-orch/>

operating system, the installation of a middleware with fixes, the deployment of an application, etc. IN this case, it is provided by TPM.

- A resource repository: a database which describes the available resources. In this case, it is a component of TPM, the Data Center Model (DCM). DCM contains all the servers and resources description that make up an application from a TPM point of view.
- ▶ The objective analyzer (OA) which uses resource information from the DCM and performance data from a performance manager that can be used by the TIO. The objective analyzer has two main functions:
 - One function is to pull resource information from the resource repository
 - Another function is to process performance data and link with the resource optimizer. It is often called: the capacity manager (CA). CA will process the performance data and turn it into recommendations.

TIO can use its own objective analyzer or an external OA. An EWLM objective analyzer enables TIO to make provisioning decisions based on performance information received from EWLM.

There is one OA instance per service class. The values processed by the OA represent the probability that the service goal will be missed.

- ▶ The performance manager is the function that defines business goals, provides an end-to-end view of actual performance relative to the defined goals and may manage resources to meet the performance goals. This is the definition of EWLM.

With the integration of EWLM and TIO, the EWLM performance information is passed to TIO through the OA; the OA will evaluate how the running cluster will be able, or not, to meet its performance goals; then the provisioning manager can automatically perform the provisioning action. Resources not in use may be provisioned into the cluster; if no resources are available, TIO will determine which clusters have a lower priority to provision higher-priority clusters.

5.9 EWLM V2.1 enhancements

ibm.com/redbooks | International Technical Support Organization **IBM**

EWLM V2R1 enhancements

- ❑ **EWLM LPAR CPU management**
 - Partition class
- ❑ **Support platform expansion**
 - EWLM domain manager on Linux (x, p, i, z)
 - Including EWLM control center
 - EWLM managed server on Linux (x, p, i, z)
 - EWLM domain manager on z/OS
 - Including EWLM control center
- ❑ **ARM 4.0 instrumented middleware**
 - WAS 6.0
- ❑ **Tivoli Intelligent Orchestrator Integration capability**
- ❑ **Other enhancements**

Redbooks Workshop
IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.


Figure 5-23 EWLM V2R1 enhancements

Figure 5-23 summarizes the difference between EWLM V1 and the new available EWLM V2.1.

- ▶ EWLM partition management is a newly supported facility of EWLM V2R1. It enables automatic CPU allocation management of p5/i5 LPAR partitions. The partition class is a new concept in V2R1 to support LPAR CPU management.
- ▶ In EWLM V2R1, some platforms are additionally supported. It includes Linux and z/OS support for EWLM domain manager. WebSphere Application Server V6.0 now supports ARM 4.0 instrumentation and can join the EWLM response time monitoring environment.
- ▶ Tivoli Intelligent Orchestrator (TIO) Integration is also newly supported concept of EWLM V2R1.

There are some minor enhancements in user interface and functionality of EWLM control center. It includes reporting enhancements, policy editor enhancements. The supported scalability of a EWLM management domain is also extended in V2R1.

EWLM is one of the strategic component of IBM Virtualization Engine. It can monitor and optimize the application performance of heterogeneous server, heterogeneous operating system enhancements. Its application oriented concept such as domain policy, service class, workload classification, performance goal and PI are highly abstracted and therefore it can manage the heterogeneous server environment. This abstracted application oriented approach is necessary to manage the performance of Ondemand Operating Environment and EWLM, which will also be highly continuously enhanced in the future, is the key component to realize ODOE.



Integrated Virtualization Manager for pSeries and Virtual Partition Manager for iSeries

This chapter overviews:

- ▶ Integrated Virtualization Manager(IVM) for IBM System p5 and IBM @server p5 and OpenPower solutions
- ▶ Virtual Partition Manager (VPM) for IBM System i5 and IBM @server i5 systems

On their respective hardware platforms these products enable LPAR configuration and basic management without requiring the use and expense of an independent physical console - Hardware Management Console (HMC) for multiple partitions (LPAR) configuration and management.

IVM and VPM provide partitioning capabilities intended for those small and medium business customers needing LPAR configuration and management for a small number of partitions that do not require the rich function set of LPAR configuration and management and virtualization options available only through the HMC.

In this chapter, the following topics are discussed:


- ▶ IVM and VPM benefits
- ▶ General concepts of IVM and VPM
- ▶ LPAR configuration considerations using these components
- ▶ Example LPAR configuration steps for IVM and VPM

6.1 Benefits from IVM and VPM

ibm.com/redbooks
International Technical Support Organization
IBM

Benefits from IVM and VPM

- **IVM and VPM provide**
 - Cost Effective Ways to consolidate multiple partitions onto a single server.
 - IVM is included at no additional charge with the purchase of the Advanced Power Virtualization feature
 - VPM is no additional charge, included with i5/OS
 - IVM & VPM create partitions using a browser/workstation without the additional cost of the Hardware Management Console (HMC)
 - Simplified steps to create and manage partitions (“LPARs”)
 - IVM, VPM offer an integrated means of managing virtual devices and partitions (integrated storage and partition management)
 - IVM provides intuitive, easy to use web-based user interfaces
 - VPM uses a 5250 workstation interface, similar to LPAR configuration on POWER4 and earlier iSeries systems
 - IVM, VPM can significantly reduce the time and effort required to create and manage virtual devices and partitions



Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 6-1 Benefits from IVM & VPM

IVM for p5 systems and VPM for i5 systems are not replacements for the HMC in all environments. Rather, IVM and VPM are additional offerings intended to lower the cost of entry into POWER5 virtualization capabilities - in customer environments where the richer partitioning and virtualization capabilities of the HMC are not required.

Topics and figures later in this chapter provide more details contrasting the IVM and VPM capabilities as compared to those available using an HMC - so the correct decision is made to use IVM or VPM relative to using the HMC.

Figure 6-1 summarizes the primary LPAR configuration and management benefits using IVM and VPM for their respective POWER 5 based pSeries and iSeries systems. In simple terms IVM and VPM provide a significant set of LPAR configuration and management capabilities with a reduced cost and learning time required as compared to the Hardware Management Console (HMC) with its “full function” LPAR and capacity on demand capabilities.

Important: In this chapter, we use term “LPAR” and “Partition” to mean the same thing - one or more partition.

Assuming the respective IVM and VPM LPAR configuration and management capabilities (detailed later in this document) are sufficient for customers who want up to five AIX, i5/OS, and Linux POWER partitions active, you can get your partitioned environment up and running in a short time.

Cost effective ways to consolidate multiple partitions

IVM and VPM can be viewed as POWER5-based equivalents to the Windows operating system VMWare and Microsoft Virtual Server products for supported hardware platforms. Additionally, with IVM, you can configure and manage a partition for Bladecenter which does not support the external HMC.

IVM and VPM make it easier to implement and lower the cost of entry into @server POWER5 virtualization capabilities:

- ▶ Supported combinations of IVM managed partitions running AIX 5L V5.3 or supported Linux distributions.
- ▶ IVM is included with IBM Virtual I/O Server (VIOS) V1.2, which itself is included at no additional charge with purchase of the Advanced Power Virtualization (APV) hardware feature
- ▶ Supported combinations of VPM i5/OS V5R3 or later partitions and supported Linux distributions listed later in this document. VPM supports one i5/OS partition and multiple Linux partitions
- ▶ VPM is included with i5/OS Version 5 Release 3.
- ▶ IVM and VPM are ideal for SMB environments who do not need the range of LPAR and virtualization capabilities available with the Hardware Management Console.
- ▶ IVM and VPM help minimize LPAR setup time
- ▶ IVM and VPM provide an integrated means of managing virtual devices and partitions (integrated storage and partition management)

Simplify steps to create and manage LPARs

IVM enables you to create and manage LPARs easily. IVM has an intuitive, Web-based user interface enabling great productivity to configure and manage logical partitions, manage virtual storage and virtual ethernet on the managed system, and also view service information related to the managed system.

VPM support is provided through either Dedicated Service Tools (DST) or System Service Tools (SST) interfaces. These interfaces include support for partition configuration and management. DST and SST interfaces are already familiar to iSeries administrators who have configured and managed partitions on iSeries systems that are not using POWER5 technology. iSeries administrators already familiar with DST and SST can setup and manage Linux partitions using VPM on IBM eServer i5 systems without the additional need to learn and use the Hardware Management Console.

Client partitions are considered those that have predefined virtual devices such as virtual ethernet and virtual SCSI. These client partitions can add, remove or delete virtual devices without being concerned with the actual physical devices because IVM and VPM provide a virtualization layer between physical devices and virtual devices. This shields the physical hardware placement complexity from the client partition.

IVM and VPM also simplify the answers to the questions you must respond with during configuration steps by providing well defined default values, such as when configuring to use shared processors and uncapped mode. We discuss more about this later in the chapter.

We take a look first at the HMC and IBM System p5 and eServer p5 feature Advanced POWER Virtualization hardware feature and Virtual I/O Server (VIOS), and IVM.

Then later in this chapter we discuss the IBM System i5 and eServer i5 feature, VPM.

6.2 HMC, APV feature, and VIOS

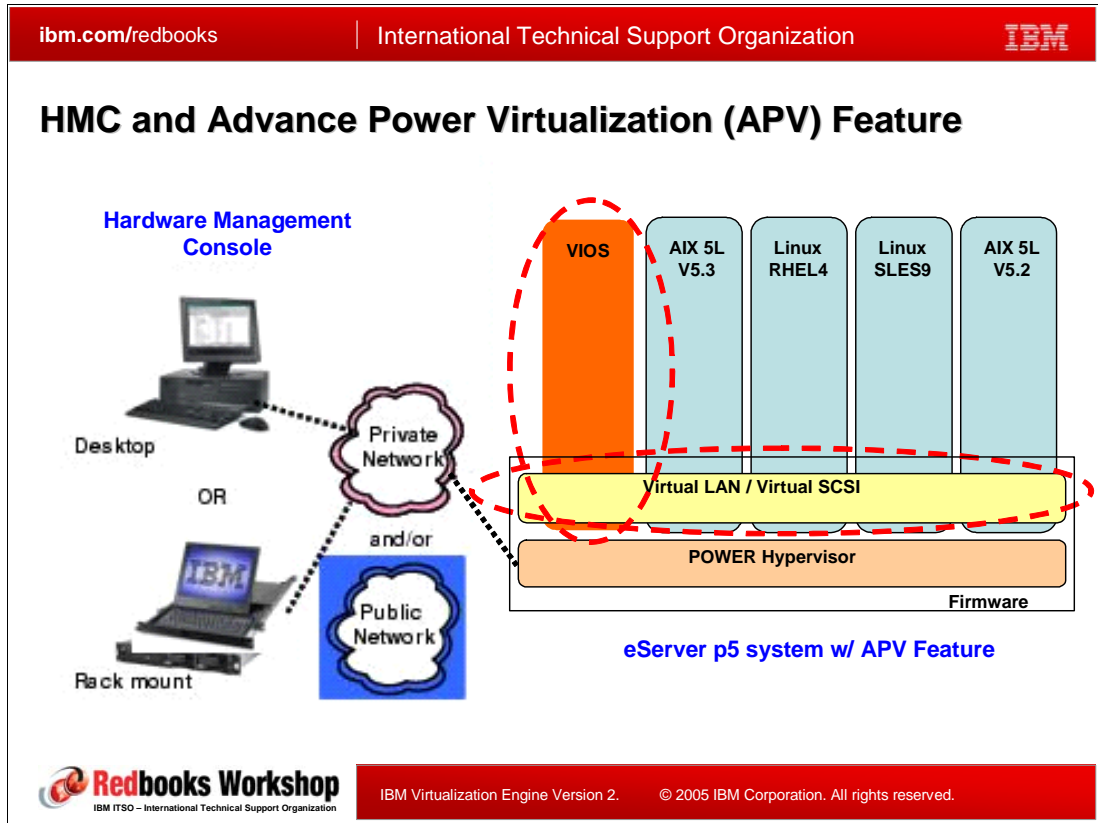


Figure 6-2 Hardware Management Console - HMC

Before going directly to the Integrated Virtualization Memory capabilities, we start with an overview of the use of the existing Hardware Management Console and the IBM System p5 and @server p5 systems and the Advanced POWER Virtualization hardware feature capabilities.

Note: Throughout the remainder of this topic, whenever we use the term *p5 systems* to include IBM System p5 and @server p5 systems.

In Figure 6-2, we show the sample HMC managed environment configuration with p5 systems, including the two types (desktop and rack mounted) of HMC devices on the left. In the system you can see the key partition and virtualization components: firmware, a VIOS partition, a Linux partition, and an AIX partition. As previously stated, VIOS is a part of Advanced Power Virtualization(APV) Feature.

Before we talk about IVM for p5 systems which is our main subject, we discuss prerequisite concepts and functions - Hardware Management Console (HMC), Advanced Virtualization Feature, and VIOS to help understand.

In Virtualization technology enable to manage hardware independently to operating system. Therefore system administrators may focus on their managing jobs effectively without paying attention to hardware management.

For this independence, hardware management needs a separate computing environment to access, configure, control, monitor, and maintain the server hardware and firmware. IBM has

developed several solutions to support this requirement and to manage hardware that target different environments depending on the complexity of hardware setup.

Hardware Management Console (HMC)

HMC is the interface between the system administrator and the POWER Hypervisor and Virtual I/O Server(VIOS) technologies giving a centralized point of hardware control. It's a separate management server that can communicate with multiple servers and partitions at once. A single HMC can manage multiple POWER5 systems and two HMCs may manage the same set of servers in a dual-active configuration designed for high availability.

To implement full server virtualization, at least one HMC is needed. HMC is a hardware system independent of the machine which allows the HMC to remain operational even when the managed system is not, and vice-versa. This provides independency of hardware management from operating management system.

Key HMC functions and tasks include:

- ▶ Creating partition profiles that define the processor, memory, and I/O resources allocated to an individual partition. You can optionally define a System profile that is an order list of partition profiles that can be started when a system profile is activated.
- ▶ LPAR management included
 - Starting, stopping, resetting, and shutting down a partition
 - Opening a virtual console for each partition on a connected IBM System p5 or p5 or i5 system
 - Booting, starting, and stopping the connected IBM System p5 or IBM @server p5 or i5 system.
- ▶ Displaying and monitoring system resources and status.
- ▶ Management of Capacity on Demand.
- ▶ Dynamic resource reconfiguration of partitions
- ▶ Creating and managing virtual ethernet among partitions.

With the *Advanced POWER Virtualization feature* on IBM System p5, or IBM @server p5 systems, or @server *POWER Hypervisor and Virtual I/O Server feature* on OpenPower systems, HMC provides the following additional functions:

- ▶ Creation of shared processor partitions
- ▶ Creation of Virtual I/O Server(VIOS) partition for physical I/O virtualization.
- ▶ Creation of virtual devices for VIOS and client partitions.

Note that, for IBM @server i5 systems, these additional capabilities are included with i5/OS V5R3.

Using an HMC you can change partition configuration dynamically via commands or its Graphical User Interface (GUI). To do this, some extra configurations are required for activating dynamic reconfiguration. HMC needs an Ethernet connection between the HMC and the involved LPAR besides the basic connection with Service Processor to interact securely with operating system on LPAR. This is a significant physical configuration difference from IVM, which communicates with its own LPARs via virtual Ethernet. Through this physical Ethernet, the HMC can securely communicate with LPARs using Remote Monitoring and Control (RMC) protocol and acquire resources and coordinate these actions with hardware configuration changes.

If you need more information about HMC and RMC protocol, which enable dynamic resource reconfiguration, refer to the following URL:

<http://www.redbooks.ibm.com/abstracts/sg247038.html?Open>

The Advanced Power Virtualization Feature

Advanced Power Virtualization (APV) is the required base for IVM. IVM is integrated into the Virtual I/O Server (VIOS), a part of the APV hardware feature. To active VIOS with IVM, users need to order the “Advanced Power Virtualization” Feature. You can purchased IVM only as part of APV, not separately.

The APV feature includes:

- ▶ Micro partitioning support for a single processor being shared by up to 10 logical partitions.
- ▶ Virtual I/O Server

VIOS provides virtual SCSI target and shared ethernet adapter(SEA) virtual I/O functions to client partitions (LPARs).

At the time this document was written, the following operating systems are supported as virtual I/O clients:

- AIX 5L 5.3 or later
- SUSE Linux Enterprise Server 9 for POWER
- Red Hat Enterprise Linux AS 3 for POWER (update 2 or later)
- Red Hat Enterprise Linux AS 4 for POWER

Use the following URL to find the latest information on Linux distribution levels support on p5 systems:

<http://www-03.ibm.com/systems/p/linux/Linux>

Select the link “Learn more about Linux on System p5 and eServer p5 products

- ▶ Automated CPU and memory reconfiguration using the Partition Load Manager (PLM) product.
- ▶ Real-time partition configuration and load statistics.
- ▶ Support for dedicated and shared processor partitions appropriately configured
- ▶ Support for manual provisioning of resources.

VIOS

@server p5 system with Advanced POWER Virtualization hardware feature and OpenPower system with VIOS feature provide VIOS. VIOS is a part of POWER hypervisor and resides in an IBM POWER5 partition. It facilitates the sharing of physical I/O resources between client partitions (AIX and/or Linux) within the server.

VIOS makes it possible to create partitions without requiring additional physical I/O resources owned by the new partition, enabling the creation of more partitions than available I/O slots or physical devices would normally support. The use of Virtual I/O increases the utilization of physical resources providing virtual SCSI target and shared Ethernet adapter (SEA) virtual I/O function to client LPARs:

- ▶ Virtual SCSI (VSCSI): VSCSI enables the sharing of physical storage adapters (SCSI and Fibre Channel) and storage devices (disk and optical) between logical partitions. It is a standard storage subsystem that provide standard SCSI-compliant Logical Units (LUNs). with VSCSIS you can have a pool of heterogeneous physical storage that can be treated as a homogeneous pool of storage blocks in the form of SCSI disks - “virtual disks.”
- ▶ Virtual networking: A VIOS can have a Shared Ethernet Adapter (SEA) that allows connection between internal and external Virtual LANs (VLANs) to share the access to a

physical Ethernet adapter and communicate with systems outside the server. Virtual Ethernet provides high-speed connections between partitions.

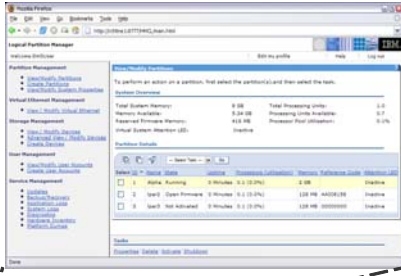
IBM Virtual I/O Server V1.2 or later is required to support IVM.

Note: The Advanced POWER Virtualization feature is available for i5 systems as well as p5 systems. It delivers the VIOS and PLM functions similar to that provided on a p5 system. However, the IVM functions cannot be used with VIOS when it is running on an i5 system. This is because IVM requires all the I/O devices on the hardware system to be owned by the VIOS partition and i5/OS requires at least one disk to be owned by i5/OS.

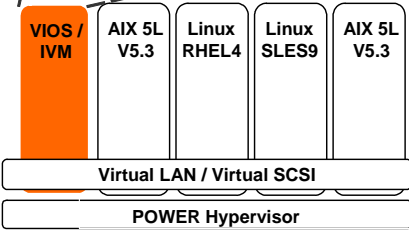
6.3 Integrated Virtualization Manager

ibm.com/redbooks
International Technical Support Organization

Hardware Management - IVM



- Web-based, intuitive/user-friendly interface
- Shipped with the Virtual I/O Server (VIOS) – provides tight integration of LPAR management and virtual storage configuration
- Provides LPAR support *without* a physical HMC – lower entry point
- Simple, Entry-level Partitioning
- Supports creating/management of I/O and LPARs within a single physical server



IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 6-3 Integrated Virtual Manager

IVM helps customers exploit the cost effective use of virtualization on IBM p5 systems. IVM provides a web-based, intuitive and user friendly interface that can be used for managing a single system without Hardware Management Console(HMC). While it does not deliver the wide scope of HMC capabilities, it does enable significant use of partitioning and other virtualization capabilities available with p5 systems.

IVM service functions are a subset of the service functions available through an HMC. The IVM partition acts as a repository and provides a management interface for “serviceable events,” and for system and service processor dumps.

The intuitive browser-based interface make it relatively easy to use (you have to have some LPAR skills). This can significantly reduce the time and effort required to set up and manage virtual devices and partitions.

IVM is shipped with the Virtual I/O Server (VIOS) V1.2 or later and provides good integration of partition management on a single managed system. It manages virtual storage and virtual Ethernet on the managed system. It also supplies service-oriented information and capabilities related to managed system.

Using IVM, the administrator can perform the following operations:

- ▶ Create and manage logical partitions.
- ▶ Configure the virtual Ethernet networks.
- ▶ Manage the storage resources.

- ▶ Download and install updates to device microcode and to VIOS software.
- ▶ Back up and restore partition configuration information.
- ▶ Display application logs and the device inventory.

At the time of writing, IVM is available on these IBM p5 systems:

- ▶ For IBM System p5 Model 505, 520, and 550
- ▶ For IBM @server p5: Model 510, 520, and 550
- ▶ For IBM @server OpenPower: Model 710 and 720

IVM Configuration

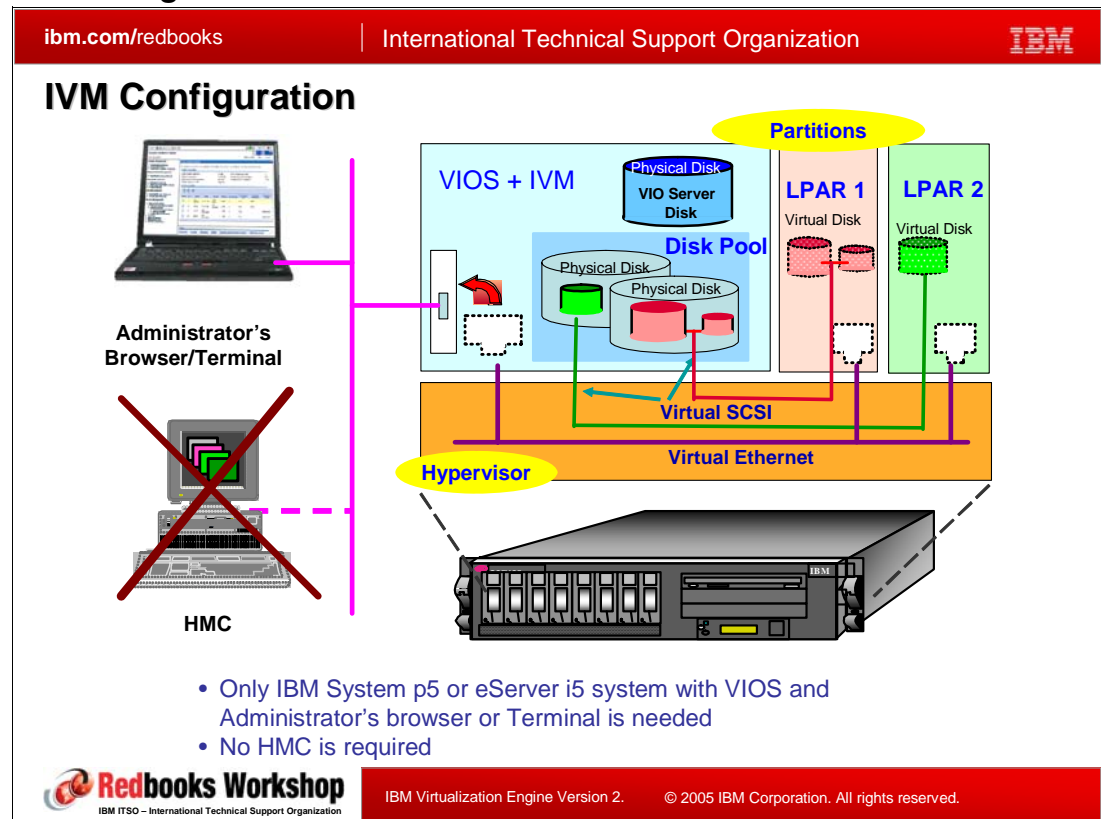


Figure 6-4 IVM Configuration

Figure 6-4 shows a sample IVM configuration. The VIOS owns all physical adapters and disks while other two client partitions are configured to use only virtual devices. The Administrator's browser or a terminal connected to IVM in the VIOS partition can be used to setup the necessary configuration.

To configure IVM, there should be no HMC connection. That is, the system is ordered from the factory to be managed by IVM - not an HMC. When you receive the new system with the APV feature, manufacturing delivers the system with IVM specified.

Switching a system between IVM and HMC management can be done, using the Advanced System Management Interface (ASMI). However, you must re-create any partition configuration and service processor personalization (such as IP configuration settings) you may have done before the change. Redpaper *Virtual I/O Server Integrated Virtualization Manager*, REDP-4061-00, discusses this (and many other IVM details). For IVM details beyond this chapter, we recommend using REDP-4061.

You can access the PDF using:

<http://www.ibm.com/redbooks>
Search for REDP4061

IVM is intended to simplify partition management, so some preset rules apply on the p5 system:

- ▶ When a system is designated to be managed by IVM, it must not already be partitioned
- ▶ VIOS needs to be the first operating system to be installed

There are a limited set of service functions for IVM because the IVM is running within VIOS. Therefore the service functions can be performed only when a VIOS partition is active. They cannot be performed if the system is powered off. To address this limitation, an authorized user can use the Advanced System Management Interface (ASMI) to perform, functions such as powering on the system. ASMI and IVM together provide a simple but effective solution for a single p5 system.

You need to initially set up and use the ASMI which comes with every POWER5 p5 and i5 system to ensure the Advanced POWER Virtualization feature is enabled. Redpaper *Virtual I/O Server Integrated Virtualization Manager*, REDP-4061-00, describes this process. You can find this redpaper using:

<http://www.ibm.com/redbooks>
Search for REDP4061

IVM manages LPARs with a basic Web interface developed to make administration tasks easier and quicker (but not as full functioned) compared to using an HMC. Since VIOS code was rewritten to integrate all virtualization tasks for IVM, IVM functions can be performed using VIOS commands.

6.3.1 Comparing VIO Server with IVM to VIO Server with HMC

ibm.com/redbooks | International Technical Support Organization **IBM**

Comparing IVM to VIO Server with HMC

- ❑ **VIO Server with IVM**
 - Easy to use
 - Workstation browser, commands interfaces - no HMC,
 - Entry-level partitioning
 - Single-server environment
 - Client LPARs have only Virtual Devices
 - Limited number of active partitions
 - Single VIO Server
 - Selected servers supported
 - Interface with Hypervisor via VMC
- ❑ **VIO Server with HMC**
 - Flexibility and comprehensive set of functions
 - HMC required
 - For redundancy, dual-HMC is possible
 - Entry and advanced partitioning
 - Dynamic LPAR etc
 - Virtual I/O and Dedicated Devices and Adapters in client LPARs
 - Large number of active partitions, multiple LPAR profiles
 - Dual-VIO Server is possible
 - Support High-Availability environments : MPIO and Shared Ethernet Adapter Failover
 - Interface with Hypervisor via HMC
- ❑ **Both IVM and VIO Server with HMC provide:**
 - Logical Partitioning (AIX or Linux Client Partitions)
 - Dedicated or Shared-CPU
 - Sub-processor CPU allocation, Capped or Uncapped, Shared Processor Pool
 - Virtual SCSI and Virtual Ethernet
 - Virtual Ethernet Bridge

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 6-5 Comparing IVM to VIO Server with HMC

IVM is not a replacement for the HMC in all environments. Rather, it is an additional offering intended to lower the cost of entry into POWER5 virtualization. Thus we need to understand the capabilities each product provides and differences between IVM and HMC to decide which solution is suitable for specific customer requirements.

Figure 6-5 is a good figure for comparing HMC with VIOS characteristics and capabilities to those available when using VIOS-IVM (no HMC). We include the Virtual I/O Server, since it is required for IVM and is required when using an MC when Virtual I/O and Virtual Ethernet are to be used.

The HMC has been designed to be the comprehensive solution for hardware management that can be used either for a small configuration or for a multi-server environment. The HMC solution may be more than a customer needs, especially in small and simple environments where partitioning is needed but only one or a few systems, and not all HMC functions are required.

IBM has developed IVM to be a simplified hardware management solution that inherits most HMC features. Its use is limited to managing a single p5 system, while avoiding the requirement for an independent personal computer - that cannot run any generally available Windows or Linux applications. IVM is designed to provide a solution that enables the administrator to reduce system setup time and to make hardware management “easy” (you do have to understand partitioning) at a lower cost compared to using the HMC.

There are differences between IVM and HMC environment in several aspects. Let’s take a look at some of the major differences.

Both IVM and VIO Server with HMC

First, at the bottom section of Figure 6-5, you see highlighted capabilities available with either choice. Logical partitioning with dedicated or shared processors, shared processor pools, capped and uncapped processor partitioning are available for supported POWER Linux distribution and AIX release levels.

Virtual I/O (Virtual SCSI) and Virtual Ethernet for TCP/IP based communication among partitions within the system itself are available. A Virtual Ethernet bridge to a real physical LAN device (adapter) is supported. The Virtual Ethernet enables the partitions to communicate with each other at maximum speeds over a “LAN” that is not exposed to security attacks and communication line errors that external lines can be subjected to.

A physical disk attached to the p5 system or logical unit number (LUN) on a storage area network (SAN) can be used for “virtual disk” support. Virtual disk protection and space management are provided.

VIO Server with IVM

As previously mentioned a browser (GUI) or virtual terminal emulator (command line interface) is used instead of an HMC. IVM as part of a VIOS Partition is considered the “controlling partition.” in contrast to an HMC managed configuration where there is no controlling partition on the system.

The client workstation using a terminal emulator or a browser, can also be running any application supported by the workstation’s operating system, for example any generally available Windows or Linux application. This is logically equivalent to a Windows or Linux workstation using remote WebSM to a powered on HMC.

Although the slide in Figure 6-5 uses the term “entry-level partitioning” you get a wide range of LPAR capabilities supporting only virtual I/O and a single p5 system. You can run up to 4 client partitions (AIX, Linux) plus the single VIOS partition.

Dynamic LPAR, is available, with constraints, when compared to DLPAR capabilities under an HMC-based configuration:

- ▶ Resources such as processors and memory can be dynamically allocated or released on the IVM (VIOS) partition. You can run those operations either on the GUI or the Command Level Interface (CLI).

The GUI does not allow dynamic operations on resources such as processor, memory, and virtual Ethernet on any of the client partitions. This is in contrast to the dynamic resource - processor, memory, and virtual Ethernet, movement among any partitions when using an HMC. This is because IVM cannot interface to the POWER5 Remote Monitoring and Control (RMC) protocol that allows the hardware management function to interact with an operating system’s resource allocation.

While IVM is capable of performing a dynamic LPAR operation on itself (the VIOS partition), changes on other partitions are not applied until the partition is restarted.

- ▶ Dynamic operations on disks, optical devices, partition name, and boot mode are allowed for the configured client partitions

IVM communicates with the Hypervisor via Virtual Management Channel (VMC) which is a new “virtualization component” recently introduced that is discussed later in 6.3.2, “IVM design” . The HMC interfaces directly to the Hypervisor.

One consideration worth mentioning here is, should the IVM/VIOS partition be shutdown, the other partitions cannot continue to run once they attempt to address a defined “virtual device” - owned by the inactive IBM/VIOS partition.

VIO Server with HMC

With the HMC managing your system you get a full range of dynamic capacity on demand and resource movement among all active partitions:

- ▶ Full partition definitions and management for all supported operating systems with up to approximately 254 partitions. The upper limit of partitions that can be activated is determined by the model of the POWER5 system being managed
- ▶ There is no “controlling” or “primary partition.” So any partition can be active or shutdown with out causing the other partition to also shutdown. This is true unless a VIOS partition is managing the virtual devices for other partitions and the VIOS partition is shutdown.
- ▶ Multiple VIO Servers can be defined. This enables another level of higher availability for partitions defined to use virtual I/O - should you have to shutdown a specific VIO Server partition.
- ▶ Each partition can use virtual I/O or directly attached dedicated devices and adapters
- ▶ Assuming appropriate partition configuration settings, processor and memory resources can be moved manually and dynamically among any of the partitions
- ▶ Multiple partition profiles can be defined and specific ones activated as needed for particular LPAR environments
- ▶ HMCs can be defined to manage the same system and an HMC can be setup to take over the HMC functions for another system, should another HMC not be operating.

You need to keep in mind that what we have covered here is not the full list of differences between managing a system with IVM and managing with an HMC. Consult IVM and HMC comparison table for more information in Appendix A, “Integrated Virtualization Manager versus Hardware Management Console” on page 361, as well as the redpaper Virtual I/O Server Integrated Virtualization Manager, REDP-4061.

6.3.2 IVM design

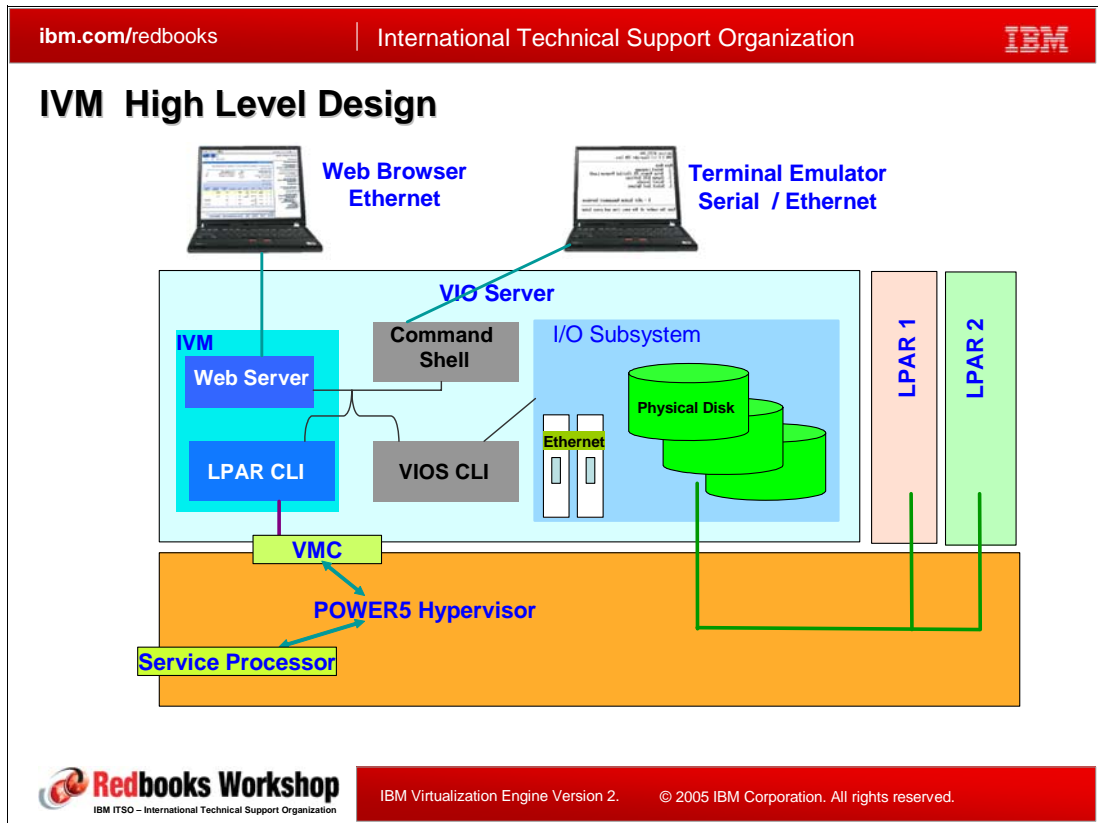


Figure 6-6 IVM High Level Design

Figure 6-6 shows a simple schema of IVM architecture.

IVM provides both a browser-based graphical interface, and a command line interface.

The only requirement to use browser-based graphical interface is to have a computer with a web browser with network access to the VIOS partition. The IBM-provided web server running in the VIOS partition supports the primary user interface through a web browser that connects to port 80 of the VIOS. The web server provides a simple GUI and executes commands using the same command line interface that can be used logging on the VIOS via a terminal emulator as shown.

The command line interface (CLI) uses similar syntax to the HMC command line interfaces and supports three sets of control functions.

- ▶ Partition Management: Commands for all partitioning functions. These are integrated into the specific IVM command set; they are available only when the system is in IVM management mode.
- ▶ Virtual I/O: Commands for virtual I/O provisioning functions that support virtual devices.
- ▶ Service and maintenance: Commands for service functions, such as VIOS partition user management, serviceable event management, and firmware/microcode management.

Virtual devices, rather than actual hardware devices and adapters are available to all client partitions in the VIOS-IVM environment.

IVM is able to perform limited functions on client LPARs such as active/deactive, send power off (EPOW) signal to the operating system, create/delete configurations, and view/change configurations.

Virtual Management Channel (VMC)

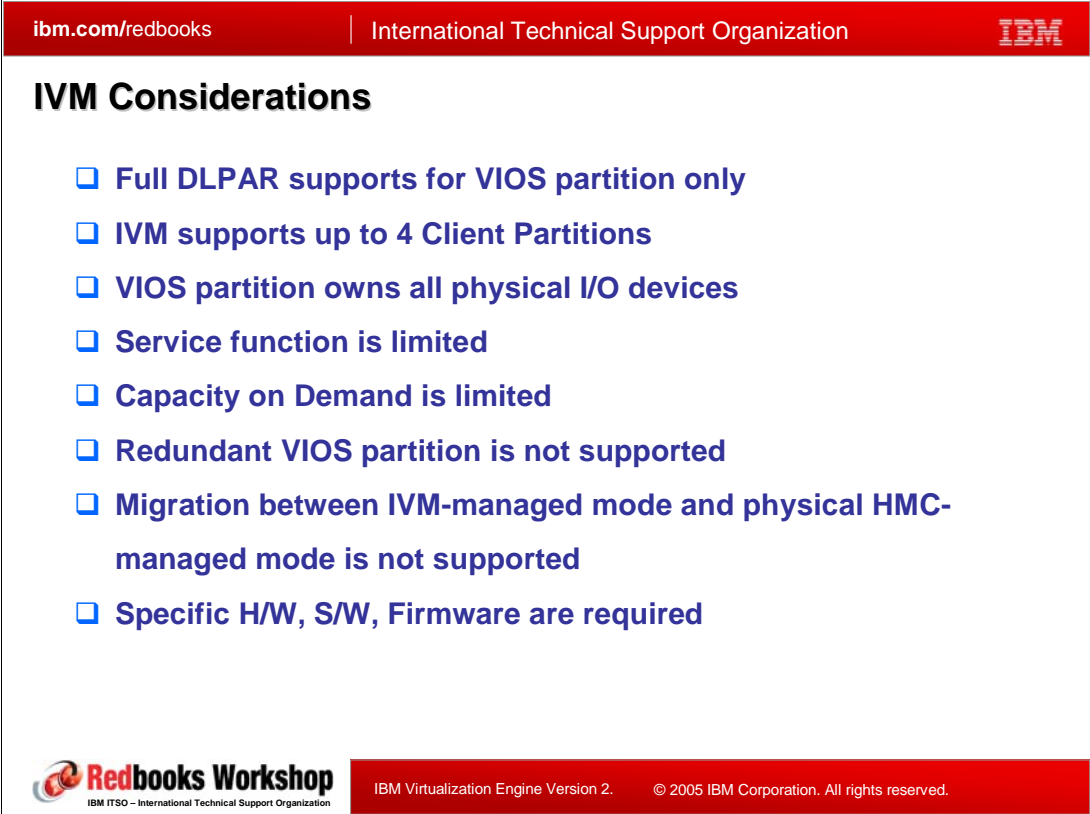
This is the special “virtual device” introduced to support IVM. The POWER5 Hypervisor is key to creation and management of partitions; therefore an interface with it is required to perform the IVM functions. The HMC uses the Service Processor to communicate with Hypervisor. Since IVM replaces the HMC, the new Virtual Management Channel enables communication between the IVM and Hypervisor. This VMC is available only on an IVM-managed System p5, an IBM eServer pSeries, and an IBM OpenPower system. Since VMC has been developed on the VIOS to enable a direct Hypervisor configuration without requiring additional network connections, IVM does not need the system’s Service Processor to interact with Hypervisor. If VIOS is installed as the first partition, VMC is activated by default.

IVM can perform basic LPAR functions using the VMC:

- ▶ Create logical partitions.
- ▶ Boot, start, and stop individual partitions.
- ▶ Controlling and monitoring partition status.
- ▶ Manage virtual devices.
- ▶ Basic system administration.

When IVM is activated, a new VMC virtual device **ibmmvc0** is automatically created and the IBM-provided Web server is started. IVM communicates with Hypervisor through VMC, so only the system on which the VIOS-IVM partition is located can be controlled. Therefore, a system administrator needs to contact each IVM managed system one by one in order to perform its management capabilities on that system.

6.3.3 IVM considerations



ibm.com/redbooks | International Technical Support Organization **IBM**

IVM Considerations

- ❑ Full DLPAR supports for VIOS partition only
- ❑ IVM supports up to 4 Client Partitions
- ❑ VIOS partition owns all physical I/O devices
- ❑ Service function is limited
- ❑ Capacity on Demand is limited
- ❑ Redundant VIOS partition is not supported
- ❑ Migration between IVM-managed mode and physical HMC-managed mode is not supported
- ❑ Specific H/W, S/W, Firmware are required

Redbooks Workshop
IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 6-7 IVM Considerations

As we discussed before in this chapter, IVM is not a replacement for the HMC in all environments but an additional offering intended to lower the cost of entry into POWER5 virtualization.

Before you choose IVM as your solution, you need to understand the limitations and considerations required to enable IVM on your @server p5 system. Review the following topics based upon the main bullets shown in Figure 6-7, to determine whether this feature is suitable for your requirements. Some of this information has been covered earlier in this chapter but it summarized together here.

Full DLPAR support for VIOS partition only

Since only VIOS can contact with Hypervisor and be able to react to hardware configuration changes, it can change its resource configuration dynamically while client lpars can't. Therefore the client lpars must keep the same resource configuration while running. However, it can configure remaining I/O configurations dynamically:

- ▶ It can assign an optical device to any virtual SCSI channel.
- ▶ It can create, delete, or assign a virtual disk to any virtual SCSI channel.
- ▶ It can create, delete, or change ethernet bridging between a virtual network and a physical adapter at any moment.

IVM supports up to 4 client partitions

Using IVM, the user can configure up to other “non-IBM partitions” than can be set up using s micro partitioning (less than a full processor capacity partition) uncapped processor

capacities (system determines a partition needs more processing power, and, assuming additional capacity is available on the system, add processor power to the “needy partition”). The client partitions can run supported levels of AIX 5L V5.3 or later and POWER Linux distributions. It does not support an i5/OS partition.

VIOS partition owns all physical I/O devices

The VIOS partition own all physical I/O resources, with the following considerations:

- ▶ Maximum of 2 VLAN adapters per client LPAR.
- ▶ Maximum of 4 VLAN adapters per system (fixed VLAN IDs:1,2,3,4)
- ▶ One VSCSI adapter per client LPAR.

Even though each client LPAR can have only one Virtual SCSI adapter, multiple virtual disks can attach to the associated host Virtual SCSI adapter, if desired. For example, a user can assign one virtual disk for the partition boot disk, and another virtual disk for a data or application disk.

Service Function is limited

The are important service limitations that may or not be a requirement for your environment.

- ▶ No Concurrent Firmware Maintenance (CFM) Firmware updates are supported, because this function requires an HMC.
- ▶ No Service Focal Point (SFP) for Client LPARs
- ▶ No hot addition or removal of I/O drawers to the system

An eserver P5 system managed by an HMC can update system firmware being supported by Concurrent Firmware maintenance(CFM). However, VIOS is not able to update any system firmware concurrently.

The IVM partition provides a subset of the service functions that are available on HMC. Instead of client LPARs itself, IVM provides a repository and a management interface for Serviceable Events, and for system and service processor dumps.

Capacity on Demand is limited

Since the Advanced System Management Interface (ASMI) is responsible for Capacity on Demand (CUoD) functions, ASMI can be activated in IVM environment. However, the IVM environment does not support the other capabilities such as On/Off or Trial capacity because it needs an HMC to monitor and activate resources.

The IVM partition is enabled to perform a limited set of actions on the other partitions, such as:

- ▶ Activate and deactivate
- ▶ Send a power off (“EPOW”) signal to the partition’s operating system
- ▶ Create and delete partitions
- ▶ View and change partition configuration

Redundant VIOS Partition is not supported

In an IVM environment only one VIOS partition owns all physical I/O devices and adapters. Therefore creating multiple VIOS partitions for redundancy is not possible.

Migration between IVM-managed and physical HMC-managed is not supported.

If HMC is connected to system running IVM, the IVM interface is immediately disabled and the HMC takes over management of the system. If you do this, your logical partition data or virtual I/O configuration is lost and there is no way to recover except restoring backup. That is any logical partition configuration done through IVM is gone and you need to recreate the partitions as if there was never any partition configuration done.

Specific H/W, S/W, Firmware are required

IVM is supported on the following hardware and software configurations

- ▶ Hardware:
 - IBM System p5: Model 505, 52A, and 55A
 - IBM @server p5: Model 510, 520, and 550
 - IBM @server OpenPower: Model 710 and 720AVP activation
- ▶ Software:
 - Virtual I/O Server(VIOS) v1.2 or later
 - AIX or Linux Client Operating Systems
 - AIX 5L V5.3 or Later
 - Suse Linux Enterprise Server 9 for Power or later
 - Red Hat Enterprise Linux AS 3 for POWER (update 2 or later)
 - Red Hat Enterprise Linux AS 4 for POWER
- ▶ Firmware:
 - Microcode level SF235 or later is required

6.3.4 VIOS- IVM user interface

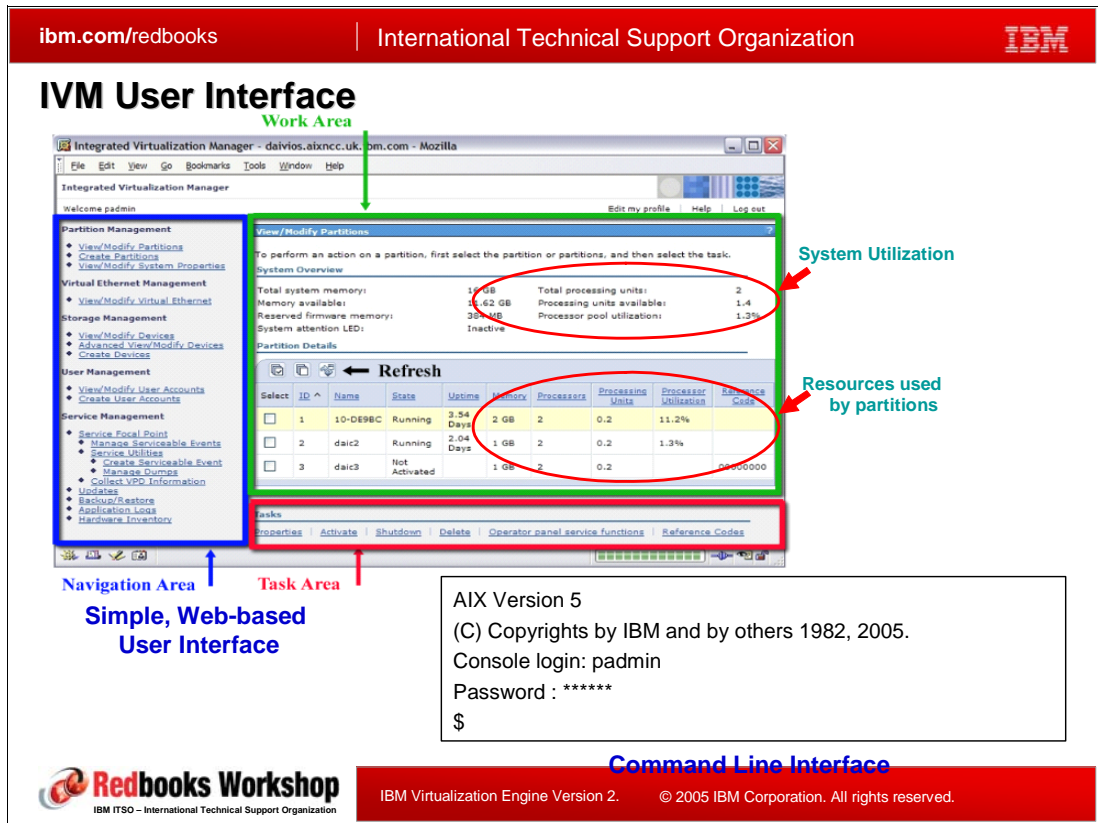


Figure 6-8 IVM User Interface

Figure 6-8 shows the two IVM User Interfaces that enable you to create and manage logical partitions. In the upper left we show the IVM graphical user interface (GUI) and in the lower right a logical representation of the Command Line Interface (CLI).

The IVM offers a simple and intuitive interface, but does have some constraints; The GUI has functions and view information grouped into three major areas:

- Navigation area** The navigation area displays the tasks that you can access in the work area.
- Work area** The work area contains information related to the management tasks that you perform using IVM and to the objects on which you can perform these management tasks.
- Task area** The task area lists the tasks that you can perform for items displayed in the work area. The tasks listed in the task area can change depending on the page that is displayed in the work area, or even depending on the tab that is selected in the work area.

The text based console Command Line Interface (CLI) is less intuitive compared to the GUI, but enables more functions with greater flexibility. You can access it through a native TTY serial port or network connectivity using telnet. The command shell login for the IVM partition is *restricted shell* because the VIO Server (and thus IVM) is considered as an “appliance” server or a “closed system”.

6.3.5 IVM and VIOS initial setup and partition preparation

VIOS preparation overview

System Overview

Total system memory:	4 GB	Total processing units:	2
Memory available:	0 MB	Processing units available:	1.8
Reserved firmware memory:	304 MB	Processor pool utilization:	0.8%
System attention LED:	Active		

Partition Details

Select	ID	Name	State	Uptime	Memory	Processors	Processing Units	Processor Utilization	Reference Code
<input type="checkbox"/>	1	10-5E9DE	Running	8.8 Minutes	3.7 GB	2	0.2	4.0%	

Initial Partition Configuration

- VIOS partition preparation
 - Initial network setup
 - VIOS partition configuration
 - Network Management
 - Storage Management

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 6-9 VIOS partition preparation overview

Initial network setup

To connect IVM Web interface to the system, a proper network configuration is needed. IVM uses web servers to communicate, so you need to be careful of both the name and IP address resolution of the IVM's hostname. We do not present here, detailed considerations for setting up the specific IP address to connect your browser to IVM. We refer you to the Installation chapter of Redpaper *Virtual I/O Server Integrated Virtualization Manager*, REDP-4061-00.

We described earlier how to find this redpaper and provide it here again

<http://www.ibm.com/redbooks>
Search for REDP4061

VIOS partition preparation

Figure 6-9 displays an example VIOS - IVM window demonstrating an “in-progress” reduction of its resources in preparation for client partition creation. In 6.3.4, “VIOS- IVM user interface” , we discuss VIOS partition configuration briefly to show you the simplicity of VIOS partition configuration and management.

Figure 6-9 shows the first screen after the first log in to IVM as the padmin user. At this initial connection, all the available memory in the system is allocated to the IVM partition and a yellow attention icon with an exclamation mark is present. When the VIOS partition is created during the installation, entire memory is assigned to VIOS partition but committed, which causes this attention icon - “warning sign” appears.

You need to change the amount of memory assigned to the VIOS partition to guarantee there is sufficient free memory to accommodate client partitions that will be created. Dynamic memory deallocation is supported for VIOS partition and the action is made gradually over several seconds. Allow this action to complete before going on to the next step.

IVM Network Management

One network device for each physical Ethernet present on the system and four virtual Ethernet adapter is created during the installation. At installation time, no bridging between a physical adapter and virtual adapter has be set up.

IVM Storage Management

The IVM uses a little different concept for storage management compared to pSeries AIX 5L. The following is IVM's view and terms for storage:

Physical volume	A physical disk or a logical unit number (LUN) on a storage area network (SAN). They are all owned by IVM.
Storage pool	A set of physical volumes treated as a single entity. There may be multiple pools but they can not share physical disks. One pool is tagged as <i>default storage pool</i> .
Logical volume	A portion of a storage pool. It may encompass more than one physical volume.
Virtual disk	Disk space that IVM provides to a single partition. It can be either a physical volume or a logical volume.

At first, VIOS has only one storage pool named *rootvg*, which is the default storage pool. Rootvg has only one physical volume usually and any other physical volumes on your total hardware configuration are unassigned to any storage pool.

To work effectively you must create at least one other storage pool and set it as the default storage pool before creating any client partitions. This is because the rootvg pool is good only for IVM management purposes - not for use by other partitions.

As stated, both physical volumes and virtual disks can be assigned to a partition to provide disk space. Each of them is represented by the LPAR operating system as a single disk. For example, assigning a 73.4 GB physical disk and a 3 GB virtual disk to an LPAR running AIX 5L makes the operating system create two *hdisk* devices.

6.3.6 Client LPAR Creation

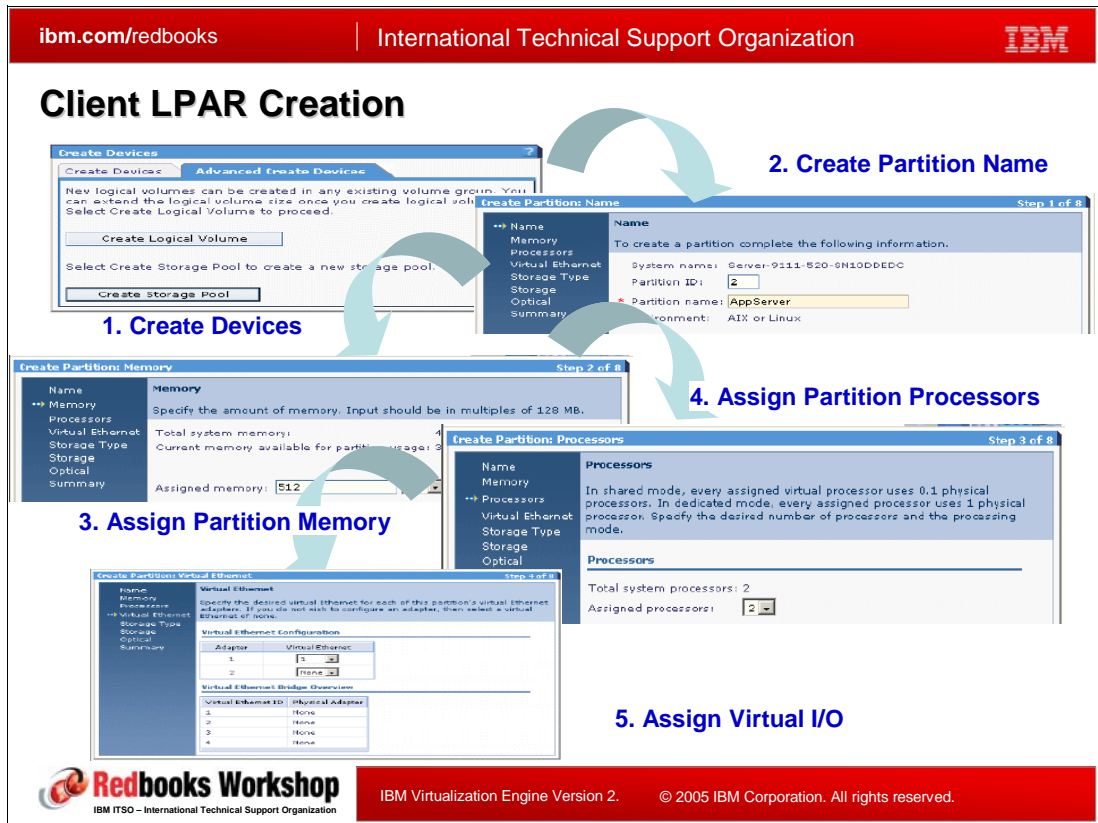


Figure 6-10 Client LPAR Creation

Figure 6-10 shows the major steps to create client LPAR. As you can see from Figure 6-10, you can make a client partition with just five simple steps which require a couple of input values from you.

Furthermore, IVM provides a default value for each entry, if possible. In most cases you need only to decide the amount of resource you think is required. Compared to the HMC environment, the IBM interface minimizes the time and energy required to create your own LPARS in p5 system.

Important: Regardless of using IVM or the HMC interface to configure a partition, you still need to do some initial sizing estimations based upon the planned workloads for each partition. You want to use your best estimation for the initial resource value to be entered for memory, processors, and virtual disks and disk storage, based upon the planned application workload in the partition. This estimation/planning needs to include the total system resources available when the identified concurrently running partitions are active.

You can certainly take advantage of the on-demand resource changes available under either IVM or the HMC capabilities, but poor planning can result in more demand than there are available resources.

“LPAR planning” is beyond the scope of this guide. This subject is addressed in many redbooks on LPAR and within IBM Information Center documents.

Create Devices

IVM can create virtual disks with a physical volume and/or logical volume. Because creating logical volumes for a client partition from the IBM provided *rootvg*, default storage pool is not recommend, you need to create at least one additional storage pool and make that the default storage pool used by non-IVM partitions.

A storage pool can have a number of physical disks with different type and size. Vastly different sizes should not impact performance, given the POWER5 technology disk controllers and disk arm. Note that although you can create multiple storage pools, the same physical disk can be included in only a single storage pool at a time.

Recall the example used previously. Assign a 73.4 GB physical disk and a 3 GB virtual disk to an LPAR running AIX 5L. This makes the AIX 5L operating system create two *hdisk* devices.

Depending on your actual hardware configuration you could add additional physical and logical disks to a partition.

Create Storage Pool

You can create a storage pool with GUI interface as following:

- ▶ In the browser's navigation area, select the **Create Device** link under **Storage Management** -> **Advanced Create Devices** in the work area → **Create Storage Pool** → input a name for **Storage pool name** → select disks as needed. Click **OK**.
- ▶ You need to set a default storage pool to storage pool you created other than *rootvg*
- ▶ In the navigation area, select **Create Device** link under **Storage Management** → **Advanced View/Modify Devices** → select the storage pool you want to be the default → click **Assign as default storage pool** in the task area. Click **OK**.

Logical volume creation

A storage pool can have multiple Logical Volumes (LVs) and serve them to a client partition as virtual disks providing disk spaces for the client partition. A client partition can have a number of virtual disks. However one LV can be assigned to one client partition at a time. You can create a Logical Volume with GUI interface as follows:

- ▶ In the navigation area, select **Create Device** link under **Storage Management** → **Advanced Create Devices** in the work area → **Create Logical Volume** → Type a name in **Logical Volume Name** → select a **Storage Pool Name** from the drop down list → define the amount of disk spaces in **Logical Volume Size**. Click **OK**.

Create Partition

A client LPAR needs a set of resources, including processors, memory, and I/O devices. Unlike LPARs managed with an HMC, IVM allocates resources to a client LPAR at the time of assignment and does not allow other client partitions to use them regardless of whether the assigned LPAR is active or down. Therefore, you need to understand the usage of these resources in defined partitions, so as to not end up overcommitting real resources.

Create Partition Name

You can create a partition and specify a Partition Name with the GUI interface as follows.

In the navigation area, select **Create Partitions** link under **Partition Management** → **Start Wizard** in the work area → Enter the client LPAR name for **Partition name** field. Click **Next**.

You do not need to supply Partition ID because it appears with default value.

Assign Partition Memory

When you click **Next** after you input the variables for **Create Partition Name**, the screen for configuring memory appears. You can define the amount of memory for a client partition with GUI interface by inputting the amount of the memory needed in the **Assigned memory** field. You can also set the unit of memory, for example MB or GB. Click **OK**.

Assign Partition Processor capacity

IVM simplifies the configuration settings for Micro-Partitioning using an allocation number of processors. A partition can be defined either with dedicated or with shared processor types. The wizard provides you with the available resources in both cases and prompts for which processor resource type you want to use with the partition.

When shared processors are selected for a partition, you are asked only to choose the number of virtual processors to be activated, with a maximum value equal to the number of system processors. For each virtual processor, 0.1 processing units are implicitly assigned. If you assign 3 virtual processors to a client LPAR then the partition has 3 virtual processors which means 0.3 physical processors.

Partitions can be defined to be in either dedicated mode or *uncapped* mode. Dedicated mode means that “n” processors have to be available to start the partition and that same value of “processor units” is available during the time that partition is active.

By default a shared processor partition is created to be in uncapped mode with a weight value of 128 (out of 256 weight values). Uncapped means that when the workload of an uncapped partition is high enough (that is consuming 100% of the currently allocated processor units), the hypervisor may be able to “add” CPU resources to that partition, assuming there is available “processor power” in the shared processor pool.

How additional processor units are dynamically added to a partition is determined by the number of processors assigned to the shared pool and the amount of “processor power” that is being consumed at that time by other uncapped and active partitions. This is where the weight value of a partition and its current processor consumption come in to play.

Complete coverage of this subject is beyond the scope of this presentation guide. However, the subject reminds us that you to take the most effective advantage of uncapped partitions, you need to estimate the amount of processor power consumed by the currently active partitions throughout the business day. Then you need to consider the weight value assigned to each partition.

In general a partition running a critical application that may need, for times of peak workload activity, more processor power, consider assigning a higher weight value to that partition than the weight value of another active partition. This presumes this “other partition” may be able to “run at a degraded performance” level - if the higher weight value partition needs the processor resource.

If you have all partitions with the same weight value and all partitions are active, the advantages of an uncapped partition could be significantly reduced. The objective of an uncapped partition is to enable that partition that needs more processing power to get the additional processor power, but you need to understand the available processor power and the needs of other active partitions defined as uncapped.

If all active partitions have the same weight value, and partition ONE is currently consuming over 100% of its originally assigned processor units and a second partition now needs more than its 100% originally assigned processor units this second (same weight value) partition may not get any additional processor units.

When you click **Next** after you input the variables for **Assign Partition Memory**, the screen for configuring process appears. You can define the number of processors for client Lpars with GUI interface by inputting the number of processors needed in the **Assigned processor** field.

You can also select one of the **Shared** or **Dedicated mode** for Processors in the **Processing Mode** parameter. Click **Next**.

Assign Virtual I/O

As IVM automatically supplies one pre-defined virtual ethernet and one virtual SCSI adapter for each of client partition, you do not need to take extra steps to create virtual I/O devices. Remember, a single client partition can have up to 2 VLAN adapters and up to one VSCSI adapter. Also, a single system managed by IVM cannot have more than four VLAN adapters.

You can define the Virtual Ethernet for client LPARS on the screen that automatically appears as soon as you finish the **Assign Partition Process** step with GUI interface. Each entry has a default value, so you can accept it without modification when only one VLAN is needed. If you want to create and additional virtual ethernet, you can do this now or do it later.

After you click **Next** in **Virtual Ethernet** screen, you see the screen for specifying virtual disks.

Select **Assign existing virtual disks and physical volume in the work area**. You may also create another new virtual disk for this client at this stage by selecting **Create virtual disk**, which is located in the default storage pool. Remember the earlier recommendation to have already created an additional storage pool to be defined as the default pool.

Click **Next** to have IVM display the list of available physical volumes and available virtual disks which we have already defined. Select physical disks and/or virtual disks in the list that you need. You also specify the size of physical volumes and virtual disks, considering the amount of disk space you have estimated you need for this partition.

Click **Next** to select optional devices such as a DVD/CD, from the **Available Optical Devices** menu.

Click **Next** and the **Summary** window display appears. This summarizes all the default values and the ones you previously entered for the client partition about to be created. Click **Finish** and the partition creation process is performed.

Dynamic relocation of optical devices is available for a client partition, so you can reallocate optical devices dynamically from or to any logical partition, whether the logical partition is running or not.

These steps we explained in this chapter but not e complete procedure. See the REDP4061 *Virtual I/O Server Integrated Virtualization Manager* redpaper for more details.

The information presented here is intended to help you understand the IVM configuration process, the values you can select, and some partition planning considerations need to get you productively started with a the partitioning environment, without requiring extensive LPAR skills.

This ends the topics on the Virtual I/O Server Integrated Virtualization Manager capabilities for IBM System p5 and IBM eServer p5, and OpenPower solutions. Remember the VIOS and IVM capabilities are included in the Advanced POWER Virtualization hardware feature.

Then next set of topics address the IBM System i5 and IBM eServer i5 Virtual Partition Manager (VPM) component that provides LPAR functions logically equivalent to VIOS/IVM functions and capabilities. Note that VPM interfaces are “in tune” with LPAR configuration support on AS/400 and iSeries systems prior to the availability of POWER5 technology.

6.4 Virtual Partition Manager

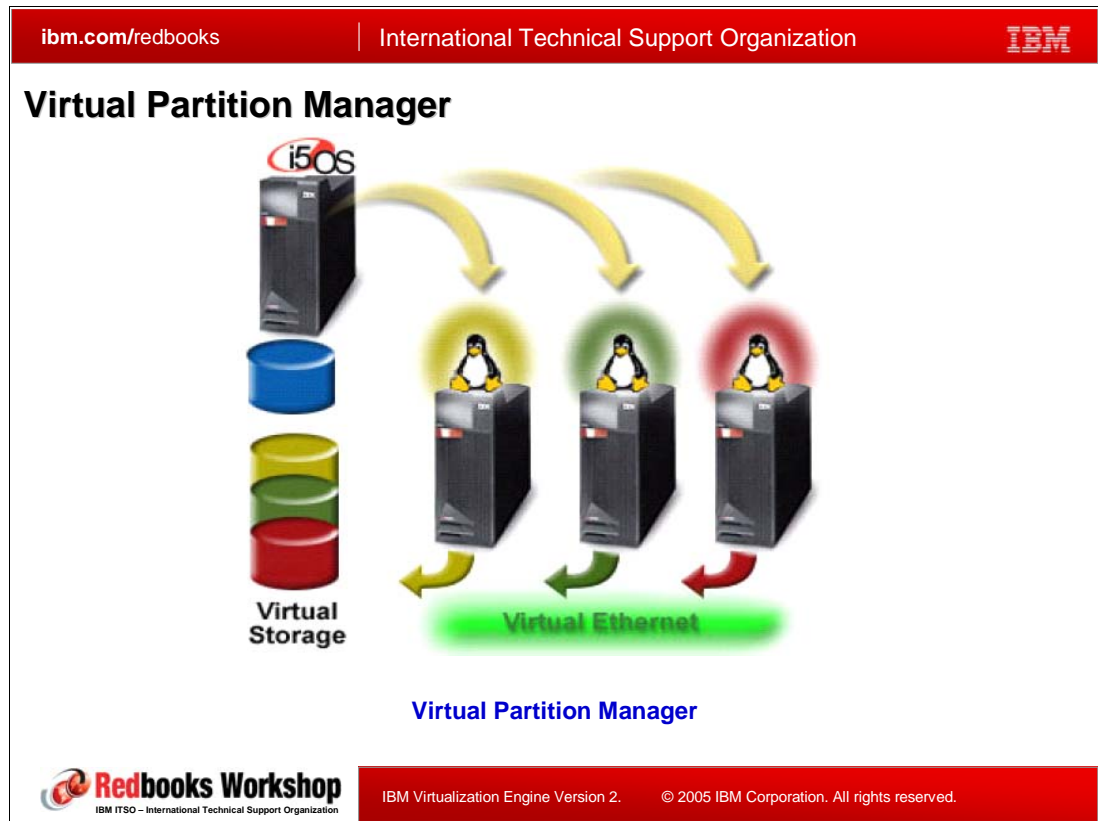


Figure 6-11 Virtual Partition Manager

Figure 6-11 depicts Virtual Partition Manager (VPM) on a POWER5 i5 system. VPM provides a good solution in a economical and easy way for small and medium iSeries-based customers who want to simplify their infrastructure with their @server i5 Server and want to run Linux partitions. This capability, is included at no additional cost as part of i5/OS.

Note that for information details beyond the content of this chapter, refer to the redpaper *Virtual Partition Manager: A Guide to Planning and Implementation*, REDP-4013-00, published April 2005. You can access the PDF for this at

You can access the PDF using:

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

Search for REDP4013 or “Virtual Partition Manager”

VPM has very similar functionality to that of IVM, which we discussed in the previous sections. It does, however, have distinctly different characteristics and interfaces. Not only is its user interface different from IVM's use of a browser to an internal HTTP server, but also its supported operating systems are different:

- ▶ VPM enables the creation and management of Linux partitions without the HMC
- ▶ A single i5/OS partition with up to four Linux client partitions
- ▶ Linux partitions can use only virtual I/O devices which are owned by the i5/OS partition
- ▶ Configuration is done through the 5250 workstation text-based interface available using the iSeries service tools interfaces

You can see an AIX 5L client partition is not supported by VPM. While IBM System i5 and IBM eServer i5 i5/OS and Linux partition support is for POWER5-based systems, its LPAR support is very close to the LPAR support provided on AS/400 and iSeries systems prior to the availability of POWER5 technology.

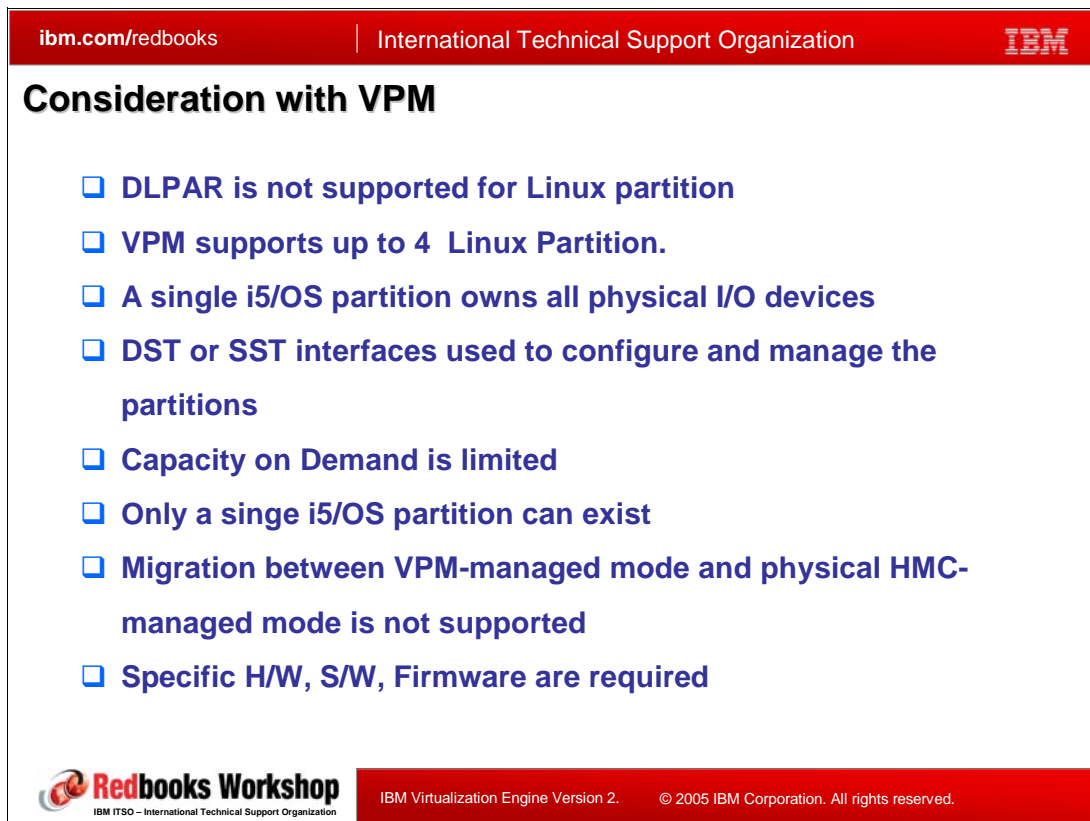
Table 6-1 summarizes the key differences between the Virtual Partition Manager(VPM) and the Hardware Management Console(HMC) in POWER5 based iSeries system.

Table 6-1 Differences between VPM and HMC

	Virtual Partition Manager	Hardware Management Console
Operating System supported	i5/OS and Linux	i5/OS, Linux and AIX
Maximum number of partitions	5 (1 i5/OS and 4 Linux)	254
Uncapped Partition support	Yes	Yes
Dynamic Resource Movement	No	Yes
I/O Support for Linux	Virtual	Virtual and Direct
Maximum # of Virtual Ethernet connections	4	4096
Maximum Virtual Disk storage per partition	64TB	64TB

If you have requirements for multiple i5/OS partitions, AIX 5L partitions, flexible hardware resource assignments, and dynamic resource movement, the full functioned Hardware Management Console is required.

6.4.1 Limitations and considerations with VPM



The screenshot shows a slide from the Redbooks Workshop. The slide has a red header with the text 'ibm.com/redbooks' on the left, 'International Technical Support Organization' in the center, and the 'IBM' logo on the right. The main title of the slide is 'Consideration with VPM'. Below the title is a list of eight bullet points, each preceded by a blue square icon. At the bottom of the slide, there is a red footer containing the 'Redbooks Workshop' logo on the left, and the text 'IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.' on the right.

ibm.com/redbooks | International Technical Support Organization IBM

Consideration with VPM

- ❑ DLPAR is not supported for Linux partition
- ❑ VPM supports up to 4 Linux Partition.
- ❑ A single i5/OS partition owns all physical I/O devices
- ❑ DST or SST interfaces used to configure and manage the partitions
- ❑ Capacity on Demand is limited
- ❑ Only a single i5/OS partition can exist
- ❑ Migration between VPM-managed mode and physical HMC-managed mode is not supported
- ❑ Specific H/W, S/W, Firmware are required

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 6-12 Consideration with VPM

Figure 6-12 gives a good summary of key VPM considerations. VPM is not intended as an HMC replacement in all environments. Rather, VPM is an additional offering intended to lower the cost of entry into iSeries-based POWER5 virtualization capabilities - in customer environments where the richer partitioning and virtualization capabilities of the HMC are not required.

Before you choose VPM as your solution, you need to understand the VPM limitations and considerations for required to enable VPM on your @server i5 Server. Review Table 6-1 on page 241 and the following topics to help determine whether VPM is suitable to your environment.

DLPAR is not supported for VPM environment

You can use uncapped processor pool which enable system to balance processor automatically. However you can not move resources such as processor, memory, and I/O dynamically. When resources are changed in i5/OS or/and Linux partition, IPL of the system is required.

VPM supports up to 4 Linux Partitions

Using VPM, the user can configure one i5/OS partition and up to 4 Linux partitions including micro partitioning with uncapped processors.

A single i5/OS partition owns all physical I/O devices

- ▶ Linux partition uses virtual I/O to access disk, tape, DVD, and Ethernet resources owned by the i5/OS partition.

- ▶ Linux partition can not have direct attached I/O devices including any additional ethernet adapters for a firewall gateway.
- ▶ Maximum of 4 VLAN adapters per Linux partition or for the i5/OS partition.

DST or SST interfaces used to configure and manage partitions

VPM support is provided through either Dedicated Service Tools (DST) or System Service Tools(SST) interfaces. These interfaces include many iSeries hardware service oriented functions as well as support for partition configuration and management. DST provides a superset of SST functions - providing more detailed hardware functions than SST does. DST and SST interfaces are already familiar to iSeries administrators who have configured and managed partitions on iSeries systems that are not using POWER5 technology.

iSeries administrators using DST or SST can setup and manage Linux partitions using VPM on IBM system i5 systems and eServer i5 systems without the additional need to learn and use the Hardware Management Console.

These DST and SST partitioning functions on i5 systems are logically similar to the p5 system IVM functions.

Using VPM prohibits the use of the iSeries Access for Windows (product 5722-XE1) iSeries Navigator LPAR management function. Additionally partition configuration data cannot be saved through DST or SST tasks. You need to keep hardcopy print out with configuration screens to recreate the partitions.

Also, VPM can not be managed through IBM services offerings such as the LPAR Toolkit or similar LPAR management tools provided by various IBM business partners. Likewise, there are no Application Programming Interfaces (APIs) or Secure Shell SSH capabilities enabled to support programmed LPAR management tools.

Capacity on Demand is limited

DST or SST do not support Capacity on Demand such as On/Off CoD and Reserved CoD because these tasks require HMC. Only permanently activated processors can be allocated to any of the partitions.

Only a single i5/OS partition can exist

In VPM environment i5/OS partition owns all physical I/O, so creating multiple i5/OS partition is not possible. This is logically equivalent to IVM supporting only one VIOS partition for the p5 systems. If need more than one i5/OS partition on you i5 system, the HMC management is required.

Migration between VPM-managed mode and physical HMC-managed mode is not supported

You can not migrate partition configuration data to the HMC. If HMC is deployed at a future stage, you need to recreate the Linux partitions. However, the data stored through virtual I/O on i5/OS remains unaffected.

Note that you cannot use Virtual Partition Manager on an i5 system that has already been configured using an HMC.

Specific hardware, software, firmware level requirements

There are important requirements for Software and Firmware that enable deployment of a Linux partition with VPM.

- ▶ Software

i5/OS Version 5 Release 3 or later

- Prerequisite PTFs MV34946 and MF34962 or supersedes, which must be permanently applied before doing a SAVSYS (save system) of the i5/OS partition.
- PTF MF34753 or its supersede PTFs

- ▶ Linux for Client Operating Systems

IBM has announced Linux distributions from Red Hat and Novell support the i5 systems. Specifically, the distributions that run in logical partitions on an i5 system are:

- Red Hat Enterprise Linux AS for POWER Version 3 and Version 4
- SUSE LINUX Enterprise Server 9 for POWER

Novell and Red Hat have changed their product names to state "for POWER" to indicate the support for eServer i5, eServer p5, iSeries, pSeries, and other POWER processor-based offerings

For the latest information on Linux distributions running on iSeries and i5 systems, refer to the following URL

<http://www.ibm.com/eserver/series/linux>

Scroll down the first page to see Linux distribution release level information.

- ▶ Firmware level

SF230_126 [PTF MH00312] or later PTF

You can find additional hardware information, logical partition planning tasks, and details on how to create a Linux logical partition using i5/OS virtual I/O resources from URLs below:

- ▶ eServer Information Center

<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/index.jsp>

Select IBM Systems Hardware Information Center and expand the left navigation area tree structure as shown in Figure 6-13.

- ▶ Redbooks

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

- Redbook *AIX 5L on IBM eServer i5 Implementation Guide*, SG24-6455-00
- Redbook *Linux on IBM eServer i5 Implementation Guide*, SG24-6388-00

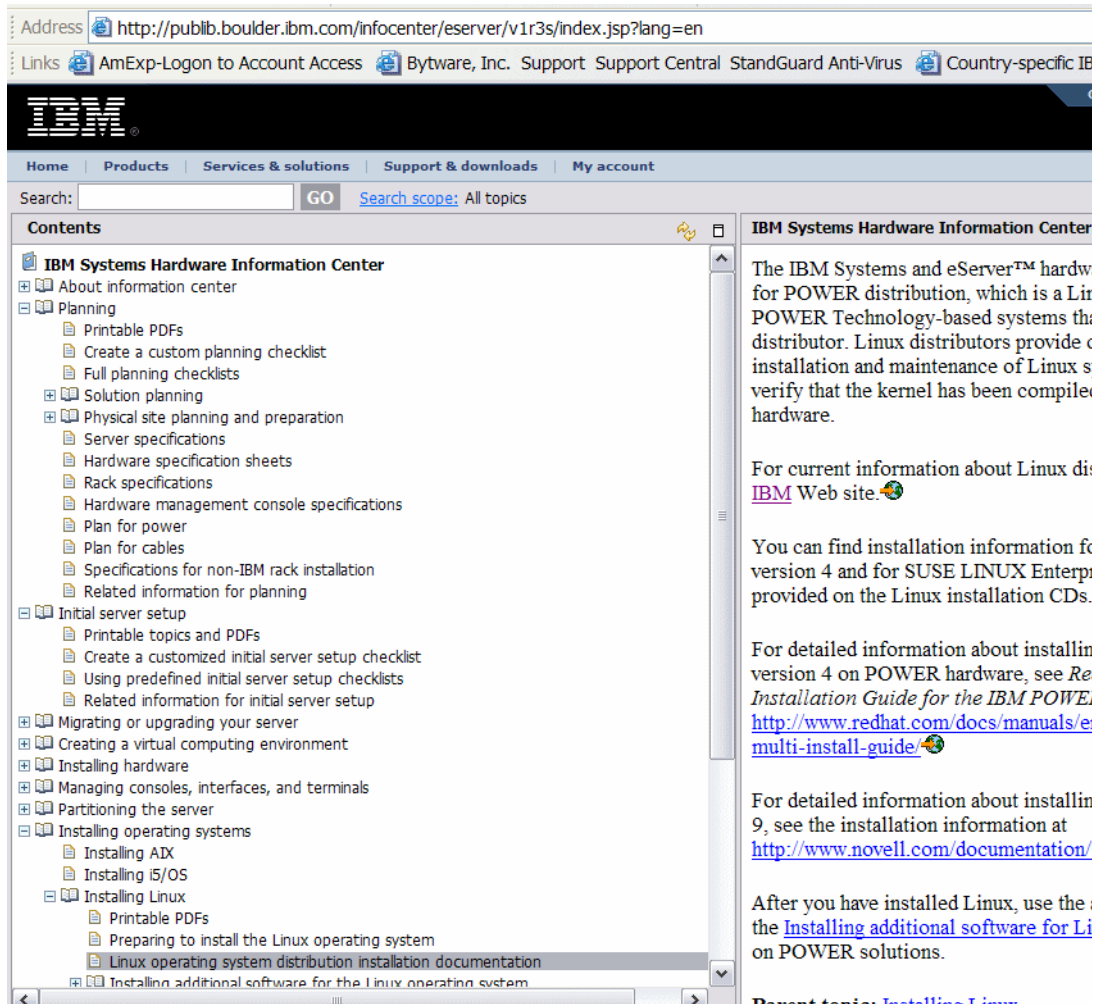


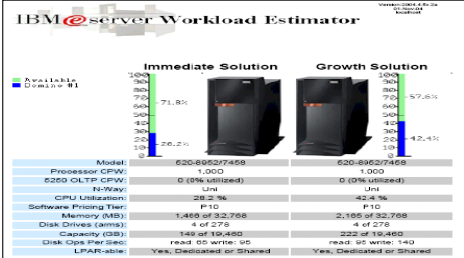
Figure 6-13 eServer Information Center - Linux partition example

6.4.2 Preparing system for VPM

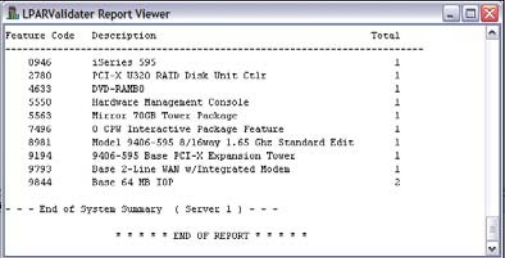
ibm.com/redbooks
International Technical Support Organization

Preparing System for VPM

- Performing capacity planning
- Designing and validating partition configuration
- Completing initial setup of your eServer i5 system
- Changing partition configuration options



Workload Estimator



Logical Partition Validation Tool

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 6-14 Preparing System for VPM.

Performing Capacity planning

You can use the IBM eServer Workload Estimator (WLE) to produce sizing guidelines for partitions with Linux workloads. Based upon the workloads specified, WLE outputs estimates resources required for processor, disk and main storage. For a Linux partition sizing information is available for file serving, web serving, and network application workloads. For more information on WLE go directly to:

<http://www-912.ibm.com/supporthome.nsf/document/16533356>

Designing and validating your hardware partition configuration

You can use the iSeries Logical Partition Tool (LVT) to design a partitioned system from a hardware configuration viewpoint. It helps you to validate your system requirements while not exceeding logical partition recommendations. You can download a copy of the tool as follows:

<http://www.ibm.com/servers/eserver/series/lpar>

- ▶ Select the Getting Started tab
- ▶ Read about the LVT
- ▶ Select to download the tool

Complete Initial Setup of Your @server i5

In this chapter we assume initial setup for @server i5 system is based upon earlier use of either the predefined or customized setup checklists. You need to initially setup as a standalone system without specifying logical partitions even though we create a Linux

partition later. Thus, no HMC is required for setup. For more information on setting up your Linux partition, refer to:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/index.jsp>

Select IBM Systems Hardware Information Center and expand the left navigation area tree structure for Initial server setup.

Changing the i5/OS partition configuration

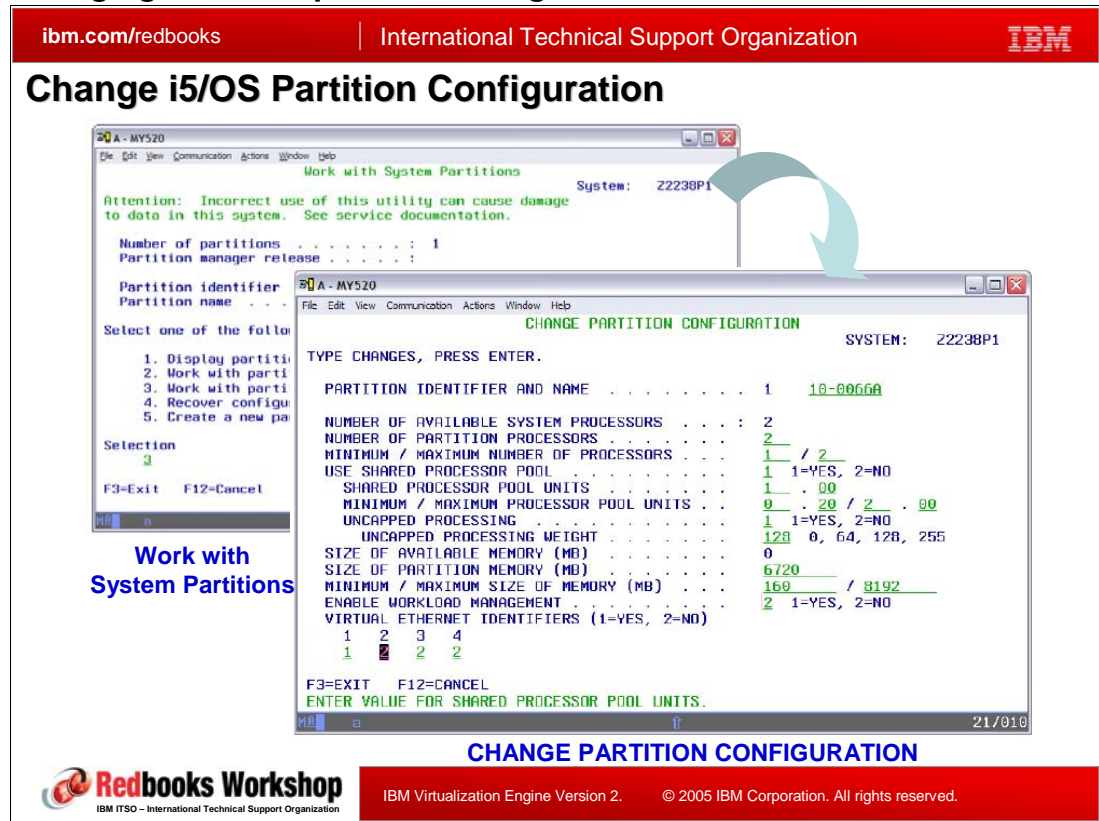


Figure 6-15 Change Partition Configuration Options (SST)

In Figure 6-15, using a 5250 workstation, we have already signed on to DST. Note that use of DST (or SST) requires a specific i5 service tools user id and password with appropriately granted privileges. A DST/SST user id and password are separate and distinct from an i5/OS user id and password.

Initially by default, i5/OS owns all of the processor, memory, and I/O resources. To accommodate a Linux partition, we need to take steps reducing resources of i5/OS. We did a similar “resource reduction” when using IVM in 6.3.5, “IVM and VIOS initial setup and partition preparation”

Key distinctions between DST and SST when using one of these interfaces to prepare and configure a Linux partition are:

- ▶ Dedicated Service Tools (DST): You can use this tool even when the server has limited capabilities. That is i5/OS does not need to be fully operational.
- ▶ System Service Tools (SST): At this mode, i5 system remains in full operational mode so other users can access to use the system concurrently while a Linux partition is being created.

Dynamic Resource Removal is not supported for i5/OS, so a system IPL is required to complete the tasks.

Removing Resources from i5/OS

In this chapter, we overview the procedure removing resources from i5/OS partition based on SST. Also, we suppose that you are familiar with SST or DST and Service Tools user IDs was configured before starting SST. If need more information for this, perform the following:

<http://www.ibm.com/eseries/series/infocenter>

- Select to iSeries Information Center - V5R3
- Expand the left directory tree for Security.
- Expand the link for Service tools user IDs and passwords

Assuming you are authorized as an i5/OS user, use the System Service Tool (STRSST) command from the main menu or the i5/OS command line as follows.

- ▶ **STRSST** -> Enter Service tools user ID and password in **Start Service Tools (STRSST) Sign On Screen** -> Select option **5. Work with system partitions** in **System Service Tools(SST)**.
- ▶ Press **Enter** to confirm using Virtual Partition Manager to partition the server in **LOGICAL PARTITIONING ENVIRONMENT SUPPORTED**. -> Select **3. Work with partition configuration** in **Work with System Partitions**
- ▶ Select **2** to Change partition configuration in **Work with partition configuration**.

Then a menu appears like Figure 6-15 in which you can remove resources from i5/OS partition to be able to create Linux partition. You need to reassign the CPU and memory allocation based on the values you determine during you planning with LVT.

Similar to IVM, VPM can define a partition to use either dedicated processors (keeps the assigned processor power while the partition is active) or as using a Shared Processor Pool. The shared processing pool enables processing capacity that can be shared among multiple logical partitions. With a shared processor pool partition the partition also, during configuration, to an Uncapped partition.

As stated earlier for IVM this uncapped support enables the system to detect the partition is using 100% of its current processor capacity and assign available processor units to this partition. The “available processor units” can be unused in a shared pool or currently assigned to another active uncapped shared processor pool partition.

Also as discussed under IVM and repeated here, a weight value of 128 is the default for an uncapped partition. When setting up an environment where multiple uncapped partitions are active concurrently you should first to size the expected “normal workload” processor requirement for such active processors. Then, if you want dynamic movement of processor units among active partitions, consider different weight values for the different partitions.

Depending on the actual processor resource consumption at a particular time, if all partitions have the same weight value, then the partition currently using its processor power may not “give up” processor units to the other partition that may need the additional processor power. This actually depends on the amount of processing power being consumed by the partition currently with the processor units assigned. A partition with the higher weight value (for example, 200) and operating at 100% current processor capacity would be able to “obtain some processor units” from a partition with, for example a weight value of 128.

Uncapped partitions work dynamically and best in a scenario when the application workload in a partition has significantly low utilization of its currently assigned processor units and another, higher weight value processor needs that processor power.

The system performs the automatic movement of processor power if partitions are defined to be uncapped. However, VPM does not support full dynamic movement of resources such as processor, memory, and I/O is not supported. If you make changes, you must restart the Linux partition for the changes to take effect.

Understanding Virtual I/O Slots and Requirement for IPLs

In the VPM environment, the single i5/OS partition creates 8 virtual I/O slots by default. Three of them are allocated to two virtual slots and one virtual serial, with five slots remaining free. When creating each partition, two of them are allocated to a virtual SCSI server adapter and virtual serial client adapter automatically. When you add a virtual Ethernet to the i5/OS partition, it uses one additional virtual slot.

If you use up all the eight available virtual I/O slots, the system sets itself up to use all of the available virtual I/O slots and enables them during the next IPL. Thus, when you create more than one Linux partition with a couple of VLANs, the Partition Configuration screen shows you that you need to IPL your system to implement the changes.

6.4.3 Defining new Linux partitions



Figure 6-16 Defining new Linux Partitions

In Figure 6-16, “Defining new Linux Partitions”, two SST windows show the important steps in initially defining values for a new Linux Partition. We discuss how to start to configure Linux partition and show you some examples for it here briefly. If you need full configuration steps and/or meaning of each entry, please refer the VPM redpaper at:

<http://www.redbooks.ibm.com/abstracts/redp4013.html?Open>

After preparing the system for VPM, you can start to create Linux Partitions with the System Service Tool (STRSST) command from the main i5/OS menu or the i5/OS command line.

- ▶ **STRSST** -> Enter Service tools user ID and password in **Start Service Tools (STRSST) Sign On Screen**
- ▶ Select **5. Work with system partitions** in **System Service Tools(SST)**
- ▶ Select **5. Create a new partition** in **Work with System Partitions**

Then a menu appears like the front right window in Figure 6-16, in which you can create the Linux partition. This screen asks you to enter input values for:

- ▶ Partition Identifier and Name
- ▶ Number of Partition Processor.
- ▶ Minimum/Maximum number of Processors
- ▶ Shared Processor Pool Units
- ▶ Minimum/Maximum Shared Processor Pool Units

- ▶ Uncapped processing
- ▶ Size of Partition Memory
- ▶ Virtual Ethernet Identifiers

Use of the LVT should have enabled you to have the values you want ready to be entered. We discussed use of the LVT in the beginning of 6.4.2, "Preparing system for VPM" .

The steps we explained in this chapter are not the complete procedure. They are shown to help you understand the key capabilities and LPAR planning considerations as they would apply to your environment.

If you want to know more complete information on VPM configuration and management, please refer to the redpaper we previously referenced - *Virtual Partition Manager - A Guide to Planning and Implementation*, REDP4013.



Security

The goal of the IBM Virtualization Engine is to support the sharing and the coordinated use of diverse resources in heterogeneous, distributed and complex environment, dynamically.

Importantly this type of environment consists of different security models (i.e. different authentication models, different authorization models, different user registries etc.). The real challenge is to bridge these different distributed security mechanisms.

The objectives of security in the Virtualization Engine Version 2 is:

- ▶ To leverage existing security mechanisms where possible and fill the gaps where needed. In addition, this must be done in a way that significantly improves the end-user experience while significantly lessening the administrative overhead associated with managing security.
- ▶ To accomplish Single Sign-on (SSO), from the browser through the console to managed servers and managed nodes

SSO is a term that is used by many people in many ways. We therefore need to define what we mean by SSO in the context of the Virtualization Engine. In this document we define SSO as the "Ability for a user to sign-on to a distributed application once and to perform required actions across multiple security domains and multiple native platforms without forcing the user to re-authenticate."

- ▶ To provide end to end security experience for applications in the Virtualization Engine.

This chapter describes the mechanisms included in the Virtualization Engine that enforce security during the communication between components but also make sure that the users are kept with a minimum of overhead when it comes to accessing systems and resource. New in this release is a single sign-on, which provides several mechanisms for authentication.

This chapter explains:

- ▶ Why is a single sign-on being implemented
- ▶ What is the architecture for the single sign-on
- ▶ How do you install Virtualization Engine and setup the security
- ▶ How do you update the security information within the Virtualization Engine console

7.1 The end to end security model

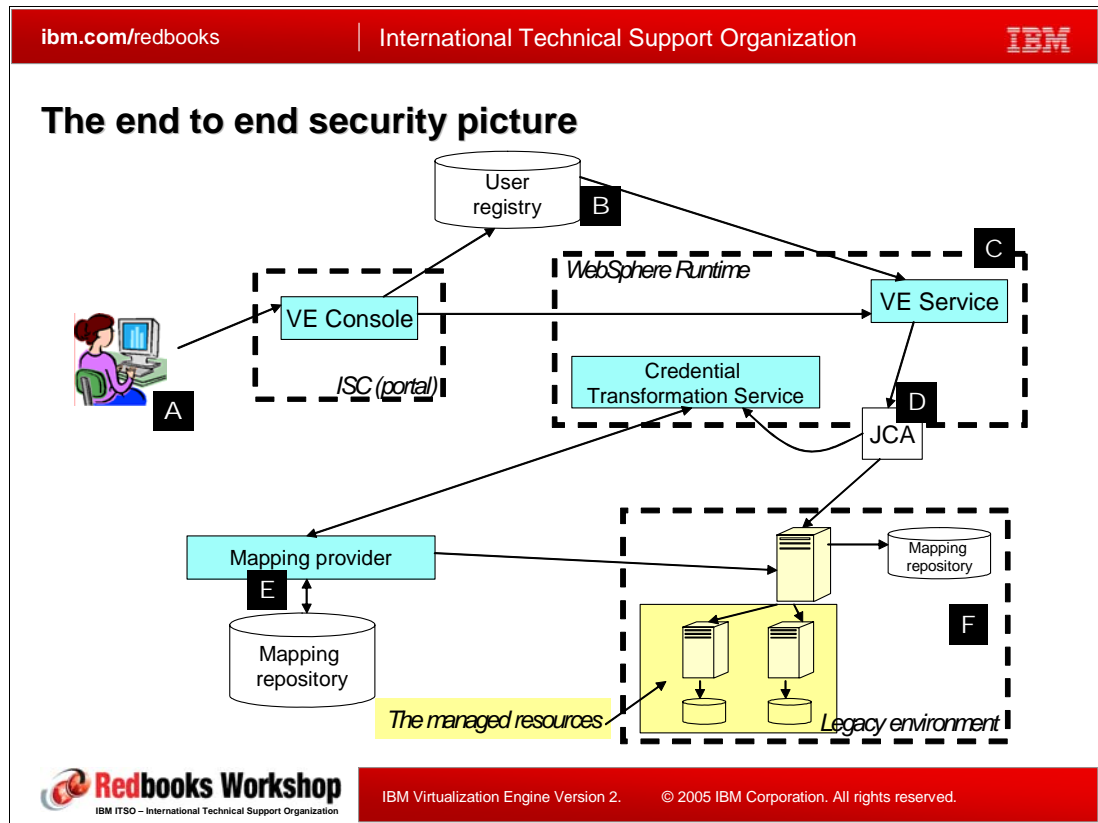


Figure 7-1

The Virtualization Engine consists of different applications that are distributed across multiple domains. The domains include client, middleware, legacy, etc. Security needs to be provided for requests beginning with the client request all the way until the access to the back end (the legacy) is completed.

In Figure 7-1 there are essentially three different security domains, where a security domain consists of one specific authentication mechanism and a user registry used for authentication/authorization:

- ▶ The portal domain
- ▶ The WebSphere environment domain
- ▶ The legacy domain

From a security point of view the different Virtualization Engine components need to be introduced:

- ▶ **A**: the client is responsible for accessing Virtualization Engine console applications. There are two types of clients available:
 - browser based clients, who can have different roles for accessing applications that reside in the portal environment.
 - web services clients, for which WS-Security is implemented
- ▶ **B**: the user registry is where the user (group) information is stored. The portal domain and the WebSphere domain use one single user registry to authenticate (authorize) the user.

- ▶ **C**: the WebSphere is the hosting environment for all the Virtualization Engine applications and provides an authentication mechanism and an authorization mechanism.
 - The authentication mechanism is responsible for creating a credential, which is an internal representation of a successfully authenticated client user.
 - Authorization is the process of checking whether the authenticated user has access to the requested resource.
- ▶ **D**: the Java Connector Architecture (JCA) is the framework component that defines the mechanisms that allow a structured and standardized way to link enterprise applications (the legacy) to a J2EE environment.
- ▶ **E**: the mapping provider is used by any legacy system that needs to obtain an appropriate mapped credential. The mapping repository is the place where identity and credential mapping are stored.
- ▶ **F**: the native operating system will determine the identity based on the unit of work; based on the identity, it will consult its internal mechanism.

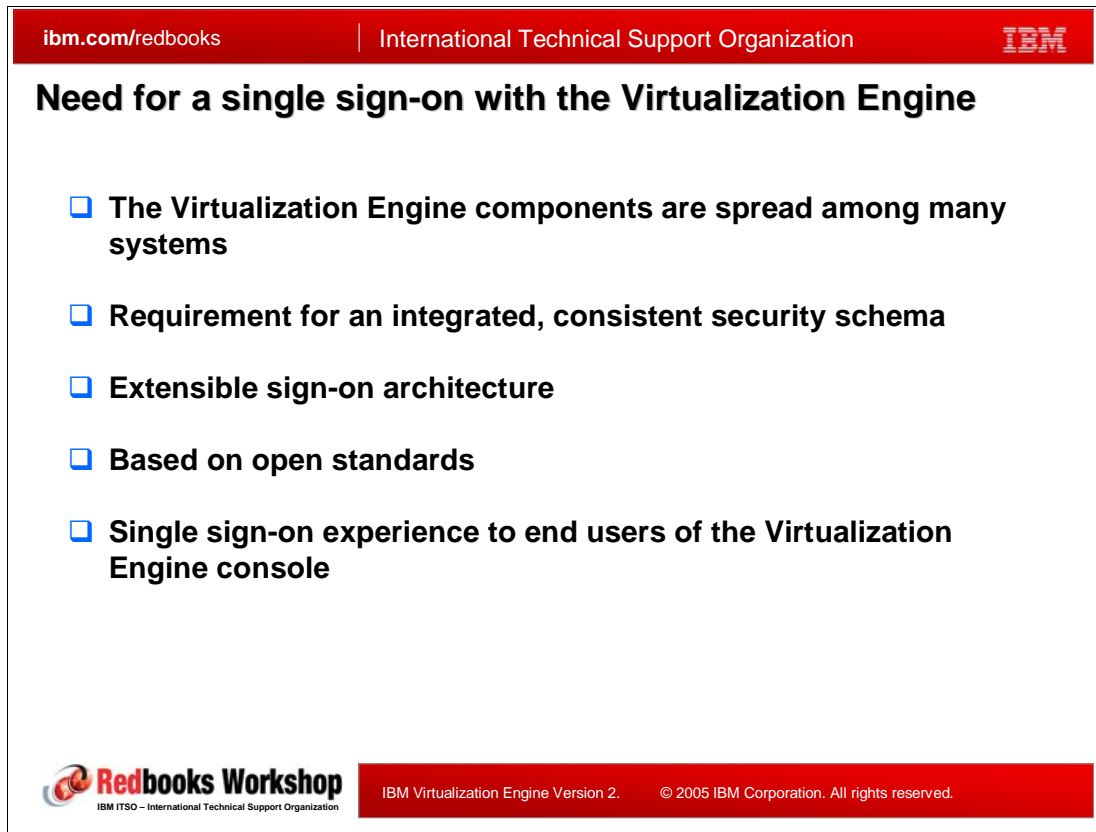
In the Virtualization Engine environment this means when a user logs into the Virtualization Engine console, he/she will be prompted for a userid and password. Once the authentication is successful, the user may select a console option that requires a request to run or access an application outside of the console environment.

Virtualization Engine Version 2 introduces *roles* for authorizations:

- ▶ The Virtualization EngineAdministrator can do everything.
- ▶ The Virtualization EngineMonitor can access some administration functions and is mainly in read-only access.
- ▶ The Virtualization EngineUser can be any authenticated user.
- ▶ The Virtualization EngineAnonymous can be any unauthenticated user.
- ▶ The Virtualization EngineManager can perform management operations.

A Virtualization Engine security administrator will have to assign roles to users depending on the authorization they need. The ISC is using *user groups which* describe a set of people who has similar responsibilities; users groups still need to be assigned to users as well.

7.2 Need for single sign-on in the virtualization environment



ibm.com/redbooks | International Technical Support Organization IBM

Need for a single sign-on with the Virtualization Engine

- ❑ The Virtualization Engine components are spread among many systems
- ❑ Requirement for an integrated, consistent security schema
- ❑ Extensible sign-on architecture
- ❑ Based on open standards
- ❑ Single sign-on experience to end users of the Virtualization Engine console

Redbooks Workshop IBM ITSO - International Technical Support Organization IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 7-2 Need of single sign-on with the Virtualization Engine

The virtualization engine components spread across many systems. Logged to the Virtualization Engine console, the users must be able to access any resources in the infrastructure, and there are unattended operations flowing between components. Security concerns are high and it is important to be able to protect resources and only allow users to access the systems for which they are entitled.

Virtualization Engine Release 2 provides an integrated, consistent security schema. It allows you to store in one repository the information regarding who can access what and help protect passwords and certificates.

This extensible single sign-on architecture starts with the components of the Virtualization Engine, but users logged to Virtualization Engine console will be able to add information on how to connect additional systems they have to connect or with which they interact.

Once the systems are set up, the logged users can get information, launch applications on remote systems, as well as start and stop them without having to enter a username or password again.

The solution for security deployed in virtualization engine release 2 is based on the Enterprise Identity Mapping technology, whose information is stored in an identity mapping repository, hosted by an IBM Directory Server, implementing the LDAP protocol. For the Virtualization Engine, the information on how to access the data is stored in the Global configuration repository, with the following definitions:

Enterprise Identity Mapping (EIM):Enterprise Identity Mapping (EIM) is an IBM@server infrastructure technology that allows administrators and application developers to address the problem of managing multiple user registries across their enterprise. This infrastructure provides a common set of APIs that can be used across platforms to develop applications that look at the relationships between user identities and a single EIM identifier that represents that user in the enterprise. Figure 7-3 provides an EIM implementation example in an enterprise. Two servers act as EIM clients and contain EIM-enabled applications that request EIM data using EIM lookup operations (6). The domain controller (1) stores information about the EIM domain (2), which includes an EIM identifier (3), associations (4) between these EIM identifiers and user identities, and EIM registry definitions (5).

Identity Mapping Repository (IMR):This repository stores the single sign-on mapping information as well as the credentials (passwords) used for SSO. In release 2, this IMR is an EIM domain configured on an IBM Directory server.

Global configuration repository (GCR): This directory based repository stores global virtualization engine configuration end runtime information. It stores the IMR configuration information as well as Virtualization Engine topology information used by the Single sign-on GUI.

IBM Directory Server:The IBM Directory Server is a Lightweight Directory Access Protocol (LDAP) directory that runs as a stand-alone daemon. It is based on a client/server model that provides client access to an LDAP server. The IBM Directory Server provides an easy way to maintain directory information in a central location for storage, updating, retrieval, and exchange.

LDAP: The Lightweight Directory Access Protocol (LDAP) is a directory service protocol that runs over TCP/IP. It is designed to provide lighter-weight and lesser-function access to a directory while not incurring the resource requirements of the X.500 Directory Access Protocol (DAP).

There are exist many books on these different components and we will not go into further details for each of those but rather how they work together within the Virtualization Engine context.

Lastly to say, lessening the cost to manage security of Virtualization Engine and exploiting security aspects that customers have already configured to protect their existing resources have been taken into consideration for this security implementation.

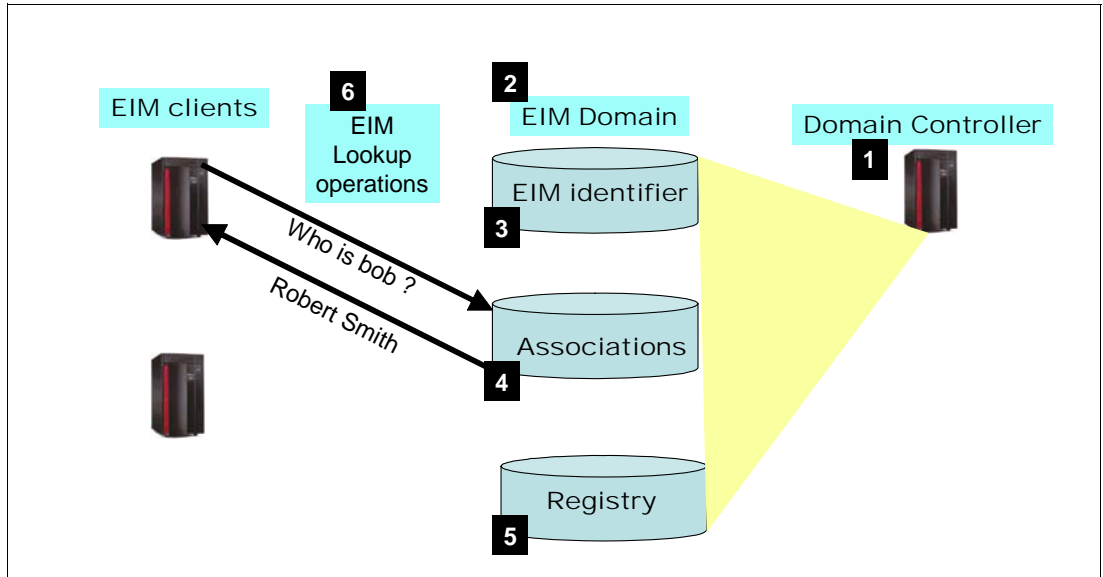


Figure 7-3 An EIM implementation example

7.3 Single Sign-On architecture in the Virtualization Engine Version 2

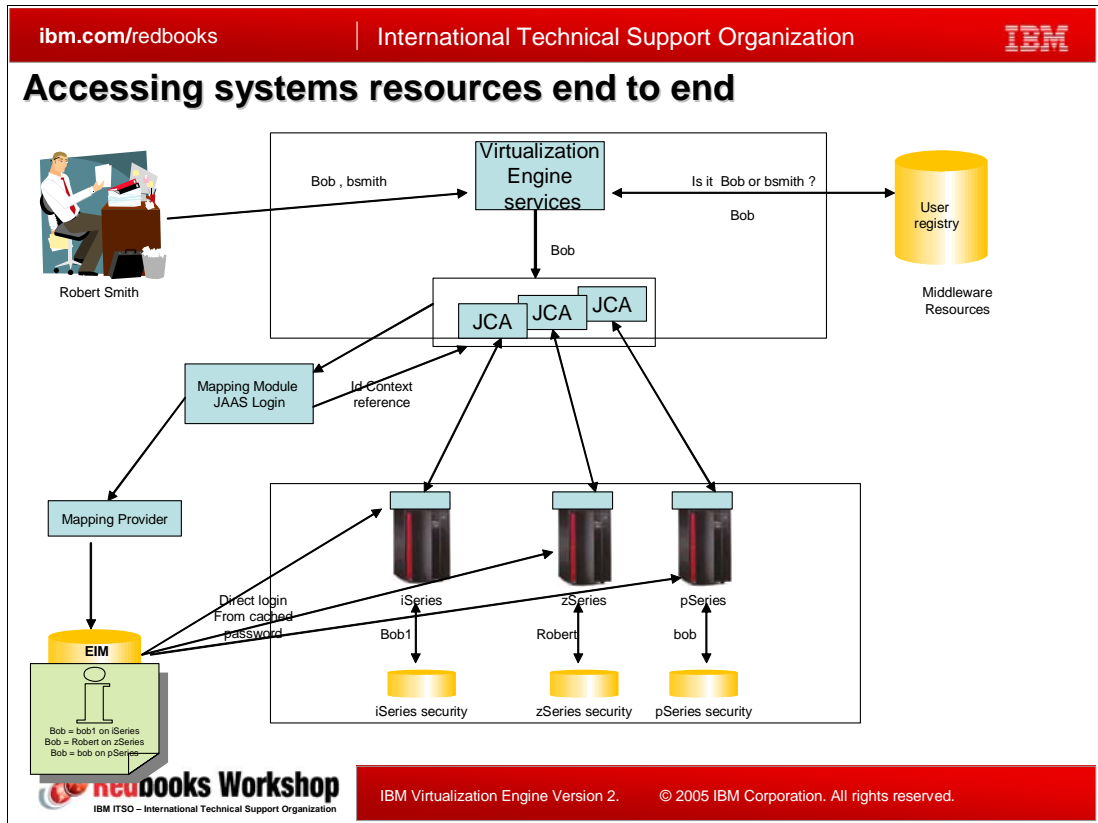


Figure 7-4 Single sign-on architecture

The SSO components

The SSO architecture needs the following basic components:

- ▶ A repository for storing the SSO information; CGR and IMR will be used.
- ▶ Associations set-up between the Virtualization Engine console users and a user on the target system
- ▶ A means to determine if a cached password is required by the target application for the target system or if identity context references are supported instead.
- ▶ An interface to know all of the target systems that are participating in SSO for the Virtualization Engine console.
- ▶ An interface to determine whether a cached password should also be established for the user on the target system.
- ▶ EWLM and Resource Dependency Service have special needs: for example Resource Dependency Service needs a specific interface to process the Resource Dependency Service credential and EWLM needs special interfaces to be used during the EWLM Domain Manager configuration process.

The types of SSO mechanisms

The single sign-on part of the virtualization engine is designed to be able to support multiple SSO mechanisms. In this release, three mechanisms have been implemented:

Lightweight third party authentication (LTPA)

The use of this mechanism is the only possibility when the communication is established between WebSphere based applications that are part of the Virtualization Engine solutions. It is based on a well known mechanism existing in the Websphere application server: the LTPA mechanism which allows the servers to communicate securely with each other, and which allows the called server to decrypt security information from the calling server.

In that scenario, when users make requests for a resource on a server, that is a bridge to one of the management servers, the LTPA cookie generated when they logged to the Virtualization Engine console, serves as an authentication token for the Web application server. It contains user identity and password information. This information is encrypted using a password-protected secret key shared between both servers. This scenario is the same even the bridge and the Virtualization Engine console are on the same logical server.

Identity context propagation

In this mechanism, the WebSphere-based applications, such as the Virtualization Engine console or the bridges, that need to access a resource outside of WebSphere Application Server will use a J2EE connector architecture (JCA) to obtain the credential needed to access the other resource. The JCA connector will be configured to use the single sign-on Java Authentication and Authorization Service mapping module (SSO JAAS MM). Most applications will use this module to obtain the credentials and then use another connection method to the server.

The SSO JAAS MM uses the SSO configuration information to obtain the necessary credentials, such as:

- ▶ Source application id
- ▶ Target application id
- ▶ Target hostname
- ▶ Target port
- ▶ Autonomic indicator: indicates whether the connection is being triggered by a real user or if is being part of an autonomic reaction from the environment.

The JAAS module retrieves the SSO configuration information for the target application by looking up into the GCR.

The possible authentication methods are:

USER: Authenticates using a user id and password. The JAAS mapping module retrieves a cache password for the target user. It is returned to the JCA connector that uses it to connect to the remote server. This methods is used by all application that run outside of a WebSphere Application Server environment.

IDENTITY_CONTEXT: Authenticates using an identity context reference. Allows authentication without having to cache a password. An authentication context is created that contains all of the source and target application information as well as the pre-mapped target userid. This authentication context is then stored in the IMR and an identity context reference is returned to the JCA. It is an 8-bytes string that identifies the authentication context. This string is passed to the remote application. The only application supporting this method is IBM Director.

The possible registry types are:

SYSTEM: credential information is for a user in the system registry. The target hostname itself determine the target registry.

APPLICATION: credential information is specific for a user in an application registry. You need then both the hostname and port to determine the target registry. The only application that currently uses application registries is EWLM

Real user ID and password via credential mapping

This mechanism has to be used for WebSphere-based application which link with an application that runs outside of a WebSphere environment.

7.4 Single sign-on set up

ibm.com/redbooks | International Technical Support Organization **IBM**

SSO Set-Up Tasks

- Configure the target systems participating in SSO**
- Register the target applications participating in SSO**
- Enroll Virtualization Engine console users to participate in SSO (GUI)**
- Associate target users on target systems with the Virtualization Engine console users (GUI)**

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 7-5 SSO set-up tasks

The installation process needs to go through the set-up tasks as defined in Figure 7-5.

The Virtualization Engine SSO support is installed with the base Virtualization Engine support; it is always installed. During the installation of the common runtime, the administrator is being prompted to enter the configuration information for the CGR, and the IMR. As part of the process, the administrator can choose to use an existing repository or create a new one. Once the installation process complete:

- ▶ The GCR is updated to include the information on how to locate the IMR, this includes the directory server name, the port number, as well as the distinguished name and password.
- ▶ For new installation, the IMR is initialized by creating a new EIM domain and a LDAP user registry representing the Virtualization Engine console users.

After the common runtime is installed, applications are being deployed. Virtualization Engine components installed via the suite installer calls a post-install java bean to take of the security part. When this process exits:

- ▶ the GCR is being updated with the application id, the default authentication method (such as userid/password or authentication context reference), registry type (system, application external), additional information to locate the external registry in case of external registry type, and indicator if this application uses Virtualization Engine applications association.

If this application uses a registry type of system, then a registry is created in the IMR with the same name as the hostname. An alias for this hostname is added to the registry.

If this application uses a registry type of application, then a registry is created in the IMR with the name of hostname:port. An alias of this same name is added to the registry,

If the application uses a registry type of external, then the additional information in the CGR is used to create the necessary registry and to add the alias.

Note: If you have not selected the automatic creation option during the install interview, the registry information has to be set up manually before using the SSO GUI.

The code for installing the SSO GUI in the console is an integral part of the console, and you don't have to select to install it. Identically, you cannot de-install the security component only.

7.5 Configuring the single sign-on in the Virtualization Engine console

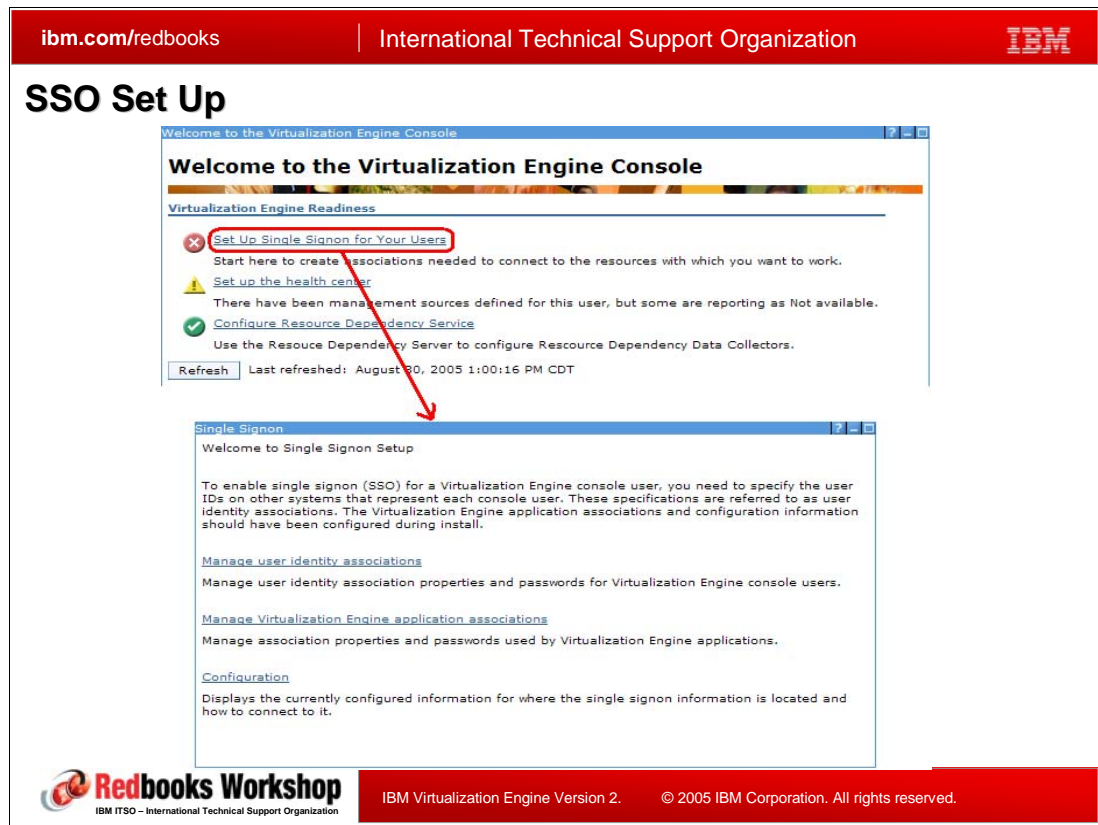


Figure 7-6 SSO set up from the Virtualization Engine console welcome page

The configuration screen for the single sign-on within the Virtualization Engine console can be reached from different places. The easiest one might be from the welcome page where a link directs you to this page as shown in Figure 7-6.

As shown in Figure 7-6, in the left frame, under the virtualization engine setup menu, you find the link to the single sign-on management item, . This is the main page for single sign-on configuration. Virtualization Engine Administrators can manage everything from this portlet though Virtualization Engine users cannot create new user identity associations, edit existing identity associations (other than setting the password for their own associations, manage application associations (they will not see this link), or edit SSO Configuration settings (they have read only access to this info).

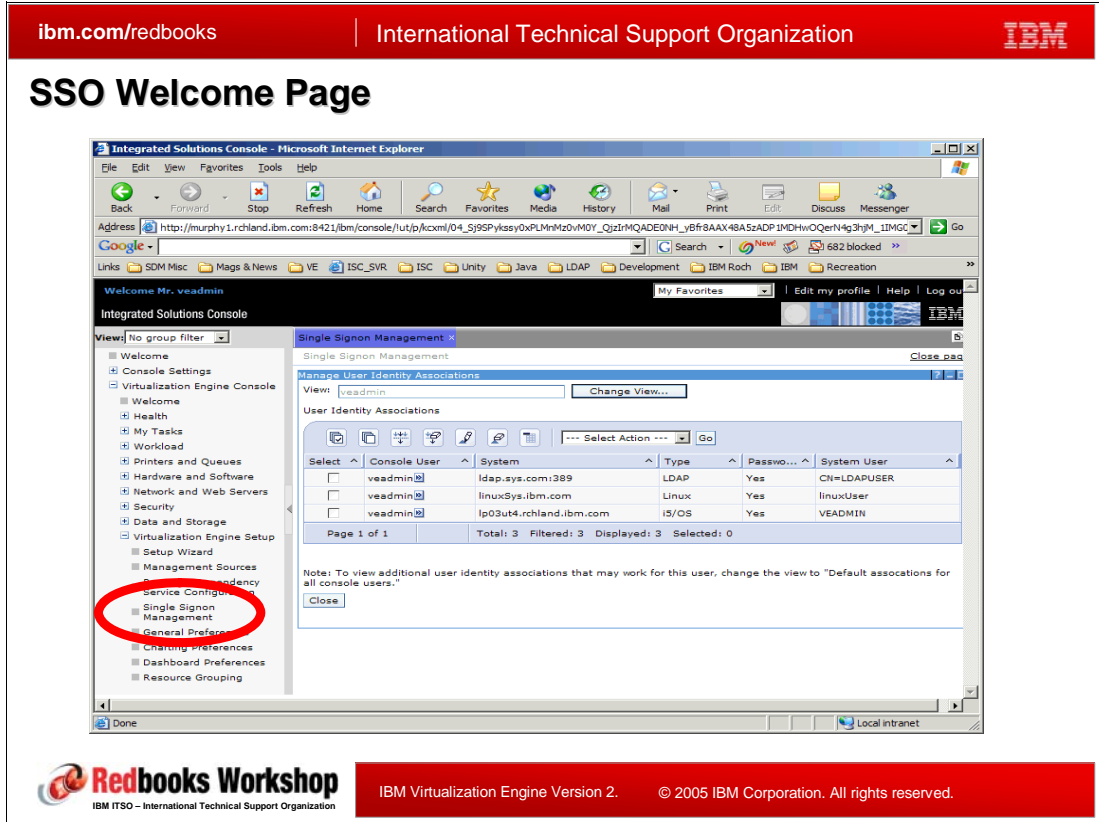


Figure 7-7 SSO welcome page

The portlet includes three entries.

- ▶ One to manage user identity associations, as shown in Figure 7-8, from where you can add multiple associations at a time from this panel. Only one association can be created for a given "system" (to prevent ambiguous results during mapping resolution). Note that if a password needs to be cached/specified, a subsequent update of the association is required.

If you are logged on the Virtualization Engine console with administrative privileges, you can see and modify your own entries, and you can also manage entries for other users defined on the console.

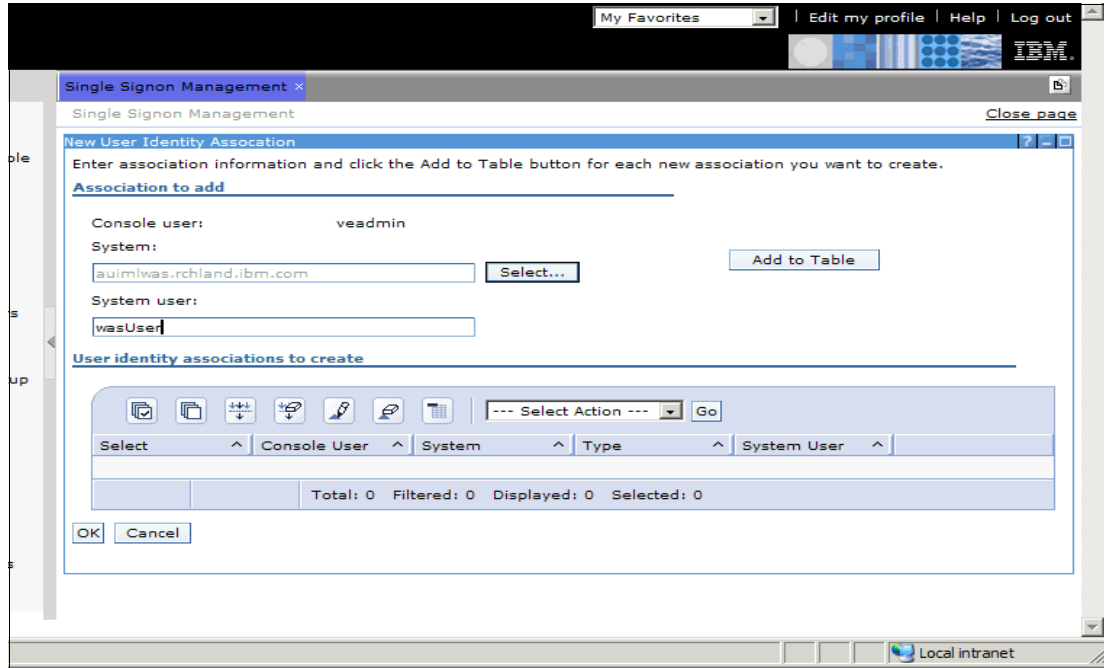


Figure 7-8 User identity association

- ▶ Another one to manage virtualization engine application associations, as shown in Figure 7-9, from where you can create an association that will authorize inter-application communication. Currently only Resource Dependency Service supports application associations. The Resource Dependency Service GUI only can create the Application Association support. Virtualization Engine administrators only can manage these application associations. The context menu "Set Password" option can be used to cache/update several passwords at once if the passwords are the same.

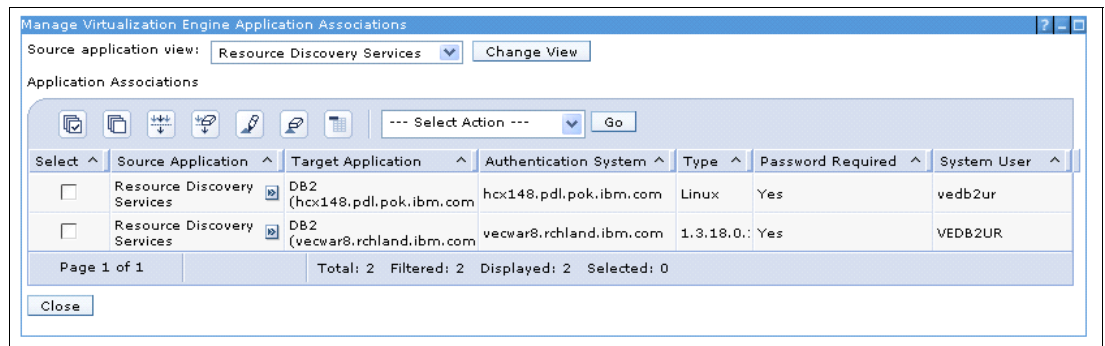


Figure 7-9 Manage Virtualization Engine Application Associations

- ▶ A third one to configure, as shown in Figure 7-10. The configuration panel lets you review and change the information to access the IMR repository. you will have to specify the LDAP information such as the distinguished names, hostname, port number as well as the user id for the directory server administrator.

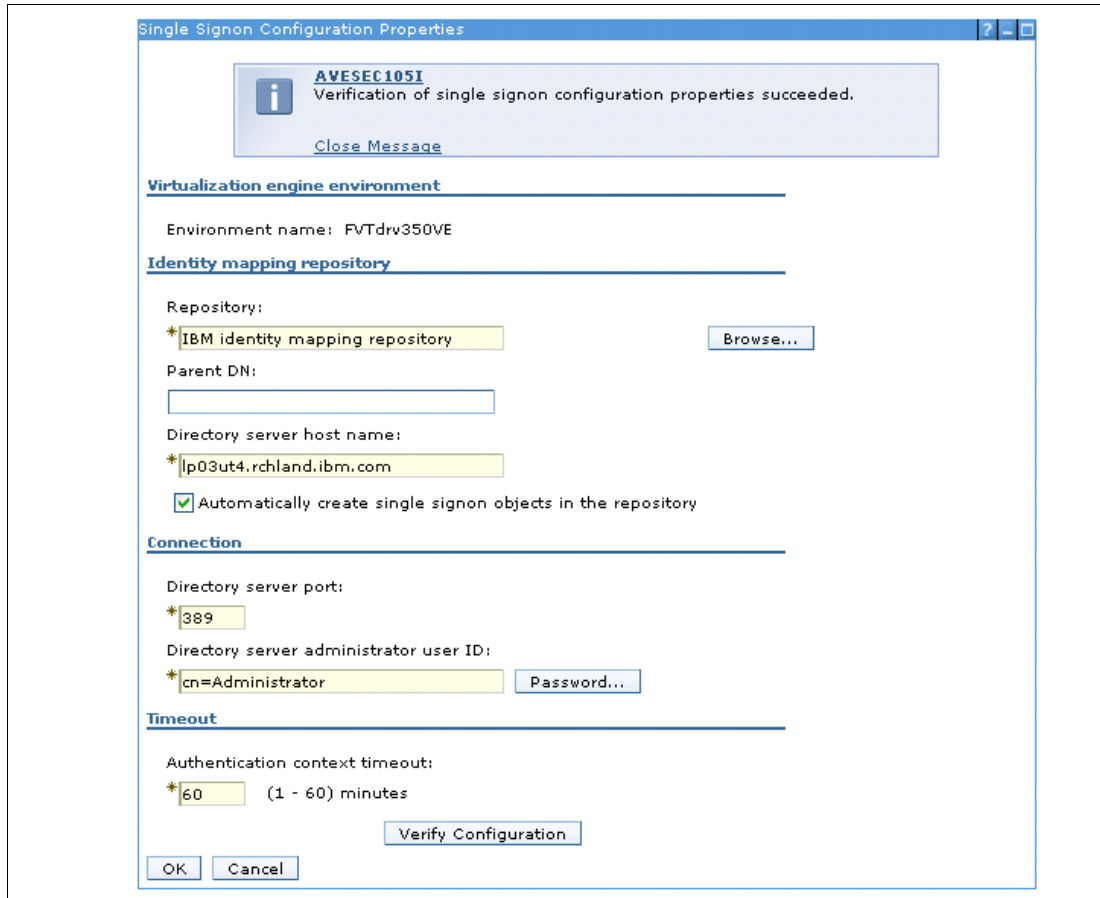


Figure 7-10 Configuration panel for the SSO environment

7.6 IBM Director security

Separate from operating system security IBM Director offers its own security, primarily in two separate “connections:”

1. Use of SSL between the IBM Director server and IBM Director console

SSL between the console and the server is separate from the encryption between the server and the managed nodes.

The IBM Director Console sign-on prompt, where you select the server that you want to connect to, allows the use of SSL or non-SSL between console and server. The IBM Director server has SSL and encryption parameters that must “fit” with the IBM Director console SSL selection.

IBM Director 5.10 has SSL enabled by default and is much easier to set up than in previous releases of IBM Director. For example, on i5/OS, no i5/OS digital certificate needs to be created through the Digital Certificate Manager (DCM) interface and the default certificate does not need to be assigned in DCM as was required with IBM Director 4.2n.

If you do “slip install” or an upgrade from IBM Director 4.2 to IBM Director 5.10 the first start of the IBM Director 5.10 converts any previous release digital certificate set up.

The IBM Director console for 5.10 or later has an **Options** → **Server Preferences** option. Server preferences include a Connections tab. Selecting the Connections tab lets you allow or disallow using SSL between the console and the server. If you select both check boxes then the server allows either a non secured console session or an SSL protected console session - as selected by the IBM Director console when doing an initial connection to the IBM Director server.

2. Encryption between the IBM Director server and IBM Director managed nodes (agents)

This connection uses encryption or does not use encryption. Encryption is enabled by default for IBM Director 5.10. You can view and change use of encryption by the server using

the `cfgsecurity` command before the IBM Director is started or while the server is running using the IBM Director console **Options** → **Encryption Administration**. The default is Advanced Encryption Standard.

With Encryption Administration you specify managing encryption settings and keys for the server. These settings apply to communication between the server and agents that are managed by this server. Note that not all agents support encrypted communications. On some managed agents, use of encryption is optional. The IBM Director server will not encrypt data destined for an agent that does not support it.

The IBM Director server supports three types of encryption:

- ▶ Data Encryption Standard
- ▶ Triple Data Encryption Standard
- ▶ Advanced Encryption Standard

Encryption can be enabled on a managed node agent using the `cfgsecurity` command before starting IBM Director. Under i5/OS you run `cfgsecurity` in a Qshell session. For example:

- ▶ `qsh`
- ▶ `cd /qibm/userdata/director/bin`
- ▶ `twgend` (IBM Director agent must be inactive)
- ▶ `cfgsecurity`

Respond to the questions you are asked. The IBM Director Agent for i5/OS supports Data Encryption Standard, Triple Data Encryption Standard, or Advanced Encryption Standard.

If using `cfgsecurity`, the “secured automatically” response will indicate you must do a request access to unlock the managed node (right click the managed object).

Summary of IBM Director supported SSL and data encryption standards

IBM Director lets you select one of three supported encryption algorithms:

- ▶ **Data Encryption Standard (56-bit key):** The Data Encryption Standard (DES) is the name of the Federal Information Processing Standard (FIPS) 46-3 standard, which describes the data encryption algorithm (DEA). The DEA also is defined in the American National Standards Institute (ANSI) standard X9.32. The terms “DES” and “DEA” often are used interchangeably, but we will use “DES” throughout this book. DES has been studied extensively since its publication and is the best known and most widely used symmetric encryption algorithm in the world.

DES is a robust algorithm used in many applications where security is a prime concern. The obvious method of attack is a brute-force exhaustive search of the key space; this process takes 255 steps on average. Many experts consider DES to be insecure, and DES encryption is no longer allowed for US government.

- ▶ **Triple-DES:** Triple-DES uses three DES algorithms in parallel, which effectively lengthens the key to 168-bit. However, the way Triple-DES works is that the #1 and #3 keys are the same, which results in a 112-bit key (128-bit with parity). The cryptographic community at large feels that Triple-DES is more secure than DES.

IBM Director can be set to use Triple-DES encryption, using the Diffie-Hellman key exchange protocol to ensure that a new secure key is generated for each session between IBM Director Agent and IBM Director Server.

- ▶ **Advanced Encryption Standard:** Advanced Encryption Standard (AES) is a block cipher created as a result of a worldwide call for submissions of encryption algorithms issued by the US Government's National Institute of Standards and Technology (NIST) in 1997. After nearly five years of standardization, AES was adopted by NIST as FIPS-197 in November 2001. In June 2003 the U.S. National Security Agency (NSA) announced that AES is secure enough to protect classified information up to the TOP SECRET level, which is the highest security level.

AES was developed to replace DES and Triple DES.

The AES algorithm uses one of three cipher key strengths: a 128-, 192-, or 256-bit encryption key. Each encryption key size causes the algorithm to behave slightly differently, so increasing the key size not only offers a larger number of bits with which you can scramble the data, but also increases the complexity of the cipher algorithm.

- ▶ **Secure Sockets Layer (SSL)** is a protocol designed to enable secure communications on an insecure network such as the Internet. SSL provides encryption and integrity of communications along with strong authentication using digital certificates, such as those provided by DSA encryption. SSL encryption is available for communications between IBM Director Server and IBM Director Console.

In IBM Director version 5.10 and later, SSL encryption between the management server and management console is enabled by default. This is a change from previous versions.

Secure Shell: Secure SHell (SSH) is a protocol for creating a secure connection between two systems. In the SSH protocol, the client machine initiates a connection with a server machine. After an initial connection, the client verifies it is connecting to the same server during subsequent sessions. The client transmits its authentication information to the

server, such as a username and password, in an encrypted format. All data sent and received during the connection is transferred using strong, 128-bit encryption.

Secure SHell is available for and may come standard with many UNIX-based and Window operating systems. Free and customized downloads are also available on the Web. Most search engines will find many “hits” when searching for “secure shell.”

When using i5/OS to install Secure Shell support you must install the product:

- IBM Portable Utilities for i5/OS, 5733SC1, with its option 1 - OpenSSH, OpenSSL, zlib

IBM Director can make use of SSH, rather than the simple and insecure Telnet protocol when using the Remote Session task. IBM Director can also download its agent software to a system that does not already have a version of the agent installed - if that system has SSH active (commonly listening on port 22).



Getting Started

This chapter provides the information you need when you want to install the Virtualization Engine Version 2 or some of its components:

- ▶ Section 8.1, “The Information Center” on page 272 describes how to use the Information Center that will help you to get the latest information about the Virtualization Engine Version 2.
- ▶ Section 8.2, “Planning advisor” on page 280 describes how to use the Planning Advisor that will help you plan your installation
- ▶ Section 8.3, “The common runtime components” on page 291 describes the installation process
- ▶ Section 8.5, “Products packaging” on page 308 describes which product number to get
- ▶ Section 8.6, “Architecture, performance, and sizing” on page 310 provides you with some performance information that will help you choose your infrastructure support.

8.1 The Information Center

IBM Systems Software Information Center is the single location of the repository for the IBM Virtualization Engine documentation.

In this chapter, the following topics are discussed:

- ▶ How to access Virtualization Engine documentation through the information center
- ▶ What information center provides
- ▶ What is the benefit of using the information center

Note: In this chapter the term *information center* and *infocenter* will be used to mean the “IBM Systems Software Information Center”

8.1.1 Overview

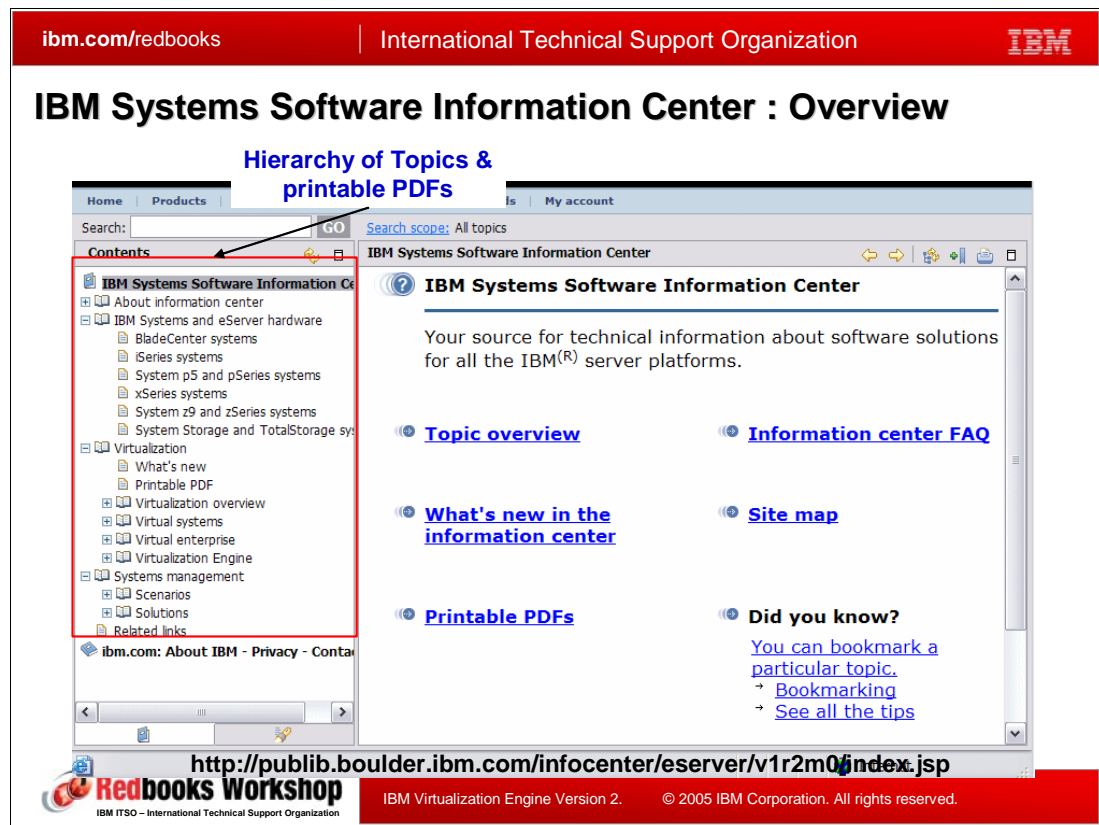


Figure 8-1 Information center entry panel

IBM Virtualization Engine covers a very wide range of functions, underlying technologies, components, and specific products. IBM Systems Software Information Center is the place to start to get the perspective of all of these capabilities and drill down to specifics.

Two examples include:

- ▶ Learning about and using the Virtualization Engine planning advisor functions
- ▶ The IBM Director Installation and Configuration Guide PDF

Figure 8-1 shows the first screen you can see when you access IBM Systems Software Information Center - with the directory tree expanded. Information Center on the Web is a single location of the repository for the Virtualization Engine documentation which was previously called @server Software Information Center.

It organizes the topics into navigation hierarchy on the left tree list. You can use this hierarchy to access printable PDF files as well as a fast path for Virtualization Engine information quickly. The prominent characteristic of the information center is that you can go over informations based on your goal and tasks. For example, you can get information walking through the scenarios for how to manage virtual systems and also virtual enterprises. These include how to implement the Virtualization Engine to best suit the needs of your business from the beginning such as planning, installation, configuration, management, use, troubleshooting services and so on.

Information center delivers customers the latest technical information. Thus customers can use it as a reliable source of the latest technical information to effectively integrate IBM Systems solutions into their business.

Information center fulfills your requirements from three points of view:

► What you need

It is the repository of information for Virtualization Engine which is usable, technically accurate, retrievable, and complete. It also supports several multinational languages. With the newest “version 2” information center uses the eclipse framework to examine your browser and determine which language information center should be displayed in.

If you need to access older information you may need to tell information center the language you want to use.

It provides information in several formats. You can get the information online or download the printable format as PDF on the same URL for offline or to print it.

► When you need it

You can access resources at anytime on the web at the URL shown Figure 8-1. The contents are optimized for Web delivery for users to make it easy to read and retrieve.

► Where you need it

You can access the information through multiple retrieval and delivery methods even including product user interface. We will discuss it in detail later in this chapter.

8.1.2 Access

ibm.com/redbooks
International Technical Support Organization
IBM

Information Center: Access

❑ With Virtualization Engine Version 2, you can access the Information Center at any time:

- From [ibm.com](http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp), or directly at <http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp>
- From marketing and educational materials
- From materials shipped or downloaded with the Virtualization Engine products
- **From the product user interface...**

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-2 Information Center - Access

Note to Jim: This fig is unreadable and understandable from the text below

Figure 8-2 lists various locations of Virtualization Engine information. You can access the information center anytime through several ways.

- ▶ You can access the Information Center through the URL shown on the slide as well as through ibm.com. Marketing and education materials you might receive also contain links to the system Software Information Center. Fulfillment materials shipped with Virtualization Engine products are another good resource for information.
- ▶ You can access the information directly from the Virtualization Engine products within the product user interface.

As shown in the Figure 8-2 example windows, when you are working with the Virtualization Engine console, you might need to refer to Virtualization Engine information of the information center. The right window shown is an Information Center example for the Virtualization Engine console Launch-in-context capability.

- ▶ You can also use the online help in the Virtualization Engine console by clicking the Help button. This is typically the first help information you look for, before, for example, going directly to the information center.

8.1.3 General topics

The screenshot shows the IBM Redbooks Workshop interface for the 'Virtualization Engine – General topics'. The top navigation bar includes 'ibm.com/redbooks', 'International Technical Support Organization', and the IBM logo. The main title is 'Virtualization Engine – General topics'. On the left, a 'Contents' sidebar displays a tree view of the information hierarchy. On the right, a bulleted list provides an abstracted view of the information hierarchy. The footer contains the 'Redbooks Workshop' logo, 'IBM ITSO – International Technical Support Organization', and copyright information: 'IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.'

Contents

- Virtualization Engine
 - Systems Edition
 - What's new
 - Printable PDF
 - iSeries
 - pSeries
 - xSeries and BladeCenter
 - Linux on System z9 and zSeries systems
 - Related information
 - Management collection
 - What's new
 - Printable PDF
 - Getting started
 - Requirements
 - Planning
 - Installing and configuring
 - Managing
 - Uninstalling
 - Troubleshooting the Virtualization Engine
 - Fixes and updates
 - Virtualization Engine console
 - Enterprise Workload Manager
 - Resource Dependency Service
 - Dynamic Infrastructure
 - Related information
 - Systems management
 - Scenarios
 - Scenario: Determine the impact of a failing resource
 - Scenario: Dynamically shifting processing capacity
 - Solutions
 - IBM Director
 - Virtualization Engine console
 - Enterprise Workload Manager
 - Resource Dependency Service
 - Systems provisioning
 - Dynamic Infrastructure
 - Enterprise Identity Mapping

Figure 8-3 Information Center general topics

Figure 8-3 depicts how the information center classifies types of information, the intended content, and an abstracted view of the information hierarchy on the right side. Using IBM Director as an example of a Virtualization Engine product, there is a topic collection in the information center information center for IBM Director.

Typically you think of one idea and the information center provides several topic types of procedures, concepts, examples, references, and details:

- ▶ The procedure documentation helps you understand how you can complete a task with step-by-step instruction.
- ▶ The conceptual documentation is helpful for understanding what something is by providing concepts related to the Virtualization Engine products
- ▶ Reference lists the product manuals and IBM Redbooks (in PDF format), Web sites, and information center topics that relate to the Virtualization Engine management collection topic. You can view or print any of the PDFs.
- ▶ Examples present a lot of information at a high level of how businesses might deploy the Virtualization Engine in order to take advantage of the value the Virtualization Engine can provide. Examples also provide details about how to carry out such a **scenario** in a recipe format.
- ▶ Details provide other secondary articles such as more background and further information in a topic.
- ▶ An **Overview and getting started** topic provides high-level conceptual information about what the Virtualization Engine is, what the components that make up the Virtualization

Engineare. If a decision has been made to install the Virtualization Engine, it provides some quick getting started, roadmaps and directions on how to go through the process of ordering, installing, and deploying the Virtualization Engine products.

- ▶ The **Planning** topic helps you understand functional capability of the virtualization products and guides you to understand all the tasks needed in order to install and deploy the Virtualization Engine. The interactive Virtualization Engine Planning Advisor is discussed in Section 8.2, “Planning advisor” on page 280.
- ▶ The **Scenarios** topic details business solution scenarios and illustrates the values of Virtualization Engine and what a business can actually gain by taking advantage of various Virtualization Engine products and capabilities.
- ▶ **Installation, Management, and Troubleshooting** topics provide procedural documentation on installing the Virtualization Engine software, details on the day-to-day management tasks that might need to be done with a virtualization product, and troubleshooting information.
- ▶ **Virtualizing a system, Managing a virtual system, and Managing a virtual enterprise** topics provide a higher level approach by expanding not only into Virtualization Engine products but also addressing the hardware side of Virtualization within an enterprise.
- ▶ The **Systems management** topic includes scenarios for effective systems management and solutions for a single platform, multiplatforms, and cross-platform systems management.

The Information Center also provides various other IBM Systems-related software topics.

8.1.4 Interactive Resources

ibm.com/redbooks | International Technical Support Organization **IBM**

Information Center : Interactive Resources

More than just traditional text
 ➤ Much interactive and tailored to individual customers.

Interactive Information	Description
eLearning	Allows the user to learn about task and concepts in an engaging, interactive, and user-driven manner. This is available for Virtualization Engine Version 1 information center via the Virtualization Engine eLearning module!
Advisors	Walks users through complex decisions; makes smart recommendations based on user input. For example, watch for the Virtualization Engine Planning Advisor!
Conceptual animations	Combines text, graphics, and sound to create a rich user experience. Illustrates complex concepts and relationships in an engaging format. For example, look for EWLM conceptual animations!
Custom checklists	Provides personalized procedures based on customer input.
Interactive tables	Allows the user to sort and filter complex sets of data in an interactive manner. No more sorting through pages and pages of data!
Finders	Allows the user to look up information fast; a mini-search.

IBM ITSO - International Technical Support Organization
 IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-4 Information Center: Interactive Resources

Compared to traditional text information resources, the Information Center has a more advanced format than just traditional plain text. It has much more information which is interactive and tailored to individual customers. Figure 8-4 lists the types interactive information contained in the information center.

- ▶ *eLearning* (Virtualization Engine release 1) provides animations that contain a path for users to learn about task and concepts in an engaging, interactive and user-driven way. For example, there's a Virtualization Engine elearning module which a customer can access, walk through and see in an animated way how Virtualization Engine works, what the Virtualization Engine is, and how they can benefit of business.
- ▶ *Advisors* seem similar to traditional wizards you have seen in software products, but what the advisors do is help customers make decisions and provide output containing formal recommendations based upon the input that the customer provides. The Planning Advisor explained in Section 8.2.2, "Overview" on page 282 is a good example of it.
- ▶ *Conceptual animations* is another offering in the information center and very similar to eLearning modules but they combine animation, text, graphics, and sound. It provides a richer conceptual explanation of a concept than traditional concept textual documentation does.
- ▶ The *Custom checklist* included in the Virtualization Engine Planning Advisor provides you with a personalized procedure, based on the information you give. In this document we provide an overview of Performance advisor capabilities in Section 8.2.2, "Overview" on page 282.

- ▶ *Interactive tables* allow you to sort and filter out a large amount of data based upon your needs. For example, using the supported operating systems topic, if you want to see the IBM only IBM Director functions and products, you can filter the table to only see those you want to see.

If you want more details regarding IBM Director supported operating systems for the management server component, you can select to see only the supported operating systems on which the IBM Director server can run.

- ▶ *Finders* help you quickly lookup and get to the information that you need. Finders are similar to interactive tables, but have more search and retrieve mechanism to look up information. The Application Programming Interface (API) finder is one example.

8.1.5 Printing

Figure 8-5 Information Center: Printing

The information center organizes all individual topics into topic collections, a group of common topics we discussed previously. This information is also available in PDF format for easy printing.

Each topic collection has a printable PDF. This can be seen in the first part of the Web page which contains the same topic information as the product’s online help information.

For example, Enterprise Workload Manager (EWLM) is a topic collection in the Information Center. If you need EWLM related information, you can go to the EWLM topic collection in the Information Center and access all the documentation online. If you need offline materials or need to print them, access the PDFs, that contain all data on the EWLM topic. This would include information such as getting started, planning, installing, and so on. You can download the content and use it offline or print it for later review.

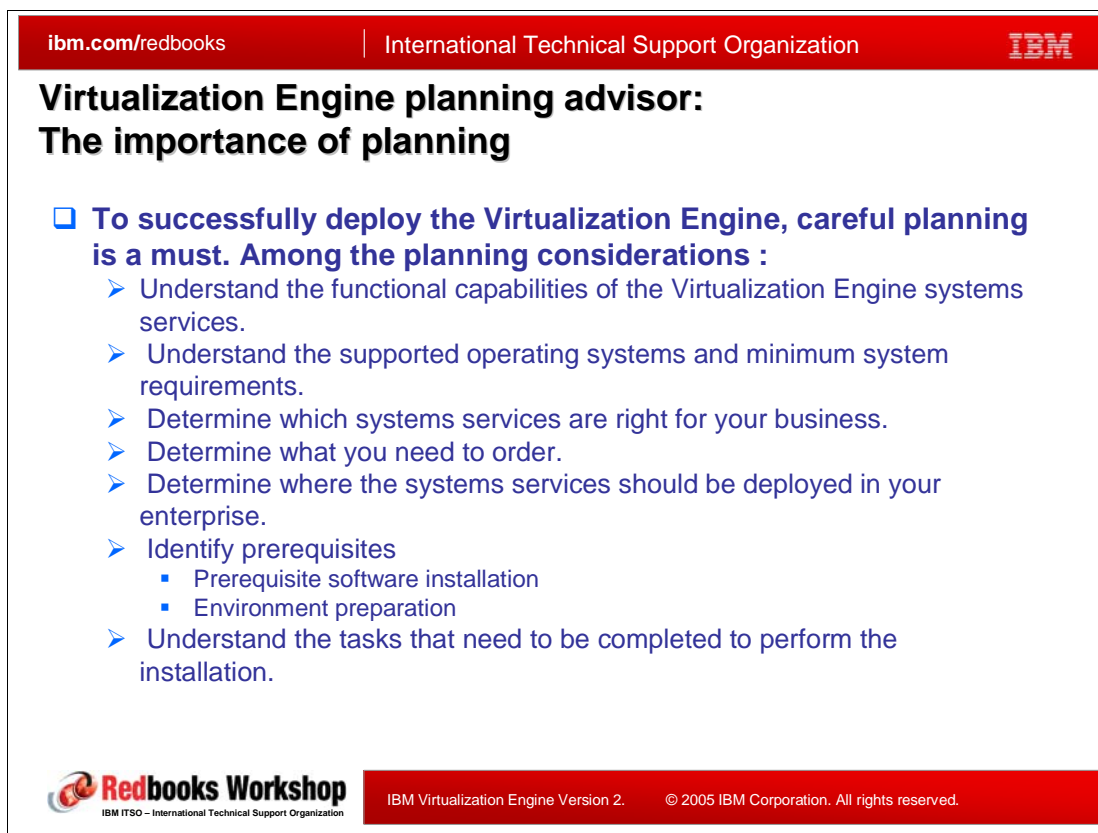
Remember, if you have downloaded the PDF, you do not need to be actively connected to the internet.

8.2 Planning advisor

In this section we discuss:

- ▶ The importance of planning your enterprise's use of Virtualization Engine components and products
- ▶ How the Virtualization Engine's Information Center-based planning advisor can assist you
- ▶ Examples using the Virtualization Engine planning advisor
 - Build Your Engine
 - Customized Installation Plan

8.2.1 Role of the planning advisor



The screenshot shows a document header with 'ibm.com/redbooks' on the left, 'International Technical Support Organization' in the center, and the 'IBM' logo on the right. The main title is 'Virtualization Engine planning advisor: The importance of planning'. A blue square icon precedes the text: 'To successfully deploy the Virtualization Engine, careful planning is a must. Among the planning considerations :'. This is followed by a list of seven blue arrow icons pointing to the following text: 'Understand the functional capabilities of the Virtualization Engine systems services.', 'Understand the supported operating systems and minimum system requirements.', 'Determine which systems services are right for your business.', 'Determine what you need to order.', 'Determine where the systems services should be deployed in your enterprise.', 'Identify prerequisites' (with two sub-bullets: 'Prerequisite software installation' and 'Environment preparation'), and 'Understand the tasks that need to be completed to perform the installation.'. The footer contains the 'Redbooks Workshop' logo, 'IBM ITSO - International Technical Support Organization', 'IBM Virtualization Engine Version 2.', and '© 2005 IBM Corporation. All rights reserved.'

Figure 8-6 Virtualization Engine planning advisor: The importance of planning

Deployment of the Virtualization Engine capabilities within your enterprise can be a fairly complex undertaking depending upon your environment, how many Virtualization Engine products you want to use, and the size of the network (for example the number of systems and workstations you want to manage, and so forth).

To deploy the Virtualization Engine successfully with this possible complexity, proper planning in advance is essential. You need to understand functional capabilities of the virtualization capability and which systems and workstations you want to use as a server (managing node) or as a managed servers.

- ▶ You need to research and evaluate which products you are interested in. The Virtualization Engine information within the Information Center can assist you in determine which products to use.
- ▶ Once that is done you need to understand the supported operating systems and minimum system processor, memory and disk storage capacity requirements of each virtualization product. Then, you need to determine where the components of the Virtualization Engine product should be installed and deployed in your enterprise.
- ▶ You also need to identify prerequisite information and understand software prerequisites, and environment preparation prerequisites that must be satisfied. For example, EWLM requires a WebSphere Application Server product somewhere in your network. IBM Director requires at least one server system, at least one IBM Director agent being managed, and at least one IBM Director console component installed. Depending on the operating system the IBM Director component is being installed on this can involve a single hardware platform and operating system or multiple hardware platforms and operating systems.

For example an i5/OS partition can run the IBM Director server and agent components, but not the IBM Director console component. This means at least one other hardware platform or operating system is required for the IBM Director console. In a simple case, this additional operating system could be on one of the following:

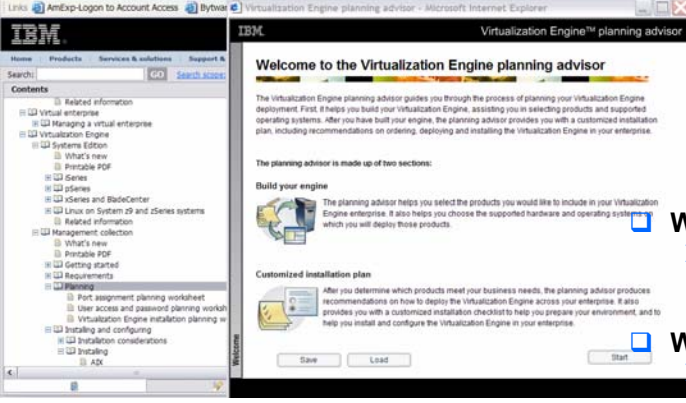
- One AIX or Linux partition on an iSeries or IBM System i5 family system, or a separate hardware platform.
 - One xSeries server running Windows or Linux on an integrated xSeries Server for iSeries or connected to an iSeries or IBM System i5 family system via the Integrated xSeries Adapter (IXA).
 - A separate hardware platform running a Windows operating system, a Linux distribution, or IBM AIX.
- ▶ Finally, you need to understand all the tasks that need to be completed to perform the installation.

How can you prepare and determine all these complex questions and steps above and complete proper planning? The answer is the Virtualization Engine planning advisor.


8.2.2 Overview

ibm.com/redbooks
International Technical Support Organization

Planning advisor : Overview



- ❑ **What is it?**
 - An interactive, Flash-based tool that asks you questions about your enterprise and how you plan to use the Virtualization Engine, then translates your input into recommendations on ordering, deploying, and installing.
- ❑ **Where do I find it?**
 - In the IBM Systems Software Information Center on the Web
- ❑ **When do I use it?**
 - When you are ready to begin the process of implementing the Virtualization Engine in your enterprise, even before you place your order if you like.



IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-7 Planning advisor: Overview

The Virtualization Engine planning advisor is a tool to help you complete your planning, by walking you through the appropriate questions and considerations. This provides a more interactive, customized approach to planning than that is provided by traditional planning manuals. It is most valuable when you are interested in installing and deploying a combination (more than one) of Virtualization Engine products in your enterprise. It provides a comprehensive look at planning, no matter what combination of Virtualization Engine products you desire, customizing the output to fit your configuration. Therefore, the planning advisor can help users understand what to order, how to deploy the various components of the respective systems services, and how to install the systems services on the systems in their enterprise.

What is it?

It is a interactive flash-based tool that guides you through a series of items you can select and it gathers information from entries about the Virtualization Engine products you are interested in, your network, and the hardware and operating system combinations you have within your enterprise. Then it guides you through the process of building a Virtualization Engine configuration.

After you complete the *Build Your Engine* Web pages, it summarizes your responses, enabling you to go backward or proceed through pages to help you order the products and provides recommendations for proceeding to install the products you would order.

Thus, the planning advisor enables you to build a customized look at your Virtualization Engine network.

The Planning advisor includes:

- ▶ Inclusion of new Virtualization Engine V2 capabilities.
- ▶ Rich content to illustrate the value of each component to aid you in selecting the right products for your network. There are conceptual animations for each product.
- ▶ Detailed explanation of the licensing and packaging structures.
- ▶ Ability to save the planning advisor recommendations to HTML format.
- ▶ Performance recommendations based upon previous experiences with Virtualization Engine products.
- ▶ Ties with planning-related content in the documentation found within the Information Center
- ▶ The capability to print your planning results for further study and use

Where do I find it?

It is housed in the Information Center. See the left tree structure shown in Figure 8-7 on page 282. This shows the Management collection selection → Planning path to the Advisor information.

When do I use it?

The planning advisor can be used anytime. It can be used for exploratory purposes, for discovery purposes, for the customers that need to learn more about the products described within information center before they start making purchasing decisions. It can also be used after the decision has been made, before an order has been placed or even after an order has been placed

The following sections provide some examples of planning advisor use.

8.2.3 Planning advisor flow

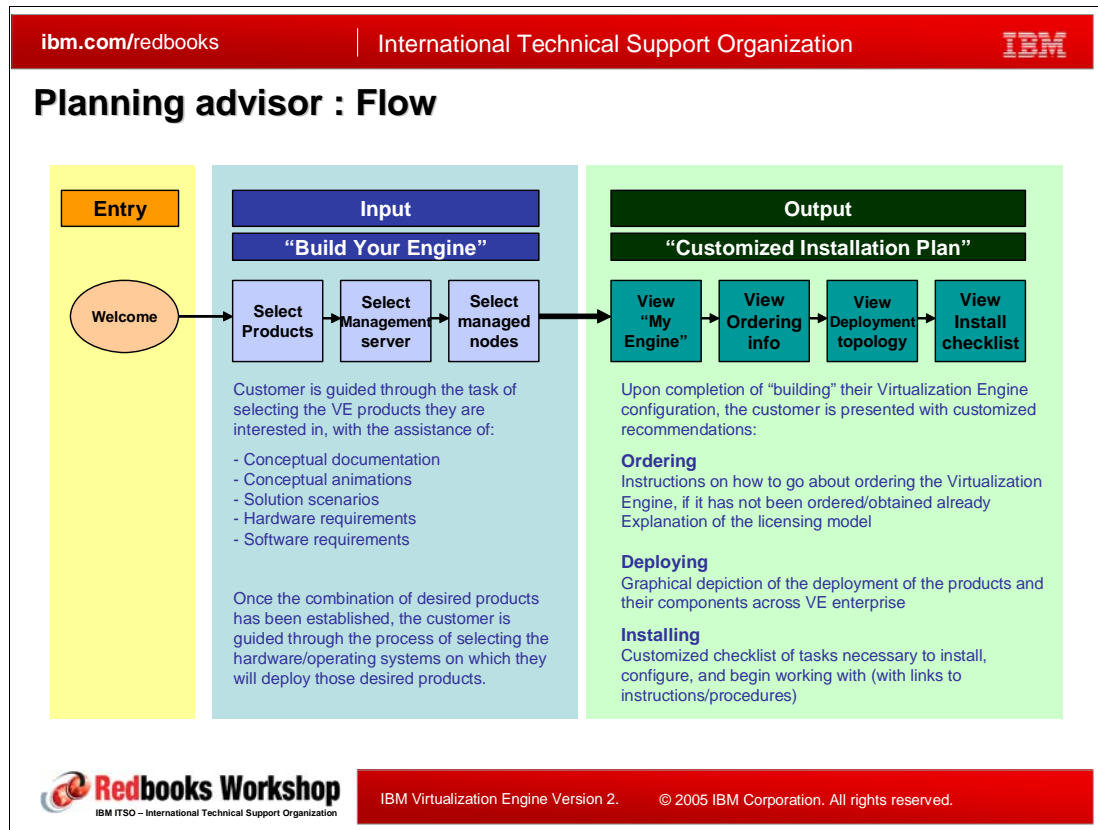


Figure 8-8 Planning advisor: Flow

The Planning advisor consists of two main sections which are *Build Your Engine* and *Customized Installation Plan*.

- ▶ Build your Engine is where you select products, hardware system, where appropriate which one is the server and which are agents, and which operating systems are to be used.
- ▶ The Customized Installation Plan is the output of building your engine, which leads you to ordering the appropriate products, later to installing them.

You access the planning advisor from the information center and it guides you through the task of selecting the Virtualization Engine products you are interested in, with assistance of conceptual documentations, conceptual animations, solution scenarios, hardware requirements, and software requirements. It ultimately helps you figure out where you can install virtual engine products on which hardware, and what hardware, operating system, operating system version will be appropriate by showing you the operating system supported, hardware requirement for the products that you are going to install. Therefore you can be sure that you've got the appropriate hardware, processor speed, memory capacity to deploy your Virtualization Engine products. With the set of input you have made presented in a summary you can go back and make changes, identify system hardware requirements (such as a system's minimum recommended performance ratings) and proceed to ordering the appropriate software.

Obtaining the appropriate hardware with at least the minimum performance ratings listed is a process outside of this planning advisor tool.

The input section of planning advisor consists of three main sets of information:

- ▶ Virtualization engine products.
- ▶ Your management server.
- ▶ The managed nodes.

The output section consists of the following main sections:

- ▶ My engine
- ▶ Ordering
- ▶ Installing

8.2.4 How to select your Virtualization Engine products



Figure 8-9 Planning advisor: Build Your Engine - Virtualization Engine product selection

In the Build Your Engine phase, the advisor presents users with the products that make up the Virtualization Engine, as shown in Figure 8-9. You select the ones you want to use. Though we don't show examples in this presentation, you can see the **Learn more** and **View demo** links. The windows behind each of these two links are where you can examine what the product is and get help in determining if the product is right for you.

The advisor shows you solution scenarios that describe typical business case instances in which a company uses multiple Virtualization Engine products to achieve real business goals. These scenarios demonstrate how to use virtualization technologies and solutions to solve common systems management challenges. Although these challenges may not represent every business case, they will showcase the advantages of integrating virtualization technologies and products into an IT infrastructure.

Operating system and prerequisites are covered. The **View demo** link includes a Flash demo.

As you make product selections (Add to My Engine) they are tracked in the right pane. In our screen capture you see Virtualization Engine console already selected. Since the Virtualization Engine console always requires another Virtualization Engine product, such as Enterprise Workload Manager, IBM Director, or Resource Dependency Services, this is an "in-progress" screen example.

In the example we used to build our screen captures, we next selected IBM Director. The fact that we selected IBM Director will show up in the screen captures shown in the next planning advisor topics.

8.2.5 How to select your management servers

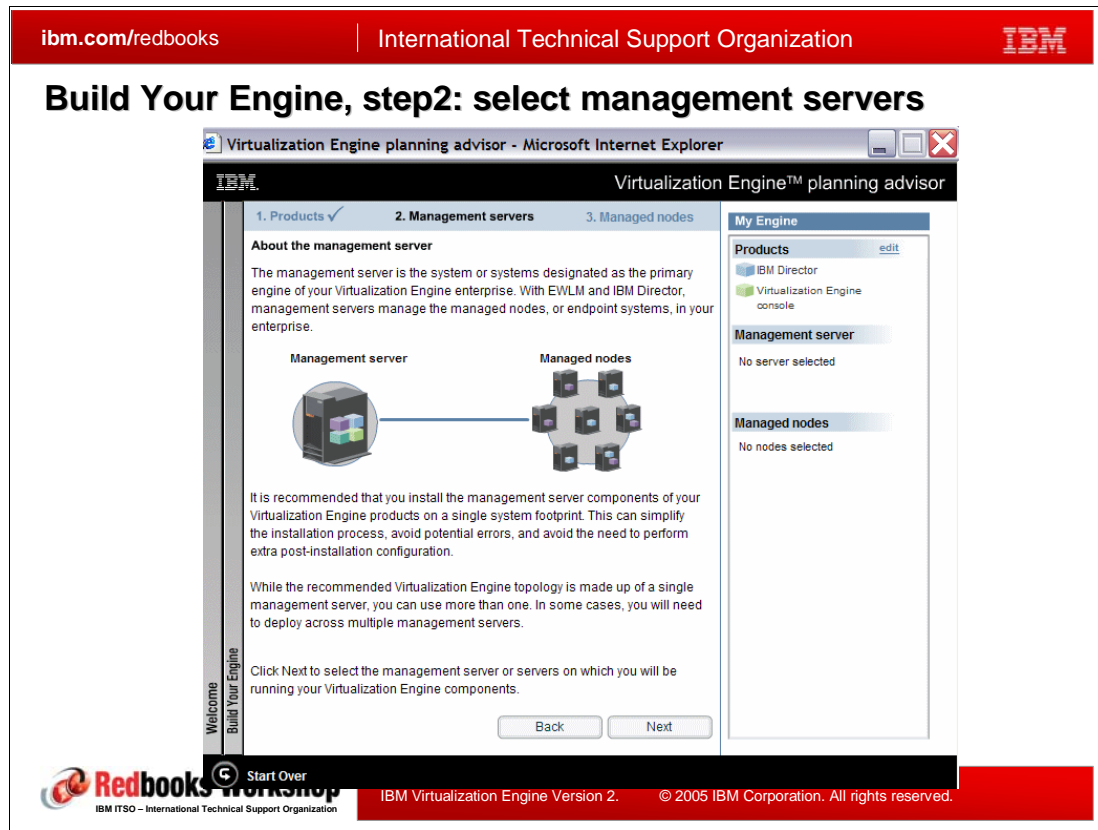


Figure 8-10 Planning advisor: Build Your Engine - Management servers

The left pane in Figure 8-10 shows the first window under the Management server's section. You can read about the considerations for a management server. The planning advisor explains the concept of a management server, including listing the supported management operating systems and which functions are supported on each operating system.

In the upper right My Engine "progress pane," you can see that we had already selected Virtualization Engine console and IBM Director.

Clicking **Next** brings you to the window shown on the right of our figure. In our example we have specified to deploy Virtualization Engine console and IBM Director on an iSeries system running i5/OS V5R3. Clicking the Define Management Server button leads to other windows and eventually to the Managed nodes section of building My Engine.

8.2.6 How to define your managed servers

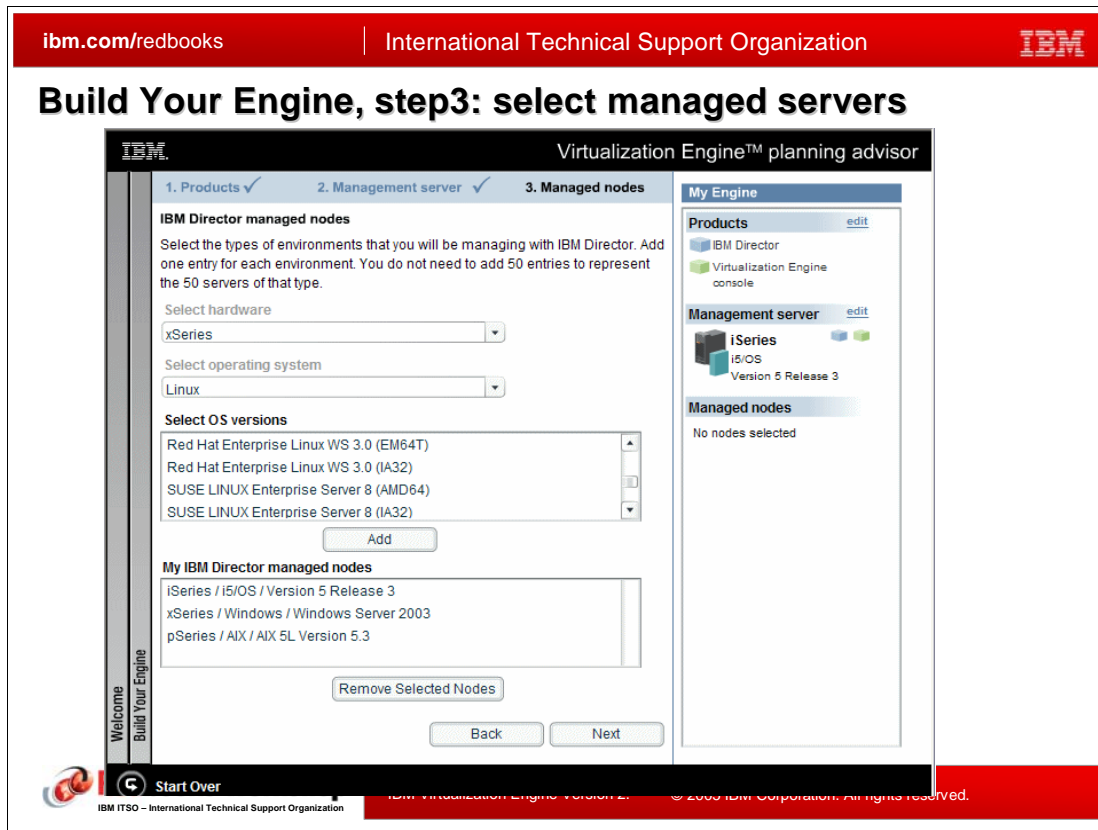


Figure 8-11 Planning advisor: Build Your Engine - Management nodes and a built engine

Completing the selection and definition of one or more management servers leads to the Managed nodes section of the planning advisor. You select the different hardware platforms, each platform's operating system that will be a managed node. You define each unique hardware platform and operating system combination that will be on your managed nodes.

You do not do this for each physical hardware system managed node in your network. Rather you select the unique combination of hardware platform and its operating system in your network for one node you actually have in your network. For example, if you have 10 xSeries systems that will be managed nodes and each one is running Windows/Windows Server 2003, you specify this combination only once.

In Figure 8-11, as shown in the left window, we have already selected:

- ▶ iSeries system running i5/OS V5R3.
- ▶ pSeries system running AIX 5L V 5.3.
- ▶ xSeries system running Windows Server 2003.

We are about to select an xSeries system running a Linux operating system distribution.

In the right window of Figure 8-11 we show the completed output of our Build Your Engine phase. In the right pane of the window, you see the summary of the Virtualization Engine products, the managed server, and the list of managed node unique hardware platform - operating system combinations in our example network.

This helps you consider how many products you need to order and deploy. You are responsible for knowing how many hardware platforms or operating system partitions the products you selected have to be deployed to.

You can make changes by clicking the Edit button.

Important: You need to aware that the managed node concept applies only for IBM Director or EWLM. It is not applicable to Resource Dependency Service or Virtualization Engine console.

The next step is to move on to a customized installation plan.

8.2.7 Your customized installation plan

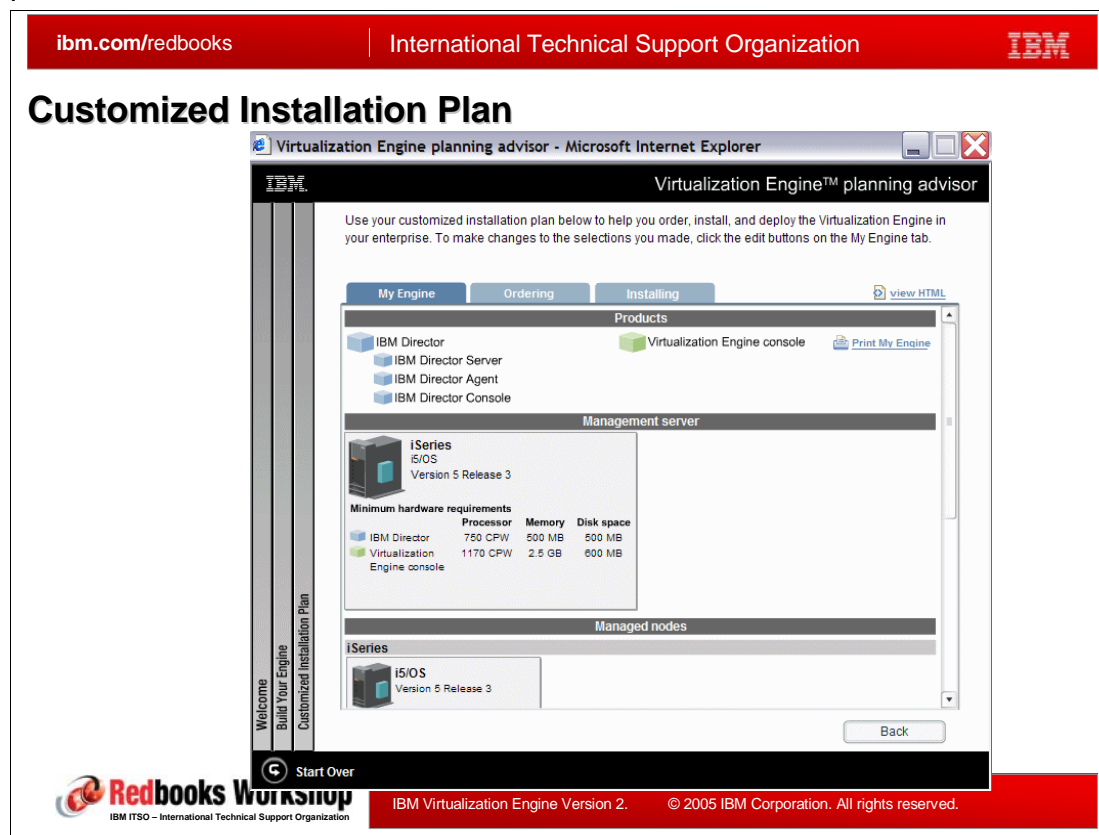


Figure 8-12 Planning advisor: Customized Installation Plan

If you finish your Virtualization Engine topology with the planning advisor, you can see the screen similar to the one shown in Figure 8-12 which has three tabs: My Engine, Ordering, and Installing. These are the outputs of your plan based on your input while “Build Your Engine” flow.

My Engine

This tab is the inventory of your configuration listing of component inventory selected by you during the “Build Your Engine.”

From this tab you can access deployment information. We summarize deployment considerations here.

You can see the graphical depiction of the customized Virtualization Engine topology. This shows the Virtualization Engine components and where the various components need to be installed in the enterprise with information about hardware and system requirements. The Build Your Engine window shown in Figure 8-9 on page 286 is an example of different types of hardware and operating systems with EWLM and IBM Director on an iSeries i5/OS management server. The Management server section on top shows two management server component for the EWLM domain manager and IBM Director server. If you select **view details** link, more information about the minimum hardware and software requirements for each of the component as well as prerequisite dependencies will be shown. The bottom half of the screen shot shows all the managed nodes defined at previous stage. This is a simple example of lots of different types of hardware and operating systems which can be pretty complicated to understand and plan to deploy without planning advisor. However planning advisor help users understand their topology with intuitive graphical deploying tab.

Ordering

This tab provides ordering recommendations for customers that haven't yet obtained the Virtualization Engine products code, if necessary. This step is optional because this information is not needed for customers who have already obtained Virtualization Engine components and products.

However, for customers who are looking for a place to order, this tab provides information on ordering tools, product ids as well as licensing model helping them figure out what they need to do to order Virtualization Engine products.

Installing

This tab provides customized checklists for you to install and configure your Virtualization Engine configuration. It includes recommended order of installation, links to information center documents on completing tasks, and level of difficulty indicators to indicate the amount of time and complexity required to complete a task.

Without use of the planning advisor, for example, if you are interested in IBM Director and EWLM, you need to review IBM Director documentation and then have to review EWLM documentation. Then you must try to figure out the order in which to perform the required tasks. The planning advisor provides the recommended ordering of tasks as well as links to the procedure documentation and how to complete those tasks.

The planning advisor output also presents the level difficulty indicators (lower right window of Figure 8-12 on page 289) which try to help users understand how difficult or how time consuming a specific task may be. Just providing a simple checklist doesn't convey which tasks may take longer or be more difficult than another task.

8.3 The common runtime components

This section introduces the Virtualization Engine common runtime components.

8.3.1 Virtualization Engine software components

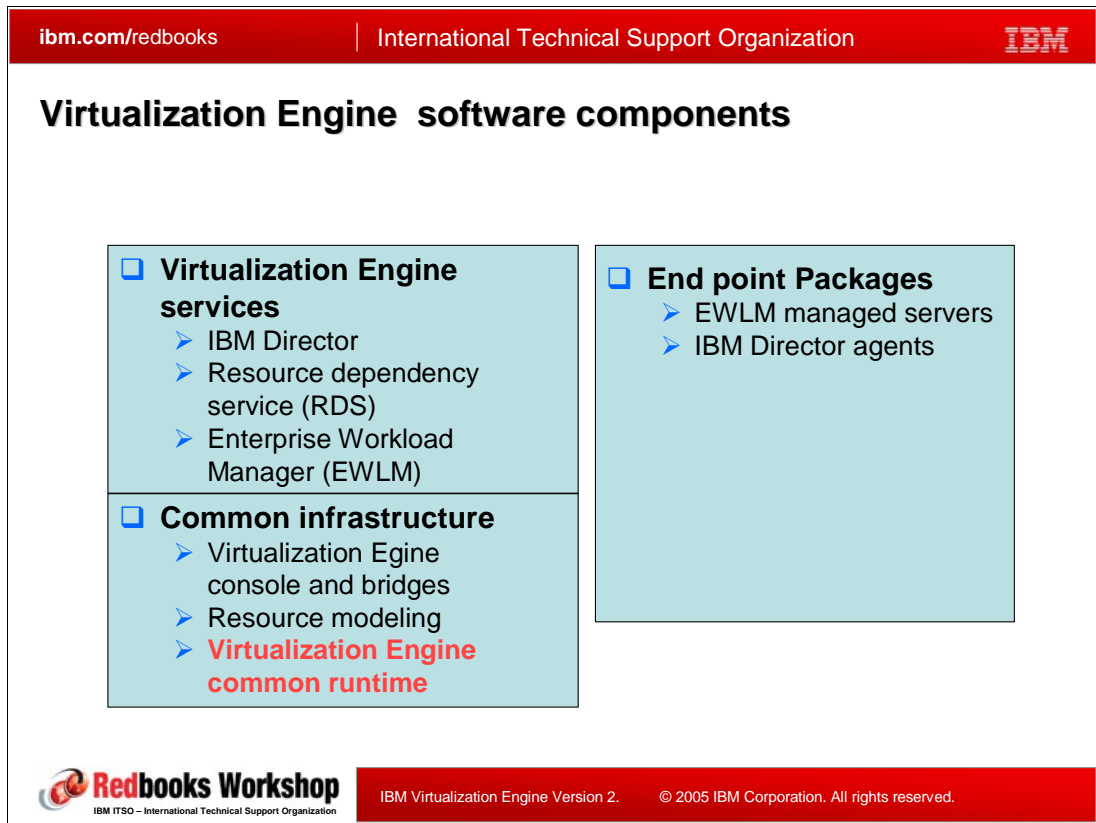


Figure 8-13 Virtualization Engine software components

The IBM Virtualization Engine is made of several components such as the Virtualization Engine console, EWLM, or the IBM Director. Each component has its own software “sub”-components such as the Virtualization Engine console server, the Virtualization Engine console bridge, the EWLM domain manager, the EWLM control center, the EWLM managed server, the IBM Director server, the IBM Director agent, etc. Some of them, being Java programs, need to run within a WebSphere Application Server instance, some of them need a database and use DB2 to host the data, and some of them need a repository and use ITDS to manage an LDAP server.

From an installation point of view, the Virtualization Engine software components are categorized into three groups. These groups do not represent a physical packaging of the components but a logical grouping from an installation point of view:

- ▶ The Virtualization Engine services are the server components of the Virtualization Engine. They include the IBM Director server, Resource Dependency Services, and the EWLM server components (the Domain manager and the control center).
- ▶ The common infrastructure is the name used to represent all the general software components which relate to the Virtualization Engine services servers and includes:
 - The Virtualization Engine console, which is an integrated console which can launch the IBM Director, the EWLM management console, or other specific consoles.

- The resource modeling framework, which provides the base facility for Resource Dependency Service.
- The Virtualization Engine common runtime components, which are the software modules used by the Virtualization Engine services.
- ▶ The end point packages are the agent software components of the Virtualization Engine services. It includes the EWLM managed server software, and He IBM Director agents.


8.3.2 Virtualization Engine common products

ibm.com/redbooks
International Technical Support Organization
IBM

Virtualization Engine common runtime

- **WAS + DB2 + Directory server + ISC**
 - WebSphere Application Server – WAS 6.0.2 + VE iFixes
 - DB2 Universal Data Base - DB2 UDB 8.2 SP1
 - IBM Tivoli Directory Server – ITDS 6.0 SP1
 - IBM Integrated Solutions Console – ISC 6.0.1

- **Core elements**
 - Java Runtime Environment – JRE 1.4.2 SP2 for DB2
 - IBM Solution Install framework – SI 1.2.1.10
 - iLog Jviews
 - International Components for Unicode - ICU4J
 - IBM Common Logging module - ICL
 - Fix acquisition services
 - Deployment services



IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-14 Virtualization Engine common runtime

The Virtualization Engine common runtime components include a set of middleware infrastructure that is commonly used by different Virtualization Engine management servers. It is only installed in the servers of the Virtualization Engine management services and they are not installed in the end points (managed servers). Using the Virtualization Engine common runtime components, the Virtualization Engine management servers can share a common version and levels of the infrastructure. The Virtualization Engine common runtime components are supported on the following operating systems:

- ▶ Windows Server 2003 Standard Edition
- ▶ AIX 5L Version 5.3
- ▶ Linux (SLES9 and RHAS 4)
- ▶ i5/OS V5R3

The Virtualization Engine common runtime components include:

- ▶ An Java application server, namely a WebSphere application server; WebSphere Application Server 6.0.2 is used.
- ▶ A database server, namely DB2; DB2 UDB 8.2 is used.
- ▶ A directory server, namely ITDS which will use the database server; ITDS 6.0 is used.
- ▶ The Integrated Solution Console (ISC), on which the Virtualization Engine console is installed; ISC 6.0.1 is used. An ISC instance consists of a framework and a set of one or more console components provided by the Virtualization Engine console. ISC is constructed on WebSphere Portal technology and the administrative activities are executed as portlets that have access to all of the standard services within the Java 2 Enterprise Edition (J2EE) environment that is provided by the underlying WebSphere Application Server
- ▶ And core elements which include the following:
 - A J2EE server needs to be used; JRE 1.4.2 is used.
 - The IBM Autonomic Deployment Engine framework. The IBM Autonomic Deployment Engine, an Autonomic Computing IBM initiative, is a framework to centrally manage software changes during product and maintenance install and uninstall using a database of XML descriptors. It enables the componentization and the deployment of component based solutions with hierarchical development and packaging of components. It is based on Open standards for cross platform and multi-product change deployment.
 - iLog Jviews¹, which is a set of Java graphics products for building Web-based GUI displays: diagrams, charts, schedules, etc.
 - The International Components for Unicode; ICU4J is a mature, portable set of C/C++ and Java libraries for Unicode support, software internationalization and globalization, giving applications the same results on all platforms.
 - The IBM Common Logging module (ICL). This module is a result of the autonomic computing initiative to develop a common logging and tracing tool.
 - And services to acquire or deploy modules for specific functions

Some fixes need to be installed with the components; refer to your installation package and to the Virtualization Engine fix Web site for the latest information:

<http://techsupport.services.ibm.com/server/VirtualizationEngine>

The common runtime components do not necessarily have to be installed in the same server on which the Virtualization Engine services (management servers) run. Table 8-1 shows the relationship of the installation of the Virtualization Engine services and the Virtualization Engine common runtime components, where

- ▶ “L” means that the common runtime component must be installed locally,
- ▶ “E” means that the common component can be installed on any server in the same environment,
- ▶ “R” means that the Virtualization Engine service can use a remote access to the common component. Note that Resource Dependency Service can access DB2 as a remote database.

¹ More information about iLog Jviews can be read at the following Web site: <http://www.ilog.com/products/jviews/>

Table 8-1 Virtualization Engine common runtime installation relationships

	EWLM domain Manager	Resource Dependency Service	Virtualization Engine console	Console bridges
WebSphere	L	L	L	L
DB2	N/A	L/R/E	E	E
Directory Server	E	E	E	E
ISC	N/A	N/A	L	N/A
Resource Manager	N/A	L	N/A	N/A
Core components	L	L	L	L

The runtime components are licensed for internal Virtualization Engine use only.

Note: IBM Director does not have Virtualization Engine common runtime dependencies. However, it is recommended to install IBM Director before other services to share control information with them. The installation of IBM Director is a separated native process except on i5/OS. On i5/OS, the installation is also included in Virtualization Engine installation wizard. IBM Director joins an Virtualization Engine environment when an IBM Director bridge to the Virtualization Engine console is installed on the same system.

8.3.3 Repositories

ibm.com/redbooks | International Technical Support Organization **IBM**

Virtualization Engine common runtime

- **Directory server repositories**
 - **Virtualization Engine environment repository**
 - Logical system group information
 - Installation/Configuration setting
 - **WAS security server repository**
 - For Virtualization Engine console
 - For Virtualization Engine console bridges
 - For Resource Dependency Service
 - **Virtualization Engine Identity Mapping Repository**
 - IMR

Redbooks Workshop
IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-15 Virtualization Engine repositories

In the Virtualization Engine common runtime directory server, three repositories are created.

- ▶ A WebSphere Application Server security user registry. It is used by the Virtualization Engine console, the Virtualization Engine console bridges, and the Resource Dependency Services.
- ▶ A Virtualization Engine identity mapping repository (IMR). It contains the Enterprise Identity Mapping identity data including the collection of defined identifiers, associations and user registries. If you already have an existing Enterprise Identity Mapping repository (EIM) you can use it as your IMR. Some Virtualization Engine components can lookup at EIM identity.
- ▶ A Global Configuration Repository (GCR). It logically groups the systems with Virtualization Engine software installed and includes their installation, configuration, and setting information. The root entity is named “Virtualization Engine environment”. Any instance of the virtualization engine can belong to only one environment from installation to uninstallation.

The Global Configuration Repository

To enable a more integrated operations management environment, the Virtualization Engine Version 2 provides a centralized repository of installation and configuration information to track all management server software deployed in the customer’s environment, based of the LDAP technology.

A GCR is a network-scoped repository of Virtualization Engine configuration information, including configuration settings and software topology. It will use specific Java classes that

provide a public API for accessing the information. The Virtualization Engine console topology viewer can be used to enable administrators to graphically display the software topology and configuration settings.

GCR will use ITDS as an LDAP server. The LDAP schema needs specific extensions to provide the support for the GCR information. Virtualization Engine Version 2 ships ITDS 6.0 for the LDAP, but Virtualization Engine Version 2 will also support the use of ITDS 5.1 and 5.2, as well as the IBM Directory Server that ships with I5/OS.

The customer can point to an existing and valid ITDS or install the ITDS supplied by the Virtualization Engine common runtime.

In this Virtualization Engine Version 2, the IBM Director does not exploit the GCR.

8.3.4 User IDs, passwords, and keys

The screenshot shows a web page from the IBM Redbooks Workshop. The page title is "Virtualization Engine common runtime". Below the title is a sub-section titled "User IDs, passwords, and keys". This sub-section contains three main items, each with a blue square icon and a list of requirements:

- WAS**
 - LTPA keys/password
- DB2**
 - Instance owner user/pass (OS)
- Directory server**
 - Owner user/password (OS)
 - Administrator DN user/password
 - Virtualization Engine DN user/password
 - Encryption seed string

At the bottom of the page, there is a red footer bar containing the "Redbooks Workshop" logo and the text "IBM ITSO - International Technical Support Organization". To the right of the logo, it says "IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved."

Figure 8-16 User IDs, passwords, and keys

When installing the Virtualization Engine common runtime, you have to specify user IDs and passwords.

- ▶ The WebSphere Application Server needs passwords for the generated LTPA keys, which relate to the WebSphere Application server single sign-on.
- ▶ DB2 needs a database instance owner user ID and a password for the operating system.
- ▶ The Directory server needs a server owner user ID and password, for the operating system.
- ▶ The Directory server also needs an administrator Domain Name (DN) user and password for both the server itself and the Virtualization Engine.

8.4 Installation tasks

This section provides the information pertaining to the installation of the Virtualization Engine services. We describe the actual installation process of the Virtualization Engine. Additional installation considerations and uninstallation are also discussed.

8.4.1 Deployment planning

The screenshot shows a web page with a red header bar containing the text 'ibm.com/redbooks', 'International Technical Support Organization', and the IBM logo. Below the header, the main content area is titled 'Deployment planning'. It contains two main sections, each with a blue square icon:

- Virtualization Engine Planning Advisor**
 - Available in the eServer Information Center
 - <http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp>
 - Produces customized recommendations for implementing the Virtualization Engine in your enterprise
- Planning work sheets**
 - Also in the eServer Information Center

At the bottom of the page, there is a red footer bar with the 'Redbooks Workshop' logo on the left and the text 'IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.' on the right.

Figure 8-17 Deployment planning

Typically, the Virtualization Engine environment consists of many heterogeneous servers and agents working in different operating system. Before installing the Virtualization Engine, you have to plan the server and agent topology such as:

- ▶ Which server should be the Virtualization Engine console server
- ▶ Which is the Director server
- ▶ Which servers should have the IBM Director agent installed
- ▶ Which servers should be in the EWLM management domain, etc.

Deployment planning here means all these tasks.

The Virtualization Engine planning advisor is a very helpful tool to be well prepared. Using the planning advisor, you can get the organized report of the Virtualization Engine deployment and get the installation procedure for each server and agent on every platform. For details, please see Section 8.2, “Planning advisor” on page 280.

The Information Center provides many work sheets, as shown in Figure 8-18, that help prepare and plan for the installation. We encourage you to print them before you start the installation, and you fill in it.

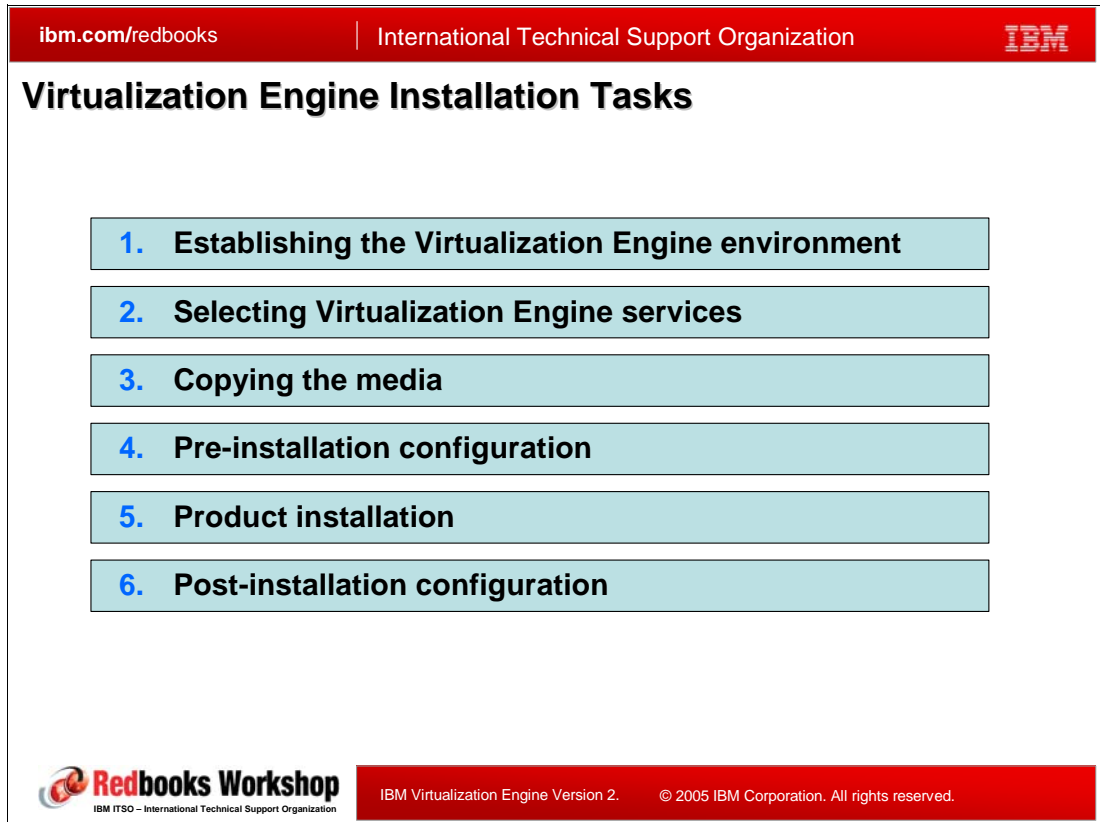
Identifier association planning work sheet
 See [Plan EIM associations](#) for an example of how to use this work sheet.

Table 5. Identifier association planning work sheet

Identifier unique name: <u>John S Day</u>		
User registry	User identity	Association types

Figure 8-18 Work sheet example

8.4.2 Virtualization Engine installation task list



ibm.com/redbooks | International Technical Support Organization **IBM**

Virtualization Engine Installation Tasks

1. Establishing the Virtualization Engine environment
2. Selecting Virtualization Engine services
3. Copying the media
4. Pre-installation configuration
5. Product installation
6. Post-installation configuration

Redbooks Workshop
 IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-19 Virtualization Engine installation steps

The actual installation of Virtualization Engine is done by the Virtualization Engine installation wizard. Figure 8-19 shows the installation steps in the wizard.

Note: This section is intended to show the outlined process of Virtualization Engine installation. Please refer to the latest Virtualization Engine information at the Information Center when you plan for the actual installation.

When using Windows, the IBM Virtualization Engine Wizard is started as an autorun process when inserting the media and a first panel is displayed as shown in Figure 8-20.

Author Comment: Christian. change fig8-20 when PA available

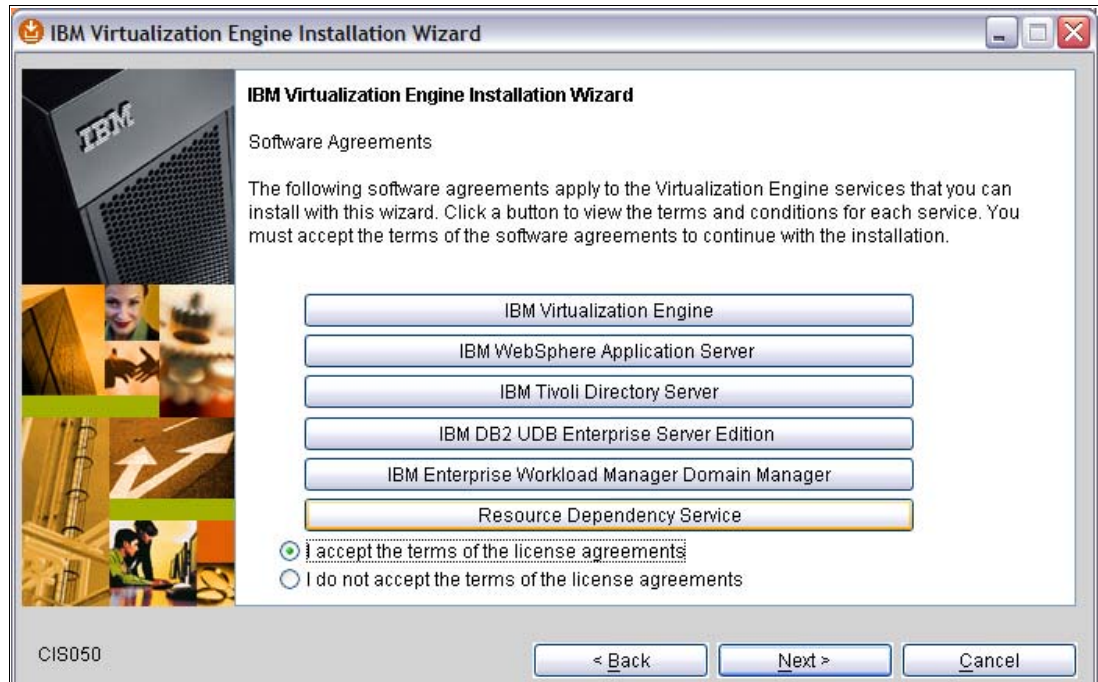


Figure 8-20 The IBM Virtualization Engine Wizard

Step 1: Establishing the Virtualization Engine environment

This step consists of the following tasks:

1. Accept software license agreements
2. Define the installation and logging directories
3. Verify the basic system requirements.

The wizard checks the environment to be sure it meets the requirements and that the process can be done.

4. Establish the LDAP directory server for installation.

In the directory server establishing step, you can choose to use an existing IBM Tivoli directory server or to install a new local directory server via the wizard. In this step, Operating System user IDs and passwords for DB2 instance owners and directory server are required. You can also create new Operating System users in this step. Please recall that Virtualization Engine supports ITDS (IBM Tivoli Directory Server) 5.1, 5.2, 6.0, and IBM Directory Server for iSeries.

5. Provide the name of the Virtualization Engine environment.

In this step, you can select either to create an environment in an existing directory server or to join a previously created environment. Virtualization Engine environment stores global/common data for an entire deployment of Virtualization Engine which includes multiple servers. The GCR stores common security settings, LTPA keys for communication between WebSphere Application Servers, WebSphere Application Server security setting, and single sign on information. Once the Virtualization Engine environment is established for the Virtualization Engine installation, it persists until all of the Virtualization Engine

software is uninstalled from that installation directory. Required operating system level and minimum memory and storage space are also checked in this step.

Step 2: Selecting the Virtualization Engine Services



Figure 8-21 Selecting Virtualization Engine services

In this step, you choose which Virtualization Engine services to install. The same selection panel is shown for all Virtualization Engine offerings for each Operating System target platform.

Product selection affects dependent installation actions such as:

- ▶ Installation of the required common components
- ▶ Media prompting
- ▶ Repository creations
- ▶ Services installed

Step 3: Copying the media

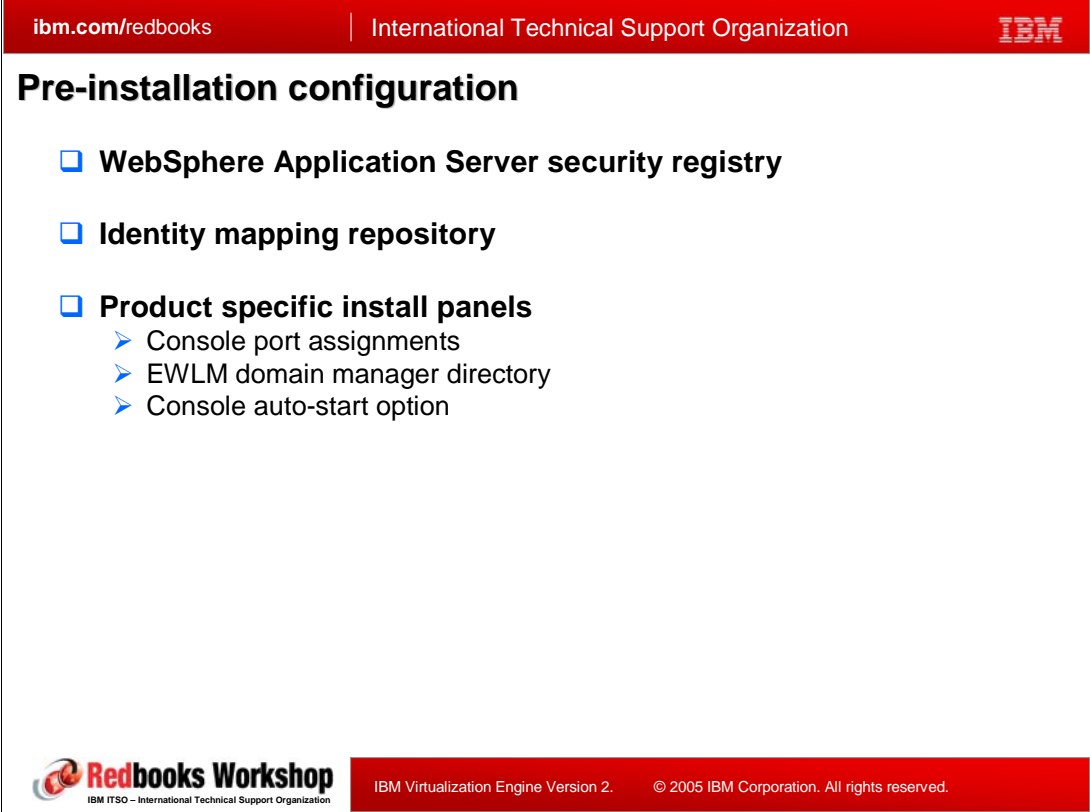
The installer prompts for each media based on the services selected. The installer first copies the necessary files to the hard disk in this step, then does the actual installation later. The staging approach provides some benefits.

- ▶ Assures all files available before starting the installation processing
- ▶ Allows core portion of installation, which takes long time, to be unattended
- ▶ Allows installation by preloading of media images to a file server

If you cancel the media copy and later restart the installer, you are only prompted for the medias which have not already been.

You should delete the media images when they are no longer needed.

Step 4: Pre-installation configuration



ibm.com/redbooks | International Technical Support Organization IBM

Pre-installation configuration

- WebSphere Application Server security registry**
- Identity mapping repository**
- Product specific install panels**
 - Console port assignments
 - EWLM domain manager directory
 - Console auto-start option

Redbooks Workshop
IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-22 Pre-installation configurations

In this step, the WebSphere Application Server security registry and the identity mapping repository are created in the directory server. Some other Virtualization Engine product specific configurations are also done.

WebSphere Application Server security Registry

In the directory server (ITDS), a WebSphere Application Server security registry is created and five user groups are created on the directory server:

- ▶ Virtualization EngineAdministrator: the administrator who can do everything
- ▶ Virtualization EngineMonitor: the user who can access some administration functions but read only.
- ▶ Virtualization EngineUser: any authenticated user
- ▶ Virtualization EngineAnonymous: any unauthenticated user
- ▶ Virtualization EngineManager: the user who can perform management operations, which are the subset of functions that the Virtualization EngineAdministrator can perform.

If you use the same directory server which hosts the Virtualization Engine environment and the WebSphere Application Server security repository, you can take the default settings. By default, the Virtualization Engine administrator named Virtualization EngineAdmin is created and is added to the Virtualization EngineAdministrator group.

Identity Mapping Registry

In the directory server, an identity mapping repository for EIM is created. If you do not utilize EIM, you can take default values, which will configure an identity mapping repository with minimal user input. By default, it uses the same directory server that hosts the Virtualization Engine environment.

Product specific install panels

Some additional panels allow to set some parameters for each of the Virtualization Engine services such as the WebSphere Application Server http port number for the ISC (Virtualization Engine console).

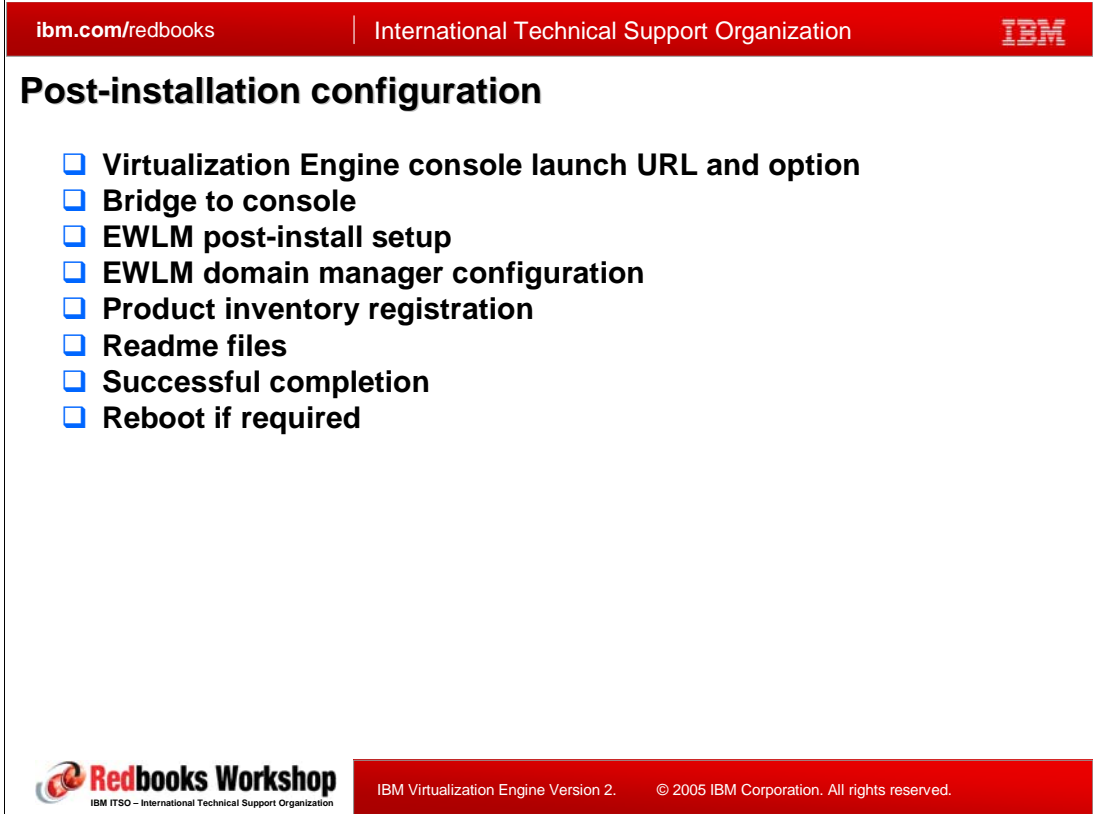
Step 5: Product installation



Figure 8-23 Product installation

In this step, the products are actually installed to the servers. Once the pre-install summary panel is shown, then products are installed with progress indicators, then a post-install summary panel is shown.

Step 6: Post-installation configuration



The screenshot shows a web page with a red header bar containing the text 'ibm.com/redbooks', 'International Technical Support Organization', and the IBM logo. Below the header, the main content area has a title 'Post-installation configuration' followed by a list of eight tasks, each preceded by a blue square icon. At the bottom of the page, there is a red footer bar with the 'Redbooks Workshop' logo and text, and a copyright notice: 'IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.'

ibm.com/redbooks | International Technical Support Organization IBM

Post-installation configuration

- Virtualization Engine console launch URL and option
- Bridge to console
- EWLM post-install setup
- EWLM domain manager configuration
- Product inventory registration
- Readme files
- Successful completion
- Reboot if required

Redbooks Workshop
IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-24 Post-installation configuration

After the installation of the products, some post installation configuration tasks must to be done. Most of them are processed automatically by the installer. It includes the necessary configuration process for each product, product inventory registration, and showing any readme files of the products.

8.4.3 Additional installation options and tasks

ibm.com/redbooks | International Technical Support Organization **IBM**

Additional installation tasks and options

- ❑ **Endpoint deployment**
 - IBM Director assistant
 - Remote Package Deployer (RPD)
 - `RPDx.sh -system <system> -userid -password <password> -p <packagedIU-file>`
 - File transfer
- ❑ **Installing from a file server**
- ❑ **Silent Install**
- ❑ **Migration**
 - IBM Director can be upgraded
 - Side-by-side for other services
- ❑ **Virtualization Engine Administration considerations**
 - Passwords and LTPA setting in GCR

Redbooks Workshop
IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-25 i5/OS Systems Edition

Endpoint deployment

After the installation of the Virtualization Engine services, you have to deploy some endpoint software for each client. They include:

- ▶ IBM Director console software
- ▶ IBM Director agent
- ▶ EWLM managed server

Endpoint software can be deployed in many different ways. IBM Director has its own update assistant. You can use the Remote Package Deployer (RPD), or you can transfer the self contained executable files and run them.

Remote Packager Deployer (RPD)

RPD is a command line utility to deploy endpoint software to other systems in the network without requiring agent software on the target system. It is installed with Virtualization Engine base modules and located in <Virtualization Engine Install>/rpd. For Windows, the command name is **RPDWin.bat**. For Linux, AIX, i5/OS, it is **RPDx.sh**. For example, the following command, Example 8-1, deploys EWLM Managed Server from a Linux machine where Virtualization Engine is installed, to a Windows server named TARGET.

Example 8-1 RPD command example

```
RPDx.sh -system TARGET -userid myUserID -password myPass123 -p  
/opt/IBM/Virtualization  
Engine2/ManagedNodes/EWLM/Win/META-INF/EWLMS_packagedIU.xml
```

RPD has some prerequisites such as secure shell, or i5/OS ToolBox for Java and host servers.

Installing from a file server

You can also install Virtualization Engine from a file server. This method is useful to install multiple servers using a single media copy activity, or to install a server which does not have media drives.

The Information Center has the information to determine the volumes you need for the products you plan to install. You can copy them all into the same directory using Operating System media copy commands or tools.

To install from the file server, issue the platform specific command to launch a graphical installation. A silent installation is also available.

Silent install

Silent mode install is accomplished by copying the media to a local storage, updating the response files, and launching the installer. There is no automated tool for silent installation. You have to update the response file manually. The password values set in the response files must be set as hashed value. You can use the utility provided with the installer to make hash values of passwords.

Migration

IBM Director server will migrate its data and programs during the installation. Other Virtualization Engine components allow a side-by-side installation. When you install two versions of Virtualization Engine services, you have to install them to different directories and can activate either at a time.

Since all Virtualization Engine servers must be at the same level to function correctly, plan to upgrade any Virtualization Engine 1.1 services to Virtualization Engine 2.1 at the same time.

The inter operability between servers and their endpoints depends on each product. The Information Center includes the latest information.

Virtualization Engine Administration considerations

To avoid multiple user inputs, the Virtualization Engine installer saves LDAP administrator password, Virtualization Engine administrator password, and LTPA settings into the Global Configuration Directory (GCR, See Section 8.3.3, “Repositories” on page 295). At the writing time, there are no tools to keep the GCR password and LTPA settings current. If you change them after the installation, the Virtualization Engine installers and uninstallers will prompt the customer for the correct passwords if it detects authentication errors. The installers will upgrade the GCR information if the correct password is provided.

8.4.4 Specific consideration for i5/OS systems edition

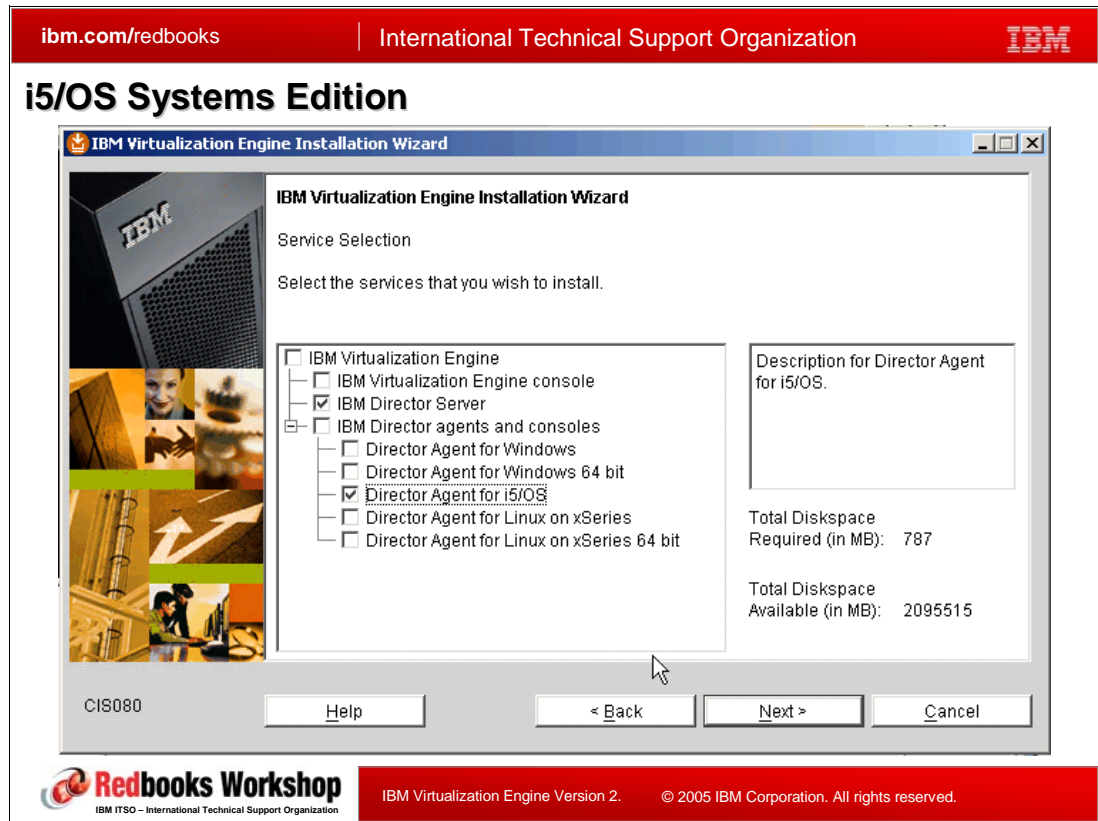


Figure 8-26 i5/OS Systems Edition

Virtualization Engine i5/OS Systems Edition is a product for eServer iSeries/i5 which is included with all i5/OS V5R3 new orders and upgrades. It includes:

- ▶ The IBM Director Server for i5/OS
- ▶ The IBM Director Agents
- ▶ The Virtualization Engine Director consoles
- ▶ The Virtualization Engine console
- ▶ The Virtualization Engine Director bridge to The Virtualization Engine console
- ▶ The Management Central bridge to the Virtualization Engine console

The installation of i5/OS Systems Edition is driven by a simplified version of Virtualization Engine installer, which has a reduced product selection panel. The installer is targeted for a single i5/OS system or partition and uses as many defaults as possible. If you want to install to multiple systems or set non-default Virtualization Engine environment, user IDs, port number values, you have to create a response file and perform silent install.

Though you can install IBM Director by Virtualization Engine installer on i5/OS Systems Edition, the Virtualization Engine environment is not established if you only select IBM Director.

If you select to install Virtualization Engine console, the c Management Central bridge to Virtualization Engine console is always installed. If IBM Director is also selected to install, or it is already installed on the server, the c IBM Director bridge to Virtualization Engine console is also installed.

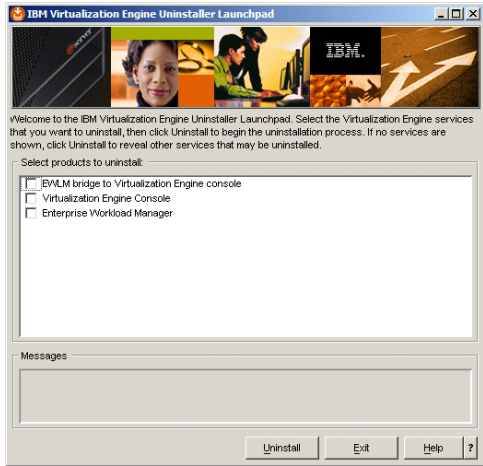
On i5/OS systems edition, when you select IBM Director agents or IBM Director console, only installation images are loaded. You can install them using IBM Director update assistant, Virtualization Engine remote package deployment or FTP and command execution to install on endpoints.

8.4.5 Uninstallation

ibm.com/redbooks
International Technical Support Organization

Virtualization Engine Uninstallation

- ❑ **Virtualization Engine Uninstaller Launchpad**
 - Windows: `uninstallVEWin.bat`
 - Linux: `uninstallVELinux.sh`
 - AIX: `uninstallVEAix.sh`
 - i5/OS: `uninstallVEi5OS.bat`



IBM Virtualization Engine Version 2.
© 2005 IBM Corporation. All rights reserved.

Figure 8-27 Virtualization Engine Uninstallation

Uninstallation of the Virtualization Engine services are available by the *Virtualization Engine Uninstaller Launchpad*. The Operating System commands to start the Virtualization Engine Uninstallation Launchpad are described in Figure 8-27. Uninstallation from i5/OS is done by a Windows machine remotely. The `uninstallVirtualization Enginei5OS` command is issued from a Windows machine which maps the i5/OS directory as a network drive.

The Virtualization Engine common runtime components can be also uninstalled individually. They will only be presented for uninstallation when all dependent services have been uninstalled and are no longer requiring the component. DB2 and ITDS can be uninstalled any time but a warning message is issued if there is a dependent Virtualization Engine environment. User can force uninstall because they have changed their environment to use DB2 and ITDS on another server after the Virtualization Engine installation.

Silent uninstallation is also supported by providing a response file for the Uninstaller Launchpad.

8.5 Products packaging



ibm.com/redbooks International Technical Support Organization 		
Virtualization Engine Version 2 products		
	Product Number	Product name
iSeries	5733-DR1	IBM Director with Virtualization Engine Console for i5/OS, V5.10 - Announce letter 205-289
	5733-EWA	IBM Virtualization Engine Enterprise Workload Manager for i5/OS, V2.1 - Announce letter 205-290 **
iSeries pSeries xSeries	5733-DIR	IBM Director with Virtualization Engine Console, V5.10 - Announce letter 205-286
	5733-EWM	IBM Virtualization Engine Enterprise Workload Manager, V2.1 - Announce letter 205-286 **
	5733-RDS	IBM Virtualization Engine Resource Dependency Service, V2.1 - Announce letter 205-287
zSeries	5655-EWM	IBM Virtualization Engine Enterprise Workload Manager for z/OS, V2.1 - Announce letter 205-284 **
	5648-F08	IBM Virtualization Engine and Infrastructure Services for Linux on System z9 and zSeries, V2.1 - Announce letter 205-283 (includes RDS - 1Q06)
The following products are not part of the Virtualization Engine products; however they deliver identical functions:		
xSeries	5697-NFD	IBM Director, V5.10
pSeries	5765-DR1	IBM Director, V5.10 for System p5
 Redbooks Workshop <small>IBM ITSO - International Technical Support Organization</small>		IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-28 Virtualization Engine version 2 products

The distribution and the packaging of the Virtualization Engine V2 is different from the Virtualization Engine V1.

- ▶ The products are fulfilled and distributed via the standard IBM series fulfillment channels (AAS for iSeries & pSeries, Passport Advantage for xSeries, ESW for zSeries).
- ▶ The offerings are comprised of the Virtualization Engine common components, which consists of the Virtualization Engine installer, the Virtualization Engine console, and the various common runtime components of WebSphere, DB/2, ITDS, ISC; and the following specific functions:
 - IBM Director
 - IBM Virtualization Engine Enterprise Workload Manager (EWLM)
 - IBM Virtualization Resource Dependency Service (RDS)
- ▶ The media can be either CD or DVD to reduce the amount of media handling.

Table 8-2 is a summary of the product number features by media type, sorted by operating system platforms.

Table 8-2 Virtualization Engine: product number, feature and media

		AIX	i5/OS	Windows	z/OS	Linux on xSeries	Linux on POWER	Linux on zSeries
EWLM: Contains Domain Manager & Virtualization Engine common runtime components		5733-EWM	5733-EWM ^a	5733-EWM	5655-EWM ^b	5733-EWM	5733-EWM	5648-F08
CD	Non-Royalty agents	5849	5809	5869		5889	5829	
DVD	Non-Royalty agents	5859	5819	5879		5899	5839	5819
	Royalty agents	5901	5901	5901		5901	5901	5829
Director and Virtualization Engine console Includes all Director Agents		5733-DIR	5733-DIR ^c	5733-DIR		5733-DIR	5733-DIR	5648-F08
CD		5849	5809	5869		5889	5829	5809
DVD		5859	5819	5879		5899	5839	5806
	Director Extension							6000
Resource Dependency Service, Includes RM & Virtualization Engine common runtime components						5733-RDS	5733-RDS	5648-F08
CD						5829	5809	
DVD						5839	5819	5839

a. Plus refer to Table 8-3

b. 5655-EWM includes EWLM Domain Manager and managed servers for z/OS; it is distributed on a cartridge.

c. Plus refer to Table 8-3

Table 8-3 gives the product number information for an iSeries platform specific offering.

Table 8-3 : Specific Virtualization Engine product number and feature

		iSeries
EWLM: Contains Domain Manager & Virtualization Engine common runtime components		5733-EWA
DVD	Non-Royalty agents	5809
Director and Virtualization Engine console Includes all Director Agents		5733-DIR
CD		5809
DVD		5819

8.6 Architecture, performance, and sizing

The IBM Virtualization Engine (Virtualization Engine) V2.1, includes more complete and practical functionality than its first version. This chapter provides some guidance for how practitioners should approach the task of designing solutions, which involve the use the Virtualization Engine as well as some preliminary performance benchmarks derived from internal IBM testing.

It is expected that, using the information provided in this section, practitioners should be able to make preliminary decisions on high-level solution design topologies and hardware specification for the use of the Virtualization Engine. While some of the performance benchmarks presented here were gathered using version 1 of the Virtualization Engine, the team felt that there was enough correlation between those results and the new release of the Virtualization Engine so they are included. Extrapolation and interpretation of those Virtualization Engine version 1 performance numbers and their applicability to the new release of the Virtualization Engine are offered where appropriate.

The reader is cautioned that, as in the design of any non-trivial infrastructure solution, the final design and performance of complete Virtualization Engine solutions depend on numerous factors, which need to be considered within the context of the customer's business and technical requirements. Among the most important factors to consider when arriving at a final hardware/software solution are the non-functional requirements for the solution. These include requirements for solution characteristics such as reliability, availability, security, maintainability, and performance.

As I/T professionals know, the non-functional requirements associated with any particular solution have the most influence over its operational model (OM). The solution's OM consists of a design model where all software components required for the solution (deployment units, or DUs) are represented with the hardware platforms where they execute (nodes). The next section (General architectural guidance) provides some guidance on the types of OMs which are supported by the IBM Virtualization Engine. Subsequent sections cover particular performance information gathered for some components of the IBM Virtualization Engine

8.6.1 General architectural guidance

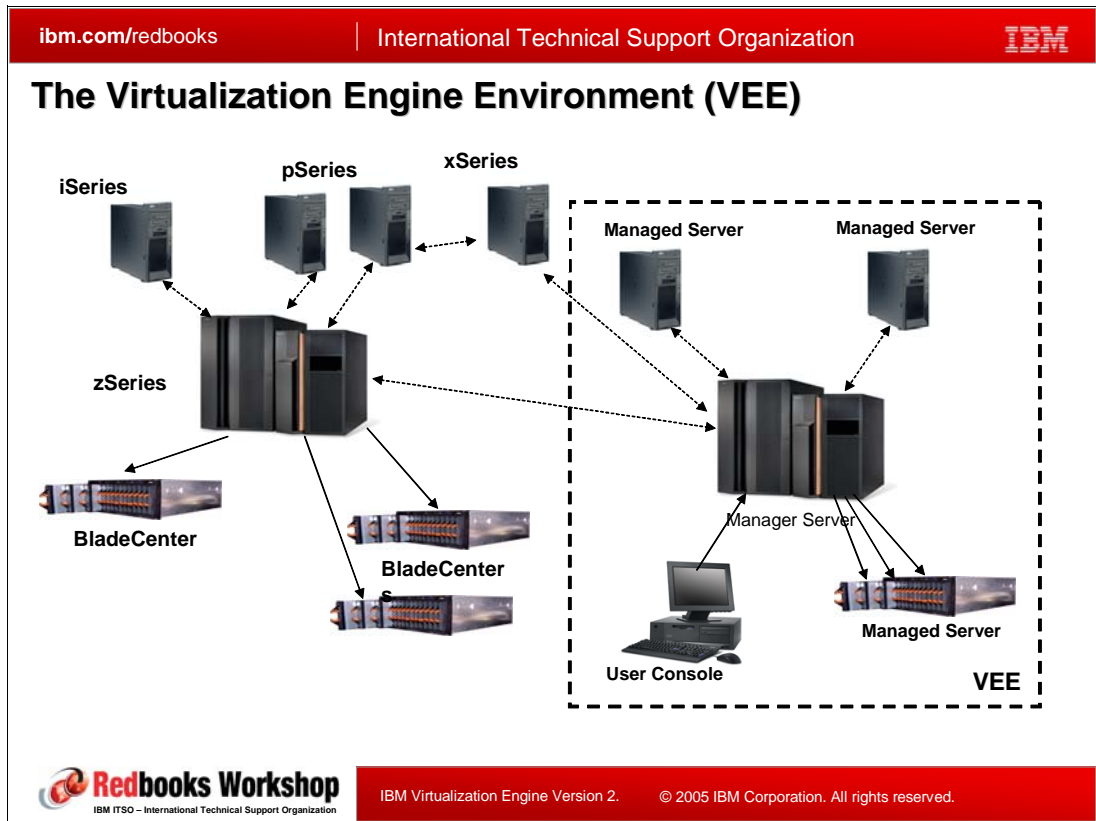


Figure 8-29 The Virtualization Engine Environment

One of the most important terms to remember when designing solutions using the Virtualization Engine is *Virtualization Engine Environment*. The Virtualization EngineE refers to a logical grouping of servers that have Virtualization Engine software installed on them. The different components that make up the entirety of the Virtualization Engine software suite can often be installed on the same server or on different servers, depending on the solution's requirements. This allows for a wide variety of design topologies that can be used for Virtualization Engine solutions.

In this section, we describe an approach that can be used when planning for and designing I/T solutions that incorporate the use of the IBM Virtualization Engine. The idea is not to cover all possible theory and guidance used when designing non-trivial hardware solutions, but to focus on those aspects that are specific to the implementation of the IBM Virtualization Engine. Additionally, we will provide some references to other resources where more general design information can be obtained.

As can be see on Figure 8-29, the Virtualization EngineE is usually comprised of a selected subset of an organization's I/T infrastructure. The selected servers host the necessary software to make the Virtualization Engine operate as desired. The servers in the Virtualization EngineE can be divided into two groups:

- ▶ **Manager Servers**

These are the servers on which the Virtualization Engine software components are loaded in order to implement the Virtualization Engine Version 2 functionality (i.e. scan, monitor, manage, view the infrastructure, etc.).

- ▶ **Managed Servers (also known as *Endpoints*)**

These are the servers which contain the organization's application and business logic. They are the ones upon which the Virtualization Engine services are applied so they can be managed.

The Virtualization Engine components that can be installed on and hosted by the Virtualization Engine can be classified into the following categories:

► Virtualization Engine Services

These are the software components that provide the basic functionality offered by the Virtualization Engine Version 2 and can be installed on one or more nodes of the Operating Model as part of the Virtualization EngineE. They are:

- IBM Director
- Resource Dependency Service
- Enterprise Workload Manager (EWLM)

► End Point Packages

These are software components that are loaded on servers which are managed and monitored using the Virtualization Engine version 2. They are:

- EWLM Managed Server Agents
- IBM Director Agents

► Common Infrastructure

These are software components on top of which the Virtualization Engine version 2 was built and are required for its operation. They are:

- Virtualization Engine console
- Bridges to Virtualization Engine console
- Resource Dependency Service Resource Modeling
- Common Components

Figure 8-30 shows a very simplistic OM where all the possible deployment units for the Manager Server are installed on one single server. There can be one or more Managed Servers and each of them has the EWLM Managed Server Agent and the IBM Director Agent installed so, both, EWLM and the IBM Director can manage them.

A simple configuration like this one, with only one Manager Server and one to many Managed Servers can be used for simple test or pilot project environments and it should operate well in those controlled scenarios. A sample hardware configuration that can work well for the Manager Server in these cases is one having, at least, a 2 Ghz 1-way Intel-based processor plus 1-2 Gigabytes of RAM memory and 2 hard disks (for improved I/O).

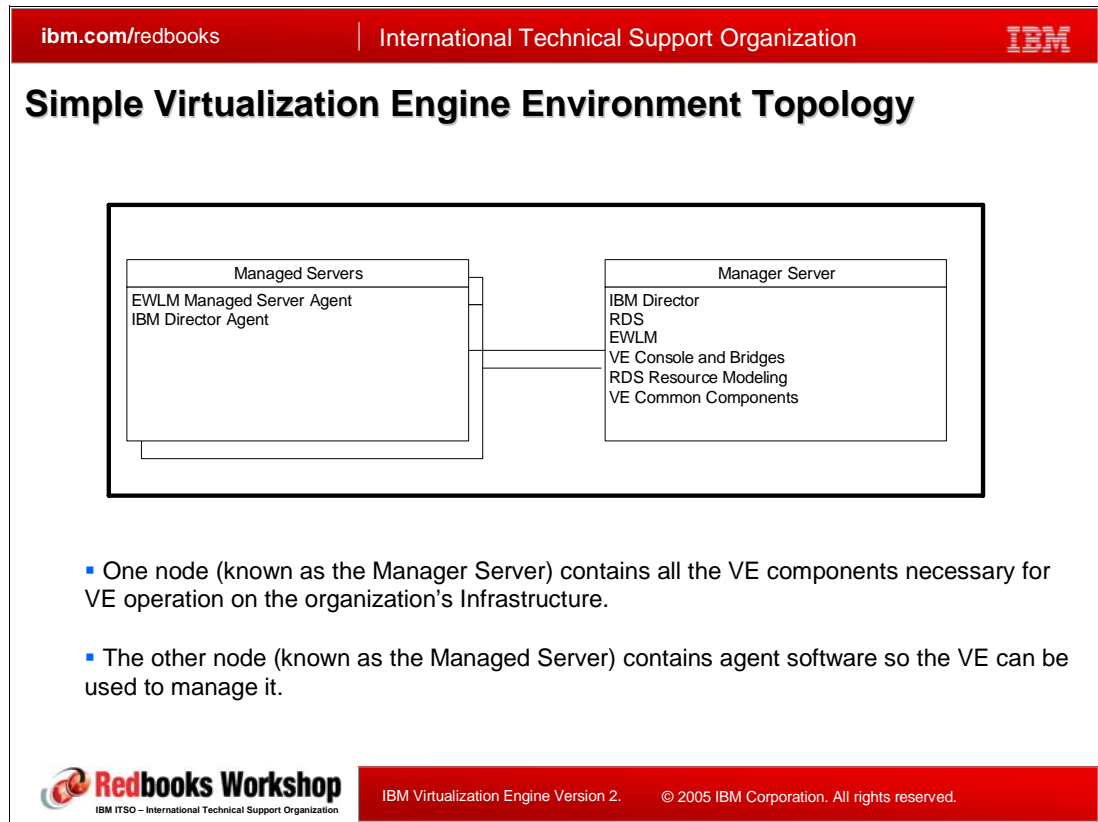


Figure 8-30 Simple Virtualization Engine Environment Topology

Figure 8-31 shows a topology where many of the Virtualization Engine components are deployed into separate nodes. This way the solution can be customized to satisfy more strict non-functional requirements (NFRs) such as high scalability, availability, and or performance. If necessary, practitioners can design each OM node as complex as it needs to be in order to satisfy the solution's NFRs. We will look into that a bit more.

All components of the Virtualization Engine, with the exception of the IBM Director, is built as a J2EE runtime and executes on top of WebSphere Application Server. Although IBM has not performed extensive testing on this yet, this means that (at least in theory) many complex design topologies supported by WebSphere Application Server are also supported by the WebSphere Application Server-based Virtualization Engine components. For instance, the use of IP-sprayers and WebSphere Application Server-managed load-balanced workloads can be exploited. Practitioners are encouraged to review other IBM publications for more information on WebSphere Application Server design topologies. The following URL provides complex design information using WebSphere Application Server and other Virtualization Engine components such as ITDS (to design the Directory Server node):

<http://www.redbooks.ibm.com/Redbooks.nsf/portals/WebSphere>

The following URL provides more information to design the Database Server node:

<http://www.redbooks.ibm.com/Redbooks.nsf/portals/Data>

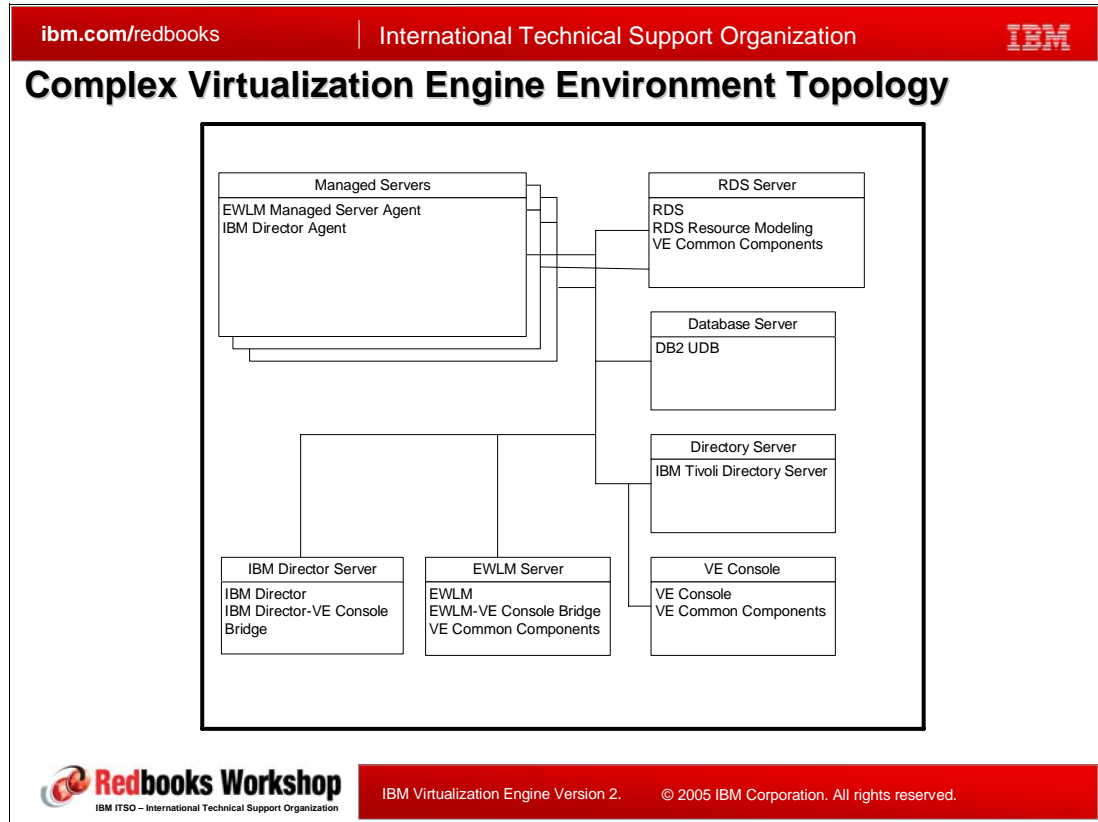


Figure 8-31 Complex Virtualization Engine Environment Topology

The only Virtualization Engine component that does not require any of the Virtualization Engine Common Components is the IBM Director, which becomes part of the Virtualization EngineE when the IBM Director to Virtualization Engine console Bridge gets installed on the same node where it executes. The Virtualization Engine Common Components are only installed on Manager Server nodes and not on Managed Servers (Endpoints).

The Virtualization Engine Common Components are:


- ▶ WebSphere Application Server – WebSphere Application Server 6.0.2 + Virtualization Engine iFixes
- ▶ DB2 Universal Data Base - DB2 UDB 8.2 SP1
- ▶ IBM Tivoli Directory Server – ITDS 6.0 SP1
- ▶ IBM Integrated Solutions Console – ISC 6.0.1
- ▶ Core elements
 - Java Runtime Environment – JRE 1.4.2 SP2
 - IBM Autonomic Deployment Engine framework
 - iLog Jviews
 - International Components for Unicode - ICU4J
 - IBM Common Logging module - ICL
 - Fix acquisition services
 - Deployment services

From Figure 8-31, we can see that WebSphere Application server will be deployed locally on all nodes that run EWLM Domain Manager, Resource Dependency Service, Virtualization Engine console, and the Virtualization Engine console Bridges. We can also see, for instance,


that Resource Dependency Service can use DB2 whether it is deployed on the same node, or it is deployed on another node by the Virtualization Engine installer, or when it is deployed remotely on any other node by the Virtualization Engine Installer, or when it is already installed on any other node accessible on the network.

The Virtualization Engine console component is built using IBM's Integrated System Console (ISC) which, in turn, runs within WebSphere Application Server (and the WebSphere Portal Server application). Therefore, we show that the ISC must be deployed locally to the same node that runs the Virtualization Engine console.

To further aid practitioners in selecting the proper type of platform to use for each node and deployment unit, we include Figure 8-32. It shows the Virtualization Engine components, how they are packaged (ovals), and which operating systems/platforms can be supported by them. Note that the ovals for the EWLM package and the IBM Director package are supported on all four platforms.

ibm.com/redbooks International Technical Support Organization 				
Virtualization Engine Component Installation Targets and Packaging				
	Windows	Linux	AIX	i5/OS
EWLM Domain Manager	X	X	X	X
Resource Dependency Service		X		
Virtualization Engine Console	X	X	X*	X
Console Bridges	X	X	X	X
IBM Director	X**	X**	X**	X

* IBM Director Bridge to VE Console equires a fix pack to be fully supported
 ** Installed outside of the VE deployment wizard

 IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-32 Virtualization Engine Component Installation Targets and Packaging

Practitioners should refer to the Virtualization Engine Planing Advisor for more planning and architectural guidance. This resource can be found at the following URL:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp>

8.6.2 Hardware sizing considerations

As discussed before, the hardware sizing information in this section reflects some performance characteristics observed of the IBM Virtualization Engine Suite Version 2, as of the writing of this book. This information is considered somewhat transient in nature and practitioners should always try to refer to the latest information available as updated by IBM in the future.

It is likely that performance results for similar or different Virtualization Engine workloads in other I/T infrastructure environments will differ from those discussed here. Additional and updated information may be available from references listed throughout this section.


Note: The minimum system sizing parameters shown in the following sections represent the smallest hardware configuration, which will make it possible for the software to operate. However, such configuration will not likely provide optimal performance, except in very small runtime requirement scenarios.

Important: The numbers provided in the tables in the following sections result of early tests. They will likely change depending on best practices, more information, and because of the product evolution. We recommend you get the latest information you can have at from the Information Center.

Virtualization Engine console sizing recommendations

Figure 8-33 shows the minimum hardware configurations recommended for the operation of the Virtualization Engine console.

ibm.com/redbooks		International Technical Support Organization		IBM
Minimum System Sizing – Virtualization Engine console				
	Processor Mini	Memory Mini	Disk space mini	Comment
AIX	1.4 GHz	1.75 GB	3 GB	add 2 GB in the /tmp directory
Linux on xSeries, System 5 and iSeries	1.4 GHz	1.75 GB	3 GB	add 2 GB in the /tmp directory
Linux on zSeries	zSeries 890 2086-210	2GB	3.5 GB	add 2 GB in the /tmp directory
Windows	1.4 GHz 1-way	1.75 GB	3 GB	add 2 GB in the temporary directory
I5/OS	1000+ CPW	3 GB	2 GB	



IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-33 Minimum System Sizing - Virtualization Engine console

It should be noted that the Virtualization Engine console will execute well in logical partitions (LPARs) with fractional parts of a processor assigned to them.


As discussed previously, the Virtualization Engine console implementation is a custom instance of the IBM Integrated Solutions Console (ISC), which uses WebSphere's Portal Server (WPS) as its foundation. Therefore, performance optimization for WPS will, in many cases, apply to the Virtualization Engine console environment.

One of the functions of Virtualization Engine console is to monitor and display continuous updates from the IBM Director monitors for resource utilization. When the Virtualization Engine console is used in this way, greater heap (memory) and processing requirements will typically apply. In fact, if these Virtualization Engine console refresh functions are used often, and by multiple users, it is very likely that the memory and processor requirements will grow beyond those listed in Figure 8-33.

EWLM Domain Manager sizing recommendations

Figure 8-34 shows the minimum hardware configurations recommended for the operation of the EWLM Virtualization Engine component.

ibm.com/redbooks		International Technical Support Organization		IBM
Minimum System Sizing – EWLM Domain Manager				
	Processor Mini	Memory mini	Disk space mini	Comment
Windows on xSeries and iSeries	2 GHz	0.5 GB	6 GB	WebSphere heapsize: 256 MB
AIX	1.4 GHz	0.5 GB	6 GB	WebSphere heapsize: 256 MB
i5/OS	1000 CPW	1.5 GB	6 GB	WebSphere heapsize: 256 MB
Linux on pSeries, iSeries, OpenPower	1.4 GHz	0.5 GB	6 GB	WebSphere heapsize: 256 MB
Linux on xSeries	2 GHz	0.5 GB	6 GB	WebSphere heapsize: 256 MB
Linux on zSeries	zSeries 890 2086-210	0.5 GB	6 GB	WebSphere heapsize: 256 MB
z/OS	zSeries 890 2086-210	1,25 GB	6 GB	WebSphere heapsize: 512 MB


 IBM ITSO – International Technical Support Organization

 IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-34 Minimum System Sizing - EWLM Domain Manager

The information presented in Figure 8-34 was determined using Virtualization Engine version 1 measurements. However, we anticipate that Virtualization Engine Version 2 requirements will be similar for the EWLM Domain Manager. During the benchmarking, measurements were taken on larger configurations than those listed, with varied workloads on 60 to 500 managed servers. From these measurements, we were able to estimate the requirements for reasonable performance on a configuration of up to 50 managed servers in a transaction environment of moderate complexity. That is the configuration (50 managed servers) on which the data shown is based.

While z/OS support is provided for Virtualization Engine version2, no measurements have been taken yet to quantify this environment.

Figure 8-34 displays minimal configuration requirements for an EWLM Domain Manager node only. Section “EWLM performance” on page 320 provides some relevant benchmark information, which will help in sizing the overall EWLM topology. Basically, the EWLM

Managed Server software execution has been measured to add a 1-8% overhead on the managed server itself.

Virtualization Engine IBM Director sizing recommendations

Figure 8-35 shows the minimum hardware configurations recommended for the operation of the IBM Director Virtualization Engine component. The recommendations shown are based on measurements taken for Virtualization Engine Version 1, but recent IBM tests on Virtualization Engine Version 2 yield similar results.

ibm.com/redbooks
International Technical Support Organization

Minimum System Sizing – IBM Director Management Server:

- Estimates based on limited experiments using IBM Director in a VE Console environment in VE V1. Initial VE V2 indications show similar results.
- Similar requirements are anticipated for other operating environments

	Processor	Memory	Disk	Comment
Windows	750 MHz	512MB	316MB	recommend 768-1024MB with Console function
Linux	750 MHz	512MB	316MB	recommend 768-1024MB with Console function
i5/OS	750 CPW	512MB	500MB	separate Console function required

- Additional resources (perhaps double) may be required if using real-time resource monitor functions

Redbooks Workshop
IBM ITSO – International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 8-35 Minimum System Sizing - Virtualization Engine IBM Director

The minimal configurations shown on Figure 8-35 assume a relatively passive use of the IBM Director Virtualization Engine component to manage a moderate number of systems without substantial real-time resource monitoring. The IBM Director offers a large suite of functions and not all of the functions are needed in every installation. However, when many of these functions are exercised, the hardware resource requirements can be much larger (e.g. double, triple, or more) than those listed. Therefore, it is recommended that practitioners allow for some experimentation in a controlled non-production I/T environment using the desired IBM Director functions. Such experimentation will help in determining the final hardware configuration for optimal IBM Director execution.

The minimum system requirements for the IBM Director Agent functions have not been included in Figure 8-35 due to lack of deterministic benchmarking results at this point. However, a limited IBM test using the i5/OS operating system indicated that most IBM Director functions, including resource monitoring, consume small extra percentage (1-8%) of the hardware system's resources.

8.6.3 Performance benchmarks

This section discusses some preliminary results derived from testing performed by IBM on some components of the Virtualization Engine Version 2. Not all Virtualization Engine Version 2 components have been thoroughly tested yet. For instance, performance results are not known at this point about the Resource Dependency Service component. More performance-related information and guidance will be made available to practitioners as it becomes available in the future via the Information Center and other IBM publications. To access the Information Center information, please visit the following URL:

<http://publib.boulder.ibm.com/eserver/>

Virtualization Engine console performance

Figure 8-36 shows a chart that illustrates how the performance of the Virtualization Engine console component in Virtualization Engine Version 2 has improved over Version 1. The y axis shows the CPU consumption (in fractional percentages) and the x axis shows different frequently used Virtualization Engine console functions, which were triggered using programmatic scripts. In Virtualization Engine Version 2, as can be seen, there is a general reduction in CPU consumption per request for the suite of Virtualization Engine console functions tested. However, not all functions improved the same way individually when contrasted with their Virtualization Engine Version 1 counterparts.

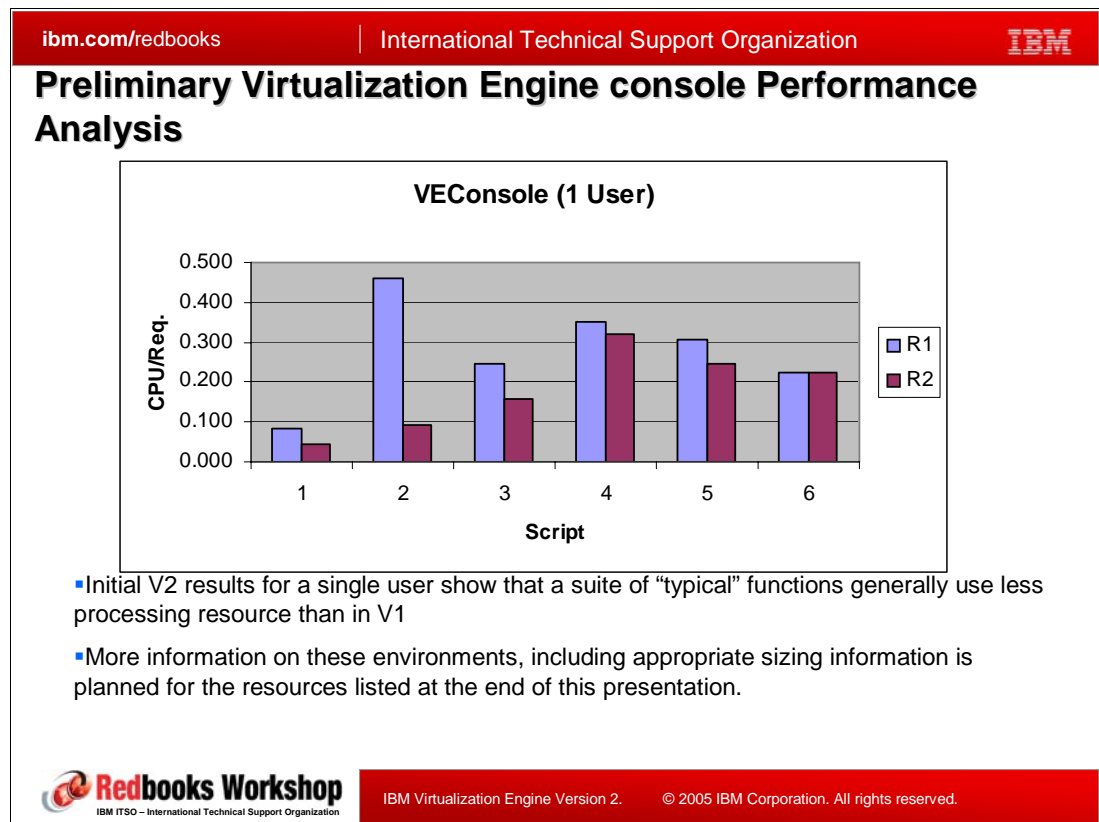


Figure 8-36 Preliminary Virtualization Engine console Performance Analysis

Figure 8-37 shows performance comparisons between the Virtualization Engine console component from Virtualization Engine Version 1 and from Version 2. The y axis shows CPU consumption (in fractional percentages) per user function request. The x axis shows the number of users triggering the requests. The chart shows that the performance improvement

shown by Virtualization Engine Version 2 remains relatively constant as we increased the number of supported users.

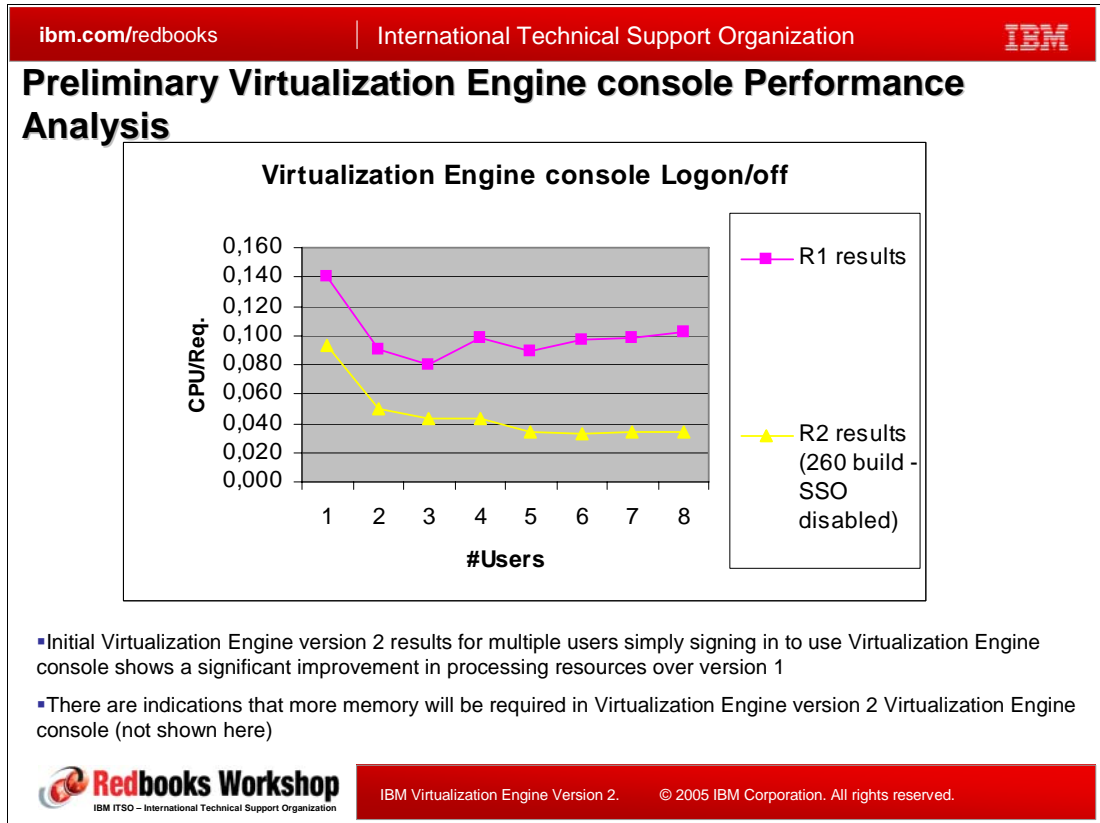


Figure 8-37 Preliminary Virtualization Engine console Performance Analysis

EWLM performance

As has been explained in other chapters of this book, EWLM has two primary functional components: a) the Domain Manager, which provides an aggregation of data from the Managed Servers, and b) the Control Center, which formats and displays the aggregated data to the user.

The Domain Manager component's execution is relatively passive in nature. Hardware resources are consumed by it in a predictable, steady manner. Therefore, its hardware resource consumption tends to be moderate and constant. On the other hand, the Control Center operation is more dynamic and this translates into more resource consumption peaks and valleys and, therefore, larger initial resource capacity requirements. In this section, we will use the term Domain Manager to refer to both, the Domain Manager and the Control Center functions, together.

Note: The performance information presented in this section was collected using the IBM Virtualization Engine Version 1. It is anticipated that the EWLM component in Virtualization Engine Version 2 will have slightly better performance characteristics, but these have not been quantified at this time.

Figure 8-38 shows some preliminary results from IBM EWLM testing using Microsoft Windows environments of varying transaction complexities. The vertical axis shows the CPU utilization percentage and the horizontal axis shows the number of servers managed plus the complexity of the transactions managed within them.

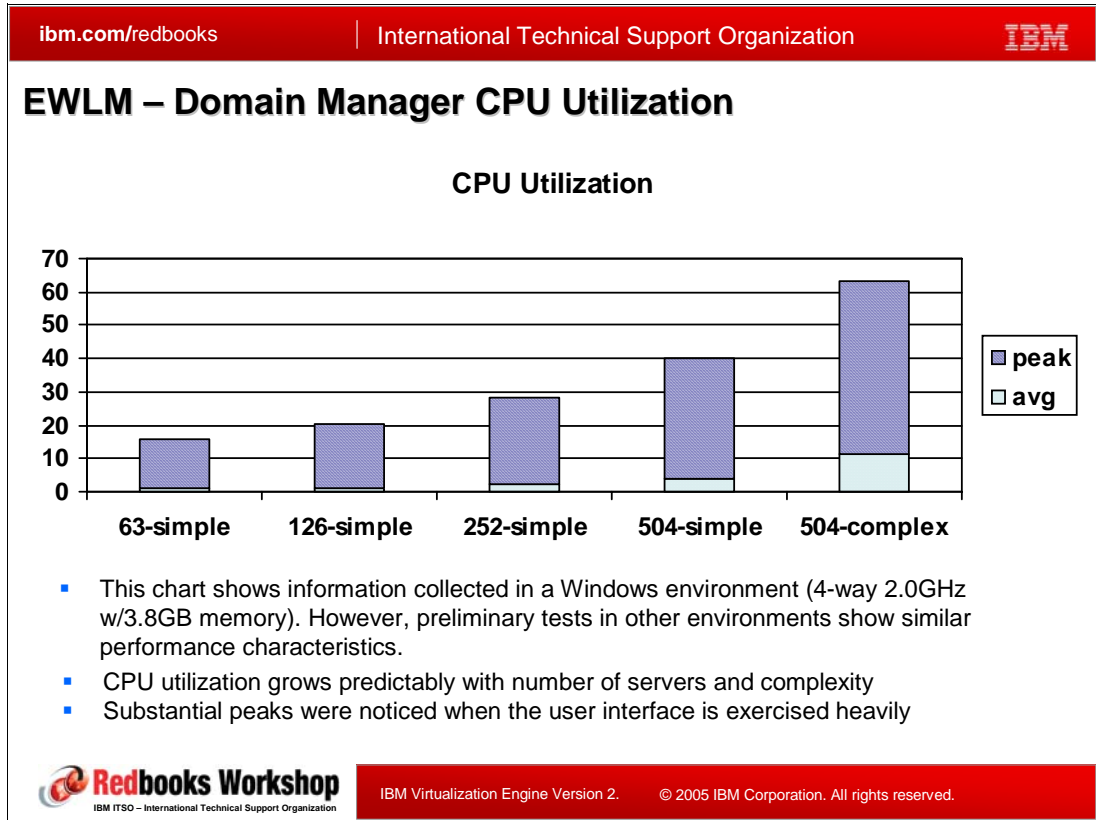


Figure 8-38 Virtualization Engine Version 1 EWLM - Domain Manager CPU Utilization

While the chart shown in Figure 8-38 was built using results from the Windows environment, many other tests using AIX, and i5/OS environments were also executed. These tests have yielded similar results. The environment workloads were simulated and managed with EWLM across 60 to 500+ managed servers using Virtualization Engine version 1 EWLM. The servers were part of a typical J2EE multi-tier topology and the simple transactions were kept within one tier, while the complex ones transcended to across multiple tiers. A variety of policy definitions for services, applications, and transactions were used in the test.

Figure 8-39 shows some preliminary results from IBM EWLM testing using Windows environments of varying transaction complexities. The vertical axis shows the RAM memory utilization in megabytes and the horizontal axis shows the number of servers managed plus the complexity of the transactions managed within them.

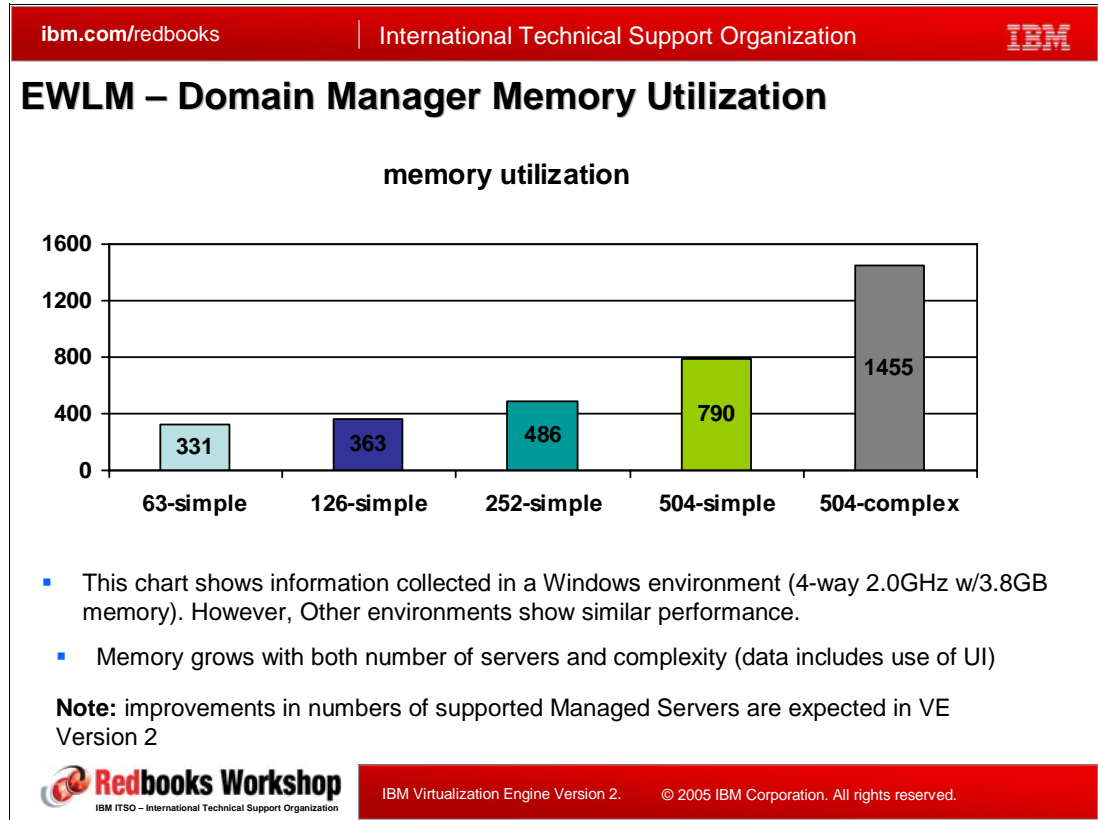


Figure 8-39 Virtualization Engine Version 1 EWLM - Domain Manager Memory Utilization

In general, the runtime execution performance of EWLM depends on many factors, which include the following:

- ▶ The number of Managed Servers

This is due to the fact that more Managed Servers simply mean more data to be analyzed and processed by the Domain Manager.
- ▶ The size of the active (not just the defined) service policies

The complexity of the service policy and the number of policy items for which data needs to be returned have a direct impact on the amount of overhead introduced by EWLM.
- ▶ The type of process classes, service classes, and transaction classes used

It is important to point out that the policy items (e.g. transaction classes, process classes, service classes, etc.) used is what matters, not the ones that are defined (but not used). As an example, defining 100 transaction classes and using 50 is likely to consume more Domain Manager resources than defining 200 and using 40.
- ▶ The transaction complexity

This refers to the number of “hops” across multiple tiers of the J2EE topology that are required to complete a given transaction. For instance, some transactions are relatively simple and are completed totally within a single server. Others may originate in a Web server, then require connections to a WebSphere Application Server container that, in turn, requires a connection to a DB2 server. In the latter case, there is more complexity to the transaction and this causes the EWLM Domain Server to consume more resources.
- ▶ The extent of use of the Control Center functionality

The use of Control Center functions place a noticeable demand on the EWLM Domain Manager system resources.

- ▶ The level of interaction between application software layers

Monitored applications that are complex tend to place more stress on the Managed Server resources. These relatively complex applications tend to have transactions that span across multiple topology tiers and tend to use multiple service calls between those tiers. For example, If an application forces its hosting application server to make 100 SQL queries to the servicing database server during a given transaction, the monitoring of that transaction is much more complex than if a single service call is placed to the database server and a Stored Procedure is used to service the call in a single action.

- ▶ Use of Application Response Measurement (ARM) standard

ARM is a standard created and managed by The Open Group for measuring application availability, application performance, application usage, and end-to-end transaction response times. Monitored applications issue ARM calls to report logic flow timings and help monitor the quality of the service delivered for each transaction. There is an increased strain on Manage Server resources as the frequency of the ARM calls increases.

Note: Some code enhancements have been made to IBM software offerings that are EWLM-enabled to improve EWLM performance on the Managed Servers. These include WebSphere Application server and DB2. It is important to try to be up to date on fix packages for, both, the host operating system and the managed IBM software products.

Additional EWLM performance information can be found in other IBM Redbooks by going to the following URL and searching for the word “EWLM”:

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

Virtualization Engine IBM Director performance

Extensive testing is currently being performed using the Virtualization Engine Version 2 Director component. These tests are examining the Director's capability to support a large number of diverse Managed Servers and its resource consumption while many of the available functions are exercised. This testing had not been completed at the time of this writing, but the results so far have yielded a few things we can report.

For the Virtualization Engine Director Management Server:

- ▶ There seems to be a limit of 5000 Managed Servers that can be managed by the Virtualization Engine Director for a given Management Server.
- ▶ We do not recommend that large numbers of Managed Servers (approaching the 5000 limit) be managed with Virtualization Engine Director using continuous monitoring. However, other short-duration functions of the Virtualization Engine Director can be less resource intensive and can be used on a very large number of Managed Servers without straining the Management Server resources.

For the Virtualization Engine Director Managed Servers:

- ▶ Preliminary testing results indicate that most Virtualization Engine Director functions create a relatively low overhead on the Managed Servers. The average of this overhead has been measure to be between 1% and 3% for things like alert management and resource monitoring.

- ▶ Some functions such as Software Update have been noted to require additional resource utilization beyond the 3% overhead average. However, this is comparable to resource requirements when using a manual process for that function.



Virtual Resources: Server Specific virtualization functions

Virtualization is not a new idea: many virtualization technologies have been already implemented at the server level; individually each of the IBM @server series include specific virtualization features. This is what is under the umbrella of the *Virtual Resources* layer in the Virtualization Engine platform Version 2. The IBM virtualization initiative tries to use these specific technologies across the servers, when it is appropriate. These specific technologies are very valuable in an homogeneous environment; the Virtualization Engine wants to extend these technologies to address heterogeneous environments.

This chapter describes some of these specific technologies, the ones that offer the most value and that be used when starting a project to virtualize the infrastructure. This chapter provides information about virtualization capabilities for each of the IBM @server series. Some of these virtualization capabilities can be implemented in several server types.

- ▶ IBM @server xSeries and BladeCenter virtualization capabilities are presented in the following sections:
 - Section 9.1, “For xSeries and BladeCenter: Capacity Manager and System Availability” on page 327.
 - Section 9.2, “For xSeries and BladeCenter: IBM Virtual Machine Manager” on page 329.
 - Section 9.12, “xSeries and BladeCenter summary” on page 353.
- ▶ IBM @server iSeries virtualization capabilities are presented in the following sections:
 - Section 9.3, “For iSeries, pSeries and zSeries: Dynamic Logical Partitioning” on page 331.
 - Section 9.5, “For iSeries and pSeries: Micro Partitioning” on page 336.
 - Section 9.6, “For iSeries and pSeries: Virtual I/O, Virtual LAN, Shared Ethernet” on page 338.
 - Section 9.7, “For iSeries, pSeries and zSeries: Capacity on Demand” on page 340.
 - Section 9.8, “For iSeries and pSeries: Simultaneous Multithreading” on page 344.
 - Section 9.9, “For iSeries: Multiple OS support” on page 346.

- Section 9.15, “iSeries summary” on page 358.
- ▶ IBM @server pSeries and System p5 virtualization capabilities are presented in the following sections:
 - Section 9.3, “For iSeries, pSeries and zSeries: Dynamic Logical Partitioning” on page 331.
 - Section 9.4, “For pSeries: Partition Load Manager” on page 333.
 - Section 9.5, “For iSeries and pSeries: Micro Partitioning” on page 336.
 - Section 9.6, “For iSeries and pSeries: Virtual I/O, Virtual LAN, Shared Ethernet” on page 338.
 - Section 9.7, “For iSeries, pSeries and zSeries: Capacity on Demand” on page 340.
 - Section 9.8, “For iSeries and pSeries: Simultaneous Multithreading” on page 344.
 - Section 9.14, “System p5 and pSeries summary” on page 356.
- ▶ IBM @server zSeries and System z9 virtualization capabilities are presented in the following sections:
 - Section 9.7, “For iSeries, pSeries and zSeries: Capacity on Demand” on page 340.
 - Section 9.10, “For zSeries: Multiple operating systems support with z/VM” on page 349.
 - Section 9.11, “For zSeries: HiperSockets” on page 351.
 - Section 9.13, “System z9 and zSeries summary” on page 354.

9.1 For xSeries and BladeCenter: Capacity Manager and System Availability

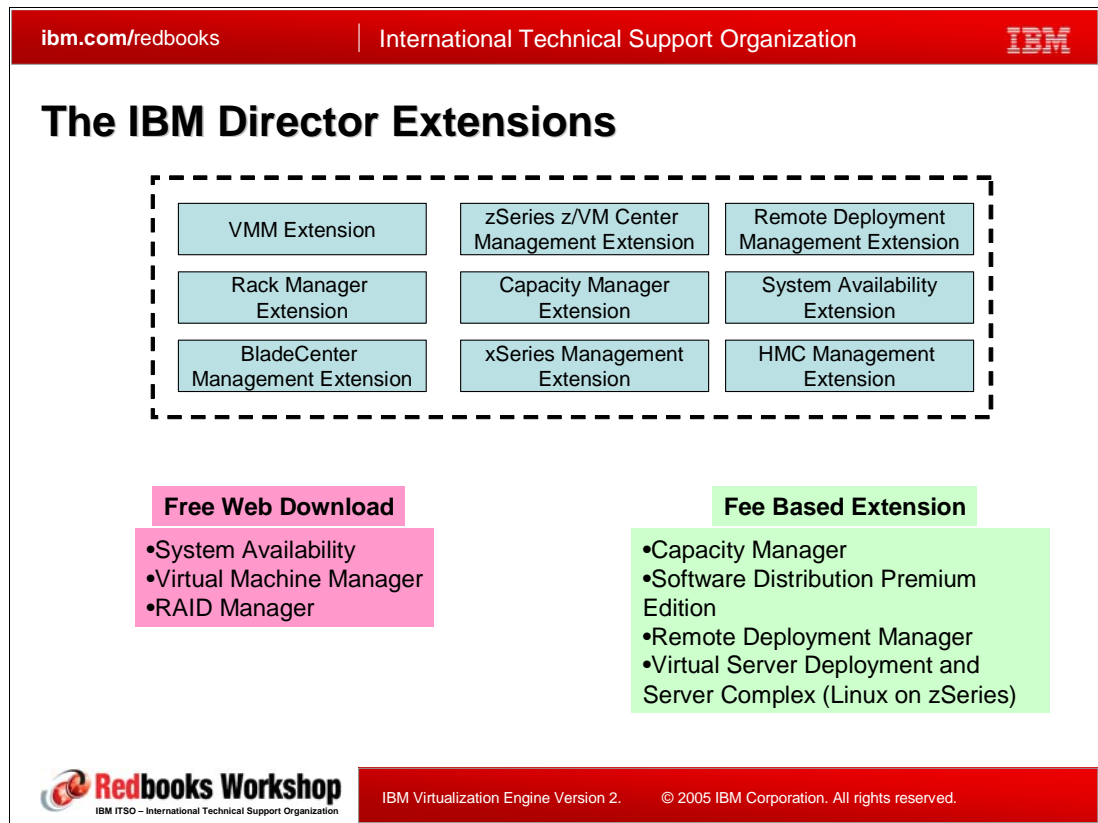


Figure 9-1 IBM Director extensions

IBM Director can be extended for greater management and optimization of xSeries and BladeCenter servers with optional tools that integrate into the IBM Director interface for consistent, single point of management. System Availability is a free-based feature of the IBM Director extensions, Capacity manager is a fee-based feature of the IBM Director extensions.

Capacity manager

Capacity manager tracks resource utilization, identifies multiple levels of existing or potential bottlenecks, makes recommendations to improve performance and generates capacity and performance reports in XML format.

Once capacity manager is installed on a managed system (running an IBM Director Agent), performance data is being collected in the background. Data is always being gathered without starting the logging of data; you simply specify what data you want retrieved from the servers and workstations in your network and it is gathered up and can be displayed graphically. Up to one month's worth of data is automatically saved by every system.

This feature enables you to automatically export performance reports on a regular basis, for example to an intranet server where the data is accessible for multiple users with a simple Web browser. These reports show potential capacity bottlenecks within the selected systems. Capacity Manager gives you the ability to plan the allocation of hardware upgrades for the systems that really need them before a capacity bottleneck occurs.

Using Capacity Manager allows you to manage systems proactively rather than in a reactive manner.

For more information, you can read *Tuning IBM eServer xSeries Servers for Performance*, SG24-5287.

System availability

System availability: tracks and offers a variety of graphical views of system downtime or uptime for an individual system or group of systems, saving labor costs associated with remotely managing and tracking server performance and availability. It can also schedule system availability reports and generate these reports in XML formats.....

9.2 For xSeries and BladeCenter: IBM Virtual Machine Manager

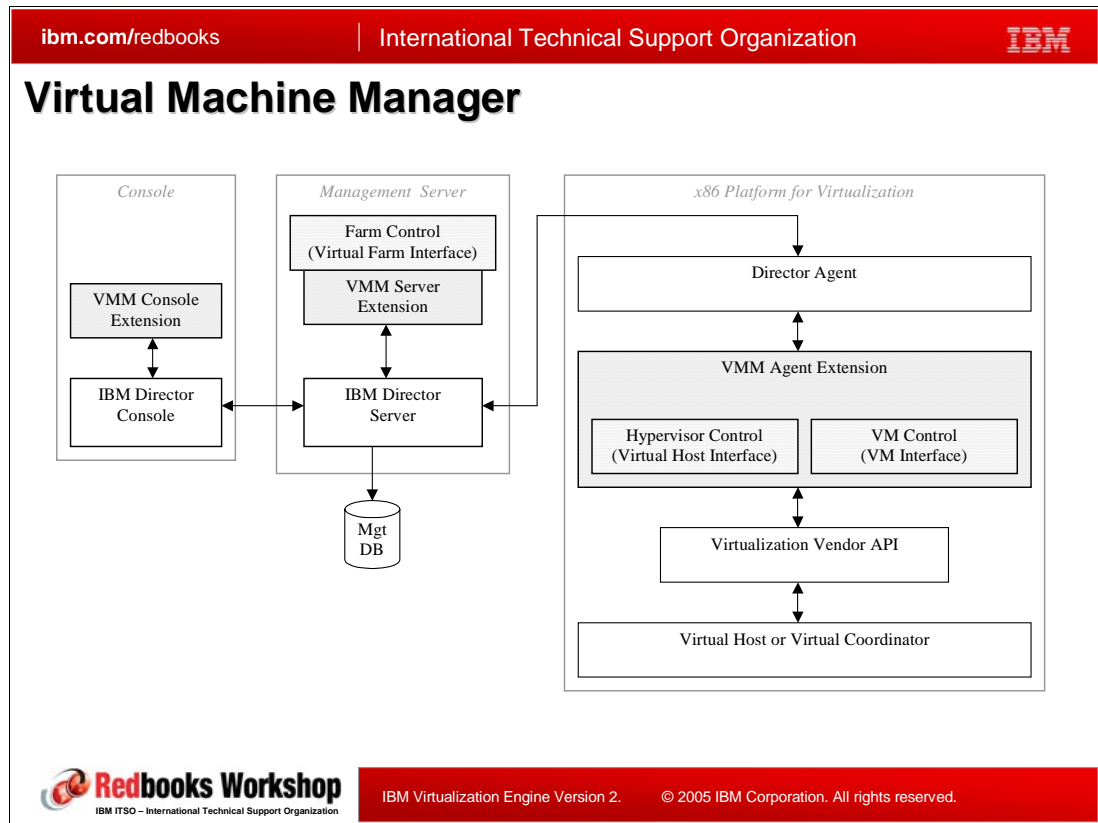


Figure 9-2 IBM Virtual Machine Manager

Virtual Machine Manager (VMM) is an IBM Director Extension that provides management of x86 platform virtualization. VMM supports management of hypervisors that operate on x86 hardware platforms.

VMM enables the use of the following virtualization applications in an IBM Director environment:

- ▶ VMware VirtualCenter
- ▶ VMware ESX Server
- ▶ VMware GSX Server
- ▶ Microsoft Virtual Server

When VMM and these virtualization applications are installed, you can perform the following tasks from IBM Director Console:

- ▶ Correlate relationships between physical platforms and virtualization components
- ▶ Report status of physical platforms and their corresponding virtualization components
- ▶ Log in to the management interface of the virtualization application
- ▶ Discover virtualization components
- ▶ Perform migration and power operations on virtual machines
- ▶ Create event action plans that involve virtualization components

When VMM is installed, IBM Director can recognize systems that contain virtualization components. VMM can create VMM objects to represent the virtualization components that were recognized. VMM uses the application programming interface (API) of the virtualization application to perform VMM operations on the virtualization component itself. All VMM

console operations, such as powering on a virtual machine, are run as background processes. Therefore, IBM Director Console can be used for other operations while VMM performs its operations.

The hardware in a VMM environment is referred to in the following ways:

- ▶ A *management server* is a server on which both IBM Director Server and VMM Server are installed.
- ▶ A *management console* is a system on which both IBM Director Console and VMM Console are installed. For example, in a VMware VirtualCenter environment, this system must also be installed with VMware VirtualCenter client, which provides a graphical user interface (GUI) to an instance of VMware VirtualCenter server.
- ▶ A *managed system* is a system on which IBM Director Agent and a virtualization application are installed.
- ▶ A *system* is a server, workstation, desktop computer, or mobile computer.

The software and its components in a VMM environment are referred to in the following ways:

- ▶ A *virtualization environment* is a managed system and its associated VMM objects that are using the following sets of software: VMware VirtualCenter server and VMM Agent for VirtualCenter (supported hosts are ESX and GSX hosts), or VMware ESX Server and VMM Agent for ESX, or VMware GSX Server and VMM Agent for GSX or Microsoft Virtual Server and VMM Agent for Virtual Serve
- ▶ A *virtualization component* is a software element that is created by a virtualization application.
- ▶ A *VMM object* is an IBM Director managed object that represents the virtualization components in a supported virtualization environment. VMM objects include coordinators, VMM farms, hosts, virtual machines, and guest operating systems.
 - A *coordinator* is a managed object that represents a system on which all of the following software is running: VMware VirtualCenter server, or VMware VirtualCenter Web service, or IBM Director Agent or VMM Agent for VirtualCenter
 - A *VMM farm* is a managed object that represents a collection of hosts and their associated virtual machines. VMM farm objects can represent farms that are defined in VMware VirtualCenter. VMM farm objects can also be a collection of hosts in other supported virtualization environments.
 - A *host* is a managed object that represents a system on which one of the following combinations of software are running: – VMware ESX Server and IBM Director Agent (VMware VirtualCenter only) – VMware GSX Server and IBM Director Agent (VMware VirtualCenter only) – VMware ESX Server, IBM Director Agent, and VMM agent for ESX – VMware GSX Server, IBM Director Agent, and VMM agent for GSX – Microsoft Virtual Server, IBM Director Agent, and VMM Agent for Virtual Serve
 - A *virtual machine* is a managed object that represents a virtual machine that is associated with a supported virtualization application.
 - A *guest operating system* is a managed object that represents an operating system that is running in a virtual machine and on which IBM Director Agent is installed.

9.3 For iSeries, pSeries and zSeries: Dynamic Logical Partitioning

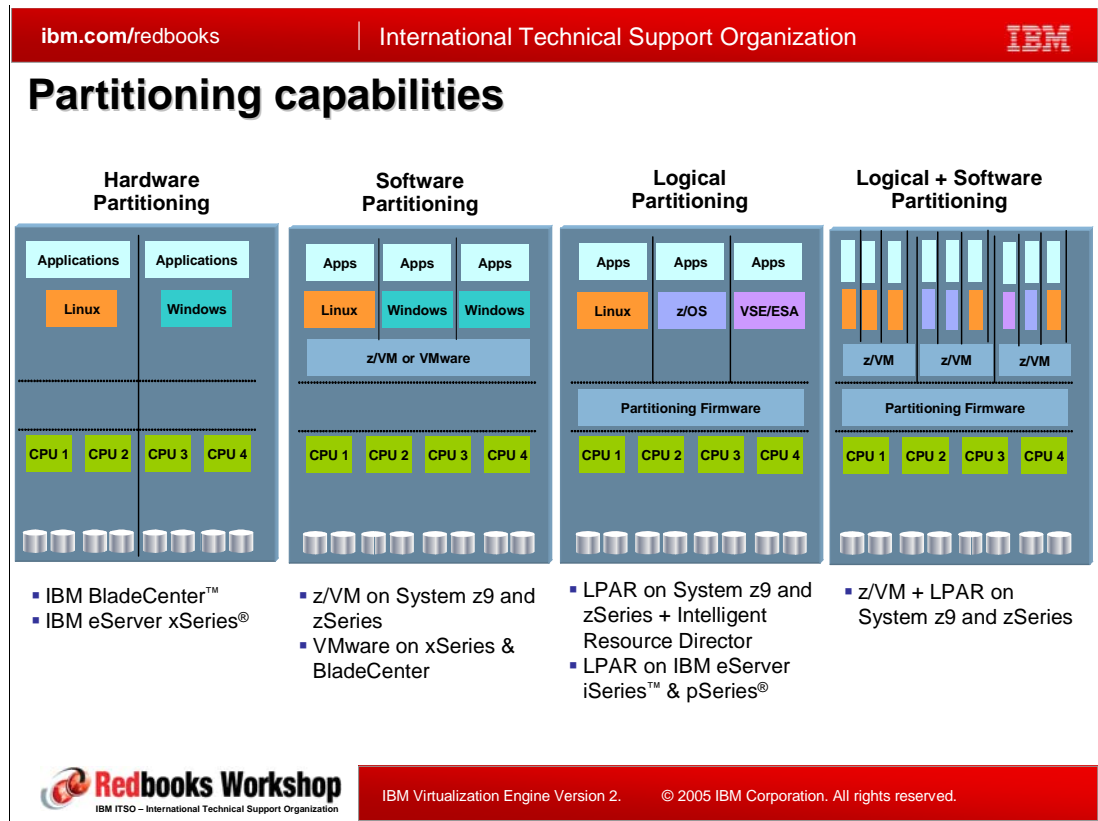


Figure 9-3 Partitioning definitions

Logical partitioning

A logical partition (LPAR) is the division of a computer's processors, memory, and hardware resources into multiple environments so that each environment can be operated independently, with its own operating system and applications. Logical partitioned servers are transparent to applications.

Dynamic logical partitioning

Dynamic Logical Partitioning (DLPAR) extends the capability of LPAR by providing the ability to logically attach and detach a managed system's resources to and from a logical hardware partition's operating system without rebooting that operating system. This resource allocation can occur not only when activating a logical partition, but also while the partition is running. Processor, memory, and I/O adapter resources can be moved directly from one partition to another, or released into a "free pool" of resource. A resource (processor, memory or I/O adapter) can be assigned to one hardware partition at one time. The movement of an LPAR resource from one hardware partition to another is managed through the HMC.

pSeries

DLPAR requires AIX 5.2 or higher; October 2002 system firmware or later; and HMC release 3 version 1 or higher. Using AIX 5.3:

- ▶ For dedicated and shared processor partitions it is possible to dynamically add, move, or remove memory in a granularity of 16 MB regions, add, move, or remove physical I/O adapter slots, and add or remove virtual I/O adapter slots
- ▶ For a dedicated processor partition it is only possible to dynamically add, move, or remove whole processors.
- ▶ For shared processor partitions it is also possible to dynamically remove, move, or add entitled shared processor capacity, change between capped and uncapped processing, change the weight of an uncapped partition and add and remove virtual processors

Alphaworks provide a dynamic LPAR Tool Set for pSeries, which provides sample scripts for monitoring LPARs and automating DLPAR operations, which customers can use and/or customize to fit their installation's needs. To know more about the tool, please refer to the following Web site:

<http://www.alphaworks.ibm.com/tech/dlpar>

iSeries

The iSeries LPAR capabilities have been improved with every new release since V4R4. Today, the POWER Hypervisor allows for multiple operating systems to run on the POWER5 based hardware: i5/OS, Linux, Windows (via Integrated xSeries Server or Integrated xSeries Adapter) and AIX 5L V5.3 and v5.2 are supported in logical partitions. Up to 254 partitions, depending on the model and features, can be defined concurrently on an IBM @server i5 servers.

The LPAR Validation Tool (LVT) can assist the user in the design of an LPAR system, and provides an LPAR validation report that reflects the user's system requirements, while not exceeding LPAR recommendations. The tool is available at the following Web site:

<http://www-03.ibm.com/servers/eserver/iseries/lpar/systemdesign.html>

zSeries

LPAR has been available on the zSeries systems since the 1980's. The latest available zSeries, the z9-109 server, can have up to 60 LPARs defined and, by the way, the basic mode (without LPARs) no longer exists on this system.

Exclusive to the IBM z/Architecture is Intelligent Resource Director (IRD), a function that optimizes processor and channel resource utilization across logical partitions based on workload priorities. IRD uses the concept of an LPAR cluster, the subset of z/OS systems in a Parallel Sysplex cluster that are running as logical partitions on the same z900 server. IRD provides the LPAR CPU Management function

This function allows the WLM for z/OS to dynamically adjust the number of logical processors within a logical partition and the processor weight, based on the WLM policy. The ability to move the CPU weights across a logical partition cluster provides processing power where it is most needed, based on WLM for z/OS goal mode policy.

9.4 For pSeries: Partition Load Manager

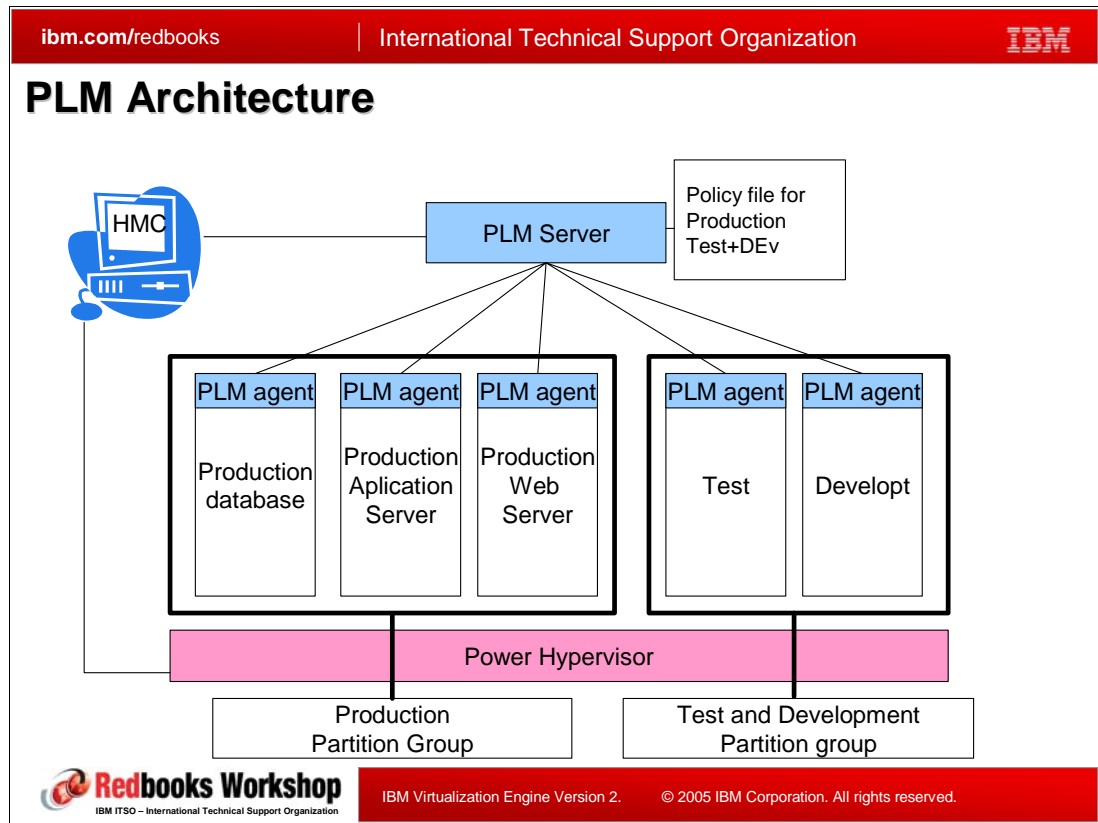


Figure 9-4 Partition Load Manager Architecture (get picture from Scott Vetter)

Owners of IBM System p5 servers using AIX 5L v5.3 with the Advanced POWER Virtualization feature have now the choice of three different workload management mechanisms:

- ▶ Shared processor, uncapped partitions. When using shared processor partitions, the POWER Hypervisor manages a pool of processors shared amongst a number of partitions. The POWER Hypervisor allocates unused processor cycles to those uncapped partitions that can make use of them.
- ▶ Workload Manager (WLM). WLM prioritizes applications' access to system resources - CPU, memory, and I/O within a partition. WLM monitors and manages the use of CPUs, memory and disk I/O rates within a standalone AIX 5L partition
- ▶ Partition Load Manager which is discussed in more details in the following.

Partition Load Manager (PLM) is a feature of the Advanced POWER Virtualization feature. PLM for AIX 5L is designed to automate the administration of memory and CPU resources across logical partitions within a single central electronics complex (CEC). To improve system resource usage, PLM automates the migration of resources between partitions based on partition load and priorities: partitions with a high demand will receive resources taken from partitions with a lower demand. A user-defined policy governs how resources are moved. PLM will not contradict the partition definitions in the HMC. PLM allows administrators to monitor resource usage across all the managed partitions. It is supported on both dedicated and shared-processor partitions running AIX 5.2 (ML4) or AIX 5.3.0 or later on IBM Systems p5 servers.

PLM is using the following definitions:

▶ PLM Agents

PLM uses agents on each of the managed partitions to monitor and manage partition resources. A PLM server configures each of the agents, setting the thresholds at which the server should be notified. The agents monitor the partition's resource usage and notify the PLM server whenever PLM-set thresholds are passed (under or over-utilized). Based on a user-defined resource management policy, the PLM server invokes dynamic reconfiguration operations through the HMC to move resources from a spare-pool to a partition or between partitions.

▶ PLM server

The PLM server may reside either in a partition on the same server as the partitions being managed or on a different machine. Multiple Partition Load Manager servers may be run on one AIX 5L system. One Partition Load Manager server can manage partitions within only one managed CEC.

▶ PLM Group

Resources are constrained into a group: A partition in one PLM group will never be given resources from another partition in another group.

▶ PLM Policy

The resource management policy for each partition group is specified in a policy file that defines both the managed environment and the parameters of the resource management policy. The policy file is static that means a partition's priority does not change upon the arrival of an higher priority work. PLM has no knowledge of the importance of any workload running in the partitions and therefore cannot readjust priority based on the changes of workload types. The priority of partitions can only be changed by loading a new policy. Policy files can be changed on-the-fly, without stopping PLM. Part of a policy definition is the relative priority of each of the partitions in a group.

PLM manages memory by moving Logical Memory Blocks (LMBs) across partitions. The size of the LMB depends on the amount of memory installed in the CEC. It varies between 16 and 256 MB. The size of the LMB can be modified with the Advanced System Management Interface (ASMI) on the HMC.

For dedicated processor partitions, PLM moves physical processors, one at a time, from partitions that are not utilizing them or that have a higher excess weight, to partitions that have demand for them. For shared processor partitions, PLM manages the entitled capacity and the number of virtual processors (VPs).

Resource Monitoring and Control (RMC) is the communications and event framework used by PLM.

PLM is only supported on AIX 5L; it is not supported on i5/OS or on Linux.

PLM has a relatively high latency - in the order of minutes. This latency makes PLM appropriate for medium and long-term changes in resource usage and ineffective for managing short-lived peaks. PLM cannot manage workload peaks that are of short duration. PLM manages the medium and long-term trends: it can handle the necessary migration of resources as operations move from the daily transactions to the overnight batch and back again. PLM can be used to optimally configure servers with stable workloads. By setting the initial partition configuration with minimal resources, leaving unassigned resources in the free pool (memory and processor), PLM will move just enough resources¹ in to each partition to satisfy the workload. This alleviates the need to perform any precise estimation regarding the distribution of the server hardware between partitions.

For more information, please refer to *Advanced POWER Virtualization on IBM System p5*, SG24-7940 and *Introduction to pSeries Provisioning*, SG24-6389.

9.5 For iSeries and pSeries: Micro Partitioning

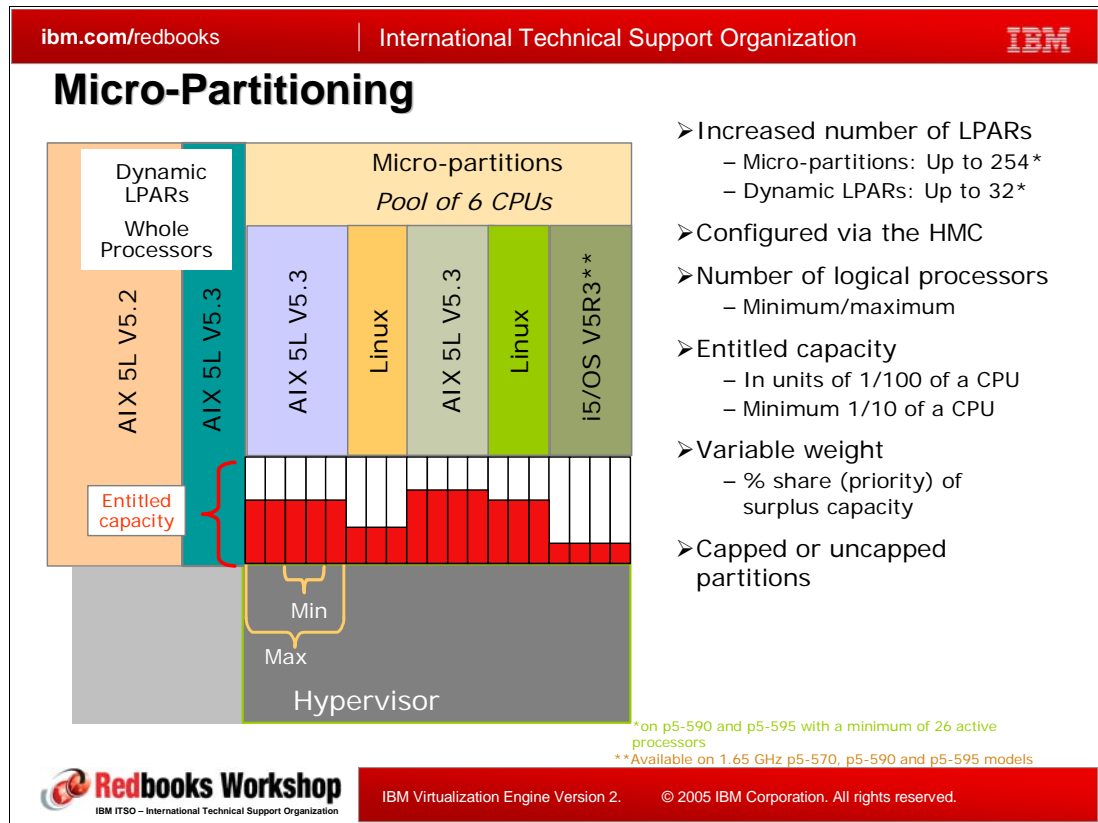


Figure 9-5 Micropartitioning

Micro-Partitioning is the ability to divide a physical processor's computing power into fractions of a processing unit and share them among multiple logical partitions.

On POWER4 systems, all partitions were considered dedicated. The processors that were assigned to a partition could only be in whole multiples and only be used by that partition. On POWER5 systems, you can choose between dedicated processor partitions and shared processor partitions using Micro-Partitioning technology. This is available for POWER5 systems with AIX 5L Version 5.3 or Linux (SUSE LINUX Enterprise Server 9 for POWER systems and Red Hat Enterprise Linux AS 3 for POWER)

One processing unit represents 1% of one physical processor. A maximum of 10 partitions for each physical processor may be defined.

Micro-Partitions can be defined with capped and uncapped attributes:

- ▶ A capped Micro-Partition is not allowed to exceed the defined capacity.
- ▶ An uncapped partitions can be configured to the total idle capacity of the server or to a percentage of the total idle capacity.

It is an option for which you must obtain and enter an activation code for most IBM System and IBM Sserver models except for the p5-590 and p5-595 models.

The benefit of Micro-Partitioning is that it allows increased overall utilization of CPU resources within the managed system. Better granularity of CPU allocation in a logical partition means efficient use of processing power.

For more information, you can read *Partitioning Implementations for IBM eServer p5 Servers*, SG24-7039.

9.6 For iSeries and pSeries: Virtual I/O, Virtual LAN, Shared Ethernet

ibm.com/redbooks
International Technical Support Organization

Virtual I/O Server

- Single function specialized appliance that facilitates the sharing of physical resources amongst LPARs within a CEC
 - virtual SCSI
 - SEA (Shared Ethernet Adapter)
- POWER5 platform only
- pSeries, iSeries, OpenPower
- VIOS bundled along with micropartitioning and PLM in the APV(Advanced POWER Virtualization) platform feature
- Supported client operating systems: AIX 5.3L and pLinux

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 9-6 Virtual I/O Server Ethernet example

Virtual I/O and Virtual I/O Server

VIO provides I/O adapters like virtual ethernet and SCSI, to client LPARs. These connections can then be routed through a physical adapter by a VIO server. VIO connections are implemented in firmware; for client LPARs, virtual I/O adapters are defined in the LPAR definition. The advanced POWER virtualization feature is needed to use VIO; VIO is only available on POWER5

The IBM Virtual I/O Server (VIOS) is a special POWER5 partition that provides the ability to implement the virtual I/O function and is not designed to run applications. It is controlled through a restricted, scriptable command line use interface. The VIOS enables partitions to share I/O resources (such as Ethernet adapters, SCSI disks, or Fibre Channel disks). Dedicated I/O resources are assigned to the VIOS, which allocates and manages I/O devices across multiple partitions. The shared resource appears as a virtual I/O device or a virtual adapter to the partition. The client partition does not communicate directly with the underlying physical I/O adapters: when the client partition uses its virtual I/O device, it is sending operation requests to the VIOS. The POWER Hypervisor provides a secure communication channel between the client partition and the VIOS, which transports the operation requests and responses between the client partition and the VIOS.

Shared Ethernet

The Shared Ethernet Adapter allows partitions to communicate outside the system without having to dedicate a physical I/O slot and a physical network adapter to a client partition.

Shared Ethernet Adapter requires the POWER Hypervisor and Advanced POWER Virtualization features of POWER5 systems; the device drivers for virtual Ethernet are available for AIX 5L Version 5.3 and Linux. A Shared Ethernet Adapter is a service to securely transport network traffic from a virtual Ethernet to a real network adapter. The Shared Ethernet Adapter service runs in the Virtual I/O Server. It cannot be run in a general purpose AIX 5L partition.

9.7 For iSeries, pSeries and zSeries: Capacity on Demand

ibm.com/redbooks
International Technical Support Organization
IBM

About Capacity On Demand

Capacity Upgrade on Demand

- Upgrade system with processors and/or memory
- No special contracts, no required monitoring (no ability to turn off the capacity)
- Purchase agreement

On/Off Capacity on Demand


- Temporary use of requested number of processors or amount of memory
- Client selects the capacity and activates the resource (registered system)
- Capacity can be turned on and off by the client
- Information captured by IBM (or reported to IBM)
- Rental agreement

Reserve Capacity on Demand

- Processor resources only (processor days)
- Capacity can be turned on and off by the client
- Prepaid debit agreement
- Requires AIX 5L V5.3

Trial Capacity on Demand

- Allow clients to test the effects of additional processors and/or memory
- Partial or total activation of processors and memory
- Resources available for fixed time
- No formal commitment required



Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 9-7

Under Capacity On Demand, IBM offers different types of Capacity on Demand capabilities for the IBM(and IBM TotalStorage) product families:

- ▶ Permanent capacity for non-disruptive growth, provides a means of planned growth for customers who know they will need increased capacity but aren't sure when.
- ▶ Temporary capacity for fluctuating workloads provides for planned and unplanned short-term growth driven by temporary processing requirements such as seasonal activity, period-end requirements or special promotions.
- ▶ Temporary capacity for workload peaks and testing, provides the flexibility to evaluate how additional resources will affect existing workloads, or to test new applications.
- ▶ Interim capacity for continued operation, temporarily supplements backup servers with the extra processing capacity they need in case of an unforeseen production server outage.

You can find detailed information on the following Web site:

<http://www-03.ibm.com/servers/eserver/about/cod/about/types.html>

Notes for pSeries

Capacity on Demand (CoD) for eServer p5 (550, 570, 590 and 595) and pSeries (650, 670 and 690) servers with dynamic logical partitioning offers clients the ability to non-disruptively activate (no boot required) processors and memory when implementing dynamic LPAR. Capacity on Demand also gives eServer p5 and pSeries owners the option to temporarily activate processors to meet intermittent performance needs, to activate additional capacity on a trial basis and to access capacity to support operations in times of crisis.

The Capacity on Demand process is also used to activate the Advanced POWER Virtualization feature on OpenPower 710, OpenPower 720, p5-510, p5-520, p5-550, p5-570 and p5-575.

Types of Capacity on Demand for IBM @server p5 and pSeries servers include:

- ▶ Capacity Upgrade on Demand for permanent capacity for non-disruptive growth.

Capacity Upgrade on Demand (CUoD) for eServer p5 and pSeries servers allows companies to install inactive CUoD processors and memory at an extremely attractive price and then bring new capacity online quickly and easily. With AIX 5L Version 5.3, Version 5.2 and Linux operating systems that support dynamic LPAR, processors and memory can be activated dynamically without interrupting system or partition operations.

The p5-550 (1.65GHz), p5-570 (1.65/1.90GHz), p5-590, and p5-595 servers offer increased activation granularity for both processors and memory (p5-570, p5-590, and p5-595). Processors can be activated in increments of 1 processor while memory can be activated in increments of 1GB (1.50GHz p5-570 support Memory CUoD). As your workload demands require more processing power, you can activate inactive processors simply by placing an order for an activation feature code. You receive over the Internet an electronically encrypted activation key that unlocks the desired amount of processors. There is no hardware to ship and install, and no additional contract is required.

Capacity Upgrade on Demand processor options for p670 and p690 servers are available in units of four active and four inactive processors with up to 50% of the system processors and memory inactive. With Capacity Upgrade on Demand for the p650 server, processors are available in pairs with a maximum of six inactive processors. To activate processors and memory on some models of p650, p670 and p690, clients may have to send current system configuration data to IBM. Memory activation works the same way. Capacity Upgrade on Demand memory is available in various sizes for the p650, p670 and p690 systems. You can activate memory in 4GB increments by ordering an activation feature code to activate the desired amount of memory.

- ▶ On/Off Capacity on Demand for temporary capacity for fluctuating workloads.

On/Off CoD provides an innovative and flexible processor and memory activation capability for planned or predictable peak processing requirements for p5-550 (1.65GHz), p5-570 (1.65/1.90GHz), p5-590, and p5-595 servers. When you order an On/Off CoD feature for p5-550, p5-570, p5-590, and p5-595 servers, you receive an enablement key that enables the administrator to turn processors and memory on and off as desired, in increments of 1 processor and 1GB of memory. The system monitors the amount and duration of the activations and generates a usage report which must be sent to IBM monthly. Billing for the activations is then based on the usage report on a quarterly basis. You pay only for the processor days and memory days after you have used them.

When you order an On/Off Capacity on Demand feature, for p650, p670, and p690 servers, you receive a prepaid 60 processor day allocation and an activation key that enables you to activate two processors for 30 days of use. You can turn processor pairs on and off whenever you need to. Deductions are made against the Processor Day allocation only when processors are activated. Increments of usage are measured in Processor Days, and the minimum usage is one day per activated processor.

- ▶ Reserve Capacity on Demand for temporary capacity for fluctuating workloads.

Reserve CoD allows p5-550 (1.65GHz), p5-570 (1.65/1.90GHz), p5-590 and p5-595 servers to have optimized, automatically managed temporary activation of CoD processors. The user purchases a block of 30 Processor Days of usage time and then assigns inactive processors to the shared processor pool. When these inactive processors are activated, all partitions in the shared processor pool will start executing on all the processors in the shared processor pool, increasing application throughput. The server then automatically manages the workload and only charges against the Processor Day

account when the workload exceeds 100% of the base (permanently activated) processors in the shared processing pool by 1/10 or a processor unit.

- ▶ Trial Capacity on Demand for temporary capacity for workload peaks and testing.

Users of IBM @server p5 and pSeries systems with Capacity Upgrade on Demand capabilities can activate all available inactive CUoD processors and memory resources one time, for up to 30 contiguous days, at no charge. Trial Capacity on Demand for pSeries p5-550 (1.65GHz), p5-570 (1.50GHz only for memory), p5-590, and p5-595 servers is enabled by registering at an IBM Web site and electronically receiving an activation key.

- ▶ Capacity BackUp for interim capacity for continued operation.

Capacity BackUp (CBU) is available for IBM @server pSeries p5-590, p5-595, p670 and p690 to provide a production system backup capability. CBU systems are configured and shipped with a full compliment of processors installed with 4 active processors and the remaining processors inactive CoD processors. With the p5-590 and p5-595 systems CoD memory may also be purchased and activated when needed. Clients may temporarily activate any number of processors or amount of memory on p5-590 and p5-595 CBU systems in one processor and 1GB memory increments. On p670 and p690 CBU systems, only processors may be temporarily activated in increments of 2 processors.

Capacity BackUp systems can be temporarily turned on at any time by using the On/Off CoD activation procedure for the needed performance during an unplanned system outage. Each Capacity BackUp configuration includes an allocation of On/Off CoD processor days which allow temporary activation of processors (p5-590 CBU includes 900 processor days, p5-595 32-way CBU includes 900 processor days, p5-595 64-way CBU includes 1,800 processor days, p670 CBU includes 240 processor days, p690 CBU includes 720 processor days) until the allocation is depleted. Capacity BackUp is intended for companies requiring a disaster recovery machine at an extremely affordable price.

Inactive processors cannot be permanently activated. Inactive processors can be used for production role swapping during an unscheduled outage, tape backup, failover testing and role swapping during upgrades or PTF installations.

With IBM HACMP V5 and HACMP/XD software (5765-F62) installed, Capacity BackUp processors can be automatically activated to provide round-the-clock business continuity and disaster recovery with no loss of data. HACMP can be configured to recognize a failing server and activate Capacity BackUp resources upon failover. When needed, HACMP can also activate dynamic LPAR and CoD resources. The combination of Capacity BackUp servers and HACMP software gives clients the optional capabilities they need to keep business critical applications up and running on a 24 x 7 basis without manual intervention. For more information on how to use HACMP with CBU for automated backup capability, see "Administration and Troubleshooting Guide Version 5.2, SC23-4862-04, Appendix H: Using DLPAR and CUoD in an HACMP Cluster" .

Capacity BackUp offerings are not intended to be used as full-time backup servers for 24x7 high-availability solutions.

- ▶ Activation of Advanced Functions on OpenPower 710, OpenPower 720, p5-510, p5-520, p5-550, p5-570, p5-575

Notes for iSeries

Types of Capacity on Demand for IBM @server iSeries servers include:

- ▶ Capacity Upgrade on Demand
- ▶ On/Off Capacity on Demand
- ▶ Reserve Capacity on Demand

- ▶ Trial Capacity on Demand

Notes for zSeries and Systemz9

Capacity on Demand offerings for IBM @server zSeries can allow for the non-disruptive addition of one or more central processors, internal coupling facilities and/or integrated facilities for Linux for predictable, permanent growth. Owners of z890, z990 and z9-109 systems also have the option of tapping inactive capacity to meet spikes in user demand, whether expected or unexpected, and to stay up and running when service outages and other unpleasant surprises occur.

Types of Capacity on Demand for IBM @server zSeries include:

- ▶ Capacity Upgrade on Demand for permanent capacity for non-disruptive growth.
Capacity Upgrade on Demand for zSeries and System z9 allows for the non-disruptive addition of one or more central processors, internal coupling facilities and/or integrated facilities for Linux for predictable, permanent growth. Using Capacity Upgrade on Demand, you can quickly add processors—up to the maximum number of available inactive engines—to dynamically meet your business needs. (Owners of select zSeries servers can also add extra memory and I/O capability on demand.)
zSeries also allows for permanent concurrent memory upgrades. This function allows you to concurrently enable memory that has been preconfigured on the server.
- ▶ On/Off Capacity on Demand for temporary capacity for fluctuating workloads.
On/Off Capacity on Demand for z890, z990 and z9-109 servers allows you to vary your hardware capacity as workloads peak and subside. While you grow your permanent capacity rationally and proportionately to accommodate the needs of your business, On/Off Capacity on Demand can give you a low-cost means of managing periodic business spikes, deploying new solutions and responding to unpredictable market demand. On/Off Capacity on Demand can help you better align your IT costs with the capacity you actually use.
- ▶ Capacity BackUp for interim capacity for continued operation.
Capacity BackUp for zSeries can provide emergency backup processing capacity for situations in which you lose capacity in part of your operation, helping you recover by adding reserved capacity on a designated zSeries system. A Capacity BackUp system is designed to operate with a "base" processing configuration and a pre-configured number of additional processing units reserved for activation in case of an emergency. You activate these backup units electronically, which can save time and eliminate the need for a technician to make a service call.

9.8 For iSeries and pSeries: Simultaneous Multithreading

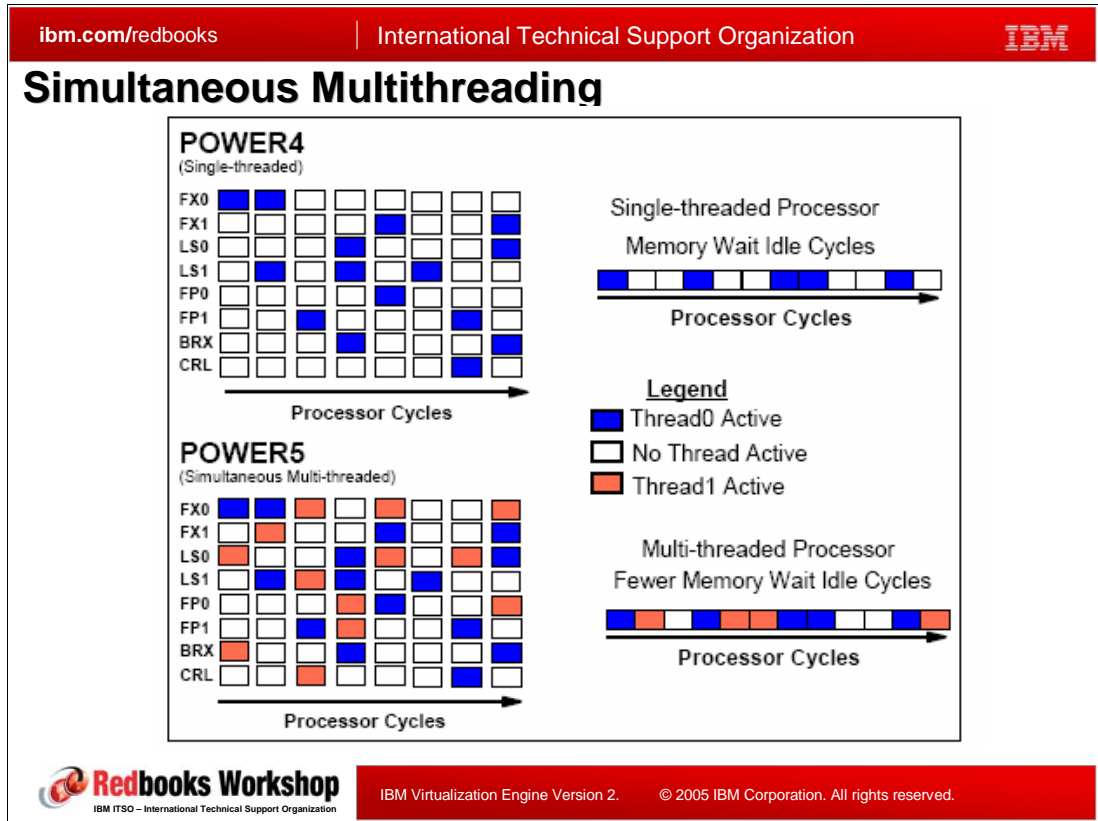


Figure 9-8 Simultaneous multithreading

Although an operating system gives the impression that it is concurrently executing a large number of tasks, each processor in a symmetric multiprocessor (SMP) traditionally executes a single task's instruction stream at any moment.

To improve performance, some (old) iSeries processors used an approach called hardware multithreading (HMT). In the hardware multithreading approach, the hardware automatically switches between the tasks on any long processing delay event,

Simultaneous multi-threading is the ability of a single physical processor to simultaneously dispatch instructions from more than one hardware thread context. Because there are two hardware threads per physical processor, additional instructions can run at the same time. Simultaneous multi-threading allows you to take full advantage of the POWER5 processor by scheduling two applications at the same time on the same processor. No single application can fully saturate the processor. Simultaneous multi-threading is a feature of the POWER5 processor and is available with shared processors. Simultaneous multi-threading is available on IBM @server eServer i5 and eServer p5 models.

Each POWER5 processor core supports the execution of two instruction streams, or threads. This capability is referred to as simultaneous multithreading (SMT). This design point provides the ability to fetch instructions from one or two threads per processor core and schedule instructions for execution from both threads concurrently.

The SMT support on POWER5 processors also introduced support for a dormant thread. When the partition does not have work to dispatch on one of the two threads of the physical processor, it can invoke an hcall to make the thread dormant. If the other thread is active, the

hypervisor makes the invoking thread dormant so that all of the register resources and cycles available on the physical processor are utilized by the remaining active thread. When an interrupt or time-out occurs for the dormant thread, the hardware revives the thread, its state is fully restored, and control is returned to the partition.

Each POWER5 processor core appears to the operating system as a two-way symmetric multiprocessor (SMP). Not all applications benefit from simultaneous multithreading. For this reason, the POWER5 supports single-threaded execution mode. In this mode, the POWER5 gives all physical resources to the active thread, enabling it to achieve higher performance than a POWER4 system at equivalent frequencies.

9.9 For iSeries: Multiple OS support

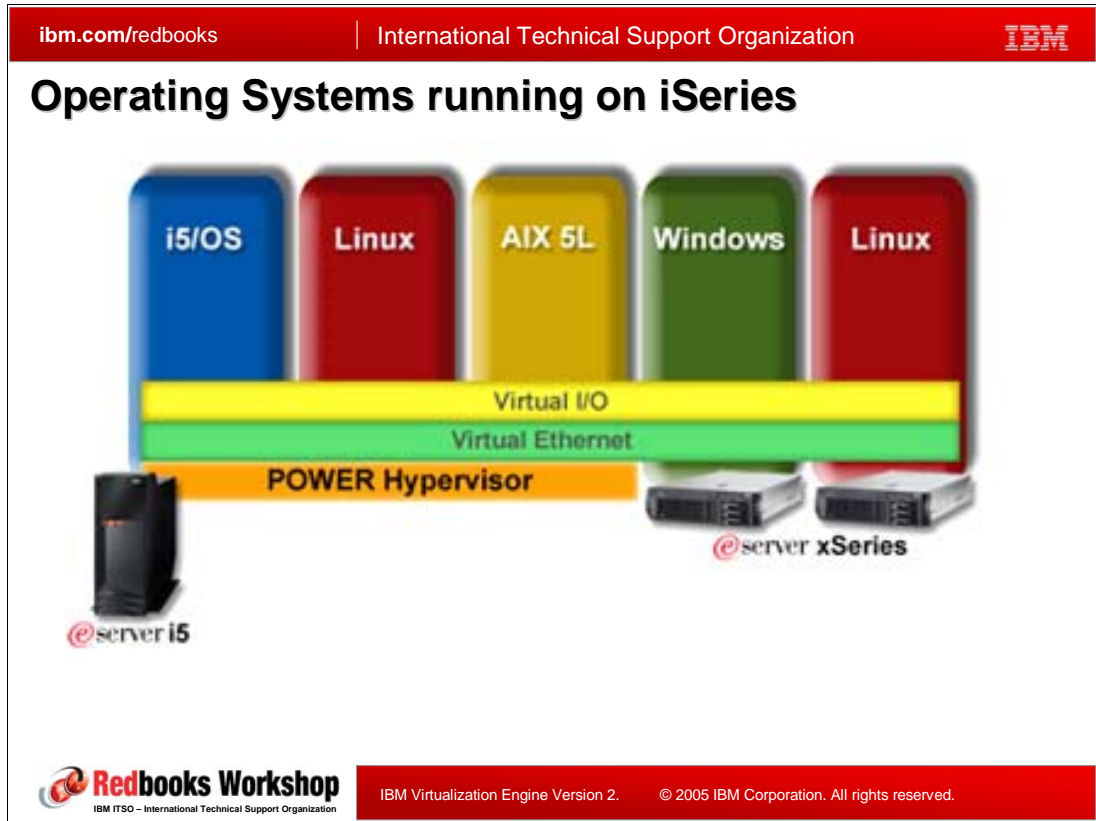


Figure 9-9 Multiple OS support on iSeries

The IBM @server i5 server runs multiple operating systems environments simultaneously, including Linux on POWER and Linux on Integrated xSeries Solutions, as well as IBM i5/OS , IBM AIX 5L , and Microsoft Windows via Integrated xSeries Solutions.

i5/OS

(Note that IBM i5/OS for V5R2 and earlier versions is called OS/400).

All of the software factors for a relational database, comprehensive security, communications with a broad range of diverse systems, including Internet capabilities, and many more components are fully integrated into i5/OS.

i5/OS contains the base operating system, additional optional feature components, and separate licensed programs bundled with the operating system at no extra charge. The client can purchase advanced features and functions that are not included in the base group of products shipped with the operating system. i5/OS is a 64-bit operating system.

AIX

AIX 5L supports the IBM @server i5, IBM @server p5, IBM @server pSeries, and IBM RS/6000 server product lines, as well as IntelliStation POWER and RS/6000 workstations. AIX 5L can run on iSeries in logical partitions, allowing you to optimize your infrastructure: share processor and memory resource, move resources to where they are needed, exploit i5/OS storage subsystem, and leverage skills and best practices. Since iSeries and pSeries p5 are built with the same POWER5 processors and server technology, AIX 5L applications

that run on pSeries p5 can run on iSeries unchanged. POWER5-, and PowerPC -based models can be executed on AIX 5L V5.3 without recompilation for the same and newer models in that processor family (POWER3 , POWER4, POWER5, or PowerPC) with a few exceptions. AIX 5L V5.3 offers simultaneous multi-threading on POWER5 systems to deliver industry leading throughput and performance levels.

Integrated xSeries Solutions

iSeries integrated xSeries solutions allow businesses to operate heterogeneous environments that include Intel -based servers running Microsoft Windows applications in addition to core business applications on iSeries. IBM @server iSeries integrated xSeries solutions allow businesses to operate heterogeneous environments that include Intel-based servers running Microsoft Windows or Linux applications. iSeries servers offer two solutions to Intel-based server farms by providing a consolidated server for iSeries applications and Windows services.

- ▶ **Integrated xSeries Server (IXS):** through the Integrated xSeries Server for iSeries, you can run Microsoft Windows Server 2003, Microsoft Windows 2000, Windows NT and Linux servers and your iSeries in a single system footprint. The Integrated xSeries Server is a unique product that puts the power of an xSeries server on a single card that can be plugged into your iSeries system.
- ▶ **Integrated xSeries Adapter (IXA):** the Integrated xSeries Adapter allows you to connect as many as 60 standard 1-way, 2-way, 4-way, or 8-way IBM xSeries servers to your iSeries through the High Speed Link, leveraging the storage consolidation and systems management advantages of the iSeries and providing greater power and flexibility than the Integrated xSeries Server.

Linux

IBM @server i5 and iSeries servers support two Linux environments:

- ▶ **Linux on POWER:** Linux runs in dynamic logical partitions leveraging IBM POWER processors on eServer i5 and iSeries servers. IBM Virtualization Engine technologies provide Micro-Partitioning, Capacity on Demand, and I/O virtualization facilities for Linux.
- ▶ **Linux on Integrated xSeries Solutions:** iSeries Integrated xSeries Solutions allow businesses to operate heterogeneous environments that include Intel -based servers running Linux applications in addition to core business applications on i5/OS or OS/400. i5/OS and OS/400 V5R2 support running Linux on the Integrated xSeries Server (IXS) and on xSeries servers that are attached to the iSeries via the Integrated xSeries Adapter (IXA).

i5/OS is enhanced to support Linux running in a secondary LPAR. The primary partition must run OS/400 V5R1 or later. Up to 31 Linux partitions are supported, depending on the iSeries model.

Linux on the iSeries can be implemented as either hosted or non-hosted. In a hosted environment, Linux uses virtual resources (such as disk, tape, and LAN), which are owned by an i5/OS partition and shared with the i5/OS partition. In a non-hosted environment, the Linux partition does not depend on a hosting i5/OS partition for any input/output (I/O) resources.

The i5 servers, based on IBM POWER5 technology, are capable of running POWER Linux distributions. Linux for iSeries is available directly from Linux distributors. RedHat Enterprise Linux AS 3, based on the 2.4 version of the 64-bit kernel supports the i5 and the earlier iSeries servers. SUSE LINUX Enterprise Server 9 (based on the 2.6 version of the 64-bit Linux kernel) supports the i5 and the earlier iSeries servers; SUSE LINUX Enterprise Server 8, uses the 2.4 kernel, and supports the earlier iSeries servers and Integrated xSeries Adapter attach.

With the Virtual Partition Manager, an eServer i5 system can support one i5/OS partition and up to four Linux partitions.

9.10 For zSeries: Multiple operating systems support with z/VM

ibm.com/redbooks | International Technical Support Organization IBM

z/VM environment

- ❑ z/VM supports a mixture of different operating systems and multiple copies of each operating system
- ❑ z/VM includes 35 years of experience to exploit virtualization technologies in the System z9 and zSeries environments
- ❑ z/VM allows for 10 to hundreds of virtual Linux servers
- ❑ Virtual Machine Resource Manager is a dynamic tuning of groups of virtual machines
- ❑ z/VM Guest LAN allows multiple virtual LAN segment in a z/VM environment

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 9-10 z/VM environment

z/VM

z/VM offers a base for customers who want to exploit IBM virtualization technology on the IBM System z9 or zSeries platforms. With z/VM, customers can easily create many virtual machines consisting of virtualized processor, communications, storage, networking, and I/O resources. z/VM includes over 35 years of innovation and invention. z/VM provides support for Linux, z/OS (and z/OS.e), VSE/ESA and TPF.

The latest version, z/VM V5.2 became available on December 16, 2005. z/VM V5.2 extends the strength of the System z9 and zSeries environments as platforms for integrating applications and data and for consolidating select UNIX, Windows, and Linux workloads on a single physical System z9 or zSeries server. z/VM supports Linux on the mainframe and enables Linux servers to share hardware resources and use internal high-speed communications (hipersockets) within the System z9 and zSeries environments. z/VM allows you to run tens to hundreds of virtual Linux servers.

z/VM provides a CIM API which is a systems management interface that simplifies the deployment and management of z/VM virtual servers. This standard interface is used by IBM Director to manage a z/VM virtual environment.

For more information about z/VM, you can visit the following Web sites:

<http://www.vm.ibm.com/library/gm130137.pdf>
<http://www.vm.ibm.com/zvm520/>

z/VM Guest LAN

Prior to z/VM 4.2, virtual connectivity options for connecting one or more virtual machines (VM guests) was limited to virtual channel-to-channel adapters (CTCA); these virtual interfaces are classified as point-to-point connections.

From z/VM 4.2 onwards, z/VM has been enhanced to provide a feature known as Guest LAN which allows you to create multiple virtual LAN segments within a z/VM environment. There is no architectural limit on the number of Guest LAN segments that you can create; however, you are limited by the amount of available machine resources. In contrast to point-to-point connections, to connect to a Guest LAN, individual guest machines create a virtual Network Interface Card (NIC). They can then connect this NIC to the LAN and communicate with other guests using standard TCP/IP protocols. A z/VM Guest LAN is solely contained within a single z/VM operating system environment and thus guests do not communicate with systems outside of the Guest LAN. Starting with z/VM 4.3, the Guest LAN can be defined to use simulated HiperSockets devices. z/VM V4.4 further enhances this virtualization technology by introducing a virtual IP switch that is capable of bridging a z/VM Guest LAN to an associated real LAN connected by an OSA-Express adapter.

Virtual machines in the z/VM Guest LAN environment can define and use simulated OSA-Express devices that support both the Internet Protocol version 4 (IPv4) and IPv6 protocols.

Virtual Machine Resource Manager

Virtual Machine Resource Manager (VMRM) is a new facility available since z/VM 4.3. It runs in a service virtual machine (VMRMSVM) and dynamically manages the performance of workloads. VMRM uses a set of user-specified workload definitions and goals, compares these with the achieved performance, and makes adjustments accordingly. (It is conceptually somewhat similar to the Workload Manager used by z/OS.) The basic idea is to allow performance objectives to be set in a manner more closely aligned with business objectives than has been the case previously.

VMRM is only effective in a constrained environment because it works by prioritizing workloads in order to enhance their access to resources, and accordingly restricts other work's access to those resources.

A workload is a collection of one or more virtual machines that are to be treated as an entity for the purpose of performance management. Each workload has certain goals defined for it. These goals are for DASD and CPU utilization. A workload can have a DASD goal, a CPU goal, or both. A goal represents the relative importance of a workload for the particular installation. Configuration changes can be made at any time, dynamically. VMRM cannot adjust users with fixed CPU or I/O priorities.

Virtual Machine Resource Manager can be described as a dynamic tuning of groups of virtual machines mechanism.

9.11 For zSeries: HiperSockets

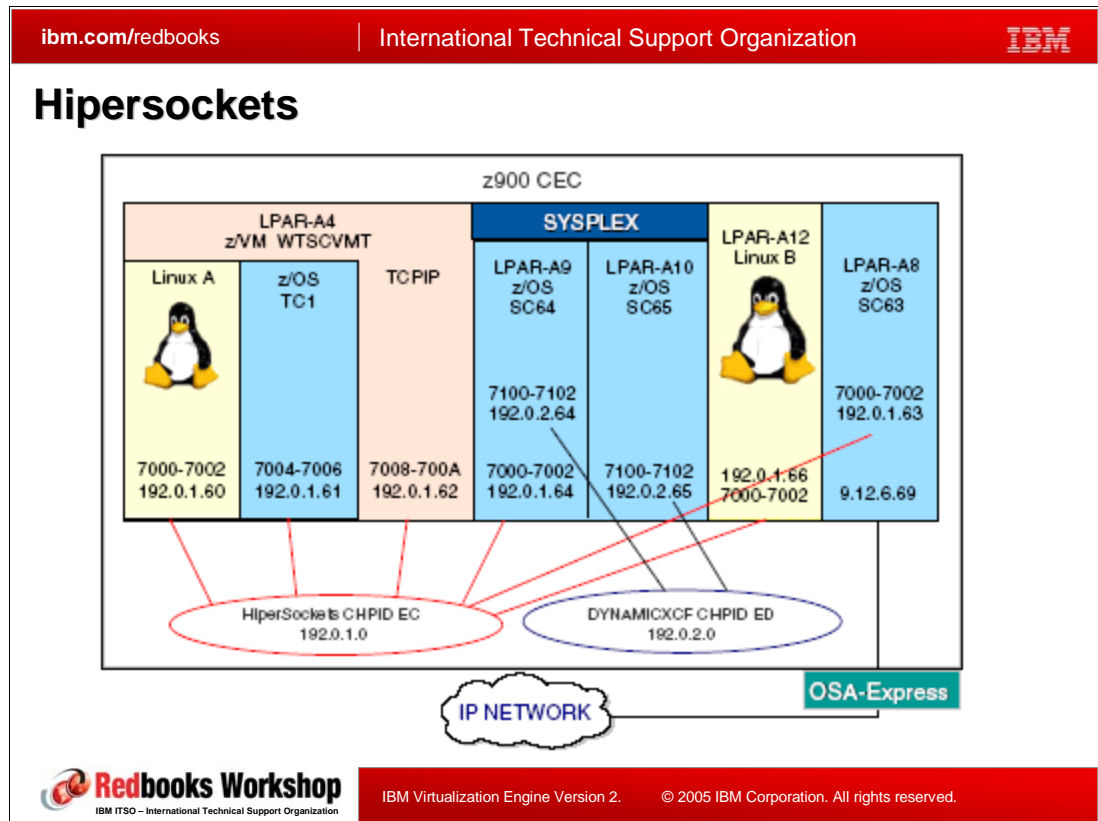


Figure 9-11 HiperSockets

HiperSockets

zSeries HiperSockets is a technology that provides high-speed TCP/IP connectivity between servers within a zSeries server (z800 and z900 servers) and IBM System z9. HiperSockets simulation capability can be installed on an S/390 G5/G6, or Multiprise® 3000 processor; however, this support is only available with z/VM V4R2.

It eliminates the need for any physical cabling or external networking connection between these virtual servers. HiperSockets provide very fast TCP/IP communications between servers running in different Logical Partitions (LPARs) on a box. The communication is through the system memory of the processor. The virtual servers that are so connected form a “virtual LAN”.

zSeries HiperSockets is a microcode function of a zSeries server and IBM System z9. It is supported by the operating systems z/OS (V1R2 and later, z/OS.e, z/VM V4R2 and later, Linux for zSeries (64-bit mode), and Linux for S/390 (31-bit mode). or later. HiperSockets supports up to four independent “virtual LANs” which operate as TCP/IP networks within a zSeries box. HiperSockets can be customized to accommodate varying traffic sizes.

Since there is no server-to-server traffic outside the zSeries CEC, a much higher level of network availability, security, simplicity, performance, and cost effectiveness is achieved as compared with servers communicating across a LAN.

HiperSockets implementation is based on the OSA-Express queued direct input/output (QDIO) protocol, hence HiperSockets is also called internal QDIO (iQDIO). The HiperSockets

microcode support up to four independent HiperSockets, up to 40 I/O devices per LPAR across all four HiperSockets (up to a total of 3072 I/O devices across all four HiperSockets on the box) and up to 4000 IP addresses across all four HiperSockets.

9.12 xSeries and BladeCenter summary

ibm.com/redbooks
International Technical Support Organization

Virtualization Engine Platform xSeries Summary

- ❑ **Virtual Resources**
 - IBM Virtual Machine Manager
 - Capacity Manager
 - System Availability
- ❑ **Management collection**
 - Windows
 - 5733-EWM, IBM Virtualization Engine Enterprise Workload Manager V2.1
 - 5733-DIR, IBM Director with VE Console
 - 5697-NFD, IBM Director
 - Linux on xSeries
 - 5733-EWM, IBM Virtualization Engine Enterprise Workload Manager V2.1
 - 5733-DIR, IBM Director with VE Console
 - 5733-RDS, IBM Virtualization Resource Dependency Service for Linux, V2.1

	Windows	Linux for x
EWLM	☑	☑
IBM Director	☑	☑
RDS		☑
VE Console	☑	☑

IBM Virtualization Engine Version 2.
© 2005 IBM Corporation. All rights reserved.

Figure 9-12 xSeries and BladeCenter virtualization capabilities

VMware is one of the key virtualization products for xSeries. It allows xSeries servers to run multiple operating systems and applications in virtual machines, allowing you to centralize IT management. VMware and the Virtualization Engine complement each other:

- ▶ VMware provides resource management, such as partitioning, for Intel compatible systems. VMware focuses on the VM level and is not application based.
- ▶ The Virtualization Engine provides access and management of resources across multiple hardware platforms. All of the Virtualization Engine products operate across different platforms. It provides a granular, application based workload management.

The following Virtualization Engine Version 2 products are now available on Windows:

- ▶ 5733-EWM, IBM Virtualization Engine Enterprise Workload Manager V2.1
- ▶ 5733-DIR, IBM Director with Virtualization Engine console
- ▶ 5697-NFD, IBM Director

9.13 System z9 and zSeries summary

ibm.com/redbooks
International Technical Support Organization

Virtualization Engine Platform zSeries Summary

- ❑ **Virtual Resources**
 - Logical Partitioning
 - z/VM
 - Hipersockets
 - Capacity on Demand
 - z/VM Guest LAN
 - Virtual Server Deployment and Server Complex
 - Virtual Machine Resource Manager
- ❑ **Management collection**
 - z/OS
 - 5655-EWM IBM Virtualization Engine Enterprise Workload Manager for z/OS V2.1
 - Prereq z/OS 1.6 minimum
 - Linux on zSeries
 - 5648-F08 IBM Virtualization Engine and Infrastructure Services for Linux on System z9 and zSeries V 2 R 1
 - Base (for free): delivers IBM Director Management Server for Linux on zSeries, IBM Director Console (for Windows) and IBM Director agents for all IBM eServer managed server platforms
 - EWLM: Domain Manager, agents for AIX, i5/OS, Windows, Sun
 - RDS, with all its components
 - Prereq SLES 9 or RHEL 4.0

	z/OS	Linux for z
EWLM	☑	☑
IBM Director		☑
RDS		☑
VE Console		☑

IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2.
© 2005 IBM Corporation. All rights reserved.

Figure 9-13 System z9 and zSeries virtualization capabilities

Virtualization is inherent to the zSeries architecture; this architectural strategy provides the advantage of incorporating virtualization functions into the hardware or software from the design phase of the system.

Logical Partitioning (LPAR) is enabled by the Processor Resource System Manager (PR/SM), part of the zSeries hardware since the 1980's ! PR/SM has been recognized being able to isolate the operating environments running with each partition and LPARs on zSeries have achieved the common criteria security certification Evaluation Assurance Level (EAL) 5 on the zSeries servers.

The combination of partitioning (PR/SM-based) and the virtualization capabilities offered by z/VM offer unique capabilities.

zSeries operating environments provide the Workload Manager for z/OS (WLM) and the Virtual Machine Resource Manager (VMRM) for z/VM.

- ▶ WLM dynamically interacts with PR/SM and virtualized resources (CP, memory, I/O) to offer a complete automated workload balancing solution (inside the zSeries environment). EWLM derives from WLM for z/OS.
- ▶ VMRM can automate the reallocation of resources, and prioritize some of the workloads to meet service goals.

Parallel Sysplex was introduced to provide a single image infrastructure to have multisystem data sharing applications with integrity, availability, and scalability. HiperSockets introduced the ability for LPARs to communicate at near memory speed and also results in reducing the

amount of hardware components that is one of the first step of infrastructure simplification. Intelligent Resource Director (IRD), another z/OS function, is a combination of the LPAR, WLM, and Parallel Sysplex facilities. IRD allows you to move dynamically workloads to where the processing resources are available. As a result, applications that support data sharing can potentially run on any system in a sysplex.

The new announced product, IBM Virtualization Engine and Infrastructure Services for Linux on System z9 and zSeries, 5648-F08, positions the System z9 and zSeries platforms to play a key role as a hub for managing cross-system resources. It is the implementation of the Virtualization Engine for zSeries running Linux. This product is structured into base product components and four optional product components.

- ▶ Base product components are delivered at no additional charge with the product (service and support are fee-based); they include:
 - IBM Director 5.10, which include:
 - The IBM Director Management Server for Linux on System z9 and zSeries
 - The IBM Director console
 - The IBM Director agents for all IBM @server managed server
 - The Management Access Point interface to z/VM
 - IBM Virtualization Engine console, which include:
 - The Virtualization Engine console
 - The common runtime components

Note: Although IBM Director and the Virtualization Engine console are delivered in one package, IBM Director installation and use are independent of the Virtualization Engine console installation and functions. The Virtualization Engine console can be used as an integration point of multiple IBM Director consoles, for example to get a central system management control point, but the IBM Director console can be used instead and does not require any Virtualization Engine console or common runtime components.

- ▶ The priced optional product components may be one or a combination of the following:
 - EWLM (EWLM Domain Manager, EWLM Control Center, and EWLM agents for AIX, i5/OS, z/OS, Linux, and HP-UX)
 - EWLM (EWLM Domain Manager, EWLM Control Center, and EWLM agents for Solaris and Windows)
 - Resource Dependency Service with all its components
 - IBM Director extensions (z/VM Center extensions, Virtual Server Deployment and Server Complexes)

IBM Virtualization Engine Enterprise Workload Manager for z/OS V2.1.0 (EWLM), 5655-EWM, extends the scope of z/OS workload management (WLM) capabilities across multiple platforms to monitor applications and resources. EWLM and z/OS WLM can run simultaneously. You can continue z/OS management with WLM and use EWLM to monitor and manage application workloads, workflows, and allocation of system resources across an heterogeneous environment (including AIX, OS/400, Linux, Windows, Solaris and HP/UX) based on business policy and priority. For more information about the EWLM on zSeries (for Release 1 at the time of writing this book), you can read *Enterprise Workload Manager z/OS Support*, REDP-3993.

The new bridge expected in 1Q2006 to link the Virtualization Engine and Tivoli Enterprise Portal Server and Tivoli Enterprise Management Server (previously titled Candle Net Portal Server (CNPS) and Candle Management Server (CMS)) will allow the integration of z/OS systems and sysplexes into the Virtualization Engine console.

9.14 System p5 and pSeries summary

ibm.com/redbooks
International Technical Support Organization

Virtualization Engine Platform pSeries Summary

- ❑ **Virtual Resources**
 - DLPAR
 - Integrated Virtualization Manager
 - Partition Load Manager
 - Micro-Partitioning
 - Virtual I/O, Virtual LAN and shared Ethernet Adapter
- ❑ **Management collection**
 - AIX
 - 5733-EWM, IBM Virtualization Engine Enterprise Workload Manager V2.1
 - 5733-DIR, IBM Director with VE Console
 - 5765-DR1, IBM Director for pSeries, 5.10
 - Linux on pSeries
 - 5733-EWM, IBM Virtualization Engine Enterprise Workload Manager V2.1
 - 5733-DIR, IBM Director with VE Console
 - 5733-RDS, IBM Virtualization Resource Dependency Service for Linux, V2.1

	AIX	Linux for p
EWLM	☑	☑
IBM Director	☑	☑
RDS		☑
VE Console	☑	☑

Redbooks Workshop
IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 9-14 System p5 and pSeries virtualization capabilities

The POWER Hypervisor is a code layer that comes with every p5 server; it provides the virtualization foundation that enables the Virtualization Engine technologies. The POWER Hypervisor distributes server resources to the logical partitions on the server according to customer specifications; it also enforces logical partition security, and can provide virtual Ethernet channels between physical partitions, reducing the need for physical Ethernet adapters and freeing I/O adapter slots.

Logical partitions use dedicated and shared processors to run concurrent operations. Micro-partitioning provides additional, advanced, partitioning support and gives you the ability to assign fractions of a processor to a logical partition.

The Virtual I/O Server provides virtual storage and shared Ethernet capability to client logical partitions on the system. It allows physical adapters with attached disks on the Virtual I/O Server to be shared by one or more client partitions. The Virtual I/O Server is installed in its own logical partition and is not intended to run applications.

The POWER5 processor-based products support i5/OS , AIX 5L , and Linux. Windows is supported if you have an Integrated xSeries Adapter (IXA) and Integrated xSeries Server (IXS) installed in an i5/OS subsystem, which requires an iSeries expansion unit or tower.

The following Virtualization Engine Version 2 products are now available on AIX:

- ▶ 5765-DR1 IBM Director for pSeries, 5.10. IBM Director for pSeries V5.10 is available to IBM eServer pSeries customers at no additional charge. Until IBM Director V5.10 becomes available, you can still use IBM Director to manage a mixed IBM eServer pSeries

and xSeries environment, but you must use an xSeries system to manage all of your pSeries servers because the management server is not supported on pSeries. With IBM Director V5.10, the management server is now supported on both pSeries and xSeries servers, so you can manage servers using a pSeries system.

- ▶ 5733-EWM, IBM Virtualization Engine Enterprise Workload Manager V2.1
- ▶ 5733-DIR, IBM Director with Virtualization Engine console

9.15 iSeries summary

ibm.com/redbooks
International Technical Support Organization

Virtualization Engine Platform iSeries Summary

- ❑ **Virtual Resources**
 - LPAR Simplification Tool
 - DLPAR
 - Micro partitioning
 - Virtual I/O, Virtual LAN
 - Capacity on Demand
 - Simultaneous Multithreading
 - Multiple OS support
- ❑ **Management collection**
 - I5/OS
 - 5733-EWM, IBM Virtualization Engine Enterprise Workload Manager V2.1
 - 5733-DIR, IBM Director with VE Console
 - Linux on iSeries
 - 5733-EWM, IBM Virtualization Engine Enterprise Workload Manager V2.1
 - 5733-DIR, IBM Director with VE Console
 - 5733-RDS, IBM Virtualization Resource Dependency Service for Linux, V2.1

	I5/OS	Linux
EWLM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
IBM Director	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
RDS		<input checked="" type="checkbox"/>
VE Console	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Redbooks Workshop

IBM ITSO - International Technical Support Organization

IBM Virtualization Engine Version 2. © 2005 IBM Corporation. All rights reserved.

Figure 9-15 iSeries virtualization capabilities

eServer i5 servers, combined with the POWER Hypervisor feature provides dynamic logical partitioning (DLPAR), including the latest enhancement that enables automatic processor movement processor using 'uncapped' partitions. The eServer i5 will eventually support up to 254 partitions, with a maximum of 10 per processor.

By supporting AIX 5L, the eServer i5 opened consolidation opportunities to deploy UNIX solutions in a single infrastructure server alongside i5/OS and other operating environments such as, Linux for POWER, and Windows 2003 Server via the Integrated xSeries Server or Integrated xSeries Adapter. Additionally, IBM also plans to enhance the Integrated xSeries Server or Integrated xSeries Adapter solution to enable Intel processor-based Linux applications on both eServer i5 and iSeries servers.

The capability to run AIX 5L is available on all eServer i5 models 520, 550, 570 and 595. AIX 5L 5.3 supports the capability for micro-partitioning, virtual storage and virtual Ethernet. While AIX 5.2 is supported on the eServer i5, it requires at least one processor per partition and direct attached storage. AIX 5L runs natively on the eServer i5, it is the same AIX 5L that runs on pSeries, running on its native hypervisor and processor. We recommend that the LPAR Validation Tool be used to understand and plan for deploying AIX partitions on eServer i5. For V5R3, Windows environment on iSeries has several new functions: for example users enrolled in the Windows environment from i5/OS can now manage their passwords in Windows; Enterprise Identity Mapping (EIM) user enrollment support allows easier setup for Windows single sign-on and also allows enrolled OS/400 user profiles to be different than Windows users profiles; the maximum network server storage space size has been increased from 64 GB to 1TB with a maximum of 32TB on a single Windows server.

IBM High Availability Cluster MultiProcessing for AIX 5L (HACMP V5.1 and HACMP V5.2) supports eServer i5 servers running either AIX 5L V5.2 or V5.3.

Red Hat Enterprise Linux AS for POWER Version 3 and SUSE LINUX Enterprise Server 9 for POWER are also available. iSeries Navigator is enhanced with functions for managing Linux in a logical partition, in the same way that Windows servers are managed today. The Linux support on the Integrated xSeries Server and Adapter is available.

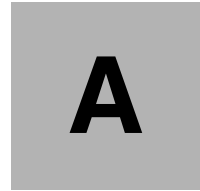
With Capacity Upgrade on Demand, you can turn processors on when you need them – turn them off when you don't. You pay only for what you activate.

IBM Director is designed to provide a consistent systems management infrastructure so that heterogeneous systems and their resources may be managed together. Coupled with the Virtualization Engine console which provides a common Web-based console for monitoring/managing the overall health of an on demand operating environment, IBM Director provides a powerful range of system management capabilities. IBM Director will complement iSeries Navigator and Management Central functions and initially, will have similar functionality such as hardware and software inventory, system monitors such as CPU utilization, threshold triggering, and event logs monitoring across multiple systems or logical partitions. Additionally, users will be able to review system-wide, cross-partition performance information that can be combined with eServer PM for iSeries for capacity planning across multiple operating systems.

The following Virtualization Engine Version 2 products are now available:

- ▶ Available for iSeries only:
 - 5733-DR1, IBM Director with Virtualization Engine console for i5/OS, V5.10
 - 5733-EWA, IBM Virtualization Engine Enterprise Workload Manager for i5/OS, V2.1
- ▶ Available for iSeries, pSeries and xSeries:
 - 5733-DIR, IBM Director with Virtualization Engine console, V5.10
 - 5733-EWM, IBM Virtualization Engine Enterprise Workload Manager, V2.1
 - 5733-RDS, IBM Virtualization Engine Resource Dependency Service, V2.1 (availability February 2006)

IBM Director and Virtualization Engine Console for I5/OS are included with every copy of i5/OS (I5/OS server only); IBM Director and Virtualization Engine Console are included in the i5/OS (5722-SS1) Software Maintenance. IBM Director and Virtualization Engine Console are available separately in AAS for i5/OS, Linux on POWER, Windows, Linux on xSeries with priced software maintenance. They are available for free download on the web.



Integrated Virtualization Manager versus Hardware Management Console

This appendix provides a comparison about the functionality offered by the Integrated Virtualization Manager (IVM) versus the Hardware Management Console (HMC).

Table A-1 IVM versus HMC

	Integrated Virtualization Manager (IVM)	Hardware Management Console (HMC)
Physical footprint	Integrated into the server	A desktop or rack-mounted appliance
Installation	Installed with the VIOS (optical or network). Preinstall option available on some systems.	Appliance is preinstalled. Reinstall via optical media or network is supported.
Managed Operating Systems supported	AIX and Linux	i5/OS, Linux, and AIX
Virtual console support	AIX and Linux virtual console support	i5/OS, Linux, and AIX virtual console support
User security	Password authentication with support for either full or ready only authorities	Password authentication with granular control of task based authorities and object based authorities
Network security	-No integrated firewall support -Web server SSL support	-Integrated firewall -SSL support for clients and for communications with managed systems
Supported Hardware	p5-505, p5-510, p5-520, p5-550, p5-550Q, OpenPower 710, and OpenPower 720	All POWER5 iSeries, pSeries, and OpenPower servers

	Integrated Virtualization Manager (IVM)	Hardware Management Console (HMC)
Multiple system support	One IVM per server	One HMC can manage multiple servers
Redundancy	One IVM per server	Multiple HMCs can manage the same system for HMC redundancy
Maximum number of partitions supported	Firmware maximum	Firmware maximum
Uncapped Partition Support	Yes	Yes
Dynamic Resource Movement (DLPAR)	No - dynamic support is only available for the management partition	Yes - full support
I/O Support for AIX and Linux	Virtual optical, disk, Ethernet, and console.	Virtual and Direct
I/O Support for i5/OS	None	Virtual and Direct
Maximum # of virtual LANs	4	4096
Fix/Update process for Manager	VIOS fixes and updates	HMC e-fixes and release updates
Adapter microcode updates	Inventory scout	Inventory scout
Firmware updates	VIOS firmware update tools	Service Focal Point with concurrent firmware updates
I/O Concurrent Maintenance	VIOS support for slot and device level concurrent maintenance via the diag hot plug support	Guided support in the "Repair and Verify" function on the HMC.
Scripting and Automation	VIOS CLI and HMC compatible CLI	HMC command line interface
Capacity on Demand	Limited support through ASMI	Full Support
User Interface	Web Browser (no local graphical display)	WebSM (local or remote)
Workload Management (WLM) Groups Supported	1	254
LPAR Configuration Data Backup and Restore	Yes	Yes
Support for multiple profiles per partition	No	Yes
Serviceable event management	Service Focal Point Light - Consolidated management of firmware and management partition detected errors.	Service Focal Point support for consolidated management of OS and firmware detected errors
Hypervisor and service processor dump support	Dump collection with support to do manual dump downloads	Dump collection and call home support

	Integrated Virtualization Manager (IVM)	Hardware Management Console (HMC)
Remote support	No remote support connectivity	Full remote support for the HMC and connectivity for firmware remote support

Abbreviations and acronyms

AAS	Advanced Administrative System	EJB	Enterprise Java Beans
AMC	Autonomic Monitoring Engine	EPR	Endpoint Reference
API	Application Programming Interface	ERD	Enterprise Resource Director
ARM	Application Response Measurement	ESB	Enterprise Service Bus
ARP	Address Resolution Protocol	ESS	Enterprise Storage Server
AUIML	Abstract User Interface Markup Language	EWLM	Enterprise Workload Manager
BPEL4WS	Business Process Execution Language for Web Services	FTP	File Transfer Program
CBE	Common Base Event	GCR	Global Configuration Repository
CBM	Component Business Modeling	GUI	Graphical User Interface
CCI	Common Client Interface	HLD	High Level Design
CCMDB	Common Configuration Management Database	HMC	Hardware Management Console
CEC	Central Electronic Complex	HPI	Hosting Partition Infrastructure
CICS	Customer Information Control System (HTML	Hypertext Markup Language
CIM	Common Information Model	HTTP	Hypertext Transfer Protocol
CIM Provider	Common Information Model Provider	HTTPS	Hypertext Transfer Protocol Secure
CIMOM	Common Information Model Object Manager	IBM	International Business Machines Corporation
CMDB	Common Model Database Schema	ICL	IBM Common Logging
CMR	Converged ID Mapping Repository	ICMP	Internet Control Message Processing
CP	Control Processor	IDD	Intelligent Device Discovery
CRT	Common Runtime	IGT	IBM Grid Toolbox
CSM	Cluster System Manager	IMR	Identity Mapping Repository
DASD	Direct Access Storage Device	IMS	Information Management System
DB2	DATABASE 2	IP	Internet Protocol
DBCTL	Data Base Control Subsystem	IRD	Intelligent Resource Director
DBRC	Data Base Recovery Control	ISC	Integrated Solution Console
DCCTL	Data Communication Control	ISV	Independent Software Vendor
DCM	Data Center Model	IT	Information Technology
Director MP	Director Multiplatform	ITDS	IBM Tivoli Directory Services
DLPAR	Dynamic Logical Partitioning	ITSO	International Technical Support Organization
DMTF	Distributed Management Task Force	IUDD	Installable Unit Deployment Descriptor
DN	Domain Name	IVM	Integrated Virtualization Manager
DNS	Domain name System	iXA	Integrated xSeries Adapter
EAP	Event Action Plan	J2EE	Java 2 Platform Enterprise Edition
EIM	Enterprise Identity Mapping	J2SE	Java 2 Platform, Standard Edition
		JAAS	Java Authentication and Authorization Service
		JCA	Java Connector Architecture

JCL	Job Control Language	PICS	Platform Installation and Configuration Services
JDBC	Java DataBase Connector	PLM	Partition Load Manager
JMX	Java Management Extensions	PMAC	Policy Management for Autonomic Computing
JNI	Java Native Interface	PR/SM	Processor Resource System Manager
JNLP	Java Network Launching Protocol	RDBMS	Relational Database Management System
JRE	Java Runtime Environment	RDM	Remote Deployment Manager
JVM	Java Virtual Machine	RDS	Resource Dependency Services
JWS	Java Web Start	REXX	Restructured EXtended eXecutor language
LAN	Local Area Network	RMC	Resource Monitoring and Control
LDAP	Lightweight Directory Access Protocol	RMC	Remote Monitoring and Control
LDW	LPAR Deployment Wizard	RMS	Resource Mapping Services
LIC	Launch in Context	RMTR	Resource Modeling Type Registry
LMB	Logical Memory Block	RTDS	Resource Topology and Discovery Services
LPAR	Logical Partition	SAN	Storage Area Network
LPW	LPAR Planning Wizard	SASP	Server Application State Protocol
LTA	Log and Trace Analyzer	SDK	Software Development Kit
LTPA	Lightweight Third Party Authentication	SLP	Service Location Protocol
LVT	LPAR Validation Tool	SMASH	Systems Management Architecture for Server Hardware
MAC	Media Access Control	SMB	Server Message Block
MC	Management Central	SMF	System Management Framework
MIB	Management Information Base	SMI-S	Storage Management Initiative Specification
MOM	Microsoft Operation Manager	SNIA	Storage Network Industry Association
MOWS	Management Of Web Services	SNMP	Simple Network Management Protocol
MP	Multi platform	SOA	Service-Oriented Architecture
MQAI	Message Queuing Administration Interface	SOAP	Simple Object Access Protocol
MUWS	Management Using Web Services	SPX	IBM Software Platform Extension
NCP	Network Control Program	SSO	Single Sign On
NRM	Network Resource Model	SVC	SAN Volume Controller
OASIS	Organization for the Advancement of Structured Information Standards	TBSM	Tivoli Business Service Manager
ODI	On Demand Infrastructure	TCM	Tivoli Configuration Manager
ODI-RM	On Demand Infrastructure Resource Modeling	TCP	Transmission Control Protocol
ODOE	On Demand Operating Environment	TCP/IP	Transmission Control Protocol /Internet Protocol
OGSA	Open Grid Services Architecture	TCP/IP	Transmission Control Program/Internet protocol
OGSI	Open Grid Services Infrastructure	TIM	Tivoli Monitoring
OSA	Open Systems Adapter	TIO	Tivoli Intelligent Orchestrator
OSGi	Open Services Gateway Initiative		
PAM	Pluggable Authentication Modules		
PAP	Provisioning Automation Pack		

TM	Transaction Manager
TME	Tivoli Management Environment
TPC	Total Storage Productivity Center
TPF	Transaction Processing Facility
TPM	Tivoli Provisioning Manager
UCD	User Centered Design
UDDI	Universal Description, Discovery, and Integration
UDP	User Datagram Protocol
URI	Unviversal Resource Identifier
URL	Uniform resource Locator
VA	Virtualization Agent
Virtualization Engine	Virtualization Engine
Virtualization Engine	Virtualization Engine console
Virtualization Engine	Virtualization Engine Environment
VIOS	Virtual I/O Server
VIPA	Virtual IP Addressing
VM	Virtualization Manager
VMM	Virtualization Machine Manager
VMM	Virtual Machine Manager
VSC	Virtual Server Complex
VSD	Virtual Server Deployment
VTS	Virtual Tape Server
WBEM	Web Based Enterprise Management
WII	WebSphere Information Integrator
WLM	Workload Manager
WMI	Windows Management Instrumentation
WPS	WebSphere Portal Server
WSA	Web Services Addressing
WSDL	Web Service Definition Language
WSN	Web Service Notification
WS-RF	Web Services Resource Framework
WSRL	Web Resource Lifetime
WUWS	Management Using Web Services
XML	eXtensible Markup Language

Glossary

Abstract User Interface Markup Language (AUIML).

An XML implementation that provides a platform and technology-neutral method of representing windows, wizards, property sheets, and other user interface elements. It defines the intent of the user interface, such that it can be described once and rendered to the user in multiple environments and on various devices. AUIML is being developed and defined by the team who is developing the IBM Autonomic Computing Common Systems Administration core capability.

Advanced Administrative System. The AAS Order Fulfillment system provides the order data, and the associated invoice data for an order.

Application Programming Interface (API). An interface that allows an application program that is written in a high-level language to use specific data or functions of the operating system or another program.

Application Response Measurement (ARM). A standard of Open Group. Application can use the ARM API set to report the information to calculate the response time or monitor the status of work processed. In Virtualization Engine R1, an application must be instrumented using ARM4.0 standard to be monitored by EWLM. In Virtualization Engine R2, non-ARM instrumented clients are supported by EWLM in a different way.

BM Software Platform Extension (SPX). The collection of architectures and specifications that define how the IBM software componentization strategy is implemented, and guide the changes to IBM software development processes, management systems, and culture.

CIM Server. A CIM Server provides access to instrumented CIM resources. Pegasus is an open source implementation of a CIM Server that has been ported onto all IBM servers.

Common Base Event (CBE). A standard, XML-based format that defines the structure of an event. The Common Base Event includes logging, tracing, management, and business events.

Common Information Model (CIM). The CIM is a conceptual information model for describing management properties that is not bound to a particular implementation. This allows for the interchange of management information between management systems and applications through the Common Information Model Object Manager (CIMOM).

Common Information Model Object Manager (CIMOM).

CIMOM is an object management engine that exists between the managed system and the management application. Management applications can be local or remote, but currently CIMOM and the programs that collect management data, or "providers, must be located on the machine that is being instrumented. See also: Pegasus

Common Information Model Provider (CIM provider). Program which corrects the management data of CIM and reports it to CIMOM.

Common Model Database schema (CMDB). The resource repository of RMS (RMS database) is based on CMDB. CMDB is converged with ODI:RM?

Data Base Control Subsystem (DBCTL). An interface between CICS Transaction Server and IMS/ESA that allows access to IMS DL/I full-function databases and to data entry databases (DEDBs) from one or more CICS systems without the need for data sharing. It also provides release independence, virtual storage constraint relief, operational flexibility, and failure isolation.

Data Base Recovery Control (DBRC). A feature of the IMS Database Manager that facilitates easier recovery of IMS databases. DBRC maintains information required for database recoveries, generates recovery control statements, verifies recovery input, maintains a separate change log for database data sets, and supports sharing of IMS databases and areas by multiple IMS subsystems.

Data Communication Control (DCCTL). A subsystem that allows IMS TM to act as a stand-alone, full-function transaction manager that can connect to DB2 or other external subsystems.

Distributed Management Task Force (DMTF). A standardization organization which designed CIM or WBEM.

Domain Name System (DNS). The distributed database system used to map domain names to IP addresses.

End Point Reference (EPR). The combination of the address of a Web service that implements the manageability interface for a managed resource type and the properties that allow that Web service to identify the managed resource.

Enterprise Identity Mapping (EIM). A mechanism for associating a person or entity to the appropriate user identities in various user registries throughout an enterprise. EIM provides application programming interfaces (APIs) for creating and managing these identity mapping relationships, as well as APIs that applications use to query this information.

Enterprise Java Beans (EJB). A component architecture defined by Sun Microsystems for the development and deployment of object-oriented, distributed, enterprise-level application.

EWLM Resource Director (ERD). A function of EWLM which monitors the CPU resource of the LPAR partition of a EWLM managed server.

eXtensible Markup Language (XML). A text-based tag language used for document processing and for publishing information on the Web.

Hardware Management Console (HMC). A system that controls managed systems, including the management of logical partitions and use of Capacity Upgrade on Demand. Using service applications, the HMC communicates with managed systems to detect, consolidate, and send information to IBM for analysis.

Hypertext Markup Language (HTML). A markup language that conforms to the SGML standard and was designed primarily to support the online display of textual and graphical information that includes hypertext links.

Hypertext Transfer Protocol (HTTP). An Internet protocol that is used to transfer and display hypertext and XML documents on the Web.

Hypertext Transfer Protocol Secure (HTTPS). A Web protocol for secure transactions that encrypts and decrypts user page requests and pages returned by the Web server.

IBM Grid Toolbox (IGT). GT in Virtualization Engine Rel.1 is factored and integrated as "IBM Grid Services" and "Virtualization Engine Common Services and Components" in Virtualization Engine Rel.2.

IBM Software Platform Extension (SPX). IBM Software Platform Extension Software Componentization Architecture. A packaged Software brand products to install and support a solution, such as Virtualization Engine, easily.

Integrated Solution Console (ISC). ISC framework is a common console technology for systems administration based on WebSphere Portal technology. It is also used for Virtualization Engine console.

Intelligent Device Discovery (IDD). A network-based IT asset discovery and categorization tool. Data discovery in Resource Dependency Service is performed through IDD.

Internet Control Message Processing (ICMP). An Internet protocol that is used by a gateway to communicate with a source host, for example, to report an error in a datagram.

Internet Protocol (IP). A protocol that routes data through a network or interconnected networks. Internet Protocol (IP) acts as an intermediary between the higher protocol layers and the physical network.

Java 2 Platform Enterprise Edition (J2EE). An environment for developing and deploying enterprise applications, defined by Sun Microsystems Inc. The J2EE platform consists of a set of services, application programming interfaces (APIs), and protocols that provide the functionality for developing multitiered, Web-based applications. (Sun).

Java 2 Platform Standard Edition (J2SE). The core Java technology platform.

Java Authentication and Authorization Service (JAAS). In J2EE technology, a standard API for performing security-based operations. Through JAAS, services can authenticate and authorize users while enabling the applications to remain independent from underlying technologies.

Java Connector Architecture (JCA). Java Connector Architecture is a standard architecture for connecting the J2EE platform to heterogeneous Enterprise Information Systems (EIS). A resource adapter compliant with this architecture can plug into a J2EE server and provide connectivity between the app server, the EIS and the enterprise application.

Java Database Connector (JDBC). An industry standard for database-independent connectivity between the Java platform and a wide range of databases. The JDBC interface provides a call-level API for SQL-based database access.

Java Management Extensions (JMX). A means of doing management of and through Java technology. JMX was developed through the Java Community ProcessSM program, by Sun Microsystems, Inc. and some leading companies in the management field. JMX is a universal, open extension of the Java programming language for management that can be deployed across all industries, wherever management is needed. definition.

Java Native Interface (JNI). A programming interface that allows Java code to interoperate with functions that are written in other programming languages.

Java Networking Language Protocol (JNLP). To enable your Java application to execute using Java Web Start, you must create a Java Networking Language Protocol (JNLP) file for the application. The JNLP is an XML file that provides the basic elements and descriptions for your application. The JNLP conforms to the Java Community Process (JCP), under specification number JSR 000056 (see Resources). The purpose of the JNLP file is as follows:

- To specify JAR files to download from the Web server
- To provide other possible package requirements
- To specify system properties
- To allow for any necessary run-time parameters
- To specify which version of the Java 2 platform to use .

Java Runtime Environment (JRE). JRE provides the libraries, the Java Virtual Machine, and other components to run applets and applications written in the Java programming language. In addition, two key deployment technologies are part of the JRE: Java Plug-in, which enables applets to run in popular browsers; and Java Web Start, which deploys standalone applications over a network.

Java Virtual Management (JVM). A software implementation of a central processing unit that runs compiled Java code (applets and applications).

Lightweight Directory Access Protocol (LDAP). An open protocol that uses TCP/IP to provide access to directories that support an X.500 model and that does not incur the resource requirements of the more complex X.500 Directory Access Protocol (DAP). For example, LDAP can be used to locate people, organizations, and other resources in an Internet or intranet directory.

Lightweight Third Party Authentication (LTPA). A protocol that uses cryptography to support security in a distributed environment.

Local Area Network (LAN). A network that connects several devices in a limited area (such as a single building or campus) and that can be connected to a larger network.

Logical Partition (LPAR). A subset of a single system that contains resources (processors, memory, and input/output devices). A logical partition operates as an independent system. If hardware requirements are met, multiple logical partitions can exist within a system.

Message Queuing Administration Interface (MQAI). A programming interface that performs administration tasks on a WebSphere MQ queue manager through the use of data bags. Data bags allow the user to handle properties (or parameters) of WebSphere MQ objects.

Network Control Program (NCP). A program that controls the operation of a communication controller.

On Demand Infrastructure Resource Modeling (ODI:RM). ODI-RM(ODI:RM) is an architecture intended to facilitate an integration of the IBM products relevant to On Demand. ODI:RM is converged to RMS.

Open Grid Services Infrastructure (OGSI). The set of interfaces that enable developers to create, discover, manage, and exchange information among grid services.

Pegasus. Pegasus is one of the open source implementations of the CIMOM that adheres to the Distributed Management Task Force (DMTF) CIM and Web-based Enterprise Management (WBEM) standards. Pegasus is designed to be inherently portable and builds and runs on the AIX, Linux, and Windows operating systems.

Processor Resource System Manager (PR/SM). A feature that allows a processor to use several MVS images simultaneously and provides logical partitioning capability.

Resource Mapping Service (RMS). The architecture for Resource Mapping Service focuses on the discovery and mining of information from existing applications, tools, and IT systems, and presenting a static view of those assets on the RMS database.

Server Application State Protocol (SASP). The communication protocol between EWLM domain manager and a load balancer . The load balancer obtains the weight of each server or the way to distribute requests to the managed servers from the EWLM domain manager.

Service Oriented Architecture (SOA). A conceptual description of the structure of a software system in terms of its components and the services they provide, without regard for the underlying implementation of these components, services and connections between components.

Simple Object Access Protocol (SOAP) SOAP provides an open, extensible way for applications to communicate using XML-based messages over the Web, regardless of what operating system, object model, or language they use.

Simple Object Access Protocol (SOAP). A lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment. SOAP can be used to query and return information and invoke services across the Internet. The protocol consists of a framework for describing what is in a message and how to process it, a set of encoding rules for expressing application-defined data types, and a convention for representing remote procedure calls and responses.

Software Development Kit (SDK). A set of tools, APIs, and documentation to assist with the development of software in a specific computer language or for a particular operating environment.

Storage Area Network (SAN). A dedicated storage network tailored to a specific environment, combining servers, storage products, networking products, software, and services.

Transmission Control Program/Internet Protocol (TCP/IP). An industry-standard non proprietary set of communication protocols that provide reliable end-to-end connections between applications over interconnected networks of different types.

Transmission Control Protocol (TCP). A communication protocol used in the Internet and in any network that follows the Internet Engineering Task Force (IETF) standards for internet network protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It uses the Internet Protocol (IP) as the underlying protocol.

Uniform Resource Locator (URL). The unique address of a file that is accessible in a network such as the Internet. The URL includes the abbreviated name of the protocol used to access the information resource and the information used by the protocol to locate the information resource.

Universal Description, Discovery, and Integration (UDDI) UDDI is the service discovery protocol for Web services through which Web service providers and endpoints can be located.

Universal Description, Discovery, and Integration (UDDI). A set of standards-based specifications that enables companies and applications to quickly and easily find and use Web services over the Internet.

Universal Description, Discovery, and Integration (UDDI). A set of standards-based specifications that enables companies and applications to quickly and easily find and use Web services over the Internet.

Universal Resource Identifier (URI). A unique address that is used to identify content on the Web, such as a page of text, a video or sound clip, a still or animated image, or a program. The most common form of URI is the Web page address, which is a particular form or subset of URI called a Uniform Resource Locator (URL). A URI typically describes how to access the resource, the computer that contains the resource, and the name of the resource (a file name) on the computer.

Virtualization Agent (VA). A component of EWLM which runs in each p/i LPAR partition, handles interactions between EWLM domain manager and the p/i hypervisor.

Virtualization Manager (VM). A component of EWLM which runs on EWLM domain manager, the resource arbitrator for virtual LPAR groups.

Web Based Enterprise Management (WBEM). A set of Distributed Management Task Force (DMTF) standards that defines the protocols used to communicate with a particular Common Information Model (CIM) implementation that uses CIM servers. This architecture provides the management design framework that consists of devices, device providers, the object manager, and the messaging protocol for the communication between client applications and the object manager.

Web Service Definition Language (WSDL). An XML-based specification for describing networked services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.

Web Service Definition Language (WSDL). The standard format for describing a web service. Expressed in XML, a WSDL definition describes how to access a web service and what operations it will perform.

Web service. A software system identified by an URI whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service using XML based messages conveyed by Internet protocols.

Web Services Resource Framework (WS-RF) WS-RF provides a common pattern for accessing resources using stateful Web services.

Windows Management Instrumentation (WMI). An application programming interface (API) in the Windows operating system that enables devices and systems in a network to be configured and managed. WMI uses the CIM to enable network administrators to access and share management information.

Term3. Term3 definition.

Term3. Term3 definition.

Term3. Term3 definition.

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 375. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *IBM Director 5.10*, SG24-6188
- ▶ *Enterprise Workload Manager z/OS Support*, REDP-3993
- ▶ *Tuning IBM eServer xSeries Servers for Performance*, SG24-5287
- ▶ *Grid Services Programming and Application Enablement*, SG24-6100
- ▶ *Patterns: Implementing an SOA using an Enterprise Service Bus*, SG24-6346
- ▶ *Introduction to pSeries Provisioning*, SG24-6389
- ▶ *Patterns: SOA with an Enterprise Service Bus in WebSphere Application Server V6*, SG24-6494
- ▶ *Patterns: Implementing Self-Service in an SOA Environment*, SG24-6680
- ▶ *Patterns: Integrating Enterprise Service Buses in a Service-Oriented Architecture*, SG24-6773
- ▶ *Partitioning Implementations for IBM eServer p5 Servers*, SG24-7039
- ▶ *Advanced POWER Virtualization on IBM System p5*, SG24-7940
- ▶ *Enterprise Workload Manager - Interpreting Control Center Performance Reports*, REDP-3963
- ▶ *Enterprise Workload Manager Overview*, REDP-3989
- ▶ *Hardening the EWLM Performance Data*, REDP-4018
- ▶ *Enterprise Workload Manager z/OS Support*, REDP-3993
- ▶ *IBM Enterprise Workload Manager Release 1*, SG24-6350
- ▶ *On Demand Operating Environment: Managing the Infrastructure (Virtualization Engine Update)*, SG24-6634
- ▶ *High Performance On Demand Solutions*, SG24-6644
- ▶ *IBM Virtualization Engine Version 1 Planning and Installation Guide*, SG24-6771
- ▶ *IBM Enterprise Workload Manager Version 2 Release 1*, SG24-6785
- ▶ *Introducing the IBM Virtualization Engine TS7510: Tape Virtualization for Open Systems Servers*, SG24-7189
- ▶ *Grid Computing Products and Services*, SG24-6650

Other publications

These publications are also relevant as further information sources:

- ▶ *IBM eServer i5 and iSeries System Handbook: IBM i5/OS Version 5 Release 3, GA19-5486-26*

Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ To get the latest news about the IBM Virtualization Engine offerings:
<http://www-03.ibm.com/servers/eserver/about/virtualization/>
- ▶ To access the IBM Systems Software Information Center:
<http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jspDescription2>
- ▶ To access the Alphaworks IBM Web site:
<http://www.alphaworks.ibm.com/>
- ▶ To access the internet Web sites to get information about standards:
<http://www.opensource.org/>
<http://www.oasis-open.org>
<http://www.osgi.org>
<http://www.dmtf.org/standards/smash>
<http://www.openpegasus.org/>
<http://www.gridforum.org/>
<http://www.w3.org/>
- ▶ To use the IBM @server Workload Estimator:
<http://www.ibm.com/eserver/series/support/estimator>
- ▶ To get more information about non IBM products:
<http://www.vmware.com/>
<http://www.netapp.com>
<http://www.aurema.com>
- ▶ To download the Grid tool box:
<https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=grid>
- ▶ To know more about the IBM Director:
www.ibm.com/servers/eserver/xseries/systems_management/ibm_director
- ▶ To know more about the IBM TotalStorage solutions
<http://www-03.ibm.com/servers/storage/disk/ds/index.html>
- ▶ To know more about the IBM Electronic Services:
www.ibm.com/support/electronic/
- ▶ To know more about the autonomic IBM initiative:
<http://www-128.ibm.com/developerworks/autonomic/books/fpu1mst.htm>
- ▶ To know more about the Intelligent Device Discovery IBM tool:
<http://www.zurich.ibm.com/csc/ibi/idd.html>
- ▶ To know more about the IT Infrastructure Library:
<http://www.itil.co.uk/>
- ▶ To know more about the IBM provisioning and orchestration solutions:
<http://www-306.ibm.com/software/tivoli/products/intell-orch/>
- ▶ To know more about Linux on pSeries:

<http://www-03.ibm.com/systems/p/linux/Linux>

- ▶ To know more about the z/VM 5.2 operating system:

<http://www.vm.ibm.com/zvm520/>

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

Address Resolution Protocol (ARP) 112
 Advanced Encryption Standard (AES) 268
 Advanced POWER Virtualization
 feature 219, 333
 hardware feature 217, 220
 hardware feature capability 218
 Advanced Power Virtualization (APV) 29, 217, 333
 Advanced System Management Interface (ASMI) 223, 334
 alert standard format (ASF) 66
 American National Standards Institute (ANSI) 269
 application environment 108, 174
 Application Programming Interface (API) 243, 278, 329
 Application Response Measurement (ARM) 174, 323
 Application Workload Manager (AWM) 67
 APV feature 218
 new system 223
 ARM API 178

B

BladeCenter chassis 64

C

Candle Management Server (CMS) 155, 355
 Candle Net Portal Server (CNPS) 355
 Candle/E Net Portal Server (CNPS) 155
 Capacity BackUp 342
 configuration 342
 offering 342
 processor 342
 resource 342
 server 342
 system 342
 Capacity Manager 56, 213, 327
 Capacity on Demand (COD) 243, 340
 Capacity Upgrade 341
 pSeries systems 342
 Capacity Upgrade on Demand (CUOD) 341
 central electronics complex
 event management 39
 central electronics complex (CEC) 39, 333
 channel-to-channel adapters (CTCA) 350
 chapter XXX 164
 child transaction 179
 CIM Agent 104
 CIM provider 104
 CIM schema 105
 integrated support 112
 CIM-based repository 104
 client application 100, 196
 CIM requests 104
 client LPAR 231

VSCSI adapter 231
 client LPARs
 limited functions 229
 Virtual Ethernet 239
 Client partition 217, 338, 356
 disk spaces 237
 logical volumes 237
 physical hardware placement complexity 217
 physical I/O resources 220
 client partition
 secure communication channel 338
 Cluster Systems Manager (CSM) 151
 Common Information Management Object Model (CIMOM) 64, 104
 Common Information Model (CIM) 47, 103
 Common Information Model Object Manager
 Open Source implementation 105
 Common Information Model Object Manager (CIMOM) 104
 Common Interface Model (CIM) 18
 Concurrent Firmware Maintenance (CFM) 231
 CPU utilization 56, 80, 152, 185, 189, 201, 350
 significant bias 202
 CPU utilizations
 normal short spikes 82

D

Data collector 109
 Data compression 77
 data encryption algorithm (DEA) 269
 Data Encryption Standard (DES) 268–269
 DB2 UDB
 8.2 314
 data collector 115
 database 115
 version 8.2 SP1 132
 Dedicated Service Tools (DST) 217
 Demand Business 2
 Directory Access Protocol (DAP) 257
 Distributed Component Object Model (DCOM) 47
 Distributed Management Task Force (DMTF) 18, 103
 domain manager 108, 115, 176, 290, 309
 domain name (DN) 195, 296
 domain policy 173, 187
 service policies 187
 dynamic LPAR
 operation 226
 Dynamic LPAR (DLPAR) 206, 226, 332

E

endpoint reference 14
 only required element 14
 Enterprise Identity Mapping (EIM) 256, 295, 358
 enterprise service bus

- Services interact 9
- Enterprise Service Bus (ESB) 9, 109
- Enterprise Workload Manager 22, 27, 60, 115, 136, 173, 278
 - technical introduction 173
- eServer i5 217, 344
 - AIX partitions 358
 - IBM POWER processors 347
 - Intel processor-based Linux applications 358
- eServer p5 217, 340
 - model 344
 - product 220
- Evaluation Assurance Level (EAL) 354
- Event Action
 - Plan Builder 91
 - Plan wizard 88–89
- Event Action Plan
 - Association 93
 - Builder 91
 - entry 63
 - wizard 79, 88
- event action plan (EAP) 48, 88, 94
- event filter 88
- Event Log
 - AS55 84
 - example 80
- EWLM domain
 - management server 115
 - manager 27, 145, 177, 291, 309, 355
 - policy 186
- EWLM management
 - domain 176, 297
 - source 149
- EWLM performance
 - adjustment 189
 - information 213
 - management 175
 - monitoring 189
 - report 182
- EWLM V2R1
 - release 208
- eXtensible Markup Language (XML) 6, 169
- external OA 213

F

- Federal Information Processing Standard (FIPS) 269
- following URL
 - DMTF web site 104
- following Web site
 - detailed information 340

G

- given Management Server
 - Virtualization Engine Director 323
- Global Configuration Repository (GCR) 132, 256, 295
- Global Grid Forum (GGF) 14
- global Pi 201
- Graphical User Interface (GUI) 109, 137, 219, 330
- Guest LAN 350

H

- Hardware Management 39, 50, 136, 206, 215, 222, 361
 - comprehensive solution 225
 - Console 29, 39, 136, 206–207, 215–216, 331, 361
- hardware multithreading (HMT) 344
- hardware platform 37, 44, 215, 281, 288, 329
 - complete information 76
 - unique combination 288
- health center 133
 - custom task 153
 - single page 157
- host name 62, 195
- hypervisor 20–21, 177, 219, 332

I

- I/O adapter
 - resource 331
 - slot 332
- I/O device 21, 221, 231, 338, 352
- I/T environment 99
- I/T Infrastructure
 - become part 100
 - inventory records 99
 - Resource Dependency Service 118
 - standards-based model 100
 - unifying abstract representation 99
- I/T infrastructure 98, 311
 - device 105
 - element 103
 - environment 316
 - manageable resource 103
 - resource 107, 109
 - topology 99
- i5/OS example 77
- i5/OS partition 50, 80, 211, 217, 281, 347
 - CPU resource monitor 80
- i5/OS system 45
 - IBM Director Agent software 63
 - IBM Director installation 45
- IBM Common Logging (ICL) 293
- IBM Director 21–22, 37, 40, 51, 55, 105, 260, 272, 313, 327
 - base functions 65
 - cfgsecurity command 268
 - continuous updates 317
 - established association 93
 - following tasks 329
 - further information 40
 - infocenter documentation 144
 - installation steps 144
 - Inventory Collection Preferences page 75
 - Inventory task 74
 - many different ways 56
 - multiple workstations 45
 - new version 22
 - previous releases 268
 - Resource Monitor task 90
 - right corner 87
 - systems management capabilities 67

- IBM Director Agent 40, 48
 - IBM Director information
 - center 45
 - center web page 40
 - IBM Eserver 219
 - IBM eServer
 - i5 hardware platform 39
 - i5 Implementation Guide 244
 - i5 system 50, 219
 - i5/p5 system 175
 - iSeries 67
 - Operating System 105
 - p5 hardware platform 39
 - p5/i5 177
 - pSeries 22, 229, 356
 - pSeries customer 32, 356
 - Workload Estimator 22, 246
 - IBM eServer xSeries
 - system 66
 - IBM System
 - i5 215
 - p5 215, 336
 - p5 hardware platform 39
 - p5 server 333
 - z9 56, 349
 - IBM System p5 217
 - Model 505 223
 - IBM Tivoli
 - Monitoring management source 155
 - IBM Tivoli Directory Server (ITDS) 291
 - IBM Tivoli Monitor (ITM) 151
 - IBM TotalStorage
 - component 27, 29
 - DS300 40
 - DS8000 105
 - IBM Virtualization Engine 1, 66, 128, 133, 174, 272, 347
 - Enterprise Workload Manager 30, 308, 353
 - new component 28
 - new version 1
 - offering 4
 - platform 1, 21
 - platform Version 2 29
 - Release 1 31
 - Resource Dependency Service 32, 359
 - strategic component 214
 - Suite Version 2 315
 - V5.10 feature 66
 - Version 1 3, 28, 320
 - Version 2 1, 4, 23
 - Version 2 product 30
 - Virtual Access 22, 26
 - Wizard 298
 - IBM Web 183
 - IBM xSeries
 - server 67
 - system 66
 - identity mapping repository (IMR) 256–257, 295
 - inactive processor 341
 - industry standard 99, 120
 - Information Center 271
 - EWLM topic collection 278
 - planning advisor 284
 - prominent characteristic 273
 - topic collection 275
 - Virtualization Engine documentation 272
 - Virtualization Engine information 274
 - Infrastructure Service 8–9, 66, 355
 - Integrated Solution Console (ISC) 255, 293
 - Integrated Virtualization Manager (IVM) 24, 27, 215, 361
 - Integrated xSeries Adapter
 - IBM System i5 family system 281
 - Integrated xSeries Adapter (IXA) 281, 332
 - Integrated xSeries Server
 - Linux support 359
 - Windows 2003 Server 358
 - integrated xSeries Server (IXS) 281, 332
 - Intelligent Device Detection (IDD) 109
 - Intelligent Resource Director (IRD) 332, 355
 - Inventory subtask 85
 - IP address 47, 195, 234, 352
 - iSeries 22–23, 39, 105, 136, 281, 325
 - iSeries administrator 217
 - iSeries hardware
 - platform 39
 - service 243
 - iSeries Navigator 137, 359
 - Application Administration interface 45
 - LPAR management function 243
 - iSeries system 44, 159, 216, 239, 287, 347
 - IBM Director 287
- ## J
- J2EE runtime 313
 - Java Authentication and Authorization Service (JAAS) 260
 - Java Connector Architecture (JCA) 121, 255
 - Java Message Service
 - bus name 196
 - destination name 197
 - MDB discriminator 197
 - method selector 197
 - topic name 197
 - Java Message Service (JMS) 196
 - Java Native Interface (JNI) 178
 - Java Networking Language Protocol (JNLP) 144
 - Java Runtime Environment (JRE) 131, 160, 314
 - Java Web Start (JWS) 160
 - JDBC connection 115
 - JNLP protocol 160
- ## L
- Lightweight Directory Access Protocol (LDAP) 257
 - Lightweight third party authentication (LTPA) 155, 260
 - Linux 31, 38, 145, 198, 241, 281, 332, 361
 - Linux distribution 217, 281
 - level 220
 - release level information 244
 - Linux instance 57
 - z/VM virtual server 57

Linux partition 39, 207, 211, 217–218, 240, 281, 347
 load balancer 175, 177
 logical partition 27, 39, 217, 220, 239, 316, 331
 channel resource utilization 332
 CPU allocations 27
 CPU resources 333
 logical processors 332
 logical unit number (LUN) 226

M

manageable resource
 Anatomy 101
 different types 121
 manageable resource (MR) 16, 18, 97, 99
 Management of Web Services (MOWS) 16–17
 Management Server 32–33, 42, 149, 163, 260, 269, 278, 287, 330, 357
 IP domain 47
 management server
 first window 287
 SSL encryption 269
 Management Source
 new name 149
 management source 34, 139–140
 Management Using Web Services (MUWS) 16–17
 Managing Using Web Services (MUWS) 100
 Media Access Control (MAC) 112
 Message Queuing Administration Interface (MQAI) 116
 middleware 3, 6, 174

N

National Security Agency (NSA) 269
 Network Interface
 Card 350

O

objective analyzer (OA) 213
 On/Off Capacity 341
 On/Off CoD
 activation procedure 342
 feature 341
 processor day 342
 one server (OS) 192
 Open Grid Services Architecture (OGSA) 13
 Open Grid Services Infrastructure (OGSI) 13–14
 Open Services Gateway Initiative (OSGI) 18
 Open Systems Adapter (OSA) 25
 operating system
 release level 194
 Operating System (OS) 3, 20, 37, 111, 165, 191, 217, 281, 330, 361
 operational model (OM) 310
 optional extension 65

P

parent-child relationship 182
 Parity member 77
 partition class 186, 191

PDF 224, 272
 performance data 327
 tests various attributes 194
 performance goal 57
 performance index (PI) 185
 physical processor 206, 208, 238, 334, 336
 hardware threads 344
 planning across multiple (PM) 359
 Planning adviser 23, 271
 Managed nodes section 288
 platform function 37
 portlet 151, 264
 Power hypervisor 177, 219, 332
 POWER5 system 227, 336
 Advanced POWER Virtualization features 339
 POWER5 technology 217
 Predictive Failure Analysis (PFA) 89
 process class 186, 322
 processing power 207, 231, 332
 rounded-up number 207
 processor partition 332
 processor pool 226
 processor power 205, 231, 248
 automatic movement 249
 product name 31, 38, 244

Q

queued direct input/output (QDIO) 351

R

Red Hat
 Enterprise Linux 220, 336
 Linux EL 3.0 57
 Red Hat Enterprise Linux 359
 Remote Deployment Manager (RDM) 55
 Remote Monitoring and Control (RMC) 219
 Remote Package Deployer (RPD) 304
 Remote Supervisor Adapter (RSA) 48
 Resource Dependency Service
 base facility 292
 Event logging 130
 high-level components 108
 I/T environment 117
 interesting feature 122
 mining tools 119
 runtime environment 131
 Resource Dependency Service (RDS) 22, 26, 28, 97, 135, 259, 286, 289, 355
 Resource Dependency Services (RDS) 22, 26
 Resource Model 22, 97
 Resource Monitor 63, 80
 Resource Monitoring and Control (RMC) 334
 Resource Type
 wide variety 117
 resource type 80, 104, 117
 resource virtualization
 facilitation 29
 service 9, 11
 response file 305, 307

response time 174
 right pane 50, 286
 memory classifications 85
 runtime component 30–31, 291, 294, 355

S

SAN Volume Controller (SVC) 22, 29
 Secure Shell
 SSH capability 243
 support 270
 Secure Sockets Layer (SSL) 268
 security mechanism 132, 253
 Server Complex 57, 355
 Linux instances 57
 Server Message Block (SMB) 112
 Server/Application State Protocol (SASP) 204
 service class 186–187, 322
 importance level 189
 OA instance 213
 performance goal 188
 Service Focal Point (SFP) 231, 362
 Service Management Framework (SMF) 18
 Service Oriented Architecture (SOA) 5
 service policy 186, 322
 shared Ethernet adapter (SEA) 220, 338
 shared processor partition
 CPU allocation 208
 processor utilization 207
 shared processor pool
 spare power 207
 Simple Network Management Protocol
 other hardware 43
 Simple Network Management Protocol (SNMP) 42, 60, 112
 Simultaneous multi-threading 344
 Single Byte Character String (SBCS) 40
 Single Object Access Protocol (SOAP) 6
 Single Sign-on (SSO) 253
 SNMP device 42
 managed object 69
 Remote Control 57
 stateful resource 15
 Storage Area Network (SAN) 226
 Storage Network Industry Association (SNIA) 105
 Storage Networking Industry Association (SNIA) 18
 Storage pool 235
 SUSE Linux Enterprise Server (SLES) 130
 System p5 39, 105, 215, 326
 System z9 31, 39, 105, 326
 I/O bandwidth 57

T

Tivoli Business Service Manager (TBSM) 108
 Tivoli Configuration Manager (TCM) 115
 Tivoli Enterprise
 Console 93
 Management 155
 Management Server 155, 355
 Portal 155

Portal Server 155, 355
 Server 155
 Tivoli Enterprise Console (TEC) 93
 Tivoli Intelligent Orchestrator (TIO) 103, 175
 Tivoli Management Environment (TME) 155
 Tivoli Provisioning Manager (TPM) 29
 topology view 128, 133
 total cost of ownership (TCO) 29
 TotalStorage Productivity center (TPC) 105
 transaction name 192
 Transmission Control Protocol (TCP) 112

U

unauthenticated user 142, 255, 301
 Uncapped partition 207, 238, 332–333
 uncapped weight 209
 uncapped partition
 effective advantage 238
 Uniform Resource Identifier (URI) 8, 192
 Universal Data Base (UDB) 293
 URI value 195
 URL 158, 273
 User Datagram Protocol (UDP) 112
 User Interface 40, 110, 129, 137, 176, 217, 362
 user interface
 little pieces 151
 minor enhancements 214

V

Velocity goal 184, 201
 VIOS partition 218, 221
 Full DLPAR support 230
 virtual device 217
 virtual ethernet 25, 217, 338–339
 device drivers 339
 network traffic 339
 virtual I/O
 adapter 338
 adapter slot 332
 client 220
 configuration 232
 device 239–240, 338
 function 220, 338
 Virtual Local Area Network (VLAN) 21
 Virtual Machine
 Manager 27, 56, 329
 Manager Web site 66
 Resource Manager 350
 Resource Manager 354
 virtual machine 11, 21, 56, 121, 329–330
 several hundreds 25
 Virtual Partition Manager 215, 239, 348
 key differences 241
 Virtual processor 207, 238, 332
 Virtual Resource 21, 98, 133, 325
 virtual resource
 I/T Infrastructure 98
 virtual SCSI
 channel 230

- server adapter 249
- target 220
- virtualization capability 3, 216, 280, 325
 - functional capabilities 280
- virtualization component 29, 218, 329
 - VMM operations 329
- Virtualization Engine 1, 18, 37, 97, 133, 139, 253, 271, 325
 - actual installation 298
 - actual installation process 297
 - Associations set-up 259
 - certain functions 160
 - Health Center 152
 - health center 150
 - Information Center example 274
 - integrated functions 27
 - new release 310
 - online help 274
 - optionally installed component 59
 - other panels 118
 - release 2 141
 - Resource Dependency Service services 129
 - Resource Dependency Service specific user interface panels 109
 - Resource Dependency Service-specific user interface panels 109
 - second version 171
 - security information 260
 - server components 291
 - shows performance comparisons 319
 - single sign-on part 259
 - user logs 255
 - web services 154
- Virtualization Engine component 290, 314
- Virtualization Engine Version 1
 - counterpart 319
 - EWLM 321
- Virtualization Engine version 1
 - measurement 317
 - performance number 310
- Virtualization Engine Version 2 271
- Virtualization EngineC 26–27, 137, 139, 260, 293
 - IBM Director 27
- Virtualization EngineE 311
- virtualized environment 22, 99, 134, 142
 - essential elements 99
- VMM Agent 330
- VMM object 329
- VMware VirtualCenter 56, 329

W

- Watch List 152
- Web Based Enterprise Management (WBEM) 105
- Web Browser 59, 139, 176, 228, 327, 362
- Web browser
 - multiple users 327
- web browser
 - control center 176
 - primary user interface 228
- Web service 5–6, 11, 17, 101, 128, 197

- data communication protocol 6
 - manageability capabilities 17
 - WSDM-manageable resources 104
- Web Services Description Language (WSDL) 6
- Web site 3, 6, 275
- WebSphere Application Server 281
 - 5.0 170
 - 6.0 170
 - 6.0.2 293
 - admin 184
 - container 322
 - design topology 313
 - environment 260
 - http port number 302
 - instance 60, 155, 291
 - security registry 301
 - security repository 301
 - security setting 299
 - security user registry 295
 - V6.0 214
 - Web 184
- WebSphere Application server
 - data collector 115
 - version 6.0.2 132
- weight value 238
- Windows Management Instrumentation (WMI) 112
- work area 233
 - physical volume 239
- Workspace area 162
 - displayed contents 167
 - easy, select two resources 165
 - relationship table view 162
- WS-Addressing specification 14
- WSDL document 6, 14
- WSDM specification 16–17

X

- XML format 6, 327
 - performance reports 327
- XML vocabulary 8
- xSeries system 66, 288, 357
 - CPU utilization resource 80

Z

- z/OS Management Console (Z/MC) 155
- z/OS system 155, 332, 355
- z/VM Center 39
 - extension 355
 - Manager 56
 - task 56

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5" spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(->Hide->Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine:fm still open and **File>Import>Formats** the Conditional Text Settings (ONLY) to the book files.

Draft Document for Review December 19, 2005 9:53 am

7112spine.fm 383



IBM Virtualization Engine Platform Version 2

(1.5" spine)
1.5" <-> 1.998"
789 <-> 1051 pages



IBM Virtualization Engine Platform Version 2

(1.0" spine)
0.875" <-> 1.498"
460 <-> 788 pages



IBM Virtualization Engine Platform Version 2

(0.5" spine)
0.475" <-> 0.873"
250 <-> 459 pages



IBM Virtualization Engine Platform Version 2

(0.2" spine)
0.17" <-> 0.473"
90 <-> 249 pages

(0.1" spine)
0.1" <-> 0.169"
53 <-> 89 pages

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5" spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(->Hide:)->Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine:fm still open and **File>Import>Formats** the Conditional Text Settings (ONLY!) to the book files.

Draft Document for Review December 19, 2005 9:53 am

7112spine.fm 384



IBM Virtualization Engine Platform Version 2

(2.5" spine)
2.5" <-> nnn.n"
1315 <-> nnnn pages



IBM Virtualization Engine Platform Version 2

(2.0" spine)
2.0" <-> 2.498"
1052 <-> 1314 pages



IBM Virtualization Engine Platform Version 2 Technical Presentation Guide



**Describes the
Virtualization Engine
management
collection
components**

**Describes the Virtual
resources for all IBM
eServers**

**Provides slides with
speaker notes**

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**